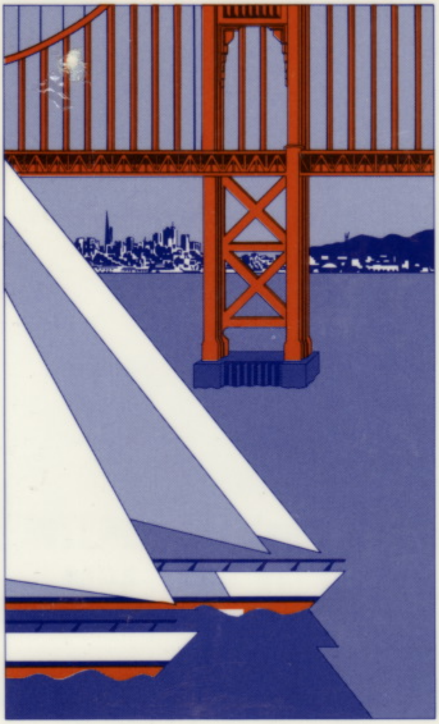
I N T E R E X



San Francisco '89

INTEREX HP Users Conference

September 11-14 1989

MPE, MPE XL, O.A.

PROCEEDINGS

Volume 1

Sponsored by INTEREX
The International Association of Hewlett-Packard Computer Users

INTERIX

The International Association of
Hewlett-Packard Computer Users

Proceedings

of the
1989 INTEREX HP Users Conference
MPE, MPE XL, O.A.
at
San Francisco, California
September 11-14, 1989

Volume 1

Louis Gorewitz, Editor

NOTES

Harnessing the Power of HP Desk using the NEW HP DeskManager Intrinsics

How your applications can provide the right information, to the right people at the right time - AUTOMATICALLY

Callum Johnston
Office Productivity Division
Hewlett-Packard
Nine Mile Ride
Wokingham, Berkshire RG11 3LL
UK

Introduction

The purpose of this paper is to demonstrate how you can integrate your applications to HP DeskManager using HP DeskManager Intrinsics and the many benefits that you will achieve from doing so.

The paper will cover the following:

A brief outline as to why our customers requested application integration to HP DeskManager and hence why we produced HP DeskManager Intrinsics.

A product overview of HP DeskManager Intrinsics and some simple scenarios.

An investigation as to how Hewlett-Packard is using the product to solve a common financial problem.

A more detailed investigation into the many possibilities that are available from integrating HP DeskManager to both applications and other mailsystems.

Product Evolution

Since the information technology explosion, most organizations have suffered the unfortunate side affect of an "information crises". Managers and decision makers are receiving too much irrelevant information. A classic example is the manager who receives a large computer printout and subsequently throws it away since he does not have the time to carefully go through and check to see if there is anything of relevance. Hence, any vital information contained in the printout is lost forever. In effect, what we have is a situation where critical information is not being delivered to the appropriate decision makers and if it is delivered it often arrives too late to be useful.

The implications of such a problem can be very great indeed. Time delays in reporting potential problems can be extremely expensive. This is demonstrated best by the example of a manufacturing organization running out of a vital component - for some organizations the potential loss in earnings could be staggering. As well as short term decisions being affected, a lack of critical information ultimately leads to poor long term and strategic decisions being made since vital pieces of information go unnoticed or unreported.

When our customers considered these problems, they realized that much of the information they required was in fact available. It was simply "sitting" in their applications. The problem was that there was no way of distributing this information from the application to the appropriate individuals. The obvious solution to this problem was to provide applications with a fast and efficient information distribution and collection mechanism, such as HP DeskManager, and ensure that the right information is delivered to the right people at the right time.

Hence, we produced HP DeskManager Intrinsics.

Product Overview

HP DeskManager Intrinsics is an add on product to HP DeskManager, therefore HP DeskManager is required.

HP DeskManager Intrinsics provides three types of intrinsic access to HP DeskManager - User Intrinsics, Gateway Intrinsics and Supporting Intrinsics. These are basically a set of tools that provide you with the means of integrating your applications to HP DeskManager.

The User Intrinsics allow an HP 3000 application to access, in a simple way, the mailing and calendar services of HP DeskManager. Hence, an application can log on to HP DeskManager as if it were a user and then send and receive items, such as reports or messages. The User Intrinsics can also provide access to the calendars of HP DeskManager users and the ability to insert and read standard calendar items.

The Gateway Intrinsics allow a simple and efficient link from HP DeskManager to foreign mail systems. The Gateway Intrinsics are a superior alternative to the existing HP DeskManager Foreign Service Connection (FSC) and should be used to develop applications that connect HP DeskManager to other messaging systems.

The Supporting Intrinsics provide support for both the User and Gateway Intrinsics. For example they provide your application with the ability to look in the HP DeskManager directories for a given name or list of names, and the ability to convert a document from one format to another.

Details of how to use these three types of intrinsics will be given later.

Simple Scenarios

HP DeskManager Intrinsics is a fairly complex product to understand and is best demonstrated by some simple scenarios. The first three examples demonstrate the use of the User Intrinsics - the ability for your application to access the features of HP DeskManager.

1. Exception Reporting

An excellent use of the product is for exception reporting. This is where, for instance, a manufacturing or stock control application could automatically inform the appropriate personnel of any stock shortage or production problem. As soon as the stock reaches a certain level, the application would send an appropriate message, via HP DeskManager, to the purchasing manager who could immediately take the necessary corrective action.

By providing your applications with the ability to distribute exception reports, decision makers are assured of receiving the right information and not simply more information.

2. Automatic Task Scheduling

The second example represents how HP DeskManager Intrinsics can provide solutions in a Sales or Service environment. Call Planning Applications, that schedule Sales or Service personnel's time with customers, could automatically send ToDo lists either to the appropriate personnel's in tray or enter tasks in their electronic calendar.

This would increase the productivity of your Sales and Service personnel, reduce administration and communication overheads, lead to a more effective management of resources and improve the service that you can offer your customers.

3. Project Resource Consolidation

The third example would provide a cure for a major headache for accounting and finance departments - that of tracking the costs and resources of various projects. The financial application would automatically distribute project resource forms at the end of each month, via HP DeskManager, to the various project managers. These can then be completed and returned to the application for consolidation and re-distribution.

This leads to improved resource management and budgetary control as well as a saving of finance personnel's time.

4. Other Messaging Systems

The fourth example demonstrates the use of the Gateway Intrinsics. These are a set of tools that allow your organization to develop applications that connect HP DeskManager to public messaging services and other proprietary mail systems. The Gateway Intrinsics substantially reduce the amount of engineering time necessary for this type of integration and provides your organization with the opportunity to have a simple and efficient link to all your messaging systems.

How Hewlett-Packard is using HP DeskManager Intrinsics

Here we have an example of how the Finance department at Hewlett-Packard's Office Productivity Division (OPD) have used HP DeskManager Intrinsics as part of a solution to solve an accounting and finance problem.

Situation

Hewlett-Packard employs many localization engineers in different European countries who translate our software products into the necessary native languages. These engineers are required to submit details of their expenses and costs to OPD's finance department for consolidation with other data on a monthly basis.

However, this monthly cost data tended to arrive on an ad hoc basis with the data for each of the individual countries being presented in an inconsistent format and requiring the time consuming task of manually comparing and consolidating the monthly localization cost data with the budgeted cost data.

Implication

Due to a lack of close monitoring, the engineers often overspent their budgets. This was due to the time delay in receiving, processing and consolidating the localization data. The finance department therefore had problems with their budgetary control since the information was processed too late for corrective action to be taken. Also much time was spent manually consolidating these costs with other data.

Solution

OPD's Management Information Services (MIS) department had developed a system whereby the localization engineers do not have to complete seaport details of their costs and expenses. This is because the necessary information is removed automatically from the Localization Application as the engineers use it on a day to day basis. At the end of the month the engineers simply mail the file, via HP DeskManager, to OPD where the Localization Application immediately consolidates the localization cost data with the localization cost budgets.

By developing a Localization Application that automatically removes the required data and integrating it to HP DeskManager using HP DeskManager Intrinsics, the information provided by the engineers is consolidated automatically as soon as it is mailed to the application.

Benefits

By integrating HP DeskManager to the new Localization Application using HP DeskManager Intrinsics the Office Productivity Division has experienced the following benefits.

Dollars

The bottom line is that HP DeskManager Intrinsics has proved to be a vital link in solving the finance departments problems. This is due to savings that are a direct result of integrating the new Localization Application with HP DeskManager using HP DeskManager Intrinsics:

Firstly, the department has been able to reallocate the relevant finance personnel's time from manually consolidating the data to other tasks.

Secondly, the division is making savings from the reduction in "over-spending" by the engineers due to the tighter budgetary control that can now be exercised.

Effectiveness

The solution has proved to be extremely effective. By consolidating the resource information as soon as it is sent to the application, the finance department has been able to keep a tighter budgetary control on the localization process and greatly improve their resource management decision making process.

Value

In addition to these quantifiable benefits, the finance department has also experienced several other less tangible benefits. The system is now easier to use since the whole process is now automated. Finance personnel no longer have to carry out the repetitive and time consuming task of manually consolidating the localization data and they no longer have the frustration of waiting indefinitely for the information to be sent.

This real life example demonstrates how even a relatively simple level of application integration to HP DeskManager has provided a straight forward, yet vital, solution in solving a difficult and costly problem.

A More Complex Example

We are now going to explore in a little more detail a hypothetical example of a stock control system. The idea is to expose you to some uses of the intrinsics that you may not have considered and also to provide a clearer understanding of what is involved in using them.

Let us first look at an example of a stock control system and the typical problems encountered with such a system.

A large organization involved in the manufacture and supply of radio equipment has a stock control system (Slide 1). At the end of the day a report is printed containing a list of all the stock items that have reached their 'low stock level'. The report is sent to the purchasing department and distributed amongst the purchasing clerks by the supervisor. The ordering process is now completely outside the control of the stock control application, and it will continue passively regardless of the state of the parts on order.

If one of the clerks loses an order, takes time off sick etc. and an order is not placed, there is no automatic procedure to chase this (Slide 2). Therefore mistakes will inevitably occur and the production manager is not likely to hear about any shortfalls in stock level until it is too late.

There is a need for a system capable of monitoring the process and starting some form of escalation procedure in the early stages of problem development.

The HPDesk Intrinsics provide the application with this monitoring and reporting capability. We will see how the problems could be avoided using the Intrinsics (Slide 3/4).

The stock control system logs onto HPDesk using the HPDesk Intrinsics and sends a detailed order to the appropriate purchasing clerk to start the order process rolling. The application also puts a READ acknowledgment on the message, and makes a note of the time at which the item reached its low stock level position. This will be used later as part of the escalation process.

The application has been configured to check the status of this order after a preconfigured time period. To do this it must signon to HPDesk using the Intrinsics and perform a list of its PENDING TRAY, checking to see if the item has been READ (Slide 5).

If the request has been READ then everything is O.K. subject to a further check we will see later.

If the request for new parts has not been READ then the application will alert the purchasing supervisor (Slide 6). It may be that the purchasing clerk is off sick, snowed under with too much work etc. This escalation process automatically informs the supervisor and thus the task can be reallocated, avoiding delays.

We have already mentioned that the time at which the component became a 'low level item' has been noted. We will also have configured another variable; this will be the maximum amount of time that may elapse between the component part becoming a 'low level item' and the item being reordered. So the application will know if any item has not been reordered.

If this time limit is exceeded our new stock control/escalation system can automatically alert the manufacturing manager through the HPDesk Intrinsics, thus each manager affected by the problem is informed by the application before the problem becomes critical. The scope for improved efficiency and disaster aversion is enormous.

Slides 7, 8 and 9 demonstrate in detail how the Intrinsics would send such a message.

This is all very well as long as all the people involved are users of HPDesk, however if any of these people are not on the HPDesk network but on some foreign electronic mail system the intrinsics provide a means of linking to this system via the Gateway Intrinsics.

Assuming the foreign system has acknowledgment support the Gateway Intrinsics allow us to take full advantage of this.

Conclusion

Although these are all theoretical applications, they are all possible for HPDesk users now that the Intrinsics are available. By effectively adding a complete transport mechanism (HP DeskManager) to your applications, we have opened the door to a whole new generation of intelligent software. The speed at which these new applications evolve will be driven by your demand to your software suppliers and your own in-house creativity.

Don't Be Afraid, The Computer Won't Bite

Donna Clark
MagneTek Universal Electric
300 East Main Street
Owosso, Michigan 48867
(517)723-7866

There I was, in southern Tennessee, Ripley to be exact, in what would be my first user training experience since I had become a computer programmer just one year before. If I can recall correctly, there were fourteen people to be trained during the course of one week. We would begin with a demonstration or the 'big picture' of how this new purchasing, receiving, inventory system would fit together. We used a personal computer, an overhead projector, and a datashow. We gathered in groups of four or five people and gave three demos the first day. This was to give them a general idea of what was to evolve or what they would be exposed to during the next few days.

It really wasn't until the next day that I began to sense the fears that individuals can have when it comes to computers. I happened to overhear a conversation some of the office personnel were having regarding the demo given the previous day. Those involved were kind enough to include me in on the discussion. Apparently one of the users was so overwhelmed by the demo that she had a nightmare about it that night. Thus begins the fear. Her description was so vivid and realistic. When she spoke of the experience, her eyes were wide with expression showing it had a devastating impact.

I had another experience a month later with a person in Altavista, Virginia that helped prompt me to write this paper. He was a man near retirement age who had apparent health problems. He attended the demo in body but not in mind or spirit. I knew what he was thinking. I'm going to retire soon, why do I need to learn something new now? The way I've been doing my job has worked for me so far, why change? These natural fears are mole hills for young changeable people, but mountains to older people. This training was obviously creating extra stress in his life. Because of this added stress and his health problems, I was afraid I might witness a heart attack or stroke.

These are two of many experiences that reflect the point I am trying to make. People are afraid of these newfangled computers.

There are ways to deal with these fears. In the next few pages I will discuss what I consider items to avoid and items of importance to become an effective user trainer.

First items on the list are what I call "ignorant training methods". Hopefully by mentioning these, you will be mentally aware of them and try not to incorporate them into your training skills.

I believe most of us have sat in a class or been caught in a conversation in which the instructor is in the stratosphere and we are lost somewhere else. If you are like most, yawning begins and your brain shuts down. Always be careful not to speak over anyone's head.

If your dealing with computer illiterate people then for heaven's sake, do not talk computer jargon. You might just as well be talking a foreign language because that's exactly what it sounds like as far as users are concerned. For instance, a record to them is something you play on a stereo. A field is where you grow crops. A flush is something you do to a toilet. I think you get the picture.

People skills would rate as number one on my list of effective trainer skills. The ability to work with people. I'm not just talking about communication skills. I'm talking about emotional skills, like compassion. You want these people to realize you are human too. Get to know them. Pick up on their interests. Let them know that there was a time when you had to be trained just like they are being trained now.

Be patient. Everyone works at a different pace. Adjust to that pace. You don't want to create extra pressure by making them hurry.

Have a positive attitude. Be energetic, happy and always enthusiastic. These are contagious and tend to rub off on people.

Treat these people as equals. No one likes to feel they are lesser than someone else.

Build confidence in your users. Always be positive in your support. It may take several times before they remember how to enter something without being told. Keep enforcing the fact, it will come.

I don't know if I can stress personal appearance enough. The way you dress and your personal hygiene, are both important factors in training people. If you portray a professional image, people will take you in a more serious manner. Respect for the trainer will create a learning environment.

Limit your groups for training. One to one is obviously the best or one to two. This will depend, of course, on the time and resources available.

Provide your users with any available printed documents to help them with their new applications. Once your training is complete, and you have left the premises, it's nice to have a document to refer to for help if you are having problems.

Last but not least, provide follow-up for your trained users. Let them know how, where, and when they can reach you. Make them understand no problem or question should be considered stupid or unimportant. Keep in touch with them. If time lapses and you've not heard from them, call them and see how everything is going. Let them know how important they are to you.

If you are put in a position as a user trainer, then it is your responsibility to become as knowledgeable as possible so you can communicate effectively to the users who depend upon you for comprehensive and usable information.

What about the 'big picture'? The big picture is how the system fits together as a whole. Most of the user training I have been involved in has provided a demonstration or 'big picture' if you will, and in some cases I agree it's useful and in some cases it is not. Sometimes you can develop unnecessary fear. Remember the woman with the nightmare about the volume of material covered? Once she realized she had to learn only a small portion of what was covered in the demo, she was able to go into the individual training environment without fear. Basically users are only concerned with the portion of the system that involves them. They don't really care how their daily tasks or anyone else affects the system as a whole. On the other hand, I believe any information you can obtain makes you a more valuable asset to your company. I would suggest that you provide a demo or 'big picture'. Users who are ambitious and want to absorb more can benefit from all the information you have provided. To ease the minds of those who are only concerned with just their daily function, point out to them that they are not required to know the 'big picture'.

Items of importance:

1. Powerful people skills
2. Be patient
3. Maintain a positive attitude
4. Treat everyone as an equal
5. Build confidence in users
6. Portray professional appearance
7. Limit training group size
8. Produce printed documents
9. Be knowledgeable
10. Provide follow-up

Items to avoid:

1. Speaking over people's heads
2. Computer jargon

Fears are a part of everyday life; they are unavoidable. If we can combat just one of those fears in a persons life then we have truly accomplished a great task. If this discussion can help you become a more effective user trainer in the computer world, then we have begun to bridge the gap between those who are computer literate and those who are not.

PCs: Choices For the Future
or
Who's Been Pirating This Software?

By Mark W. Miller

Exlog, Inc.
P.O. Box 40265
Houston, Tx 77240
(713) 744-4794

Oh, no! Engineering just bought three more PCs and none of them are compatible with each other! Accounting just got another PC and it came with DOS, but no manuals with it. That PC over in Production and Inventory Control has a bogus copy of some software. Everybody is treating software like freebies around here. What do we do now?

This was commonplace when I first came to this Fortune 400 company. Everybody wanted a PC, but nobody was coordinating their purchases. Each department would independently go out and buy what they perceived as useful for them. No consideration was given towards compatibility, standardized software choices, future needs or cost effectiveness. So, how do you get a handle on a scene like this?

This paper will discuss the process of developing sound PC policies and procedures as well as planning for the future of the company. The pitfalls are numerous and poor judgement now could cost tens or hundreds of thousands in the future. Proper 'reading' of the trends in PC developments becomes a critical point in determining what directions your company should take. Moving into the 90's isn't simply the changing of calendars; it requires a certain amount of predicting what will be needed to remain competitive. This includes the appropriate PC hardware and software combinations to make your employees the most productive possible.

Other topics covered in this paper include phases of implementation for standardization, enforcing policies and procedures, educating the users, training and shopping for the best deals.

Policies and Procedures

Developing sound policies and procedures requires determining what is best for your company. No set of policies or procedures can be expected to last forever, however, without them, chaos will prevail. Getting started is just like any other monumental task: break it down into small pieces that are more manageable.

Policies

Start with the policies since they are what drive the actual procedures. Policies for when to purchase, what to purchase and should the company purchase at all come to mind as a good beginning. Once the policies have been determined, remember to keep them general as all good policies should be. A certain generality in policies allows flexibility in decision-making (just like our Constitution).

The final set of policies should be kept at a central location with someone who can interpret them clearly. Some companies like to give a copy to department managers and hand out a brief synopsis of their content to all employees, answering questions as they arise. All new employees should be apprised of these policies as well as all other company policies.

Responsible Authority

First, establish the authority for controlling PCs and software, their justification, purpose, acquisition, maintenance and installation. This authority will be responsible for establishing the procedures and implementing them. They must have enough authority to not only set them up, but also enforce them. Without the teeth to keep the policies followed, they will serve no true purpose.

The authority will traditionally be a specific department, but some companies prefer a separate or new department to tackle this vast responsibility. In either case, the department must be prepared to delve into the world of support for all other departments in the company. Such tasks as answering questions and recommending specifications will be only a small part of supporting the entire company's fleet of PCs.

Lease vs. Purchase

Cover the decisions concerning whether to purchase or lease. This is really an accounting issue. The main difference is the depreciation write-offs versus the lease write-offs. This should be decided based on the nature of the business. Which method would prove to be more beneficial to the company? Keep in mind that it could remain flexible, to be determined on an individual basis. It does not have to be only one way or the other.

When to Purchase

Next, work on the when to purchase issue. When is a PC needed and when is it just a preference or status symbol? In looking at the needs of the personnel, justification for a PC purchase is more than just the cost of the PC. There's software, maintenance, training, learning curves and perceived usefulness barriers.

The average middle manager makes about $40,000 a year. The average learning curve to productivity is eight weeks. A 1 week training course including DOS basics, word processing, spreadsheet and database will cost about $1,000. To calculate the approximate cost for putting a manager on a PC, figure you spend $1,000 to train him, get no productivity during the week of training, and get low productivity for 7 weeks at $1,000 per week of salary and benefits comes to $9,000.

Add in the hardware costs of $2,500 and software costs of $1,000 and you have $12,000. All this expense to put a manager on a PC. If you are lucky, the manager already knows how to use a PC (becoming more commonplace these days, especially with younger managers). That would save the company $2,000 to $9,000 depending on whether the manager will use familiar software.

Most companies consider secretaries as the logical people to have PCs. After all, they are the ones who do most of the word processing. But isn't a PC more than just a word processor? What about spreadsheets, quotation systems, CAD, graphics and other applications. Isn't it a bit of a waste to buy PCs to be used solely as word processors? In some cases, yes, others not. If a secretary can meld the use of word processing and graphics successfully, along with a calendar package and possibly a client list database, the PC can become a powerful tool in their hands.

The real people that should have PCs performing much of their work are managers. The people who manage the business and the people are the ones who can get the most productivity out of computers. This would include accountants, sales staff, shop floor foremen, supervisors and engineers. These are not the only ones, either. Many of the staff that think their jobs would not lend itself to using PCs are overlooking the opportunities for producing better quality results.

These people are needed to bring the company from an average operation to an outstanding "class" act. Decisions being made by these people are critical to business; when to buy more stock, who to assign tasks or projects to, where to store the merchandise, etc. While your company may already have application systems making these decisions, aren't there other tasks that can be performed more efficiently with a PC? A tickler system is always helpful for the busy person. Memorandums to one's self concerning ideas, observations and understandings can make the difference between a mediocre manager and an outstanding one.

Justifying the PC quickly becomes more than just the money issues. Ask yourself: Can the company afford not to supply PCs to the key people? A small company may not be able to justify PCs, but larger companies must seriously consider whether they can contend with their competition efficiently without them. Remember, the same basic needs can be satisfied on the HP3000 with a little work and the right applications. It may not be able to efficiently produce fancy graphics and pretty color screens with windows emulation and pop-up menus. However, the fundamentals can be addressed if needed.

State in the policy that a purchasing justification must be filed and approved before any PCs are to be acquired. This form will be produced by the responsible PC control department. It must be maintained to include reasons for requiring a PC to accomplish the work load.

What to Purchase

Determining what will be the standards of the company is not an easy task. Hardware and software trends will be discussed in more detail later in this paper. Keep in mind that there are quality clones available out there and that name brands for whole PCs can become a large capital trap. However, there's something to be said for sticking with one name brand as opposed to several or a wide variety of clones. Beware that there are some clone components that are not up to snuff when it comes to certain software packages.

Specify that a compiled list of specifications will be maintained by the PC coordinating department. This list will be the guidelines by which all PC purchases will be made. It will remain the authority for selection of features and it must be maintained to include any technological advances that have been proven to be acceptable and reliable.

Other Policy Issues

Only the basics have been covered so far. Other issues concern software, security and property. These are essential to properly controlling purchases and compatibility.

Software

Specify that a list of acceptable standard applications will be published by the responsible PC control department. It must list specific software applications that meet the company's needs and are compatible with other company standard applications. By specifying the manufacturers, you lend all your PC users the compatibility for easily swapping files and information. Be sure to select for word processing, spreadsheet, database and graphics applications. Others might include CAD, calendar, mailbox, desktop publishing and project management.

Since there will probably already be existing packages which are not compatible, include a grandfather clause to allow their continued operation. Be sure, though to demand that all PC software is registered and licensed properly. Make it clear that bootlegging is illegal.

Do not allow any software from home (data is OK, but not software) to be loaded on any company machines. Many times these packages are either bootleg, copied off a bullitin board, or copied from a friend. In any case, they could contain viruses or could destroy company information. Do not allow copying of licensed software packages to "take home just for tonight". When your company bought the software packages, it was with the express agreement that it would be copied only for back-up purposes.

The enforcement of these software policies is tougher than composing them and this will be discussed later in this paper. In any case they should be general, but specific enough to preclude any violations due to loopholes.

Security

Securing company information is also important. The company's competitors would love to get ahold of your financials, shipping schedules, customer list, etc. Most PCs have a key lock option. Pay extra for it if you have to. Require that any PC which contains sensitive or confidential information be locked at night and on weekends. Even better, if the information can be stored onto floppies or tapes and kept in locked cabinets it will be safer.

Require a list of employees who are authorized to carry data home or off company property. Require that employees notify their supervisor when they intend to do so and have security staff require them to sign out the data when leaving. Require that computers which are to hold sensitive information be identified and when possible, put them in a secure area inaccessible to visitors or unauthorized personnel.

Make it a policy that any employee violating any of these policies is subject to penalties or possible termination. Make it clear that these policies will be strictly enforced. Make it clear that ignorance of these policies is not an excuse. Keep in mind when making these that the tighter the security, the more difficult it is on the staff. Ensure security, but without impinging on the ease of access for authorized employees. There is a fine line between safety and security that must be measured carefully to avoid frustration over access.

Property

Require that all departments purchase all PCs and software through a central source or only with a central source's approval. This way, control can be achieved for compatibility and compliance with policies. Specify that without this approval, no PC capital expenditures will be approved. Be sure to get administrative backing for this policy.

Make it a policy that no hardware is to leave company property without express written intention in advance. Require some type of materials issue form and require it to be used when any equipment is being removed from the premises.

Finally, require procedures for disasters, equipment care, breakdown notification procedures and supplies request procedures. These may seem trivial, but defining these matters ahead of time can save grief later. Make them sensible and easy to follow.

Retirement Issues

One of the issues rarely discussed when establishing PC policies is what to do with one when it becomes obsolete or is replaced with a newer one. This issue remains obscure in many companies. The main idea to keep in mind involves determining the usefulness of the older technology. At one time, it was the state of the art. Now, it is too slow to serve the increased needs of the user.

The objective here is to establish some guidelines concerning the displacement. The ultimate authority should be with the PC control department to decide what will happen to the hardware. Possible choices include shared PCs, network servers, network slaves, monitoring duties, charitible gifts and sales to employees. Don't be too specific about what to do, just outline that it should be used or disposed of in the best interest of the company.

Procedures

Now, given the above policies, develop a sound set of procedures to implement and enforce them. Here are some suggested guidelines.

PC Justification

Build a PC justification form with room for emotional as well as factual statements. Keep in mind that not all equipment can be justified monetarily and in concrete factual form. Most statements will include one or more items from the following list:

Reduce Waste (Efficiency)

Decrease workload
Reduce overtime
Free Skilled people from routine work
Increase computer efficiency/productivity
Provide better resources with which to make decisions

Better Information (Accuracy)

Increase accuracy of input/output
Improve distribution of data
Increase flexibility of processing
Increase capacity of equipment

Reduce Cost (Economy)

Eliminates redundant or unnecessary operations
Reduce size or quantity of equipment used
Reduce manhour requirements
Reduce number of reports produced
Reduce distribution of reports

More Current Information (Timeliness)

Decrease throughput time
Reduce processing turnaround time
Reduce distribution/transmission times
Provide more frequent reports

Require one of these forms to be filled out prior to submission for approval. Also include the exact and specific applications known that the PC will be used for at present and in the future. Make sure it is clear that the PC must be capable of performing the desired applications. No unrealistic expectations should be harbored for the simple appeal of idealism.

PC Purchasing

Deciding what to purchase is the next step in establishing procedures. There are many choices in the world of PCs, but the main objective is to meet the company's needs with the best possible match-up of hardware with the user and their applications in mind. Depending on what policies are set for purchases, set up a chart for selecting the features necessary to accomplish the given tasks.

Concentrate on specifying the actual internals such as the processor, the bus, the motherboard, the hard disk size, etc. Use the following check list for completeness. Remember that decisions being made now will affect your business as much as five years down the road. Most businesses are depreciating/leasing on a three or four year basis, deciding that hardware will change rapidly over the near term.

Motherboard - (number of slots, bus compatibility)
CPU - (286,386,?)
Clock Speed - (10,12,16,20,25,33,? MHz)
Coprocessor - (287,387,?)

Memory - (expanded,extended),(1M,2M,3M,4M,?)
Ports - (serial,parallel),(1,2,3,?)
Graphics controller card - (MGA,CGA,EGA,VGA,?)
Floppy Drive - (DD,HD),(360K,1.2M,740K,1.44M)
Hard Drive - (10,20,30,40,80,?),(seek times)
Mouse - (serial,bus),(physical,infra-red)
Modem - (1200,2400,4800,?),(auto-answer,auto-dial,memory),
(internal,external)
Monitor - (color,mono),(low-res,hi-res)
Keyboard - (standard,extended)
Printer - (graphics,dot matrix,daisy wheel,spray)
Peripherals - (graphics tablet,bar code reader,scanner,?)

Create another form for ordering a PC and associated software. Include the check items as listed in the charts, allow some room for flexibility and leave room for special attachments. Ensure that the PC meets the requirements outlined by the department purchasing the PC. Make sure that it is powerful enough for the applications desired, but not overpowered.

You wouldn't use an 8088 for a CAD application, but you also wouldn't use a 386-25 for a word processor only PC. Ensure that the applications requested are available on the PC (as opposed to only available on the Apple MacIntosh). This will happen. Users will say they want this package and it won't be available for the IBM compatible.

Software

Test and select specific software packages from specific manufacturers. Develop a list of these applications. This list will be published and distributed to all departments. If someone requests WordPerfect® and your company policy specifies Microsoft Word®, for example, this should be noted, and when possible, discouraged. Not every department will use the same brand packages, but make sure that files can be transferred and converted to each other's formats.

Also, be sure that the packages requested will be able to perform the tasks that the department wishes to accomplish. Software is like a tool: never use the wrong package for the wrong purpose. It is a waste of company resources. Finally, make sure that the requested packages are available in a timeframe suitable to the department. If it won't be available for six months, it probably won't help the department now. Even though it promises to make coffee each morning, it may never be as good as it sounds (pre-announced software or "vaporware").

Security

Procedures for security involve primarily monitoring PC user's activities. Making sure they comply with policies such as using the key to lock their PC is easy. Moving sensitive PC operations into secured rooms is easy. Copying sensitive data to media and storing it away could become tiring and employees may tend to be lax in following procedures by not doing it or not erasing files on their PCs after doing it.

Acquire a list of employees who can carry home company data. Post this list and give a copy of it to security. Take precautions as necessary without putting too much burden on employees because they may refrain from taking work home if it's too much trouble for them. The same goes for taking hardware home. Enforcing bootleg policies will be discussed later in this paper.

Property

Make a set of procedures to follow in the event of a disaster, breakdown, equipment care and supplies requests. Disaster procedures should be posted in a central location, updated as conditions change and be distributable when a disaster occurs. Nothing is worse in a disaster than confusion and uncertainty about what to do.

It is often desirable to report equipment breakdowns to a central location for coordination of repairs and replacement. This is especially important in large companies. Bigger discounts are usually given when a maintenance agreement can be worked out for all equipment as opposed to piecemeal repair calls. Sometimes, repair is not necessarily needed and an on-site technician can solve the problem without calling in an outside repair service.

Equipment care procedures should include a schedule of when to clean and when to replace consumables. Additionally, rules about keeping liquids, small objects and cigarettes away from equipment should be included. Rules about movement of equipment should specify that a technician should move equipment or be available in case of problems.

Supplies ordering procedures should include an order blank sufficient to specify quantities and when needed. Centralized purchase and/or storage of these items will greatly quantify available discounts and provide for timely delivery of supplies. Specify advance order lead times for hard-to-get items.

PC Replacements

Many companies make the mistake of shuffling off slower, older technology computers to the secretarial and clerical staff when new upgrades are made. If this is an improvement over their past working tools (i.e. a typewriter), this may be suitable. However, if the secretary already has valuable computer skills, it could be a major mistake. Many companies are finding other ways to "retire" old technology.

Some are using them as spare stock, to be shared by staff that don't need a full-time computer. Others are using them as network servers. Still others are sending them to field offices. Finally, some give them away to charitable organizations as a write-off. In any case, many don't see any advantage with pawning them off on clerical staff when that staff could have more advanced technology which will make them more productive.

Reading the Trends in PCs

Keeping up with the trends in PC usage is a formidable task at best. To understand the problem, walk along the path of history to see what has occurred in the phenomenon of personal computers.

History of Trends

In the late seventies, small home computers began springing up to meet new consumer demand for the fascination of these miracle devices. These were simple computers, able to perform simplistic tasks with relative ease for the amateur. A plethora of companies were created overnight to take advantage of this demand, with little thought and even less agreement about any standards within the industry. Each manufacturer had their own architecture, operating system, storage methods and user interfaces.

Some of these manufacturers began settling on a common operating system for limited compatibility, but that is where the similarity ended. Then, just as these "toys" began to become less of a novelty and more serious, IBM became interested in providing these newer style machines to their large corporate customers. They even thought of it as a limited market, grossly underestimating the perceived advantages and quantities that would be demanded.

Because of the new "standard" established, many other manufacturers began attempting to copy the giant's machine to make new-found profits. The IBM PC soon became the recognized standard for corporate America. Within a short period of time, millions were sold and installed within businesses ranging from a Mom & Pop store to the major conglomerates. Everyone was sure the only standard was set. IBM, though, couldn't stand allowing any of the profits to be going to other companies when they had set the standard.

Thus was born the PS/2. The architecture of this machine was closely guarded and must be licensed to be copied. A monopoly on the market! Who could ask for more? Well, many consuming companies thought this was a dirty trick and decided that the old architecture was good enough to keep. A consortium was formed and the Extended Industry Standard Architecture (EISA) was born. It portended to be the alternate path for those who would stay with the old guard. An extension of the old architecture, the EISA promised to allow much of the new functionality of the PS/2 without compromising the stock of add-in boards already available.

Then there's the software side of the issue. In the early days, there was a congealing of opinions focusing on the CP/M operating system. It could easily have been the defacto standard, except IBM chose instead to use Microsoft's DOS as the operating system for their new PC. This set the new standard, leaving the CP/M operating system to slowly become a has-been in the race for the standard. It quickly became obvious that the future of operating systems on PCs would be dictated by IBM.

When the PS/2 became IBM's new standard, along with it came a new operating system to handle the new features and functionality available with the new architecture. Thus was born the OS/2 operating system to match the hardware. This new operating system would have many new features built into it including a fancy graphics-based user interface, multi-tasking and it taking advantage of the higher speed and more sophisticated processors available.

Some say that EISA will not survive, others say that PS/2 won't be practical. Still others say that neither is as good as some existing workstations. The decisions concerning which architecture to choose are not easy. When reading the literature, articles appear concerning the issues, but many of them are non-conclusive or strictly opinion based. Reading more articles only clouds the vision, creating more confusion.

Discovering Trends

To formulate an opinion about which path to take could mean sudden death for the company or individual. With such pressures bearing weight on the shoulders, which way should be taken? Relax, trends are usually slow to catch on and even slower to become the standard. In this case, however, most companies have too much invested to follow any trend instantly. Most will take on new technology as capital is available and the needs arise. The phenomenon of an exploding population of PCs is nearing an end.

To determine such sage wisdom, turn to the magazines and periodicals available. Everyone is talking about the advances in technology, but few companies try to keep up with every new piece of hardware or software. It is the job of these media to report new things, review them and offer advice for what to do with them. They must stay on the cutting edge to remain competitive in their own market segment. After all, if they didn't, they would lose advertisers and readership.

So, how does one keep up with the trends? Reading helps, but most of the better magazines are saying the same things. Here is a sample of what is being said in various publications.

"OS/2 is being threatened by both UNIX and Apple's MacIntosh." (1)

"MS-DOS [is] currently being used by 25 million people" (2)

"Many applications slated to run under OS/2 will not take full advantage of the hardware or Presentation Manager for several years. ...users who are not satisfied with early Presentation Manager [applications] may turn to the cheaper Microsoft Windows-based environments." (3)

"If one considers how many PC-compatibles and PC/AT-compatibles have been sold since the PS/2 series was introduced, the older architecture is obviously still the technology of choice." (4)

"OS/2 is still mired in high costs and little value for the PC user." (5)

See a pattern beginning to emerge? Most articles speak about the new technology, but few are willing to commit to endearing it for today's needs. Few see the PS/2 or OS/2 as being the panacea for computing woes of the average or even above-average users. The availability of applications that take true advantage of the new architecture and its' features was cited most as being a good reason for taking a wait-and-see attitude.

Since most of the media agree on the state of the art, one should settle on a single or dual source of information to avoid any confusion or diluting of the issues. There are four or five quality publications available that all say about the same thing. Choose one or two of these, read them religiously and heed the advice and warnings given by them.

Many expect that some OS/2 applications will be available in 1990, but don't look for much more than the major packages until 1991 or 1992. So, what does this mean? Software has been slower in developing than hardware because of the element of time. Time to think of new techniques for making fast applications, take advantage of architectural features and meet user expectations simultaneously. These tasks are easier talked about than conceived and implemented.

The primary reason for this is that users are becoming more difficult to please as the applications become more sophisticated and user friendly. Thus, more complex software must be produced to meet the demands and naturally, the more complex, the more problems involved. Longer development lead times, longer testing periods and longer packaging and marketing preparation is needed to turn out a successful product. After all, the stakes get higher as the competition grows tougher.

What it all boils down to is that most are recommending to stick with the older technology until the market offers more advanced applications which take advantage of the newer architecture. This time-frame appears to be coming around mid to late 1990 at the earliest. Until then, the hardware and software will only be able to depreciate and run the old applications. Besides, why retire hardware that may have a few more years left in it?

The new OS/2 operating system is still young and immature, offering little advantage at this time over the old DOS. The new one requires at least 2 MBytes of RAM with a recommended minimum of 4 MBytes. It has difficulty running smoothly at any configuration less than this. Another consideration to keep in mind is that the OS/2 operating system requires about 10 MBytes of hard disk space for storage. This is substantial considering the small increase in operating systems advances.

Stick with the old systems until such time as the new ones are proven to have a distinct advantage over the old. Nobody is going to lose out because they aren't state of the art. More precisely, the new systems won't be able to do anything more than the old ones except maybe do it faster and fancier. After all, who needs to keep up with technology just for the sake of keeping up with technology.

But What About EISA?

The EISA architecture is being introduced later this year and should prove to be interesting. It is promising to be the alternate answer to IBM's Micro Channel Architecture (MCA) used with their new PS/2 machines. Since it is an extension of the older PC architecture, it will offer more features than them. Unfortunately, by retaining the same drawbacks inherent in the older PCs, it doesn't seem to offer much improvement.

Here's what some authors have to offer about it:

"EISA ... is at best a step between the old AT architecture and the Micro Channel Architecture. Because of its limited bus speeds, it will lack the ability to transfer data at rates sufficient to satisfy power hungry applications." (6)

"Personally, I'll be surprised if it [EISA] ever sees the light of day. If it does, it will be crushed in the market." (7)

"Now comes EISA, replete with the hidden message that the machines [PC architecture], if not the boards, you are buying today are obsolescent. It reminds of something we heard from the folks at IBM... How well did it work for them?" (8)

As can be seen from this small sample, not many people are overly excited with the EISA machines. Admittedly, however, most reserve their opinions until the machines begin rolling off the assembly line and into the testing labs. Until then, nobody really knows what their performance will be. One thing is certainly clear, though, and that is the machine will be 100% upward compatible with the old PC architecture. These machines could prove to be useful in filling the gap between the old and the new.

Hardware and Software Choices

To determine what to standardize on for the company, look at what is currently being utilized. Are there a considerable number of old 8086 based machines? Or are there mostly 80286 and up processors? Do they typically have a hard disk or are many of them floppy based? Determine what the current trend is within the company in order to decide where to aim for the future. This is the base to start with when discussing considerations.

Hard Decisions

The current recommendations for new hardware concern the purpose for which it will be used. If the applications are number and interface intensive, the higher end 80386 processor could be called for. However, if it will only be used for word processing and small spreadsheets, the 80286 (at least 10 MHz) would easily suffice. Some number-crunching applications may get by with the 80286 and a 80287 co-processor.

For a network server, however, get a high end 80386 or a high end PS/2. Preferably, using the PS/2, preparation for the future is already in place. In either case, OS/2 would give the distinct advantage of being prepared for the vast memory addressing needs required with a network. Further, it is far more capable of providing a spooling environment suitable for accomodating multiple output devices to be shared by the network nodes. The server should contain at least 8 MBytes of RAM to allow room for the buffering necessary to keep a network operating smoothly and quickly.

Memory for an 80286 model should start with at least 1 MByte of RAM to accommodate the network requirements. Be sure that the 384 KBytes of extra memory are Expanded memory to let the network use it to monitor the communications. Higher usage machines should contain at least 2 MBytes of RAM to allow most of the popular applications to be run. Increasing this to 4 MBytes will allow a Windows® environment and popular applications as well as the network software to operate in a robust manner. For stand-alone PCs, consider the applications that will be run and base the RAM requirements on the recommendations of the particular applications.

The hard disk requirements would dictate that at least a 40 MByte capacity should be recommended. With the prices of storage still falling, a larger one could be justified. The server should have at least a 120 MByte hard disk with a preferable 250 or more to accommodate considerable file storage for the network. For a floppy drive, settle on either the 1.2 MByte 5 1/4" or the 1.44 MByte 3 1/2" drive. Don't mix them too much, though, or compatibility problems will arise. In either case, have one available with both drives for transferring data and programs from one to the other.

Monitors come in three basic flavors. There's the monochrome graphics, the color graphics and the high resolution graphics monitors and adapter cards. For simple word processing and spreadsheeting, the monochrome graphics adapters will probably suffice. For more advanced applications such as graphics production, CAD and use of graphics user interfaces (e.g. Windows® or Excel®) the color or high resolution monitors and adapters will probably be more satisfactory. In either case, base the decision on the types of applications and suitability of the adapter and monitor based on recommendations of the applications themselves.

Other hardware attachments and boards should be selected based on individual and application requirements. A mouse can prove to be extremely productive, especially if the application is oriented towards it. Icons and graphics based applications are particularly productive when combined with a mouse interface. When selecting a mouse, consider using a bus type to facilitate independent interface handling as opposed to a serial mouse which requires more system interaction and slower response with the slower clock speeds and processors. On the faster systems, a serial mouse is acceptable since most of the interaction will rarely outrun the processor.

Printer requirements vary as much as users needs do. It is not generally economical to justify one laser printer per each PC. Usually, no one user can keep a laser printer busy. Consider a network or printer sharing device for sharing the more expensive printers. Less expensive printers can be strategically located on more user's PCs to facilitate internal memos and draft copies of documents to be later produced on a laser or letter quality printer.

Modems, graphics tablets, specialty boards and secondary floppy drives should be considered on an individual installation basis. These add-ons are not necessarily needed for all PCs, but could be used by some of them. Keep the computer as basic as possible but fulfill the needs of the user involved.

Softer Decisions

Software choices are probably even more difficult to make than the hardware decisions. There are literally thousands of applications to choose from in the commercial sector and thousands more in the personal sector. Which ones are the best suited for the purpose at hand? Are there more than one that would fulfill the need? How does one select which packages to purchase? These questions and more arise when attempting to analyze the needs and meet the requirements of the users involved.

The operating system of choice for 1989 would have to be the MS DOS system. It is still quite functional for the current set of applications available for the PC. When more applications take advantage of the OS/2 environment, it will become the operating system of choice. Until then, unless the PC has 4 MBytes or more of RAM, forget it. This status will be changing as 1989 wears on and the 1990's approach. Most companies, however, will retain the substantial investment placed in the old operating system until the perceived advantages become more realistic and viable.

There are many directions to follow when considering application possibilities. The single most productive approach to take appears to be the Windows® environment. This environment allows multiple applications to be running while the user flips back and forth between them. It also allows sharing information, graphics and data between them in a transparent manner. If the applications are designed to run under windows, they will accommodate maximum productivity for the user. The graphics symbols used allow ease of pointing with a mouse and clicking to cause actions. This interface facilitates a more natural interaction with the programs and produces satisfactory results in a more timely manner than without it.

When considering a new installation, strongly consider the Windows® environment and compatible applications. There are more and more Windows® applications available each month. Many of these take full advantage of the environment and allow ease of passing objects between them. The word processing applications that work in this environment allow intensive use of icons and graphics symbols to facilitate selection of options and actions appropriate for providing fully featured documents and letters.

Spreadsheets come in two flavors: those that work in the Windows® environment and those that don't. The ones that operate under Windows® are graphics based with icons and pop-up menus for simplistic user interface. These provide a much more productive environment for the user to work under than the old keyboard intensive programs. The user learns to coordinate their hand motions with the view of the pointer on the monitor. By not needing to look at the keyboard, the user makes changes with less movements and fewer wasted steps.

Graphics software comes in many styles. Beware of the way vendors use the word "graphics". Four basic types of packages are available: Presentation Graphics, Freehand Graphics, Vector Graphics and Drawing Graphics. Presentation Graphics usually consists of graphs, charts and symbols prevalent in business presentations. Freehand Graphics allows "bit-mapped" drawings of objects and symbols. Vector Graphics is math oriented allowing the drawing of circles, rectangles, curves and other manipulatible formula drawings. Drawing Graphics consists of drafting quality, engineering and design drawings.

Select the type of graphics that best suits the application needs of the user or department. Marketing may need a combination of Presentation and Freehand Graphics while engineering may need Vector and Drawing Graphics. Still another department may need Presentation Graphics alone. Keep the needs separate from the wants and desires of users because most will say they need one thing when they need less. Make the department justify the need for an expensive package. Keep in mind that many graphics packages can be used to produce slides and slide shows which may be sent out for final production or produced on the monitor for that live action feel.

Other software needs may vary from utility packages to add-on packages to enhance the current stable of software. Keep in mind that if an HP3000 connection is desired, a good terminal emulation package will be necessary. There are some relatively inexpensive but functional communications packages and there are some expensive and extensive emulation packages also. If the user needs the full features of the extended package, by all means, provide it. However, if the user only needs to carry a session and interact with the HP3000, a simpler package should suffice.

When setting the software standards, consider the currently installed base. Do not attempt to obsolete the old applications immediately. The users would become angry and it would cause more disgruntled employees than necessary. Rather, set the standard and encourage users to explore the advantages of each application as it becomes available. Future purchases should include the packages that have been set as standard recommendations. This set of standard applications does not necessarily need to be limited to one package in each category. In fact, allowing a selection of two or more would more likely be acceptable to all departments involved.

Finally, keep in mind that forcing standards on users is difficult and can only cause consternation. Allowing feedback from users concerning their perceived ideas on applications and their usefulness can reap valuable insights into the actual needs. Keep a log of requested applications, logging each request for each package. Keep a log of which users have which applications and their level of expertise for these applications. They may become useful in determining application selections for future purchases and be used as references when recommending new software or training new users for a particular package.

What Does it All Add Up To?

The primary question should be: Is the company ready to invest in the future with hopes of seeing results in a few years or does the company need the increased productivity of what is currently available? The answer is not a simple one. If the company stays with current software, the invested learning curve will be difficult to overcome when changing to new standards in the future. However, if all new applications are introduced at once, confusion and disarray will prevail.

The sensible approach is to stick with what is already proven and gradually migrate to newer technology as it becomes available and practical to introduce. This route allows room for growth without jeopardizing the company's productivity. The eventuality of advancing technology dictates that it will occur anyway. Why not take it as it comes and in stride?

Phases of Implementation

So far, the discussion has concerned preparation for implementing sound PC policies and procedures. Now, the focus will turn to actually putting them into practice. The preliminary work is completed (keep in mind that policies and procedures will need reviews and updates periodically) and it is time to proceed with the business of controlling and monitoring PC and software purchases, installation and maintenance.

To implement all aspects of the policies and procedures at once would only cause chaos and misunderstandings. Instead, implement in phases to allow user adjustment to the new ways of accomplishing goals. The following guidelines are for a stratified implementation of policies and procedures.

Purchases

The most logical place to start with is in the purchasing process. Armed with the new standards, start using procedures for final approval/criticism of PC and software purchases. Be sure that the new items are truly needed and that there are not existing solutions already in place for the given problems. Encourage compliance with standards, allowing little deviation from the norm. Once departments become aware that there are new procedures for the acquisitions, they will begin utilizing the new resource as an advising group for future decisions.

Keep in mind that some departments may already have their own standards and that they may resist corporate influence. This can be resolved, however, through pleasant communications and gentle reminders about the approval process. Educating the departments about the standards becomes tantamount to proper implementation of policies. Make them aware of the advantages concerning central control of these massive and diverse capital resources. Keep them informed about the advances in technology as well as any changes in policies and procedures.

Adherence to common interests promotes the concept of eventual corporate alignment for standards and commonality. To be most effective in becoming competitive, the company must be able to communicate ideas, prospects, information, data and concepts easily and readily amongst the departments and remote locations. This is best accomplished by setting standards and complying with them consistently. Corporate standards make this possible even if their implementation is difficult.

Security

Issue a memorandum concerning the security issues involved with the use of PCs and various software in the office environment. Make the users aware of possible security breaches and the dangers of introducing non-standard software packages. If they are aware of these things, they can be responsible for their actions when ignored. Keep them informed about any problems that have been encountered in other departments and give them tips on avoiding these pitfalls.

Work with each department in establishing data access authority lists, security priorities and potential losses of data or machines. Prepare a consistent backup plan for each department and encourage compliance to avoid unnecessary time loss. Keep the security actions within practical limits to prevent any loss of enthusiasm in job performance.

Installation Practices

When a new PC arrives, ensure that an experienced technician actually sets the computer up for the user. More problems with new installations are the result of sloppy setups than any other single factor. Pay close attention to software being installed for special instructions concerning use of extra memory, mice, special video adapters, etc. These factors can mean the difference between an application just running and an application running well.

Inform the user about safety practices, re-boot procedures, path navigation and problem reporting procedures. Let the user know that questions should be routed to the support desk. Keep them on a list of supported installations so that they are given priority when their calls come in. Offer assistance for start-up and step-through of applications and operating system utilities. Keep them confident that support is only a phone call away. This level of support will assure them that there is nothing to be afraid of and that help is always available.

Maintenance and Care

The next important step in implementing sound PC procedures is to offer some type of hardware support. An on-hand technician should be available to check out problems before making a maintenance call. Several types of maintenance contracts are available for PCs. Some offer on-site 4 hour response, others offer on-site next day response. Some allow on-site once a week response and still others offer off-site support of varying degrees. Select the type which is most suitable and affordable to the company.

Before maintenance can be practical and useful, the care and feeding of PCs should be established. Schedules for cleaning (including floppy disks and tape drives) should be drawn up and posted or distributed to all pertinent personnel. Do's and Don'ts should also be prepared for a general matter and posted as well. Maintain a trouble reporting procedure for all departments and make it available to all concerned.

By establishing a central supplies ordering depot, the company expenditures may be reduced to allow easier ordering and keeping a small stock of supplies. Be sure that everyone knows not to move equipment without having an experienced technician on-hand to resolve any potential technical problems.

Expert Advice

By establishing the department as an authority on PCs and the associated software, the standard is set for providing the central problem resolution site. The importance of supporting installations is substantial and requires a certain amount of expertise in PC operation. Developing quality personnel for this task takes time and patience. Experience is a good teacher, but technical courses make for better learning.

The department may not be able to offer this level of support, but may instead contract an outside agency to handle it. Offering training classes or referral to training resources can also establish a good foundation for support and understanding. Each department will probably develop its' own expert among their user staff. This usually occurs in medium to large companies but is not uncommon in smaller companies also. As more users become trained, less support will be necessary and the departmental resources can be reorganized to focus on other areas of concern.

Enforcing Policies and Procedures

Probably the most difficult task ahead is the enforcement of the newly established policies and procedures. Just like state and federal laws, if the policies have no teeth, they will go mainly unheeded and chaos remains. If they are not enforced sufficiently, nobody will take them seriously and nothing significant has been accomplished. The following guidelines are merely one means of aligning personnel to follow practices closely.

Software Piracy

The single most difficult policy to enforce is the one concerning illegal reproduction of licensed software. This problem is caused mostly by the misconception of personnel that software is a freely reproducible commodity. When purchased, it comes on a floppy with a license agreement which is rarely read by the consumer. Most people believe that it is simply OK to copy it onto as many machines as is convenient.

While this view is most prevalent in society today, it is strictly illegal. If a person within the company proliferates a single licensed copy of software and the company does nothing about it, the company is guilty of copyright infringements as well as the perpetrator. Thus, the company should take a hard-line stance against such behavior. The policy does well to state that position, but enforcing it is another issue.

There are several techniques to prevent such activities. First, make the policy against it known to all affected personnel. Second, make it known that violations of the policy may result in disciplinary action up to and including possible termination. Publish a memo stating that all illegal software should be removed from all currently installed PCs immediately. Specify that any company data should be transferred to licensed software packages before taking such action. Third, let it be known that random audits will be taken of all PCs and if any un-licensed software is located on a specific machine, the user will be held responsible. Fourth, establish a list of applications that are legally licensed and installed on each PC. This reduces any disputes as to legal possession of any software. Fifth, establish a central deposit for all original licensed diskettes.

Next, keep a list of all authorized users for each installed PC for reference when an audit is taken. This dispells any disputes concerning who uses each PC. By keeping this rigidly established, a major reduction in pirating software will occur. If the users are well enough informed about the illegal aspects of performing such acts, they will soon learn to honor it.

Finally, establish a list of known programs to be associated with each software application for reference when performing these audits. Establish a rapport with the company's software supplier to allow evaluation copies of software to avoid any attempted illegal copying. By making this resource known, the users will learn to acquire software legally and without any possible risk.

Unauthorized Software

The next most difficult policy to enforce will probably be keeping users from bringing software from home. This is easily accomplished by the user and without the knowledge of company personnel. This is most dangerous to the company in that it does not prevent a stray virus from invading company PCs. All that is needed to bring in a virus is to load a program and run it. Even if the program is then removed, contamination may have already taken place. Let users know that this practice is strictly forbidden. There is no foolproof method of enforcing this policy.

The best defense against this infringement is to educate the users about the dangers involved. By making the policy common knowledge and promoting healthy practices among the user community, most people will learn tc follow the lead and refrain from introducing unknowns into the environment. Remember, data diskettes will rarely carry any dangerous side-effects, so don't attempt to stop all magnetic media from entering or leaving the premises within the guidelines of security.

Purchasing Control

By requiring each department to justify their PC needs, most of the control is in place. Since most capital spending requests are controlled through one source in most companies, that is the key point to intervene on a purchase. This works well with PC purchases, but does not necessarily control software, add-on and peripheral purchases.

To gain control over these types of purchases, it is best to alert the purchasing department or whoever issues purchase orders to intercept them and re-route them through the controlling department. Another good interception point is in the accounts payable department. If an invoice comes in for software or PC related hardware, re-route them through the controlling department. If both of these fail, detect who the culpable departments are and confront them about it. This technique usually effective for thwarting any incompatible purchases.

Data Security

As previously mentioned, controlling the company's data is important and is the most sensitive point for keeping company integrity. Armed with the authorization lists and maintaining sufficient company security, this should keep itself under control, depending upon the level of security desired to guard against loss of data.

If a person is suspected of divulging company proprietary information, that it is a risk to allow that person near any of the company computers. Some companies go so far as to install magnetic media detectors to thwart anybody from departing with company data. Decide the importance for the company and take sufficient measures to secure what is necessary.

Hardware Security

Other than hardware accessibility, be sure that any PC with a connected modem is not left connected to an outside phone line at all times if sensitive data is located on it. If it must be, secure it with either a dial-back facility or some password access control scheme. Hackers love to face a challenge and PCs offer this challenge when connected to a phone line.

Keeping the hardware can become another problem within the company. If the hardware disappears, it will have to be replaced. Sufficient security measures will prevent any major pieces of equipment from being lost or stolen. Make sure a materials issue form is completed for everything that leaves the premises. A copy of this form should reside in the employee's personnel file such that upon permanent departure from the company, recovery of the hardware issued to that personnel is possible.

Educating the Users

Making the users knowledgeable about PCs and PC usage is key in affecting the productivity of the company. If the users only use their PCs to emulate a terminal, it is probably a waste of resources. If they re-key data from the mainframe onto a spreadsheet or data base on the PC, time and energy is being wasted. If they only use the PC thirty minutes each day, they don't really need one dedicated to them.

The best way to educate the users is to set good examples, offer good advice on usage and offer training alternatives for serious users. If users aren't taking advantage of the available resources, generally, they are either already well-educated or don't want to learn more. If they don't want to learn more, the resource is probably wasted.

Start a company newsletter to keep users informed about policies, procedures, tips, tricks and new software and hardware availability. This newsletter will become a vehicle for promoting good practices and at the same time provide a means of communicating with the users. Encourage users to submit articles, no matter how trivial, to be printed in the newsletter. Offer incentives for the submitting these articles.

The best teachers in the company will come from the ranks of co-workers. They are trusted people working alongside each other and are the most accessible to those who need them. Encourage people who display enthusiasm and nurture this attitude by allowing them to check out manuals and experiment (within reason) with the operating system and application software to discover hidden features and shortcuts.

Another good way to promote a healthy user community is to run a bulletin board for the company. This can serve as a stand-alone or added vehicle for keeping users informed. However, it does not offer the high visibility of a newsletter and should be considered as an added feature as opposed to the only one. Keep in mind that visibility is key in providing quality services.

Maintain a library of manuals and educational books concerning PCs and software applications. This library should be accessible to all employees, allowing them to learn for themselves. Encourage the users to reference the library when a difficulty is encountered. Keep the library as up to date as possible to prevent it from becoming obsolete.

Training that is offered should encompass everything from beginner to advanced, leaving no major gaps uncovered for those mediocre users who are marginal. Encourage participation whether it is company sponsored or not. Offer some type of incentive for those who do participate and be sure to keep the list of available classes appropriate as possible for the company's needs.

Answering questions for users usually provides enough input for them to continue learning. Keep in mind that this may be a new experience for them and they probably want to learn as much as possible to do their jobs better. By learning to be a good listener the technician can become a good teacher. Take the stance that everyone wants to learn and nobody is just being a pest. The users will return for more help as they encounter newer yet higher levels of expertise.

Shopping For the Best Deals

To say that every purchase is made at the best possible price is a falacy in a free enterprise system. Every company is in business to make a profit. If they give the best prices to every customer and always beat the competition, chances are the business will not survive. Thus, every business gives the best possible price while still making a profit.

In the free enterprise system, the lowest price does not necessarily indicate the best value. Something is not equal in quality in most low-price cases. However, shopping for the best deal is still the best way to perform purchasing. Getting quotes and specifications of what is included from competing vendors allows leveraging maximum discounts.

Bargain Hunting for Hardware

If the company has a policy to purchase only specific brands of PCs, then shop for the best deal on that brand. If the company does not have such a policy, try checking out clone dealers. Not all clone dealers are created equal, however, and shopping becomes critical when considering their practices. Consider the quality of services available as well as the company's reputation. Check into references to determine other customer's satisfaction with service and delivery.

Be sure to specify all components when purchasing clones. Some dealers will substitute hardware of "equal value" when a specific component is not in stock. Be as specific as necessary for each component and don't assume that the specifications will be followed religiously. Spot check the dealer to determine their consistency in following specifications. Many will not reveal when they have not followed instructions and assume that the substitutions were acceptable. Above all, make sure that the vendor supplies the licensed operating system software with every PC purchased, including the diskettes and manuals, too.

Software Values

Software vendors are easier to shop for. They will give discounts based on the quantity of purchases and many offer good services to back up their sales. Look for a vendor who will provide evaluation copies of software for user acceptance. Also, look for one that has a liberal return policy. Even when a software package appears to fit the bill it may not work out exactly as planned and being able to return it will save the company money.

Be sure that the vendor stocks most of the applications that the company will be using and that they stock the latest versions. Another key ingredient is the vendor's ability to handle upgrades when they become available. This feature will save time when updating multiple packages. Check to see that the vendor also provides needed services. Such services as software search and recommendation can save countless hours of work for the company.

When looking for the vendor, keep in mind that the company could qualify for quantity discounts. These discounts will save funds and allow leveraging to obtain extra copies for non-planned or future installations. The extra buying power can also enable the company to request the vendor to stock specific packages.

Bibliography

(1)Douglas Barney, "Microsoft Biting its' Nose to Spite its Face", *Computerworld*, 13 Mar. 1989, pp. 39,47.

(2)ibid

(3)Stephen Jones, "All Eyes on Graphics in Application Arena", *Computerworld*, 14 Nov. 1988, p. 134.

(4)Larry Liang, "Performance and Compatibility Keys to Choosing the Right PC", *MISWeek*, 19 Dec. 1988, p. 31.

(5)Jim Seymour, "The /2nd (superscript nd) Anniversary", *PC Magazine*, 11 Apr. 1989, pp. 77-78.

(6)Winn L. Rosch, "The Great Divide - EISA vs. Micro Channel", *PC Magazine*, 27 Dec. 1988, pp. 165-186.

(7)John C. Dvorak, "EISA Come, EISA Go!", *PC Magazine*, 27 Dec. 1988, p. 71.

(8)Bill Machrone, "Computer Buyers Survival Guide", *PC Magazine*, 13 Dec. 1988, p. 65.

auditability and the micro
or
I don't think we're in Kansas anymore, Toto!

robert a. karlin
karlins' korner
7628 Van Noord Ave.
N. Hollywood Ca.
91605

"Is this a Micro I see before me
With its keyboard towards my hand"
mac-beth.

"To key or not to key, that is the question.
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take backups against a sea of troubles,
And by opposing, end them?"

with apologies to Bill the Bard.

INTRODUCTION

As I sit to write this paper, I reflect that the machine that I am keying upon is more powerful than most of the HP3000 line of computers. I have more memory than was considered possible in a mainframe not too many years ago, and my disk drive has more than twice the disk storage capacity of the first machine I programmed in 1971.

And yet, we were responsible, as a service bureau, for the processing of an enormous amount of data. We had a staff of fifteen programmer analysts available to produce any report that our clients could need. Even though our MTBF (Mean Time Between Failures, or in other words, the average time we could expect our machine to continue to run uninterrupted before crashing) was approximately 6 weeks, our clients were assured of continuous service, with reports delivered on time, and the accuracy of the data never in question.

The power of that service bureau now sits on most clerks' desks. And yet, even with an MTBF measured in half years and years, with computer speeds an order of magnitude greater than our service bureau's, major disasters happen daily. Important corporate data is continually lost or altered beyond redemption. Even worse, important business decisions are made on incorrect summaries, and these mistakes will never be uncovered or corrected.

The most tragic aspect of the above tragedies is their preventability. This papers attempts to provide tools, techniques and advice in preventing tragedy and assuring auditability of the PC environment.

HARD DISK BACKUPS

The first and most important area of auditability is the protection of both the current and historical software environments. Data and program errors tend to propagate over time. If an error is not discovered immediately, other errors can be built on the first, until it becomes difficult to spot the original error. Backups can be an important part of the audit trail needed to find these errors.

There are two types of backups. Full backups are backups of everything on your disk. There are two types of full backups, image and file. Image backups backup your hard disk on a sector by sector basis. This type of backup can only be restored as a single unit. File backups backup each file as a discreet entity, and can therefore restore single files from the backup.

Incremental or partial backups backup only those files which have changed since the last backup. If all of your data is kept in a few large files, incremental backups can take as long as full backups.

Most microcomputer users back up their systems exactly once. After their first backup, they never seem to be able to find the time to do so. Micros now tend to be purchased with a standard forty to eighty megabytes of hard disk storage. In order to back this up to floppies, one would need at least twenty to thirty high density floppies (1.44Mb apiece) even with the best of the backup compression algorithms. Backing up to low density floppies would take four times as many disks. A streaming tape unit, however, can cost from $1,000 to $3,000.

One company I know solved the problem for its programming staff by purchasing one tape unit for the department, and installing controller cards in each PC. Once a week, an operator went from machine to machine backing up hard disks.

Another solution can be implemented in offices that have installed local area networks. In these offices, backups can be taken across the LAN. This is feasible only if all local data is accessible to the LAN, and also requires a streaming tape drive of some sort.

These solutions may be feasible in larger offices, but can be difficult to implement in offices where the microcomputers are spread out in a large area, or where is no one who can be assigned the responsibilities of ensuring backups.

If a microcomputer user is responsible for his own backups, there are a few techniques that can be used to help lessen the pain. First, segment your hard disk into at least two drives. The first drive should be as small as possible and still be able to hold all of your software for your applications and utilities. This drive can then be backed up twice (never trust a single backup of anything important) and the backups archived (more about archives later). You should only need to backup this drive if you change software or operating systems.

Second, keep as little data as possible in your active files. In other words, if you can keep only this quarter's data in your current database and copy anything older into another database for reference, your incremental backups can be kept small. If you wish, you can segment your disk again to keep your inactive data separate.

Third, keep all your input data, notes, etc. until after you have backed up your system. Your backups should be scheduled based on the amount of time that it would take you to recover from a system disaster. Remember, if you must recover, you need to be able to reconstruct everything you have entered from the last backup, and this time must be added to the time necessary to restore your system backups as well.

ARCHIVAL STORAGE

It does very little good to backup your system religiously if you keep your backups on the desk beside your microcomputer. The spilled coke that wipes out your hard disk will do wonders for your backup diskettes.

At least one copy of your backups should be kept in a fireproof safe. It would also be a good idea to store the original copies of all of your software in the same place. If you have a secondary site, the safe should be kept in that location. If not, keep it somewhere away from the general hustle and bustle of the office place. The location should be cool and dry, and should not be near any large machinery.

Floppy disks deteriorate rather rapidly. A single diskette is rated at about 300 hours of use. Stored in a safe place, out of both heat and moisture, a floppy can last over a year. But it is still a good idea to copy the data on your archived diskettes about once every six months.

Hard floppies, that is the three and one half inch diskettes, are safer for archival storage than the standard five and a quarter inch diskettes, since they will not bend, and the drive mechanism has less of a chance to warp the diskette (if you place a five and a quarter inch floppy into a drive incorrectly, the disk drive spindle can warp the center hole, making the diskette unreadable). The media used in the diskettes is the same, however, and will still deteriorate in time.

Cartridge tape backups will last slightly longer than floppies if they are kept in a controlled environment. Do not reuse these tapes too often, since they will stretch in time.

It is most important to be aware of the fact that these backups are on magnetic media. I have seen cases where a floppy was placed in a purse with a magnetic clasp, destroying the data on the diskette. Even placing the media on a strong motor such as a desk fan, or air ionizer, can make magnetic media unreadable. Keep your floppies away from your printer, and your telephone (the ringer is a small electromagnet). In fact, anything that moves by electricity is suspect. Also, do not leave magnetic media anyplace where someone may accidentally place a magnetic screwdriver on top of them. And never, never, use refrigerator magnets to hold your diskettes up.

SOFTWARE SELECTION AND AUDITABILITY

The software that you use on your system should be selected with auditability in mind. There are auditability considerations in any software purchase, no matter how small it seems.

First and foremost, purchase all software that you are planning to use in a business environment. Software can and will have bugs. You have no recourse to any form of recompense if software that you have not purchased. destroys valuable data. Even though the software that you buy has large disclaimers papered all over them, many states have implied warranties, and even in those that don't, the courts have awarded damages if it can be shown that the software vendor has been negligent.

Also, by purchasing the software you are also purchasing support from the software vendor. This can be invaluable if you make a mistake that will be difficult to recover from, or if you find yourself in need of something that your package just can not do.

Never download software from bulletin boards. Even the ones that screen their software products often can not identify a virus until it strikes. Most computer viruses are harmless, but there have been a few that will do things like erase your hard disk on alternate Mondays. These viruses can lie dormant for months, until you have contaminated all of your backups, before they activate. Always purchase your software from a reputable source.

As far as types of software to purchase, this depends totally on the application you have in mind. Sometimes the simplest of spreadsheets will fit the needs of your department, other times you will need a complete package designed for a specific task, such as a general ledger or a restaurant management package. Sometimes you will decide to purchase a "4gl" high level language such as dBASE or a relational data base such as Informix. In all these cases, you need to be certain of your software source, and you need to be concerned with the audit trail.

THE AUDIT TRAIL

If you are purchasing a word processing package, it may be enough to know that the package will not write over an existing file without asking you if you wish this to happen. Any audit trail that is needed will have to be produced by you. This may consist of a simple list of files and what they were, and on which backup they can be found. Unless you are keeping the minutes of board meetings, or other files with legal requirements, the audit trail can be very simple.

On the other hand, if you are keeping a general ledger or an accounts payable system, you must establish a very rigid audit trail. All changes to your data must be logged by the software in a transaction log. The software should allow no changes to summary information, such as year to date payable information, without logging to the transaction log. The software should allow no changes to detail information. All modifications should create offsetting detail information instead. Every summary entry should have the date and time of last modification. Every detail entry should contain the date and time of creation. Only with this information can a company ensure the auditability of its books.

One of the most insidious tools available to a microcomputer user is the spreadsheet. Accountants that would be horrified at the thought of pencil ledgers, or using white out on the companies books routinely make changes to spreadsheets that contain the same data and are used for the same purpose. Spreadsheets are an excellent "what if" tool, allowing corporate management to see what changes in cash flow or expenditures would do to their bottom line. On the other hand, keeping corporate books in spreadsheets should be done very carefully, and with great trepidation. Since there is no change control, every version of a spreadsheet should be archived. Numbers on a spreadsheet should not be changed, entries should only be added. All supporting documentation for a spreadsheet should be kept against future audits.

One of the largest problems with spreadsheets, and indeed with other high level comes from propagation error. This occurs when computed summary results from one spreadsheet or 4gl program are fed into subsequent spreadsheets or programs without going back to the original data. An easy example of this can be illustrated as follows:

Spreadsheet one contains the following figures:

	$2
	$1
	$3
	$7
	$9
for a grand total of	$22.

Our next spreadsheet assumes this figure to be 1% of gross earnings, and calculates a figure of $2200.

In actuality, the above figures are rounded from the following:

	$2.35
	$1.45
	$3.45
	$7.40
	$9.30
for a grand total of	$23.95.

This figure gives us a total of $2395 for gross earnings, or in other words, a greater difference by almost five hundred percent over the original spreadsheet. If we were trying to justify this as an expenditure against earnings, we may have just wiped out an entire department.

This type of error is very common when using 4gls and spreadsheets to do ones accounting and budgets, since these types of programs are specifically designed to hide the internal workings of their calculations.

All in all, if you are keeping corporate books on your microcomputer, be certain that you can identify any transactions that have been applied against your data, when they were applied, and why they were applied. Print a hard copy of your master information at regular intervals, and keep it in your safe. If the size of your master list makes it cumbersome to deal with, create the list as a file on diskette, and have it microfiched. There may be a time when all heck breaks loose and the only available copy of your data that can be acquired in a timely fashion will not be in machine readable form, because you will have no machine to read it on.

LANS

The introduction of the LAN into the business office complicates the task of auditability enormously. The interplay of multiple users can cause even a seasoned DP professional to shudder. In order to adequately protect the corporate environment, certain steps are essential.

First, there must be a single person in charge of the LAN. This LAN manager must be responsible for establishing the protocal necessary for multiple users to interact. The allocation of all LAN accessible files is part of his domain, as well as controlling access to all shared files and directories.

Second, all software that runs on the LAN must provide certain basic audit requirements. There must be a record or sector locking facility as part of the LAN structure, and the software that is running must make use of it. This should be physical locking, that is, locking that is enforced by the LAN software, as opposed to logical locking and semaphore schemes that are enforced only by software agreement. There must be some form of either roll forward or roll backward recovery. (Roll forward recovery makes use of a snapshot backup of the data files, and recovers transactions by applying a transaction log that has been updated each time the data file has been updated. Roll backward recovery involves keeping track of all portions of a transaction until it is completely applied. If the transaction is not "committed", all portions that have already been applied are reversed). The software should also keep statistics on what has occured to the data, that is, the number of times each file is accessed or updated.

Third, the file server machine of the LAN must be kept in a secure area, preferably with an Uninteruptable Power Supply (UPS) but at least with proper surge and power loss protection. In order to enhance the security of the system, many LAN vendors supply either mirrored or duplex hard drive access software. Mirrored access duplicates all file writes and posts them to dual disk drives on a single controller, while duplex access uses two complete and separate hard drive units, including controller, power supply, interface, etc. Other LAN vendors supply software to connect two separate file servers with a high speed bus, allowing the backup file server to take over if the primary file server crashes.

Finally, access to important information should be controlled on a need to know basis, as opposed to the "he doesn't need it, therefore he won't bother it" scheme of file security. The DOS command shell, as well as most LAN software support both file and directory access protection.

TO SLEEP, PERCHANCE TO DREAM

Microcomputers will probably revolutionalize the office place as much as the automatic 150 key calculator of the thirties and forties. But we must take care in the use of these new tools. Computers can multiply our mistakes and oversights much faster that we are used to. If we are to become a community of computer literate employees, we must establish habit patterns that will be as important to us as the habit of buckling a seat belt is important to an automobile driver. The seat belt seems to serve no purpose except discomfort until the first accident. Auditability techniques exist to provide the same increase in safety.

Buckle up for your own peace of mind.

CD ROM as an Alternative Media

Karlie Arkin
Hewlett Packard Company
100 Mayfield Avenue
Mountain View, California

This paper presents a perspective on using the CD ROM as an alternative media. Reader interest in this paper is proof of one of two things. Either A) Currently prevalent technologies are not meeting the needs for the quantity of data, or the usage of information we would like to have today. Or, B) CD ROM is an attractive and somewhat mysterious creature, and the mention of "CD ROM" in the title was enough to arouse interest - after all, the title doesn't even mention the context in which CD ROM is a viable alternative.

The information in this paper should serve to dispel some of the mystery surrounding the CD ROM, as well as provide data to show where the technology can serve to improve our existing processes and manufacturing costs.

Optical Technology

The sheer capacity of optical technology discs makes them an enormously appealing option. CD ROM is one of several optical disc technologies. "Optical Disc" refers to any media that is read or written to with the aid of a laser beam. The extraordinary accuracy of the laser beam positioning allows it to be reliably focused on a very small area, providing the means for a high density of information. The track density on an optical disc is about 15,000 tracks per inch. The approximately 600 megabytes that can be stored on a five inch disc is roughly equivalent to 500 high density floppies, 1500 low density floppies, 200,000 pages of documentation, 15 40-megabyte hard disks all in a row, or 176 copies of Tolstoy's "War and Peace."

There are three categories of optical media, each with a distinct set of characteristics that will tend to bind them towards specific types of applications.

CD ROM - Compact Disc Read Only Memory

CD ROM is the computer equivalent to the CD Audio discs that are commonly available in the music stores. It is the same technology, and the same manufacturing process. These discs can hold approximately 600 megabytes of data. The data is pressed onto the disc, through an automated stamping process. The manufacturing time, even for thousands of discs, can be as low as one day turnaround. The manufacturing cost per disc, particularly with high quantities, is very low. Aside from mastering costs, a per disc cost of under $3.00 is common. Standards for the CD ROM drives, CD ROM layout, and file format are well established, and provide device independence of the media. Why is this important? We would surely expect that a properly formatted floppy diskette could be read by any machine with a floppy drive. The same must be true for CD ROM, and indeed for any media, to become well established. This has certainly been key to the acceptance of the technology in the marketplace today. The other optical medias are not quite at this stage yet. By the end of 1988, over 90,000 CD ROM drives have sold. Sales are expected to continue to rise rapidly, and to more than double by the end of 1990.[1]

The major properties of CD ROM are:

o Low manufacturing cost
o High capacity
o High reliability and durability
o Read only
o Relatively slow access time

This media is preferred for large amounts of stable data, with a large distribution, where reasonable but not fast access time is acceptable. The standardization of the formats also allow for wide distribution into MS-DOS®, Apple®, and a growing UNIX™ marketplace.

WORM - Write Once Read Many

This is an optical media and corresponding WORM drive that looks and behaves similarly to the CD ROM, but has the property of being able to alter the bits on the disc exactly one time. Once a bit has been "set", it can no longer be cleared. This double sided disc comes in a five inch size and can contain about 650 megabytes of information. The WORM differs from the CD ROM in that it is written serially from the computer, and therefore does not have the very low media cost of the CD ROM. Each disc will cost around $100.

The major properties of WORM are:

o The ability to write to the disc, serially
o Low cost relative to other direct access devices
o After writing, the data is read only
o Slow access time
o High reliability, Good durability
o Drive dependant

The WORM discs are primarily targeted for archival of computer data. By the end of 1988, over 50,000 drives have been sold, roughly equivalent to 60% of the CD ROM drives sold by that date. Sales are expected to nearly triple over the next two years, although not to exceed sales of CD ROM drives.[2]

Erasable Optical

Erasable optical provides the dynamic access of a floppy or a hard disk with the large capacity of optical media. This technology generally works by using

[1] James Daly, "Erasable optical disks a step closer to the forefront," Computerworld, 10 Apr. 1989, p. 25.
[2] James Daly, p. 25.

varying powers of laser beams to modify and read the contents of the disc. The media, today, is significantly slower than the high performance hard disks, but is also significantly cheaper per megabyte. Manufacturers such as Hewlett Packard are providing erasable optical in a jukebox form, where the discs can be swapped in and out of the drive for on-line access to many gigabytes of information. There are primarily three types of erasable optical discs available or under development for commercial use.

o Magneto-optic
o Phase-change
o Dye Polymer

The magneto-optic drives are the only one of the three technologies that are commercially available at this writing. Erasable optical discs have not yet achieved the level of standardization that is seen with the CD ROM media. The CD ROM format is not suitable for erasable optical discs, primarily due to some optimizing assumptions on the CD ROM that the files will not change size or location.

Other Optical Media

Research into other forms and formats for optical media is heavily funded by several major manufacturers. As a result, the technology is in a state of rapid growth. A variety of new media materials and formats over the next five years would not be surprising.

This paper is dedicated to the understanding of the CD ROM media in particular. The high level of standards in place for the format combined with the low manufacturing cost can be a winning combination.

Today's Media

The media that is used to store information or data is greatly entwined with the intended use of the information. To illustrate this point, consider this short list of commonly used media:

o Magnetic tape reels
o Cartridge tape
o Floppy disks
o Microfiche
o On-line access
o Paper
o Video tape
o Audio cassette
o CD Audio

If you had a newsletter to distribute, it is somewhat comical to consider sending it out on 1/2" mag tape. At the other extreme, it would be absurd to consider sending a video taped training course on paper, floppy disk or microfiche. To understand the appropriateness of CD ROM, it is necessary to address each application in terms of the contents, intended usage, and audience. A number of measurable factors will come in to play when it is time to evaluate the trade-offs between different media.

1. Capacity vs. amount of data
2. Cost per unit
3. Quantity needed
4. Replication time
5. Frequency of distribution
6. Durability and reliability
7. Availability of necessary hardware and software

This list is probably similar to the questions that either implicitly or explicitly asked today to compare any choice of media to another. Only the last item on the list, availability of hardware and software, hints that the choices have expanded to include a newer technology.

The attributes of CD ROM are such that several additional factors come into play. When weighing CD ROM as an alternative, consider the following:

o **The Sizzle Factor** - aside from the hard facts of cost, time, and resources, the perceived attractiveness and appeal of the CD ROM may add to the momentum of a switch to this media.

o **Market Acceptance** - while CD ROM has rapidly gained market acceptance over the last few years, it still carries the reputation of being a 'State of the Art' technology. This perception can either be a powerful persuading factor, or just as powerful dissuader to your recommendation to move towards CD ROM, depending upon your environment.

o **Resistance to Change** - while the CD ROM access could be transparent to the user, it could also be very visible. The visible effects could be not only in the way the data is accessed and used, but may also affect the process used to generate the data. Overcoming any resistance to change on either the generating side or the accessing side may be a significant factor in deciding whether or not to switch.

o **Incremental Value** - the capacity of the CD ROM may be larger than is needed. How can you take advantage of this extra space? Extra files could be added containing speed enhancing lookup tables, or text files containing background information, documentation, or newsletters. Databases could be expanded to include additional fields, demo versions of software could be provided, or templates for existing applications. In this way, CD ROM is an enabling technology. The CD ROM attributes can enable you, as a publisher or software manufacturer, to provide information or services in ways that were simply not viable with other commonly available media.

o **Starting a New Process** - the time and cost to set up a new process may be prohibitive, and is very dependant upon the type of information and how it is to used relative to the current content and method.

CD ROM as a Direct Replacement

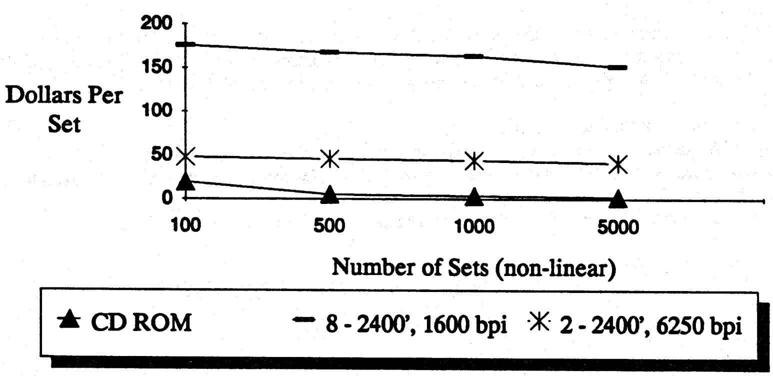
When the circumstances are right, choosing CD ROM as an alternative can have numerous benefits, including reduced manufacturing costs, reduced manufacturing time, and lower cost of distribution. But at the other end of the spectrum, CD ROM may be an expensive mis-match. Understanding the trade-offs in time and cost is key to making this decision.

From the perspective of cost of manufacturing, the comparison for replacing an existing method of distribution is straightforward. The cost will be directly tied to the amount of information, and the number of copies needed.

The following graph uses the following assumptions:

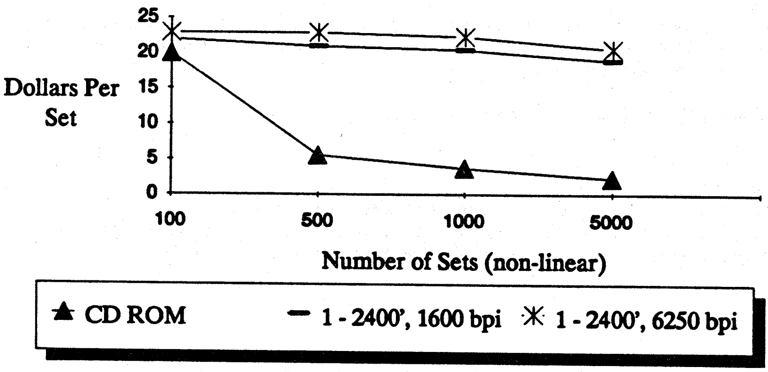
o a mastering cost of about $1800
o a per media cost of about $2.00
o a manufacturing time of 5 days (shorter time has additional costs)
o a Magnetic tape media and replication cost of ranging between $19-24.00, depending on quantity and density. These costs are without any bulk discounts for repeat business.

300 Megabytes on CD ROM vs. Magnetic Media



Of course, half-inch tape is typically used in a minicomputer or mainframe environment. The ability to access a CD ROM peripheral reader must be available before CD ROM can be considered as a viable alternative.

40 Megabytes on CD ROM vs. Magnetic Media



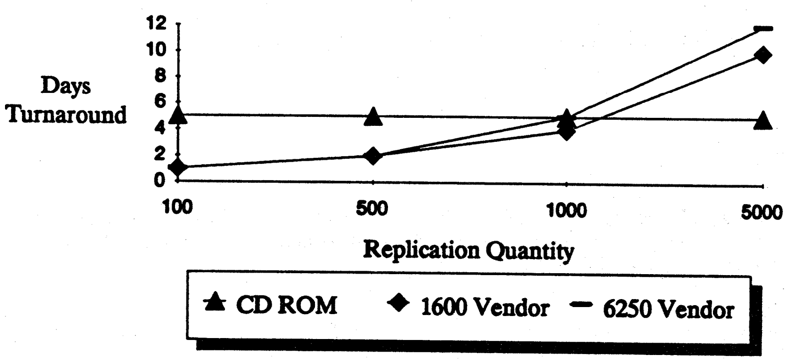
The time to replicate copies of the media for distribution, or the cost of additional equipment and labor to speed up the replication time, can be a significant factor in the choosing one medium over another, particularly when dealing with large replication quantities.

The following assumptions are made:

o CD ROM turnaround time is set at 5 days, to match the other examples in this paper. For a premium price, turnaround can be reduced to 1 day. In-between prices are also usually available.

o Vendor turnaround time is dependent upon the capacity of the vendor, and will vary.

Replication Time - Lowest Cost

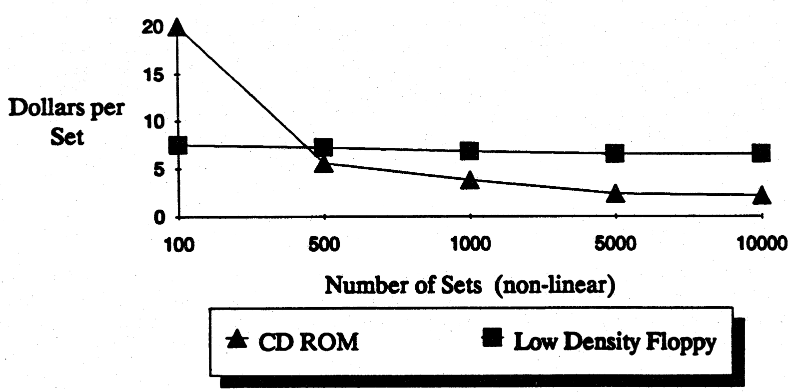


With magnetic tape, the need to provide customized tapes sometimes exists. With additional labor and time, individual tapes can be matched to a set of specifications prior to the actual copy. While the cost of this can be significantly higher, perhaps an additional $10.00 per magnetic tape copy, there is no comparable way to do this with the CD ROM media. The master CD ROM, once produced, is unalterable and will yield only exact copies. A CD ROM solution to the need for customized content can likely be devised, depending upon the nature of the customization. For instance, if each tape was a subset of a greater set of information, it would be feasible to include the entire set of information on the CD ROM in some inaccessible or otherwise secure form. The CD ROM would then be accompanied by some instructions, passwords, software, or hardware that would give the user the means to access the precise content to which they were entitled. Furthermore, the scheme could be extended such that when the user wished to have access to additional information, they might be able to make a phone call, provide a credit card or purchase order number, and immediately receive over the telephone the password or software that would entitle them to additional contents of the disc.

Replacing floppies with CD ROM has some clear advantages when you compare the convenience and cost of many floppy disks with a single CD ROM disc. Consider for an example, the run-time version of Microsoft's[R] PC software, Microsoft Windows. This is currently being distributed by a variety of manufacturers. Generally the files will located on 10 or 11 low density floppy disks. With the following assumptions, we can compare the approximate manufacturing costs of floppy diskettes vs. CD ROM.

o a CD ROM mastering cost of about $1800
o a CD ROM per media cost of about $2.00
o a CD ROM manufacturing time of 5 days (shorter time has additional costs)
o a floppy labelling, replication and media cost ranging from $6.50 to $7.50 per set of 10, depending on quantity.

Cost of CD ROM vs. Set of 10 Floppy Diskettes



In this situation, the cost of CD ROM manufacturing undercuts the cost of a floppy set when the quantities approach 500 copies.

New Applications for CD ROM

The coming of age of CD ROM provides a variety of opportunities to distribute information that was not previously distributed on electronic media, or to combine the distribution of information that was previously issued separately for one reason or another.

This is the area where the most growth will be seen for CD ROM, where the compact disc and laser technologies have extended the boundaries for electronic distribution of information and tools far beyond the limitations of the magnetic technologies commonly used today.

The most visible use for CD ROM is in that of the sales and distribution of information. Periodicals, and other types of textual documentation, are popular candidates for CD ROM distribution. This is primarily for three reasons.

1. The CD has the capacity to hold a sufficient amount of information that it is convenient to put all of the information on a single piece of material.
2. The excess capacity allows the addition of files that increase the performance and resolution of the electronic access to the extent that it is a viable alternative to paper.
3. The manufacturing cost of a CD is significantly lower than the paper publishing cost of an equivalent amount of information.

Opportunities to combine the distribution of information, previously separate, is another way to leverage the attributes of the media, and can take on a variety of forms. The following case study shows how you might take advantage of the CD ROM to reduce the cost of manufacturing the product, as well as to provide the customers with a higher level of productivity.

A Fictional Case Study

Perhaps, as an example, you are selling a large accounting software package designed for a workgroup of up to 10 users connected on a LAN. You may be currently providing the software on 10 low density floppy diskettes. The package comes with five copies of a two-volume manual set, an installation guide, and five copies of a quick reference pamphlet. You may also sell or provide some software templates or forms on 4 floppy diskettes. The templates are accompanied by a catalog showing samples, disk names, and filenames. A tutorial is provided on another floppy, with a tutorial handbook. And, the package comes with a reference book on common accounting procedures. The forecast is for 150 units per month. You plan to ship the version for 12 months before revising the software, documentation, or tutorials.

The CD ROM Role

You can certainly substitute the low-density floppies with a CD ROM. After placing the software, the templates, and the tutorial software onto the CD, you have replaced 15 low density floppies, a maximum of 5.5 megabytes of data. With about 595 megabytes left on the CD, there's a lot of room to play.

Let's say you put an electronic copy of the manuals on the disc, taking up two or three more megabytes for the 600 to 1,000 pages of text. Add the reference book on accounting procedures, for another megabyte. The manuals are not very useful unless you add some software to provide fast and easy browsing and full text keyword search and retrieval. That occupies another 5 megabytes, and will probably have additional cost in terms of development, licensing, or purchase of the software. At this point, the data and software has taken over 14 megabytes of storage space.

To add high resolution illustrations to your electronic manuals, you will need more disc space and will incur some additional costs for scanning or converting the illustrations to electronic format. The size and quantity of illustrations will determine your space needs. We will assume 50 megabytes of uncompressed black and white drawings. Obviously, this is not something you would attempt to provide on floppy diskettes. Still, with all of the above on the disc, you have used up about 64 megabytes - about 1/10 of the CD ROM's capacity.

Furthermore, with the appropriate choice of software, you have provided the customer with something they did not have before: the ability to search the software documentation, the reference manual, the tutorial, and the guides simultaneously. Plus, the means to ask complex questions such as "Find places in the documentation where there is a reference to 'taxable' and 'rate' but not 'optional'".

Additional costs:

o Data preparation, for the illustrations and text
o Search/retrieval software purchase or development
o Probable need for a new peripheral

Saving costs:

o Ship a single paper copy of the manuals to the customers - saves replication and shipping
o In a period of months, reduced manufacturing costs will be realized.
o Added functionality for the customer in the search/retrieval software

Additional considerations:

o If the software and documentation was not stable over the course of a year, the recurring mastering costs would significantly reduce the gap between CD ROM and floppy/paper manufacturing costs.
o If the software was not destined for a LAN, then placing the documentation on the CD ROM would not reduce the paper copies of the manuals from quantity of five to a quantity of one.
o The data preparation costs should be carefully scoped. This will probably be the major additional expense. If the documentation is subject to frequent modifications, then this could be a partially recurring cost.
o The search/retrieval software needs to accommodate the data. For example, if the documentation contains drawings, the software ought to be able to display those drawings.
o Resistance to new hardware purchase might be offset by coordinating a program in conjunction with a CD ROM drive manufacturer. If the success of the media is very important, hardware give-aways might be appropriate.

- o The decision to provide CD ROM may be optional to the customer. This would have impact on the forecasts, and the costs based on quantity production.
- o If you intend to sell, rather than to give, the templates and template catalog, you would need to define a scheme for protecting that information unless the customer had purchased this option. Alternatively, you might decide to continue to distribute this data on a separate media set, which could again be either on floppy diskettes or CD ROM.

This example is fairly simplistic, but is a good example to show the complexity of factors that must be balanced before making a "best media" determination.

HP LaserROM Case Study

Two sets of people have to be convinced that the CD ROM media is a superior media for a particular application. First, the publishers or manufacturers of the information must see the cost savings, revenue increase, or strategic value to the new media. Additionally, the customer or user must be convinced enough to justify the cost, if any, for the CD ROM's themselves and the hardware to read them with. In many cases the manufacturing costs alone can convince the potential CD ROM publisher that this media is cost justified. But how is the customer convinced?

HP LaserROM is a subscription service to HP documentation and support information. Several different subscriptions are offered, the one in this example is a subscription to the MPE V support information and documentation. With this subscription, the customer receives a new CD ROM every month. Each CD ROM is a full replacement of the previous months, updated with the latest information. The contents of the disc include the MPE V operating system manuals, subsystems, and languages. They also contain Software Status Bulletins, Application Notes, an HP Product Catalog, and more. The following justification was prepared and successfully used by a customer to justify two years of monthly CD ROMs. Many of these factors could be applied to a variety of applications of the CD ROM technology.

HP LaserROM Costs:

	Year 1	Year 2
Drive	$1,095	$0
Software	100	$0
Subscription	1,287*	$2,340
Total	$2,482	$2,340
Per month	$207	$195

*At the time of the study, an HP LaserROM subscription was being sold under a 45% discount promotion for the first year of purchase.

Floor Space Savings:

o One manual set takes up 20 square feet
o Estimate at $25/square foot/year

20 s.f. * $25/s.f.year = $500 savings per year

Productivity Savings:

- The study found that each reference manual access averages 31 minutes and that with HP LaserROM, this is reduced to 14 minutes. This is a savings of 17 minutes per access.
- Their software developers estimate that they spend about 1/8 of their time using reference manuals. On the average, they are spending five hours per week accessing their reference manuals.
- Their MIS group found that they averaged 17.3 reference manual accesses per year per person.

Software Developers:

Time saved	= 17/31 minutes times 300 minutes (5 hours) per week
	= 165 minutes per week
	= 2.74 hours per week
	= 137 hours per year, per software developer

o Cost and overhead for each software developer is valued at $55.00 per hour.

Amount saved	=137 hours per year * $55 per hour
	= $7,535 per software developer, per year.

MIS Support Group:

Time saved	= 17/31 minutes times 17.3 accesses
	= 4.9 hours per year per MIS Support Person

Amount saved	= 4.9 hours per year * $55 per hour
	= $270 per MIS Support person, per year.

Payback:

Situation A:
o One software developer using HP LaserROM
o Average HP LaserROM cost of $2400 per year
o Average productivity gain of $7535 per software developer per year

PAYBACK = $2400 / $7535 = 3.8 MONTHS

Situation B:
o Four 4 software developers using HP LaserROM
o Eliminate the purchase of one extra manual set
o Average HP LaserROM cost of $2400 per year
o Average productivity gain of $7535 per person per year
o Average floor space cost per manual set of $500 per year

PAYBACK = $2400 / (($7535 * 4) + 500) = .97 months = 3.9 WEEKS

Situation C:
o Eliminate the purchase of one extra manual set
o Average HP LaserROM cost of $2400 per year

o Average productivity gain of $270 per person per year
o Average floor space cost per manual set of $500 per year

How many people using HP LaserROM are needed to payback one HP LaserROM for this group?

PAYBACK = ($2400 - 500) / $270 = 7.04 MIS Support People

Additional Notes:

- o The study evaluates the use of reference manuals only. Additional value can be applied to individuals who will take advantage of the other support information on the discs.
- o The set of manuals on the current MPE V disc would currently cost over $2,000.00. By eliminating the purchase of additional sets of manuals, significant savings can be realized. The study does not take this savings into account.
- o The productivity savings during the first year should be buffered with 1 - 2 hours of time for each user's learning curve.
- o Do not forget about the time that is currently invested in maintaining and updating the current manuals sets with page inserts and replacements. The CD ROM version of the manuals are updated in their entirety, eliminating the need for update pages.

CD ROM's Position as an Alternative Media

CD ROM is a viable alternative to other media when certain conditions are satisfied. Particularly suitable applications for CD ROM are those that:

o have a high volume of information
o have data that is reasonably static between pressings
o the data does not need to be modified dynamically
o do not have a need for very fast access time
o need a durable, compact, standard format media
o are sensitive to the cost of manufacturing and shipping
o can accommodate the need for a CD ROM drive

There is an open arena for new applications that can take advantage of the optical technologies. Some applications will extend our current ways of doing things to a larger scale, such as providing huge quantities of clip art on a CD ROM. Other applications will challenge our processes as they change our way of conducting business. An example of this would be to receive all available software on a CD ROM, and then to be able to purchase access to the individual software packages or databases via a telephone call. CD ROM is far from the answer to all of our storage needs. Instead, CD ROM is an enabler. It enables us to dream of new applications and better ways to be more efficient, more accurate, and more reliable when dealing with information that is so essential to the success of our organizations.

MS-DOS and Microsoft are registered trademarks of the Microsoft Corporation.
Apple is a registered trademark of Apple Computer.
UNIX is a trademark of Bell Labratories.

SUPPORTING End-User PERSONAL COMPUTING with Departmental Office Product Coordinators

Lisa Shemwell
Hewlett-Packard Company
World Wide Customer Support Operations
100 Mayfield Avenue
Mountain View, CA 94043

The session will encompass areas a Site Information Systems department had to address in developing a decentralized Personal Computer Support model.

Problem: The support staff could not keep up with the technology and the business needs of the organization. Users were demanding different levels of support.

Solution:Bring the technology and the business needs closer together by putting into place programs that encourage an integrated departmental computing environment. By putting support in place at the local level, using a decentralized departmental Office Product Coordinator (OPC) support model, local management can see gains in local solutions to tasks and processes using PC technology.

The session will discuss:

- -Centralized vs. decentralized support models. Selecting decentralized departmental computing support model.
- -Profile of case study
- -The evolving support strategy to meet the business needs of the customers.
- -Culture changes within the site environment.
- -The Information Systems fundamental support programs ensure OPC and department success.
- -The Current Office Product Coordinator (OPC) program outline.

Take the time to look at your own organization as you read through the case study and ask yourself "What portion of this could I adopt?".

SUPPORT STRATEGY ALTERNATIVES:

Here are some advantages and disadvantages to centralized and decentralized computing support models.

CENTRALIZED COMPUTING SUPPORT

Centralized support emanates from primary source. The user calls the central group for all types of questions.

Advantage	Disadvantage
-Consistent answers	-Redundant questions are answered over and over again.
-Info Systems is in control of information	-Turn around is viewed as inconsistent

-Simple structure for user	-Other priorities are more critical -Little passing on of knowledge to user -High investment of staff,plus more staff -High level of expertise for lower level problems. -Staff turn over causes major "holes" -More reactive than proactive.

Centralized support has grown more complex than the users could follow.

DECENTRALIZED COMPUTING SUPPORT

Decentralized support comes from a source within the department. Basic support of business solutions are in place. Central resources are used for higher level support questions and consulting.

Advantages	Disadvantages
-Business Solutions are put into place -Business partners are made -Knowledge of work environment by users -Increase of local expertise -More proactive than reactive -Self-sufficiency is achieved	-More complex environment for department management to manage. -Turn over of department support -More ownership of technology

The support roles need to be defined and made clear so the customer can see the benefit of decentralized support. Then Information Systems staff moves from a purely support function to providing tools and consulting.

PROFILE OF H-P's Customer Support Center Mountain View, California

In 1986 the organization moved to the Customer Support Center which brought together all local Customer Support activities under one roof. The outcome for Site Office Systems, more users to support, but little headcount growth.

Support Staff in the Office Systems Group has grown since 1983 from 1 person supporting HPDesk and HP3000 office products for 300 internal HP employees, to, in 1989, 5 Office Systems staff with 3 people assuming the primary responsibilities for PC support, education, programs, and the support of HP3000 office products and messaging products (HPDesk, Unix-Mail, Voice-Mail) over 1000 internal HP employees.

The organization has shifted from from a mixed work force to a knowledge worker site. Site Office Systems provides services to all entities on the site. Support and program decisions are made taking into consideration each entity's business, technical expertise, Corporate recommendations and the needs of the majority of users.

In 1987 Corporate Information Systems positioned the Personal Computer as the input device of choice. With this recommendation the demand for PC implementation support issues outstripped the resources of the central organization. Thus there was a change in focus from central, one-on-one, help to a decentralized liaison focus.

Information Systems (IS) Office Systems current staff.

Now: 60 department OPCs do 50-80% of basic PC support. Software installation, ordering information, business controls, checking configurations, etc..

Current Staff total of 7

3 PC support and education activities.
- 1 person to trouble shoot and develop job aids, expertise Operating systems, graphics,
- 1 person to address training, develop end-user programs expertise word processing/Desk Top Publishing
- 1 person to coordinate OPC program and consulting expertise PC databases and spreadsheets.

2 HP3000 and Message administration.
- 1 person Unix-Mail Admin and technical HPDesk support
- 1 person HPDesk administration

Office Systems On-line function
Above people rotate as on-line support to OPCs and site users between Office Systems 5 people for 2 1/2 days

2 Project and program administration and support
- 1 person Voice Mail Project Lead, Home Loan and Excess Equipment programs.
- 1 person HPDesk Directory maintenance, training registration and administrative support.

MIS HELPLINE function
Above 2 people rotation MIS HELPLINE between 5 people across site IS departments.

Some departments do not have OPCs. Non-OPC departments are included in the support plan, but not as the major focus in support and implementation planning.

SUPPORT FUNCTION ALIGNMENT

Using the decentralized support model, ownership of PCs reside in the department and work group level. In providing support to the site, Office Systems defined what Office Systems site staff and departmental OPC support functions and responsibilities are:

CENTRALIZE Support Functions	DECENTRALIZED Support Functions
25% of support calls are turned over from OPCs.	75% of calls can be answered by OPC Knows environment and priority
Able to look at trends	
Develop strategy	Looks for specific application to automate work Group task/processes.

Develop recommendations

Develop user aids

Provides consulting

In depth understanding of technology
Helps departments with short term issues.

Insures departments are aligned with site's long term strategy.

Transfer of technical information
Gains business knowledge

Consults to help departments meet their business goals and objectives.
Keeps abreast of Corporate and industry recommendations.
Becomes proactive, not reactive.

Right product is being used to automate task.

Develop PC knowledge
Computing is a part of departmental goals and objectives

More attention is given to the integration of technology within the work group.

Someone is responsibility for IS information/recommendations at the department level.

Provides support of products outside of recommendations.

Becomes more self-sufficient

EVOLVING SUPPORT

To ensure the organization's computing needs are being met, the IS organization must build partnerships with each business segment.

Those partnerships are being built using Department Office Product Coordinators (OPCs). Program development continues at Hewlett-Packard's Customer Support Center. The OPC program allows IS to move needed technology into departmental work groups. The site IS organization continues to develop and maintain support programs, ensuring the right resources will be in place to compliment the department support resources.

As departments develop expertise and understanding on recommended applications and configurations; gains in productivity and better alignment with the business goals occur within department. In turn the technology becomes integrated into the department, enabling them to be self-sufficient as new technology needs to be implemented, i.e. LANs, work group software and network applications. Meanwhile, the IS consulting role is being developed to ensure programs, tools, and departmental computing plans are in place to ensure personnel continue to be successful.

Developing the Office Product Coordinator (OPC) provides a support foundation to be in place for implementing and supporting computing at the closest possible position to the end user. Taking recommendations from the IS organizations will ensure the departments will be able to connect and communicate in the most efficient manner by using a common integrated product platform.

Moving support and strategy is a dramatic change for most organizations. There will be resistance to change. Leading the challenge IS will see the active organizations reap productivity benefits - in more up time, apply the correct computing solutions, each person develops computer skills and growing with computing as the business grows.

CHANGING THE PAST CULTURE

Dealing with the cultural change of moving support from a central resource can be a very slow process. Customer Support Center continues to progress toward the decentralized model. Here is a sample of how the model is working. In the past the central organization would have taken all of the end user support calls. Today, taking calls is only part of their role. Passing on knowledge the central group has taken fewer basic calls and more complex calls challenging the IS staff.

More effective and self sufficient trouble shooting.
- The OPC knows the department configurations and applications being used. 75% of end customer concerns can be answered within 10 minutes. The central organization's turn-around time is 1-4 hours.

Type of Problems an OPC addresses:
- A PC becomes unplugged or loose connection. The OPC can solve this problem plus pass on the knowledge to the PC user of what to check next time. While a centralized organization will not have even made it to the customer in the time the OPC had solved the problem.
- Installing PC software. User may not know how. OPC has information and will assist user if needed, but avoids installing it for the user. Information is passed on.
- The OPC works with the central staff to recover data for a complex accounting re-run. Working with the central staff, the OPC will have the skills to do the process the next time. Skills and knowledge are passed on to the OPC and department.

WHAT DO YOU NEED TO DO TO CULTIVATE THE SUPPORT CHANGE

Decentralized support of computing means departmental managers must become knowledgeable on how to manage basic computing resource needs. Many non-technical managers have yet to acquire computing resource management skills. Just as they have learned to manage financial and personnel resources, they will need to become proficient in computing resources.

How management becomes aware:

- Management presentations selling the increase PC knowledge and productivity gains of the decentralized PC support program.

- Formal education -presented to site managers such as: "Managing your Computing Resources" from Information Ideas - Oakland, CA.

- First hand using an OPC in their department.

ROLES OF ORGANIZATIONS

Each organization within the HP structure has a role in the OPC support model. In looking at the OPC program at Customer Support Center, these are the roles that have been outlined for each organization and how they fit together.

ROLES OF MANAGEMENT/DEPARTMENTS
(Corporate IS, Entity IS, Department Management, and Department OPC)

Corporate IS Role
-Provides recommendations to ensure connectivity throughout the company.
-Sharing best practice across entities.
-Provide centralized evaluation resources.
-Ensure best use of entities resources can be focused.
-Provide linkage back to product development organization.

Entity IS Role
-Adopt and adapt Corporate recommendations.
-Develop local support strategy for recommendations.
-Provide phasing in and phasing out by providing product assistance.
-Provide technical resources to address customer problem.
-Provide a consistent customer training and programs.

Department OPC Role
-Develop a work group computing plan within the department.
-Provide task/process automation direction.
-Provide guidance for software solutions.
-Provide guidance for ordering hardware.
-Communicate training needs to manager.
-Adopt site recommendations and work with IS.
-Make local adjustment as necessary.

Department Management Role

-Ensure resources are in place for following Computing Plan in the department.

-Manage computing resources at the department level; just as personnel, business controls and Finance are managed at the department level.

-Consider technology for solutions, not solutions for technology.

Department Management with OPC

-Management needs to determine what standard department application (i.e. Word Processor) will be used by the department.

-Align with Site and Corporate recommendations.

-Purchase Hardware and Software as outlined.

-Plan training time for end users.

-If application needs do not align with central/site recommendation ensure proper support for users is arranged.

PROGRAMS TO BE IN PLACE:

As the Office Product Coordinators become a vital part of the support organization, the following programs have been established to support the OPCs. These programs help in establishing expectations of each organizations role in the total support solution.

-Product Recommendations-Following Corporate recommendations of Office Systems products, recommending the PC be the input device of choice and continue to strive for connectivity between HP organizations.

-Product Support-Focusing on recommended products and their life cycle.

-Education Program-Training users, OPCs and central support time decreases, while user satisfaction increases.

-Business Controls-Providing programs to ensure the issues associated with responsible use of PCs are managed wisely.

PRODUCT RECOMMENDATIONS STRATEGY:

Corporate and Office Systems objective is to create a totally integrated office environment providing a solution to a majority of office users. Developing guidelines for recommended packages across the organization, aids to reduce support cost and increase productivity by:

-Connectivity across organizations.

-Recommendations on hardware and software for majority of purchases.

-An in depth skill base is developed in departments of product knowledge allowing OPCs and Office systems to leverage from that knowledge base.

-Establish core customer training from recommended products.

-Enable technical support to focus on a group of products.

-Job Aids can be developed for the recommended products.

Office Systems gives product recommendations to fulfill the majority of site users needs. Recommendations for hardware, software, media, peripherals, suggested workstation configurations and suggested software package mixes for job types gives the OPC guidance in selecting the right solution for their department. If departments in the organization need a product outside the recommendation set, the department will need to make their own arrangements for support for the non-recommended product.

Office Systems rate products by:

- -Ease of Learning-Average # of hours for end-users to basic product features.
- -Ease of Use-Measure of how "friendliness" and feature complexity.
- -Performance-Speed and ability to handle complex tasks.
- -Integration-Ability to share information with other products.
- -Training for end-user and support staff
- -Support for for OPCs and technical staff
- -LAN Support-Designed and approved to be networkable.
- -Corporate Recommendation
- -Projected use-Volume of users to project support level of product.

By driving the product recommendations Office Systems has been in a proactive mode to provide training and support to the OPCs. Focusing product recommendations equal:

- -Fewer configuration problems.
- -Fewer conversion problems.
- -Greater productivity for end user.
- -Greater productivity for support staffs.
- -Greater self sufficiency

PRODUCT SUPPORT STRATEGY:

Using the recommended products Office Systems has developed 3 different product support levels. This process clarifies how and what products are supported.

Product support levels are determined by product recommendation, the volume of users on site, phase of implementation and life cycle of the product. Support for products are the current version and current version -1 software applications.

PRODUCT SUPPORT LEVELS

A level-Full support priority over B and C level products. Office systems will call back OPC within 1 hour. Office Systems is committed to resolving the problem by providing a fix, a work around and/or escalating the problem to the next level of technical support resources. The Office Systems staff has a minimum training at 200 SE level.

These are the factors to determine A-level support for a product.

- -high volume of use on site.
- -recommended by Corporate IS.
- -recommended by Site Office Systems
- -is part of the Site automation strategy
- -is a mature product, in its life cycle.
- -and Office Systems has access to on-line and Response Center support organizations.

B level-Limited support. Priority over C products. Office Systems will call back OPC within 1 hour. Office Systems works with the OPC to find a fix, a work around and/or and the OPC escalates the problem to the next level of technical resources. Products are currently a mature product with limited site use or phase-in/out implementation would be classified as B-level.

Factors to determining B-level support for a product.

<u>Mature Products</u>

- -Application is used internally on a limited basis (specialized use).
- -Products have a limited number of users on site.
- -A conservative implementation approach is recommended by Corporate.
- -Office System staff are trained at or above 100 SE level.

<u>Phasing in</u> Products (approx. time 6 months at B-level) Office Systems gains knowledge, training begins for OPCs. Conversions training is developed by Office Systems and is provided to OPC's and workgroups . This training ensures departments have the tools and skills to help users convert from the old product being phased-out, to the new replacement product being phased-in. Meanwhile more technical expertise on the product is being gained by Office Systems and OPCs. Product will be moved to A level or stay at B level depending upon recommended use.

The factors to determine a product:

- -a new product recommended by Corporate IS.
- -a new product recommended by Site Office Systems
- -is becoming part of the Mayfield site automation strategy
- -is a new product in its life cycle.

Office systems will be receive 200 SE training. Office Systems will have access to on-line and response center support organizations.

<u>Phasing out</u> Products (approx. time 6-12 months at B-level) Office Systems will put no effort in to providing product training, nor gaining more technical expertise on the product. Product will be moved B-level as a mature product or to C level depending upon the user base and life cycle of product.

The factors to determine a product:
-is no longer recommended by Corporate IS.
-is no longer recommended by Site Office Systems
-is no longer part of the Mayfield site automation strategy
-close to being obsolete in its life cycle.

Office Systems has access to on-line and response center support organizations to provide conversion support to new product.

C level-No support commitment. Office Systems will attempt to resolve problem as time and local expertise permits. OPC should contact vendor directly.

These are the factors to determine C-level support for a product:
-very limited number of users on site.
-is not recommended by Corporate IS.
-is not recommended by Site Office Systems
-is not part of the Mayfield site automation strategy
-is an obsolete product in its life cycle.

OPC should make support agreements with on-line and response center support organizations. OPC and department management should plan to upgrade or change product.

END USER EDUCATION PROGRAM

Providing ongoing education and support are the most critical factors affecting the successful implementation of the Office Systems implementation plan. Office Systems focus on end-user and computing skills management training.

OPCs are given preference for classes they need. Workgroup classes will be given to for requesting departments.

A formal education program must be in place to ensure each user has appropriate skills for applications for task or process automation be used on the job. Encouragement of self-sufficient support to ensure effective use of central and OPC support time, basic computing skills are offered and encouraged. Training a user on a product is a proactive approach to self-sufficient support and helps to build product use confidence.

OFFERED CLASSES

Training classes are offered on products in the Support A category and products being phased into the organization. Each class will be between 4-6 hours, and taught during regular business hours in the Office Systems Training room. Other types of training offered will include videotapes, teleclasses, demos and self-paced training programs.

As new products are released, Office Systems will offer classes or workshops to train the organization in the new technology. Classes are coordinated through the Office Systems Education Administrator.

LEARNING CENTER AND SOFTWARE TEST CENTER

An Employee Learning Center is available near Office Systems for employees to use for self-paced training and as a workstation to test out recommended software.

COMMUNICATIONS

A site wide newsletter is distributed quarterly. Bulletins are distributed through the HPDesk on an "as needed" basis. Demos and site-wide meetings are also conducted as the phase in/phase out strategy model and as needed.

BUSINESS CONTROLS

PCs have become an important tools at the Customer Support Center. Business controls are outlined to help the department OPC put measures into place to ensure compliance with legal, security and control issues.

AREAS TO ENSURE CONTROLS ARE IN PLACE.

Processes for controlling and monitoring the use of personal computers have been established within departments. The encouragement of good business practices for protection against theft, security of sensitive data and backup and disaster recovery plans must be in place. The OPC is the central resource person to ensure the department has measures in place to meet the legal, company and site guidelines.

- -Software copying is the responsibility of each PC user. Install and operates PC software in must be in accordance with copyright laws and software license agreements.
- -Physical security of equipment is the responsibility of site users to provide adequate protection following Site Security recommendations.
- -Data security is defined by Personnel policy and procedures.
- -Decision Tools, when changing information or how information is calculated, procedure must be set up to control and document changes using SLC.
- -Data and software back-up procedures must be in place as outlined in PC Use Guidelines.
- -Use of PC at home and while traveling must meet the controls as outlined above.

OFFICE PRODUCT COORDINATOR PROGRAM

OFFICE PRODUCT COORDINATORS have become a vital part of the Office Systems support organization. With the advent of PC use comes the responsibility of managing and supporting PCs in each department. The OPC establishes a direct link to the Office Systems group, providing the OPC with support and planning tools to manage and support PCs at the department level. The OPC assists management in developing strategic and tactical plans for business information needs and automating process within their department. Office Systems provides the OPC tools and programs to assure their success in the OPC role:

ON-LINE SUPPORT

OPCs questions have priority over any other calls.

NEW TECHNOLOGY

Office Systems should be the first to try new products on the site and with in the department. Office Systems provides first release copies of software to OPCs. Before mass distribution of product starts. This is part of the phase in/out model. The OPC can delegate her/his software copy to someone in the department. That person then would become the application expert on the product.

TRAINING/EDUCATION

The OPC has first priority in any Office Systems sponsored class, seminar or demo. End-user training will be offered on A and B level products. OPC should be proficient on the software the department uses. The OPC will be given preference to first choice in signing-up for Office Systems classes.

The OPC will need more than just end-user training. A more in-depth knowledge of recommended software, hardware, installations, de-installations and configurations with trouble shooting techniques have been put into place by Office Systems.

COMMUNICATION

OPCs must be kept abreast of new products, bug fixes and order information. This is done by:

- -Newsletters covering current strategy and support information.
- -Bulletins to inform OPCs about specific concerns over HPDesk.
- -Information Sharing Sessions provide the OPCs with a forum to share information and concerns. Office Systems provides new product announcements, program changes, and exchange of best practices during these meetings.

HP SITE LICENSE SOFTWARE

Departments that are committed to the OPC program will receive site license software and upgrades FREE. The license to copy software are purchased by the Office Systems central group from HP. The OPC network at the Customer Support Center is used for the distribution of the site license software product.

EXCESS AND LOANER EQUIPMENT

Office Systems provide the use of a pool of hardware to departments for 6 months for special needs, while waiting for additional or replacement equipment.

WORK GROUP CONSULTING

Office Systems works with an OPC to provide customized solution for departments with individual needs, planning for software/ hardware and training. While developing this plan Office Systems gain a basic understand of the department's business. Office System will also give advice on automating tasks and processes.

HOW MUCH TIME AND HOW MANY PEOPLE SHOULD THE OPC SUPPORT?

This depends on the complexity of application software and hardware set-up within the department and the level of expertise within the department. One person for 5 hours per week can support 20, 40 or 80 users depending on:

THE CONFIDENCE AND USER KNOWLEDGE OF PRODUCTS BEING USED.

Untrained users and inexperienced users will increase the support time the OPC will spend with the department. Studies have shown 4 hours of on going support training/questions can be saved by 1 hour of formal instruction.

HOW COMPLEX ARE PRODUCT FEATURES.

The the more product features that exist the more things can go wrong. If the whole department uses a simple word processor, the OPC will need to only know one product and configuration. The more complex the product the more support time the product(s) will require.

TIME

Mayfield has asked each manager to commit the OPC for five hours per week to:

- -Answer beginning application questions
- -Check configurations, trouble shoot common problems
- -Ensure PC business controls are being followed
- -Keep current on ordering recommendations
- -Keep informed on central/site strategy.
- -Assist in installation and de-installation of products.

Department OFFICE PRODUCT COORDINATORS are a vital network of the total Customer Support Center support network. The basic support, system management and business controls are in place by using OPCs as a direct link to the Information Systems organization. Office Systems group, provides the OPC with support and planning tools to manage and support PCs at the department level. The OPC assists management in developing strategic and tactical plans for business information needs and automating process within their department. This link to a two way street to ensure Office Systems and Information Systems department is a business partner with each department.

Why decentralize support?

- Unique office environments require focused support
- New complex technology increases need for local expertise
- Allows Office Systems ability to concentrate on proactive productivity solutions.

3361-14

epct3

OPC FY'88 Objectives

- Provide timely, consistent PC support i.e., consulting, troubleshooting
- Facilitate a proactive approach to PC support
- Support the strategic concept that end users have ownership for their PC/OA tools

3361-15

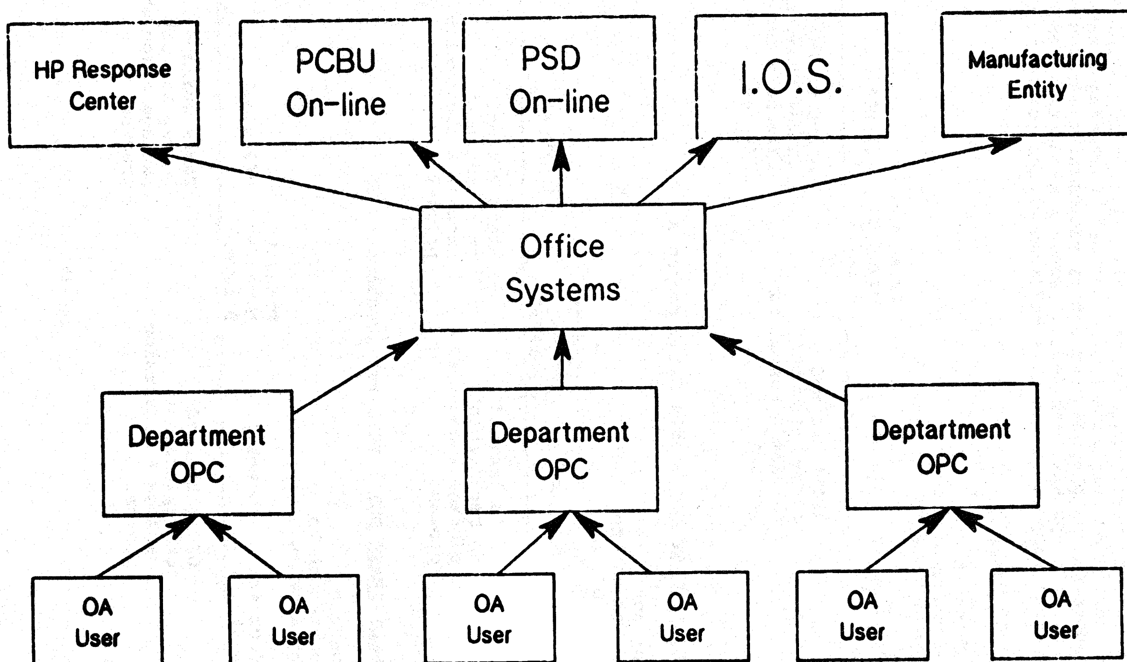
opc15

HEWLETT PACKARD

Benefits of having a department OPC

- Reduction in time spent resolving problems
- Increased productivity
- Increased satisfaction with both systems and employees jobs
- Increased autonomy in resolving problems
- Local office productivity consultant

Office Systems Support Flow



3361-17

PHASE IN/OUT OF OFFICE PRODUCTS

Products may move from one support level to another as demand is increased by user population or there is a product replacement recommended by Corporate I.O.S. and HP's marketing strategy. Other products may be phased-in or out as recommended and tested by a user/Office Systems task force. Phasing products in or out of different support levels will be based on the following guidelines.

PRODUCT PHASE IN

Phase-in
Announcement to
mgmnt & OPC's

Product at B level —(3 mos.)→ Product moves to A level —(3 mos.)→

OPC's trained on new product/release & conversion process

Users trained on new product/release & conversion process

PRODUCT PHASE OUT - 3000 based products

Phase out
announcement to
mgmnt & OPC's

Product at A level —(3 mos)→ Product moves to B level —(6 mos)→ Product moves to C level

OPC's trained on replacement package & conversion process

Users trained on replacement package & conversion process

Application removed from system

PRODUCT PHASE OUT - PC based products

Phase out
announcement to
mgmnt & OPC's

Product at A level —(3 mos.)→ Product moves to B level —(3 mos.)→ Product moves to C level

OPC's trained on replacement package & conversion process

Users trained on replacement package & conversion process

Support Level Factors

- HP Office Automation Strategy
- Volume of user requests
- Office Systems knowledge base

THANK YOU
I wish to thank Lisa Eavey, Jan Slote, Mary Nelson, Luis Hurtado-Sanches Greg Moore, Andre' Rude', Chu Chang, Wayne Innamorato, and Charlie McMenamin for their support, insight and contribution to developing the Customer Support Center OPC program.

HEWLETT
PACKARD

DOS, Advanced Memory Techniques

Interex 1989, San Fransisco CA.

Ref: *\interex\89\paper.hpw*

Author: *Jon Baker*

Version: 1.0

Printed: *May 18, 1989*

Hewlett Packard Ltd
Pinewood Information Systems Division
Nine Mile Ride,
Wokingham, Berkshire RG11 3LL
UK.

Abstract

For those who use today's modern PC software messages such as "*Not enough memory to run*" are probably familiar. Various ways in which to extend your systems performance now exist but they can be both difficult to understand and confusing. Moreover, many people using such techniques do not fully exploit their potential.

This paper provides an invaluable reference to those who are using advanced DOS software such as Microsoft Windows. It provides a grounding in common terminology used, full explanations of the DOS extensions that are available as well as advising in the best ways in which to use them.

Topics covered include an overview of the 8086 and 80286 processors, expanded and extended memory, LIM 3.2 versus 4.0, using high memory with the 286, a study of how Microsoft Windows uses memory and a look at configuring a system for Windows. We also have a brief look at the 80386 processor and the new PC operating system OS/2.

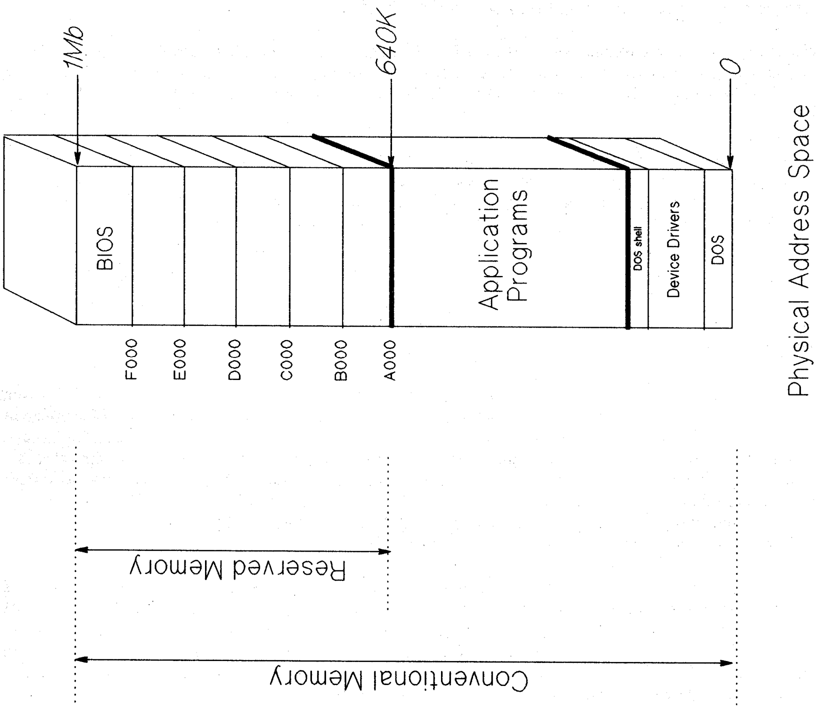
DOS, Advanced Memory Techniques is written for experienced users of MS-DOS who wish to utilize the many varied techniques now available to increase the power of their DOS based PC's. It details information necessary to configure systems that give optimum performance for the hardware available. We look specifically at Microsoft Windows, the windowing environment on which HP's NewWave is built.

History

Every PC workstation contains a certain amount of RAM (Random Access Memory) used to hold program files and data, once loaded programs can be executed by the machines CPU. The amount of RAM supplied as standard with a PC has increased significantly over the past seven years. The original machines had around 128K which at the time was thought to be quite sufficient for most peoples needs. It later became apparent that software was soon going to overrun this limit and so the second generation "XT" compatible machines (which included a hard disk) doubled the amount of RAM to 256K. Since these early days both PC hardware and software have changed dramatically, even the cheapest PC clones are now sold with a full complement of 640K, the maximum amount of base RAM configurable for an 8086 PC.

The address bus of the 8086 uses 20 signal lines to transmit the address of the memory cells and devices attached to the bus, this gives an address range of 1 Megabyte (2^{20}). DOS was designed around the 8086 architecture and inherits the 1Mb address range. The designers of the PC memory map allocated a maximum of 640K of this space for RAM, the remaining 384K reserved for use by devices and peripherals such as the display. Advanced PC software has continued to push the memory usage of applications further up, so much so that even 640K is now regarded as a limitation, ***the DOS 640K barrier***.

Figure 1: *PC Memory Map*



Expanded Memory

Terminology

The complete 1Mb address space of a PC is known as **Conventional Memory**, within that space the area from 640K up to 1Mb is called **Reserved Memory**. We usually view reserved memory as six contiguous blocks of 64Kb, note also that a popular convention for expressing addresses within the 1Mb address space is in terms of 16byte segment paragraphs e.g. physical location F0000 (approximately 983K) would be expressed as F000. In reserved memory this means that the six 64Kb blocks are located at A000, B000, C000 etc. (often abbreviated to A Block, B Block, C Block...). Figure 1 illustrates a typical PC memory map.

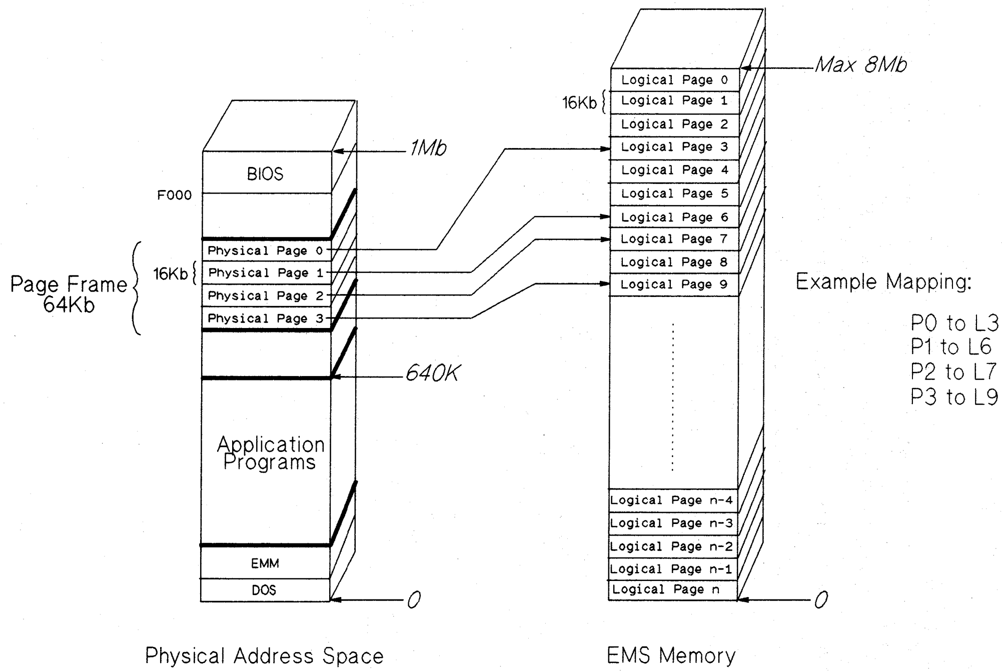
We highlighted the problem that many users face with the 640K limitation of DOS. In 1984 a group of manufacturers consisting of the Lotus Development Corp, Intel Corp and Microsoft Corp (known as the LIM group) announced a new technique for overcoming the 640K barrier with what they called **Expanded Memory**. The group devised the Expanded Memory Specification (EMS), an open specification with a programmatic interface for application designers. The first publicized draft of the specification was LIM version 3.2.

LIM 3.2

Expanded memory is RAM resident on a separate card that plugs into a standard PC slot. Remember that the 8086 can only access locations within its 1Mb physical address space, to access RAM on the card a segment of memory within the physical address space is allocated as a window into expanded memory. This window is called the **Page Frame**. Control and access to expanded memory is handled by an **Expanded Memory Manager** (EMM) which is implemented as a DOS device driver.

Figure 2 illustrates how expanded memory maps onto the physical address space. Note that expanded memory is divided up into fixed size pages, these **Logical Pages** are usually 16Kb in size although later revisions of the specification do cater for smaller pages sizes. LIM 3.2 demands that the page frame be located in a single 64Kb block of memory situated in reserved memory (i.e. between 640K and 1Mb), the 64Kb block is then divided into four 16Kb **Physical Pages**. An application can make calls to the EMM to allocate it a number of logical pages on the card. To access the pages the application must further ask the EMM to map the logical pages onto physical pages within the page frame. With this arrangement although the address space for DOS is still fixed at 1Mb applications can now access a further 8Mb of LIM 3.2 expanded memory. Figure 2 illustrates how LIM 3.2 works.

Figure 2: *LIM 3.2 EMS*



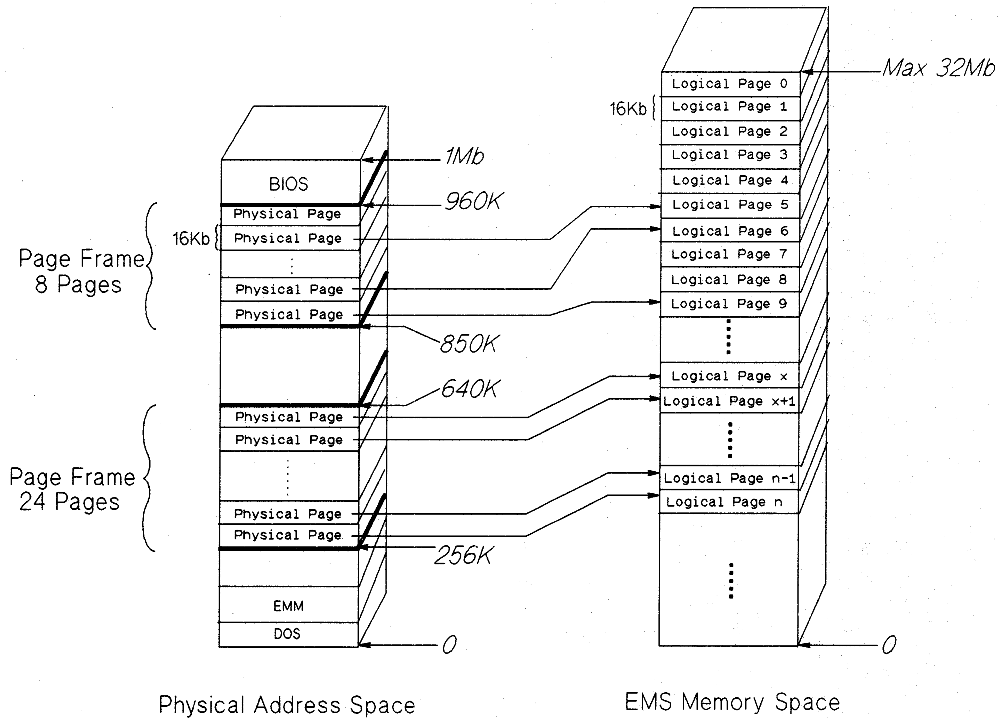
LIM 4.0

Although LIM 3.2 provided a lifeline to applications that were extremely tight on memory it had a number of limitations, we have already mentioned the fact that the page frame was limited to a single 64Kb block in reserved memory and that the maximum amount of memory addressable as expanded was 8Mb. AST Research Inc. attempted to alleviate some of these problems when they introduced their **Enhanced Expanded Memory Specification** and produced new EEMS cards that supported this specification. In 1987 both EMS and EEMS specifications were superseded by a revised specification from the LIM group called **LIM 4.0.** It was a derived from EMS 3.2 and EEMS and therefore compatible with existing EMS and EEMS cards. Note however that certain advanced features of LIM 4.0 need specific hardware support from the card i.e. an EMS 4.0 compatible card. LIM 4.0 has now become the accepted standard for expanded memory and is the one we will examine further.

One of the key contributions of LIM 4.0 is the extensions made to the definition of the page frame. A page frame can now reside anywhere within the 1Mb physical address space with a maximum of 8 pages above 640K but any amount of physical memory below 640K. The window into EMS memory may now consist of a number of separate frames within the 1Mb address space. Note that a page frame cannot be created below 640K unless a frame exists above 640K. We should also introduce another piece of commonly used terminology: the original EMS 3.2 support of a single 64Kb page frame in reserved memory is known as **Small Frame** mode whilst the EMS 4.0 support of large multiple page frames is known as **Large Frame** mode. Figure 3 illustrates a typical LIM 4.0 configuration.

Other enhancements in the LIM 4.0 specification include an improved mechanism for executing programs resident in EMS and the ability for applications to dynamically change the amount of EMS memory allocated to them.

Figure 3: LIM 4.0 EMS



System Configuration

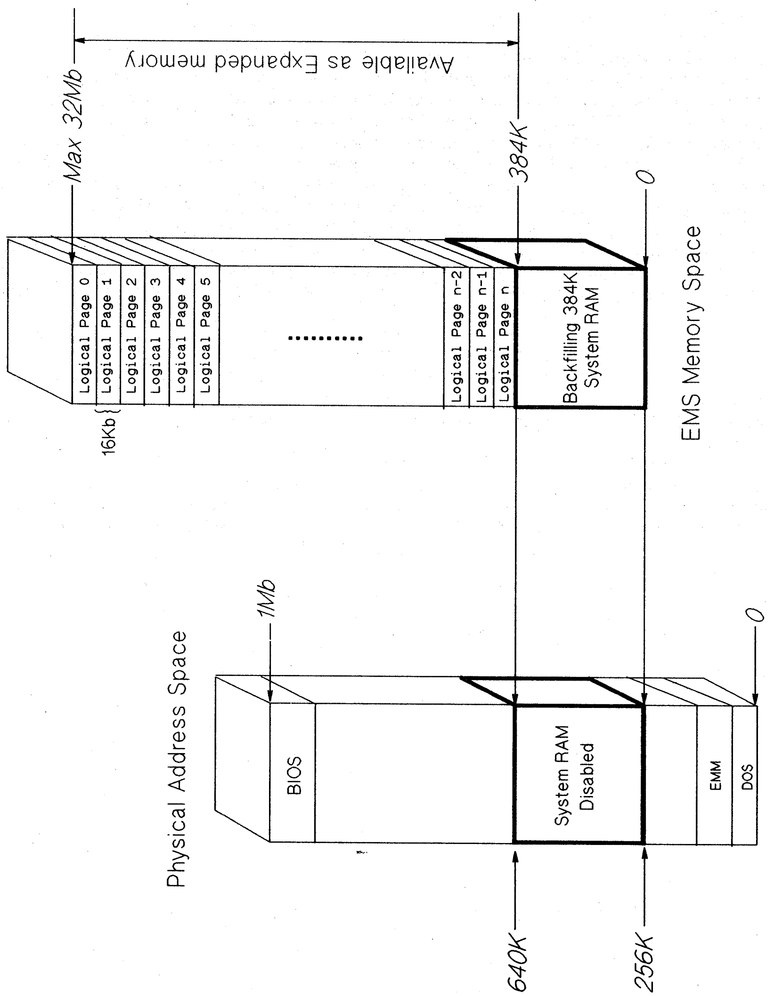
Let us consider how we might configure an ES/12 to use EMS memory. The HP 45944A Vectra ES Expanded Memory Card provides full support for both EMS 3.2 and 4.0. It is installed in a special slot in the ES/12 designed specifically for this card (other cards such as the AST RAMPage 286 fit into a normal slot). We described earlier how EMS 4.0 supports large/multiple page frames that can be located anywhere within the 1Mb address space. The ES/12 allows us to create page frames from 256K upwards, however to achieve this we need to disable the system RAM that presently occupies the space from 256K up to 640K. This is done by setting switches 1 and 2 in switch bank 3 on the PC's processor board (detailed in the ES setup manual, volume 1). To complete the configuration we also need to set switches on the ES EMS card to **Backfill** the memory from 256K to 640K in the physical address space. When the PC boots up it will find the usual 640K of RAM but now only the first 256K comes from the processor card with the remaining 384K from the EMS card. You will notice that the 384K of RAM on the processor board that we disabled is unfortunately unusable, however the performance improvements to be gained by using EMS 4.0 with large frames far outweighs this redundancy.

The reason for backfilling memory is that the 8086 cannot permit more than one block of memory to be located at the same physical address (e.g. blocks of system RAM and EMS RAM located at the same address). However once part of the EMS address space occupies the physical address range from 256K to 640K we can freely choose which areas of memory on the EMS card should fill this space. Note that backfilling is unnecessary with LIM 3.2 since the page frame is mapped to an *empty* space within reserved memory.

The block at E0000 in the physical address space is reserved for use by option ROMS that some accessory cards have, however many users do not have cards of this sort and instead can use this area as extra page space. To take advantage of this you have to inform the PC that you are disabling the option ROM block by setting switch 1 in switch bank 1 on the processor board to OFF (more details in the HP45944A manual) and add the parameter 'R' to the line in the CONFIG.SYS file that invokes the EMM.

Figure 4 illustrates an ES/12 configured for backfilling memory with the option ROMS disabled.

Figure 4: *Backfilling Memory*



Extended Memory

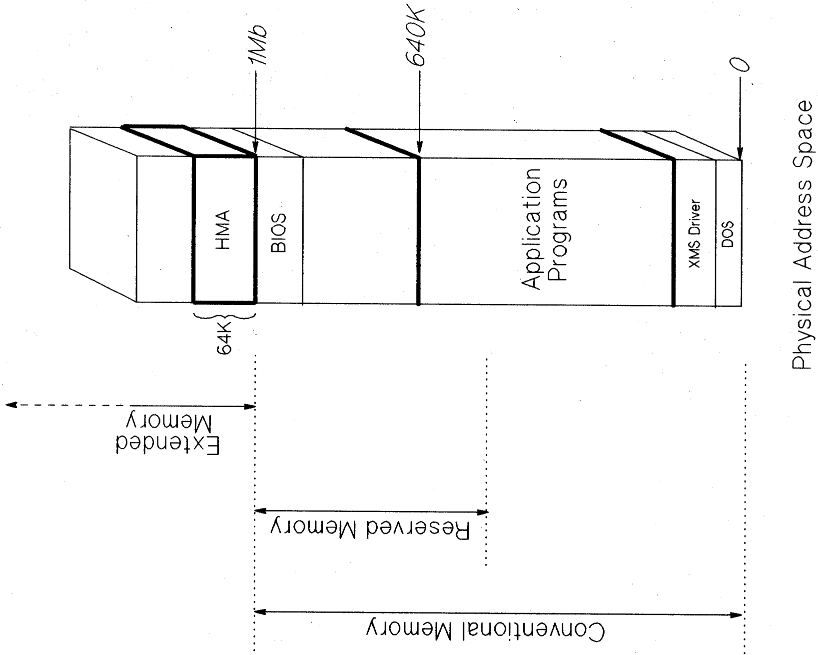
You will recall that the 8086 has a maximum address space of 1Mb and that DOS inherits this limitation, even when running on newer processors such as the 80286 and the 80386 DOS runs in **Real Mode** which is an 8086 emulation built into both CPU's. Both the 286 and 386 processors have more address lines than the 8086 and when used in some of their other modes can address locations beyond the 8086 1Mb address range. Memory located above 1Mb is referred to as **Extended Memory**. As you can see it is vastly different to Expanded Memory but unfortunately the two are often confused.

Most people also believe that the 286 and 386 processors are still limited to the same 1Mb address space when executing in real mode. However with either processor it is actually possible to address the first 64K of extended memory as well. This extra memory space is known as the **High Memory Area** (HMA). It is actually 64Kb minus 16 bytes in size and is accessed by enabling the A20 address line of the 286 and 386 whilst running in real mode. It is this ability of the 286 and 386 to let us enable A20 in real mode that allows us to access this extra 64K without wrapping back to the beginning of conventional memory.

There are a few points to note about the HMA, the main one is that the it cannot be subdivided into smaller areas and hence can only be used by one application at a time. The HMA is also difficult to manage: a mechanism is needed to be able to reserve it, prioritize its use so that its space is used more efficiently and prevent any special applications that use extended memory (e.g. RAM disc utilities) from trashing its contents. It was these difficulties that led Lotus, Intel, Microsoft and other software vendors to publish the **DOS Extended Memory Specification** (XMS) to assist DOS programmers in using the HMA more effectively. The specification also provides support for applications to store data in other areas of extended memory although we will not be covering that here. To use HMA you need an XMS driver such as the Microsoft one called HIMEM.SYS, this is installed by simply including a device reference to it in the CONFIG.SYS file i.e. DEVICE=HIMEM.SYS. For developers who wish to use the HMA in their own programs an XMS Developer's Kit is available from Microsoft. Figure 5 illustrates where the HMA is located.

Note that the HP 45944A EMS card described in the previous section can be configured so that part or all of its RAM is allocated as extended memory.

Figure 5: *High Memory Area*



Microsoft Windows Memory Management

Microsoft Windows is a good example of an environment that over the years has been modified to take advantage of the techniques that we have discussed, in this section we will briefly examine how the Windows Memory Manager uses these techniques.

Virtual Memory

Because Windows supports non-preemptive multitasking it often has a number of applications loaded simultaneously, on a machine with only 640K of RAM this can quickly fill up. Originally to help overcome this restraint a virtual memory mechanism was incorporated that allowed code and resource segments that were marked *Discardable* to be destroyed on a *Least Recently Used* basis, the segments were read back from disk when later required. Note that with this scheme the segments were not saved and hence only read only segments could be discarded. Unfortunately most Windows applications also have a fixed overhead of memory that cannot be discarded often meaning that two large applications can still not be loaded simultaneously.

It was Windows 2.0 that contained the key enhancements to use expanded memory to bank Windows applications (Windows 1.X could only bank DOS applications). Using an EMS 4.0 card and driver the Windows memory manager allocates a new *bank* of memory for each application that is loaded. The bank consists of the *minimum* number of logical pages from EMS memory needed to load the application ensuring efficient use of the EMS memory space. Windows maintains a *bank area* in memory that contains the bank pages of the currently active application. As the focus moves from one application to another the first application is *banked out* by mapping in the pages for the second application into the bank area. Only one application has control of the bank area at any time. The virtual memory mechanism mentioned above still operates and can discard segments that are both resident within the physical address space and are marked discardable, this space includes the EMS memory currently mapped in. Objects that are banked out are never discarded.

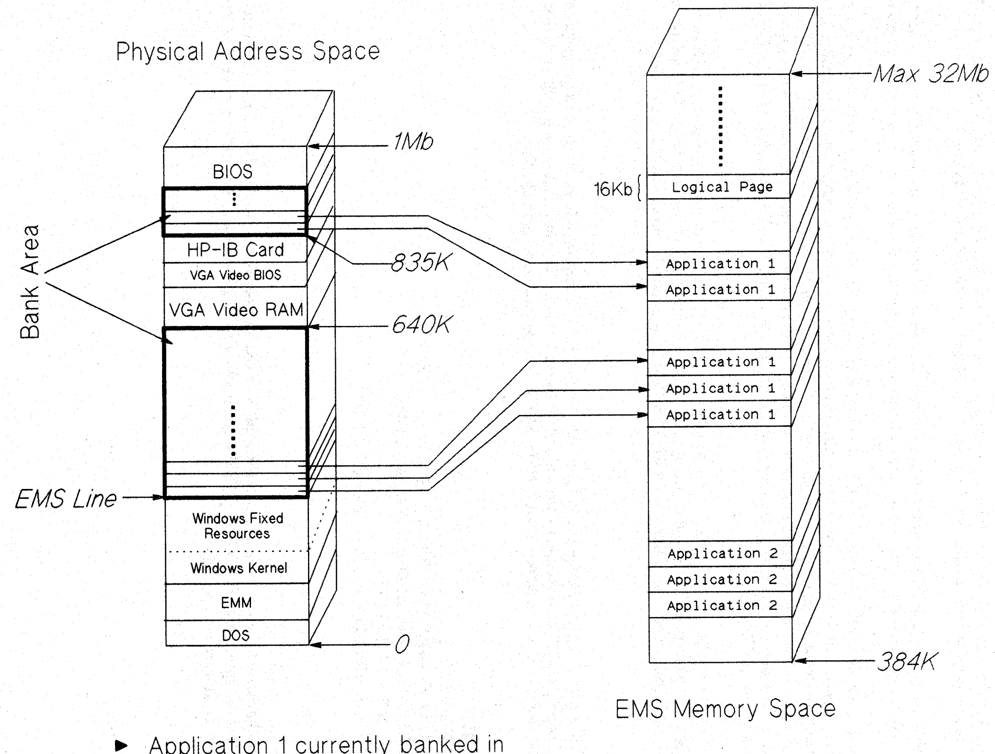
Bank Line

Windows maintains a number of resources that are not banked to EMS memory e.g. Library data segments, Library resources, Thunks etc. Windows maintains a division between bankable and non-bankable areas of memory known as the **Bank Line** or **EMS Line**. The Windows memory manager has the ability to use EMS memory in both small and large frame modes, the positioning of the EMS line determines which mode Windows will use. If Windows cannot position the line such that there is 208K available above the line and 288K below it will operate in small frame mode (Microsoft decided that if these amounts were not achievable that it would be more efficient to use small frame EMS instead of large frame). With Windows 2.11 you can alter the 288K figure by using the parameter /Ennn when you invoke Windows (where nnnKb is the space desired above the line). Windows users should try to configure their machines such that Windows has a good chance of using large frame mode and hence achieve better performance.

Windows can also use the HMA which gives it approximately another 50Kb to use. This is useful breathing space if you happen to be on the border between using large frame or small frame mode.

Figure 6 shows an ES/12 configured for Windows and running in large frame mode.

Figure 6: *ES/12 Windows Configuration*



Windows Configuration

There are a number of steps you can take to ensure that your machine is configured to run Windows at peak performance. Again we will consider configuring an ES/12 with an HP 45944A EMS card.

Firstly ensure that in CONFIG.SYS you only load drivers that are used regularly in day to day use (e.g. LAN driver). You should consider removing drivers that are used infrequently installing them only when necessary.

Attempt to make as much page space available to the EMM as you can. We have already seen a few ways of doing this, in particular see if you can use the option ROM space. Also try to compact the other spaces being used in reserved memory, you can often configure the location of a particular card in reserved memory by setting its DIP switches. When making any of these changes you must inform the EMM of what bank space is available. This is done by specifying parameters on the line in the CONFIG.SYS file that invokes the EMM (see the EMS manual for full details).

Also ensure that you have HIMEM.SYS loaded and do not allow any TSR's or drivers to grab the HMA, leave it for Windows. Remember this valuable saving can make the difference between switching into large frame mode or small frame mode particularly when large TSR's such as network software have been loaded.

Figure 6 illustrates a typical configuration for an ES/12. Note that it has two adapter cards occupying space in reserved memory: HP VGA Adapter card and HP-IB Interface card. Note also that we are backfilling from the EMS card from 256K to 640K. The CONFIG.SYS file would look something like this:

```
FILES=20
BUFFERS=16
COUNTRY=44,,C:\COUNTRY.SYS
SHELL=C:\COMMAND.COM C:\ /P /E:600
DEVICE=HIMEM.SYS
DEVICE=HPEMMGR.SYS R O X=C000-CC00
.
.
.
```

Note that the expanded memory manager (HPEMMGR.SYS) includes the parameter X=C000-CC00 which ensures that the space between C0000 and CC000 is not allocated as a page frame since this is where the VGA BIOS and HP-IB card are located.

80386 Machines

The 80386 processor has paged memory management built into the hardware, most manufacturers that provide EMS support on 386 machines make use of this powerful feature. Instead of using a special EMS board a 386 PC can use software to emulate expanded memory using extended memory. Obviously the 386 has to be configured with as much extended memory as you wish to make available as expanded. Because paged memory management is built into the CPU the performance of the emulation is as good as that of a normal EMS driver and card, it is also not necessary to backfill any of the memory below 640K.

Many of todays 386 EMS drivers have XMS support built-in and therefore do not require HIMEM.SYS to be loaded.

OS/2

OS/2 is the new multitasking operating system from IBM and Microsoft designed specifically for the PC. Contrary to popular belief OS/2 will run not only on PS/2 machines but on any AT compatible. There are two flavors of OS/2 available, **OS/2 Standard Edition** is the base version which provides the operating system and will soon include Presentation Manager, OS/2's windowing interface. Most PC vendors sell their own version of OS/2 Standard Edition that will comprise of the standard Microsoft product together with device driver support for their own peripherals. IBM have decided that as well as selling Standard Edition that they will produce their own version of OS/2 called **Extended Edition** which is basically the Standard Edition kernel with three additional components: Communications Manager, Database Manager and the LAN Requestor. Microsoft have decided not to produce their own "Extended Edition" but instead have formed agreements with other companies such as 3-Com, Ashton-Tate and Sybase to provide these extra services.

OS/2 requires a 286 or 386 based PC on which to run, operating in 80286 16-bit **Protect Mode.** This particular mode includes support for multitasking and virtual memory. This means that the 286 processor can now take advantage of all 24 address lines allowing it to address 16Mb of memory (2^{24}), considerably more than the 8086. At present OS/2 will run the 80386 in 16-bit Protect Mode where it functions in the same way as the 286 Protect mode. Microsoft have already announced that they will be producing a version of OS/2 designed specifically for the 386 which will run in its native 32-bit Protect mode.

There has been a great deal of discussion with regard the memory requirements of OS/2. For OS/2 SE 1.0 which does not include Presentation Manager the minimum recommended memory requirement is 1.5Mb, to ensure that the amount of swapping to disk is kept to a minimum I would actually recommend increasing this to 3Mb. OS/2 SE 1.1 will include Presentation Manager and as such requires more memory than 1.0, the minimum recommended amount of memory for 1.1 is 3Mb but again if multiple OS/2 applications are running the system will operate much better with 5Mb. If you are running OS/2 applications only and find yourself tight on memory then you may want to consider disabling the DOS compatibility box, a feature in OS/2 that allows you to run your old DOS applications. This can be achieved by including the line PROTECTONLY=YES in the OS/2 CONFIG.SYS file and should give you back some of the space in the first 640K of memory.

WHAT AN HP3000 USER CAN DO WITH A LASERJET

Roger Lawson
Proactive Systems Inc.
339 South San Antonio
Los Altos
CA 94022
(Tel: 415-941-9316)

ABSTRACT

The HP LaserJet when used with a PC has created the phenomenon of Desk Top Publishing. The author will argue that a similar revolution is about to take place for HP3000 users.

The conventional computer line printer is now obsolescent and can be replaced by one or more spooled LaserJets. With suitable software, users can easily create "electronic forms" to replace preprinted stationery and make all their line printer output look like professionally typeset material for no extra cost. This can include graphics as well with little extra effort.

The auther will describe how the technology can be applied and look at the various approachs that have been taken in building such software. The way it can be easily integrated into existing HP3000-based commercial applications will be explained.

To the junk heap

I believe the conventional computer line printer will be on the junk heap in a couple of years time. The chain, band, and dot matrix printers are inflexible, noisy, cumbersome to operate and less reliable than laser printers such as the HP LaserJet Series II or LaserJet 2000.

LaserJet Advantages

With LaserJets you get total flexibility in layout and typeface selection. Why stick to the fixed pitch typewriter-like look of a conventional computer printer when you can make your output look professionally typeset for no extra cost?

With LaserJets you can place graphics (e.g. bar/line charts, logos, drawings, etc.) among your text or data to enhance its quality and impact. And every copy comes out clean, crisp and clear.

LaserJets are silent, use standard sheet paper and changing paper is easy. Compare the ease of use of a LaserJet Series II and a common conventional HP printer such as the 2932 to see what I mean.

You can use a small font and print more information on less paper (and, on some models, print on both sides of the paper) so as to save paper and hence money. In fact the running costs (e.g. consumables, paper , maintenance) can be less than conventional computer printers.

Incidentally you can of course use an ordinary LaserJet as a spooled device on an HP3000 and the print speed is faster than many people realise. For example at 8 pages per minute (p.p.m.) a Series II LaserJet can be equivalent to a 400 lines per minute (l.p.m.) conventional printer. The LaserJet 2000 goes to 1200 l.p.m. and you still have the 2680 for even greater output.

The Software Problem

However most LaserJet users are not taking full advantage of these printers because of the difficulty of using their sophisticated capabilities. Unless you have some special software, the only way to format output is with long, complex escape sequences - for other than simple applications this is simply not practical or economic in labour.

What do we need in software to drive them from the standpoint of the commercial HP3000 user? My list would be:

■ Flexible font control and typesetting capabilities, e.g. the ability to justify text in a proportional type face, do multi-column output, do automatic headers and footers, etc.

■ Graphics functions such as drawing lines and boxes plus the ability to draw commonly used figures such as line, bar and pie charts. This should also extend to the easy design and production of "forms", i.e. grid based artwork (see Figure A for an example).

■ The ability to control or drive this software from existing programs written in COBOL, 4GLs, report writers, etc.

■ The ability to intercept printer output from existing programs so that source code changes are not required.

■ Software that is shareable by multiple users, that runs efficiently (due to the high volumes of data to be processed) and that can be set up to run in batch jobs without operator intervention.

Note that Desk Top Publishing (DTP) packages as used on PCs do not meet several of the above criteria so we really need a different "animal" for the typical HP3000 user.

Electronic Forms

A large volume of the paperwork in most organisations is a "form" which is basically a grid with some typeset text. The grid is used to logically organise the material and to enable a "denser" layout than would otherwise be acceptable.

If computer printing is required on a form then "preprinted stationery" is obtained where the form is printed on continuous paper. Such paper has to be specially mounted on a printer and this makes it impractical to do for one-off sheets.

Both types of form are expensive to design and print, they are expensive to distribute and hold in stock, and a lot of them get scrapped when the form needs to be changed!

Also the form is fixed in layout - you can't change it from page to page. A simple example of this requirement is an invoice on which a total line is needed - however if an invoice for one particular customer extends over several pages you only want the total printed on the last page.

All of these problems can be solved if you hold the form as an "electronic" equivalent and only print it when required. If the form needs amendment (e.g. maybe your company address has changed) this is a trivial exercise.

You can print a "blank" form; or you can set it up so that a user can fill out the form on a terminal and print it when finished (or you can get the form filled automatically by the computer from its data base before printing).

LaserJet Macro Capability

One reason why forms can be efficiently produced on a LaserJet is because they can be downloaded and held in the LaserJet memory (more than one can be held depending on the amount of memory). Therefore the form and data can be merged in the LaserJet rather than on the host system.

By holding the "form" as electronic software it is also easy to pass it around the organisation and to enforce standard usage. Figure B is an example of such an electronic form designed and printed on an HP3000.

Improving Reports

It is easy to use simple graphic design tricks such as lines and boxes to improve the appearance of computer output reports enormously. Also putting it into a decent typeface makes it more legible.

Look at figures C and D for example. The one printed on the laser printer didn't require much more effort to design and specify but it certainly looks a lot better - also the 2-column layout saves paper. Note that with suitable software you can still produce the laser printed report using a report writer such as QUERY, QUIZ, Q-GEN or BRW. You could also produce it from a COBOL program.

Dynamic Forms

By printing forms on a LaserJet using appropriate software, you can actually vary the content of the form from page to page. A couple of examples from our own customers are as follows:

■ An application form for insurance is varied depending on the basic information already known about the applicant (e.g. whether he is an existing policy holder or not) - some parts of the form that are therefore irrelevant are omitted.

■ A company that distributes office consumables, supplies each of its customers with an order form which they can distribute internally. Only the items that the customer has agreed are relevant to his organisation are listed on the form (these items are listed in the suppliers data base and the agreed purchase terms also specified). As the supplier has many customers, there are many different versions of the form which are produced automatically. If an additional item is to be added to the "orderable" list then the program that prints the order form automatically adapts it.

Typesetting Capabilities

A similar usage of such software is to produce customised letters (e.g. mailing letters, dunning letters, etc) where the words to be inserted in a standard letter vary but the output needs to look like it was printed specially for the recipient. An example of this is shown in Figure E (the original data and standard letter from which this is made up are shown in Figure F).

At this point it is worth explaining some basic concepts. The following text is printed in a fixed pitch Courier font which is a typical typeface used by typewriters or conventional computer printers.

```
This text is an example of a paragraph that
is printed in different typefaces to show
the varying capabilities of different fonts
and the ability of typesetting software to
format the text in a better manner.
```

However on a LaserJet you can use professional proportional typefaces such as Times Roman or Helvetica. The example below is the same text in Helvetica 12 point (the size of type is measured in points there are 72 points in an inch). It is called a proportional font because the amount of space ocupied by each character in the alphabet varies. Also to make it look even better we can use kerning (varying the space between characters depending on the character pair) and justification (in this case to both margins which means the space between words is varied also). The sample text then appears as:

> This text is an example of a paragraph that is printed in different typefaces to show the varying capabilities of different fonts and the ability of typesetting software to format the text in a better manner.

Note also that the type size and form can be varied within a line thus adding further complexity. Obviously with this dynamic formatting process you also need automatic hyphenation for long words as the person entering the text cannot easily predict where hyphenation will be required.

Fonts on the LaserJet can come from three sources: in-built, cartridge or downloaded soft fonts. The last is the most flexible although it obviously takes some time to download. Only on the LaserJet 2000 are proportional fonts built-in. In practice the standard LaserJet memory is sufficient to hold a number of fonts in commonly used sizes so a single download is usually sufficient. Incidentally some software packages include the commonly used proportional fonts.

To cover all the above, and more, in a flexible and sophisticated manner, requires good software. The quality of software packages aimed at LaserJet formatting varies greatly in the quality and sophistication of facilities in this area so you should look carefully at that - even a simple form tends to contain a lot of text so such capabilities are important.

Graphics

Graphics is still one of the Cinderellas of the commerical computing world. Although there are many products available with powerful facilities, it is a technology that still hasn't taken off in popularity among HP3000 users. The reasons for this are not difficult to deduce. Firstly the original emphasis of much graphics software was in producing complex artwork with a lot of flexibility and manual control. It has actually been difficult to set up and integrate it with batch reporting systems. It also used specialised, expensive and slow devices such as plotters and graphic terminals.

Now with LaserJets being both common and relatively cheap all that is required is suitable software. However you must pay attention to the way you produce graphics on a LaserJet if you want to get good throughput.

There are two ways of drawing graphics on a LaserJet. The first way is to use raster graphics where each dot to be printed is represented by a binary value. Building up a picture in this way requires transmitting a large amount of data to the printer (bearing in mind that a LaserJet prints 300 dots per inch) so using this method for routine production graphics is not a good idea, for example it could take several minutes to print one page.

The second method is to use the PCL command language built into the printer. This effectively enables you to draw lines. It cannot achieve as complex a drawing as raster but for bar, line and pie graphs it is fine.

This is a point worth considering when looking at graphics packages to use with a LaserJet as many of them use raster graphics.

We integrated graphics capabilities into our software so we can easily mix graphs with text and forms. An example of such output is shown in Figure G. This is the ideal for production jobs where human "massaging" of the graph is not needed. However, as HP has recognized, for manipulating graphics it is better to do that on a PC so we also support graphics prepared on a PC by packages such as HP Gallery or Harvard Presentation Graphics and then uploaded to the HP3000. We therefore support HP Figure Files created by the older HP software packages for the HP3000.

Logos

A particular requirement when printing forms is often to include a corporate logo. This can often be drawn with PCL based graphics. Alternatively it can be scanned as a raster graphics using an HP ScanJet attached to a PC and uploaded to the HP3000.

What's Different About the HP3000 User

If you look at software to support LaserJet printers (ie. which can provide the functions described above) it may at first appear that a PC based product would provide a solution. However the needs of an HP3000 user can differ considerably from that of a PC user and the software must differ accordingly. For example the typical HP3000 user has the following characteristics:

■ Large data volumes so performance is a priority (a good question to ask is whether the software can keep up with the 20 p.p.m. speed of a LaserJet2000 when formatting combined text, graphics and forms).

■ Multiple spooled devices, i.e. possibly several LaserJets of different models.

■ Lots of terminals (of which some will be PCs but some will also be "dumb" terminals) which should all preferably be able to use the software on a shared basis.

■ A requirement to interface to (and be useable by) software written in COBOL, 4GLs (such as Powerhouse, Speedware, Transact), or report writers such as QUERY, QUIZ, Q-GEN, BRW etc.

■ The ability to format output from existing application packages for which the source code may not be available.

■ The software must be capable of being used in an operational environment with minimal staff intervention, i.e. can run in job or session mode.

■ System management functions should be available to control and configure output devices and otherwise manage the shared resource.

Management Systems

So as to maximise system throughput and to meet the management control requirement, we incorporated extra facilities to cover these needs. For example we track forms and soft fonts that have been downloaded to each LaserJet so as to avoid repeated downloading. We also only download the ordinary ASCII character set rather than the full Roman-8 set unless the latter is specifically required. We also place forms into a catalog for ease of use and reference.

LaserJets for Quality Output

If you use LaserJets instead of conventional computer printers you can use the facilities described in this article to improve the appearance of all the printed material you produce. If you are circulating material outside your organisation you can get a higher quality image for no extra cost, or you can save external typesetting costs by no longer using a professional print shop.

Even if your d.p. output is only used internally it can be worth a lot to enhance its quality. After all, the judgement of your management on whether they get value for money from their d.p. operations may depend on their opinion of the quality of the material they receive!

FIGURE A

SUPARITE INC

Stationary Requisition Form

Suparite Inc
8855 N. 55th St.
Milwaukee
WI 53223
Tel:(414) 355-3501

QTY REQ.	UNIT	PROD CODE	DESCRIPTION	MIN QTY
Paper Products				
	Quire	04555	Blotting Paper White (25 sheets)	
	Each	52020	Book Shorthand 300p 8x5	
	Each	52023	Book Stiff cover Feint A5	
	Each	52025	Book Stiff Cover Feint A4	
	Ream	52231	Duplicator Spirit Paper A4 White 80gsm	
	Ream	52237	Duplicator Spirit Paper A4 White 70gsm	
	Ream	52327	Bank Paper SupaRite White A4	
	Ream	52349	Bond Paper White A4 70gsm	
	Ream	52354	Bond Paper White A4 70gsm	
	Each	52529	Book Shorthand 160p 8x5	
	Each	52550	Book Stiff Cover A-Z index A4	
Envelopes				
	Each	52684	M/W Manilla Gummed 381x254	
	Each	52720	L/W Manilla Gummed C6	
	Each	52721	L/W Manilla Gummed Windows C6	
	Each	52722	L/W Manilla Gummed DL	
	Each	52723	L/W Manilla Gummed Windows DL	
	Each	52726	L/W Manilla Gummed 241x165mm	
	Each	52755	White Gummed 110x220	
Filing				
	Each	02118	Storage Case, Carrier Bag Type	
	Each	02524	PVC Punched Pocket A4 Clear	
	Each	02551	Ring Binder A4 Black 2 Ring PVC	
	Each	02552	Ring Binder A4 Blue 2 Ring PVC	
	Each	02553	Ring Binder A4 Read 2 Ring PVC	
	Each	02554	Ring Binder A4 Green 2 Ring PVC	
	Each	21430	Manilla Folder L/W S/Cut F/S Buff	
	Each	52818	Flat File F'Scap Buff	
	Each	52847	Lever Arch File A4 Upright	
	Each	52852	Box File Rigid F'Scap A4	
	Each	52853	Lever Arch File Dustcover A4	
	Each	52922	Manilla Folder L/W S/Cut F/S Pink	
	Box	55255	Contract Suspension File Complete(50)	
Ribbons & Typing Accessories				
	Each	02813	Liquid Paper Thinners	
	Each	02814	Liquid Paper Fluid White	
	Each	02852	Tippex Paper White	
	Each	21006	Liquid Paper Green	
	Each	21007	Liquid Paper Pink	
	Each	21008	Liquid Paper Yellow	
	Each	21055	Liquid Paper Blue	
	Each	55248	Pentel Correction Pen	
Pens and Markers				
	Each	55055	SupaRite Pencils - B	
	Each	55332	Pilot Drymarker Black	
	Each	55334	Pilot Drymarker Blue	
	Each	55335	Pilot Drymarker Red	
	Each	55336	Pilot Drymarker Green	
	Each	55552	SupaRite Pencils - H	
Stapling & Stamp Pads				
	Bottl	55514	Endorsing Ink Bl ack	
	Each	55609	SupaRite Staple Extractor	
	Box	55712	SupaRite Staples 26/6 (No.56)	
	Each	76681	SupaRite Full Strip Stapler	
Tapes & Packing				
	Roll	04207	Dymo Tape 9mmx3mm Green	

QTY REQ.	UNIT	PROD CODE	DESCRIPTION	MIN QTY
	Roll	04208	Dymo Tape 9mmx3m Black	
	Roll	04209	Dymo Tape 9mmx3mm Red	
	Each	04218	Buff Tag Manilla Label Strung No. 5	
	Roll	04313	Carton Closure Tape 51mmx50m	
	Ball	04413	String Cotton Med 1/2 Kilo	
	Roll	08737	Clear Tape 12mmx66m	
	04313	08738	Clear Tape 25mmx66m	
	Roll	09309	Clear Tape 12mmx33m	
	Roll	09310	Clear Tape 25mmx33m	
Adhesives, Clips & Fasteners				
	Each	04108	Foldback Clip 19mm	
	Each	04109	Foldback Clip 32mm	
	Box	04125	Treasury Tag 127mm (100)	
	Box	04128	Treasury Tag 51mm (100)	
	Each	04520	Pritt Glue Stick 34 Medium	

ORIGINATED BY

Name (Person)

Address(location/dept)

Tel. No Ext

Branch Authorisation:

Date:

HEADQUARTERS

Stamp of Approval

Suparite Stationers Inc
A Subsidiary of WIDGET Manufacturing Inc.

Proactive Systems

FIGURE B

INVOICE

WIDGET MANUFACTURING, INC.
8855 N. 55th St. P.O. Box 23056, Milwaukee, Wis. 53223
Telephone (414) 355-3601 - Telex 26-9685

Order No.
Invoice No.
Invoice/Ship Date

TERMS
Net 30 Days
F.O.B. Milwaukee, WI

SOLD TO
AVCO Systems
201 Lowell Street
Wilmington, MA 01887

SHIPPED TO
AVCO Systems
1 Park West, Metro Tech Park
Teaksbury, MA 01876

Date Entered	Customer P. O. Number		Ship Via	
7/18/86	34-546		UPS	

REF. NO.	PRODUCT NUMBER	PRODUCT DESCRIPTION	QUANTITY ORDERED	QUANTITY SHIPPED	UNIT PRICE	DISCOUNT PERCENT	EXTENDED PRICE
01	6 0 41 PO	PH Probe; Submersion Epoxy; 4.5 foot cable; see data sheet S6OPR; Ser. #786-597D	1	1	295.00		295.00

INVOICE TOTAL 295.00

Proactive Systems

FIGURE C

WIDGET SOFTWARE PRODUCTS INC. - PRICE GUIDE

Effective 1 March 1989 Page 1

Product No.	Description	Price ($)
LANGUAGE COMPILERS		
1002348	Pascal Compiler for IBM PC Compatibles	340.00
1002352	Pascal Forms Management Utility	125.00
1010008	Cobol Compiler	555.00
1010009	Cobol Forms Management	75.50
1020044	Basic Interpreter	390.00
1020045	Basic Compiler	450.00
1020046	Basic Compiler Run Time System	99.00
1020047	Basic Compiler Forms Management	240.00
1030031	Fortran Compiler	350.00
1030031	Fortran Graphic Extensions	99.00
1030032	Fortran Language Extensions	75.00
DATA BASE MANAGEMENT SYSTEMS		
1102345	Superbase Relational DMBS	670.00
1102348	Superbase Run Time System	230.00
1102349	Superbase Management Utility	145.50
1102350	Superbase Forms Interface Compiler	340.00
1102351	Superbase Report Writer	265.00
1102351	Superbase SQL Language Module	288.00
1102352	Superbase Run Time Optimiser	350.00
1102353	Superbase Extended System Option 1	199.00
1102354	Superbase Extended System Option 2	199.00
1102355	Superbase Extended System Option 3	199.00
1102356	Superbase Reconfiguration System	75.60
1102357	Fastbase Network DBMS	475.00
1102358	Fastbase Distributed System Option 1	240.00
1102359	Fastbase Distributed System Option 2	240.00
1102345	Superbase/2 Relational DMBS	670.00
1102348	Superbase/2 Run Time System	230.00
1102349	Superbase/2 Management Utility	145.50
1102350	Superbase/2 Forms Interface Compiler	340.00
1102351	Superbase/2 Report Writer	265.00
1102351	Superbase/2 SQL Language Module	288.00
1102352	Superbase/2 Run Time Optimiser	350.00

Prices are subject to change without notice. Prices include delivery to US mainland locations only. Contact your Widget Software Products representative for a specific quotation.

Proactive Systems

FIGURE D

WIDGET SOFTWARE PRODUCTS INC. - PRICE GUIDE

Effective 1 March 1989

Product No.	Description	Price ($)
	Language Compilers	
1002348	Pascal Compiler for IBM PC Compatibles	340.00
1002352	Pascal Forms Management Utility	125.00
1010008	Cobol Compiler	555.00
1010009	Cobol Forms Management	75.50
1020044	Basic Interpreter	390.00
1020045	Basic Compiler	450.00
1020046	Basic Compiler Run Time System	99.00
1020047	Basic Compiler Forms Management	240.00
1030031	Fortran Compiler	350.00
1030031	Fortran Graphic Extensions	99.00
1030032	Fortran Language Extensions	75.00
	Data Base Management Systems	
1102345	Superbase Relational DMBS	670.00
1102348	Superbase Run Time System	230.00
1102349	Superbase Management Utility	145.50
1102350	Superbase Forms Interface Compiler	340.00
1102351	Superbase Report Writer	265.00
1102351	Superbase SQL Language Module	288.00
1102352	Superbase Run Time Optimiser	350.00
1102353	Superbase Extended System Option 1	199.00
1102354	Superbase Extended System Option 2	199.00
1102355	Superbase extended System Option 3	199.00
1102356	Superbase Reconfiguration System	75.60
1102357	Fastbase Network DBMS	475.00
1102358	Fastbase Distributed System Option 1	240.00
1102359	Fastbase Distributed System Option 2	240.00
1102345	Superbase/2 Relational DMBS	670.00

Product No.	Description	Price ($)
1102348	Superbase/2 Run Time System	230.00
1102349	Superbase/2 Management Utility	145.50
1102350	Superbase/2 Forms Interface Compiler	340.00
1102351	Superbase/2 Report Writer	265.00
1102351	Superbase/2 SQL Language Module	288.00
1102352	Superbase/2 Run Time Optimiser	350.00
1102353	Superbase/2 Extended System Option 1	199.00
1102354	Superbase/2 Extended System Option 2	199.00
1102355	Superbase/2 extended System Option 3	199.00
1102356	Superbase/2 Reconfiguration System	75.60
1102357	Fastbase/XL Network DBMS	475.00
1102358	Fastbase/XL Distributed System Option 1	240.00
1102359	Fastbase/XL Distributed System Option 2	240.00

Prices are subject to change without notice. Prices include delivery to US mainland locations only. Contact your Widget Software Products representative for a specific quotation.

FIGURE E

```
\left 1.5; width 5; top 2;
\just left;nowrap
\space 2
^f8^^ma
^mb
^mc
^md

^d2^
\space inch 1
Dear ^me,

\wrap;just both;indent 0.5
Please be aware that your debt of
^mf is woefully overdue at our office.
We have not heard from you since ^mg.
Despite numerous reminders by telephone,
letter, fax and carrier pigeon
we have received no response.

Unless your money is in our hands within
ten days we will proceed to take
possession of your ^mh, ^mi, and ^mj.

\indent 0
Yours truly




R.E. Grinch
```

Proactive Systems

FIGURE F

```
\left 1.5; width 5; top 2;
\just left;nowrap
\space 2
^f8^^ma
^mb
^mc
^md

^d2^
\space inch 1
Dear ^me,

\wrap;just both;indent 0.5
Please be aware that your debt of
^mf is woefully overdue at our office.
We have not heard from you since ^mg.
Despite numerous reminders by telephone,
letter, fax and carrier pigeon
we have received no response.

Unless your money is in our hands within
ten days we will proceed to take
possession of your ^mh, ^mi, and ^mj.

\indent 0
Yours truly




R.E. Grinch
                                                    \new
                                                    \ma="Mr. Mike Tree"
                                                    \mb="456 East Main"
                                                    \mc="Wastebasket"
                                                    \md="CA 99900"
                                                    \me="Mr Tree"
                                                    \mf="$90.23"
                                                    \mg="April 1, 1900"
                                                    \mh="house"
                                                    \mi="car"
                                                    \mj="dog"
                                                    \in letr01
                                                    \new
                                                    \ma="Mr. Ronald Reagan"
                                                    \mb="1205 E. Santa Barbara"
                                                    \mc="California"
                                                    \md="CA 99900"
                                                    \me="Mr Reagan"
                                                    \mf="$1000.99"
                                                    \mg="September 12, 1988"
                                                    \mh="boat"
                                                    \mi="plane"
                                                    \mj="submarine"
                                                    \in letr01
```

Proactive Systems

FIGURE G

WIDGET CORPORATION
Equipment Location
Equipment Area

Equip. Type:	General Pneumatic Compactor	Report Date:	11/17/86
Sample ID:	Z-5555A	Sample Date:	11/15/86
Sample Desc:	Q. R. sub-section 3	Equip. Hours:	243,222.9
Sample Loc.:	Outside	Run Hours:	1,234.5
Gear Type:	Widget Extra Heavy		

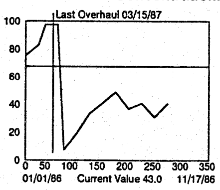
Results:

The data is extracted from the data base and prepared without effort for input to the software programmatically. The software formats the text . . . using one, two, or three columns for output in twelve, ten or eight point type according to the length of the commentary. The formatting is entirely automated.

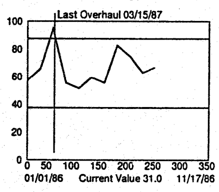
The bar charts and line graphs are drawn on the laser printer along with the text. Processing and formatting of a page of output is handled with minimal load on the HP 3000. The output is sent to a LaserJet+, LaserJet Series II, or LaserJet model 2000 without need for raster data. Transmission and printing of this page can be done with a 2400 baud line in less than 90 seconds; with a 9600 baud line it could be done in under 25 seconds.

The data displayed below is for illustration purposes only. Any relationship to real data or to labels on charts depicting real data is purely coincidental.

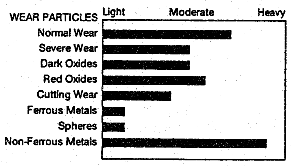
Wear Problem Concentration:

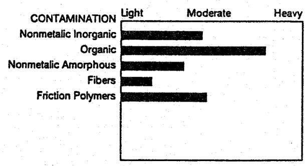


Down Time:



Analytical Results:





Proactive Systems

DESKTOP PUBLISHING:
SOLVING THE INSTALLATION MYSTERY

David A. Swinkin
Hewlett-Packard Company
3 Parkland Drive
Darien, Connecticut
06820

CONTENTS

I. INTRODUCTION

The vast proliferation of low cost laser printers that produce near type-set quality output combined with the introduction of high performance desktop microcomputers has changed the way that businesses produce documents. "Desktop publishing" refers to the use of hardware and software components that allow companies to design and produce their own business reports, newsletters, flyers, direct mail advertisements, brochures, catalogs, data sheets, books, price sheets, manuals, trade journals, etc. Desktop publishing components produce documents at a fraction of the cost of conventional printing methods and allow companies to maintain control over confidential information, publication schedules and virtually the entire production process.

A successful desktop publishing (DTP) installation involves many hardware and software products that allow a user to produce professional looking documents quickly and easily. Merging text from industry standard word processing software, graphics and scanned images in standard file formats no longer requires a publisher that uses very expensive, dedicated publishing systems. Desktop publishing is now considered a general business application, providing a user with enormous capabilities. Any business that has a personal computer, laser printer and optionally, an image scanner along with the commitment to learn basic design techniques can successfully use desktop publishing software.

In the pages that follow, the methodology of integrating the many hardware and software components of a desktop publishing solution into a successful desktop publishing workstation will be defined. In addition, the requirements for installing a complete desktop publishing workstation will be illustrated using Microsoft Windows/286, Aldus Pagemaker 3.0 and laser font options. Recommendations regarding training and literature will be also be presented.

By choosing the components of an MS-DOS personal computer desktop publishing system based on price and performance expectations, a user will not be disappointed in the quality and diversity of documents that are possible. Visually exciting business documents and reports, publications, newsletters and journals generated by in-house personnel are now extremely popular and will continue to replace conventional printing methods as new technology emerges.

II. DESKTOP PUBLISHING COMPONENTS

Configuring and choosing hardware and software for desktop publishing requires a DTP user to set objectives regarding performance and output expectations. A user can purchase a low cost 80286 personal computer and a laser printer and produce high resolution, quality output. An initial investment of $8,000.00 in hardware and DTP software can save a company thousands of dollars in typesetting costs by producing simple, camera ready output. It is not unusual for a company using DTP output to save $50.00 per page or more on layout and design, typesetting, proofreading, camera work and final paste-up. It is possible to produce a document 50 to 60 percent faster with software tools that are available today. Faster personal computers, alternative printer interfaces, image and OCR scanners, additional font capabilities and high-performance, high-capacity disk drives can be added later to produce sophisticated and visually exciting documents at a higher rate of speed.

The higher performance 80286 (12 Megahertz clock speed) and 80386 (16,20,25 Megahertz) machines allow a user to create documents considerably faster. The ability inherent in these high-performance, more expensive computers to quickly switch from one program to another using expanded memory, compile output pages with merged text and graphics at a high speed and to write data files to hard disks with 17 millisecond access times leads to increased productivity and user satisfaction. The following sections of this chapter present three levels of performance for desktop publishing. A user can easily upgrade from one level to the next by acquiring additional equipment and software. It will be shown that a small investment can yield professional results while a larger investment increases capabilities, performance and produces output that can be measured against professional publishing standards.

THE ENTRY LEVEL SYSTEM

The lower priced 80286 based personal computers (eight megahertz minimum, 12 megahertz recommended) with 640 K bytes of main memory and a 20, 30 or 40 megabyte hard disk will provide a good foundation to begin using DTP software. The computer must have a parallel printer port since serial printing will take four times longer to print on average. A 5.25 inch-1.2 megabyte, or 3.5 inch-1.44 megabyte floppy disk drive is necessary for software installation and document archiving. Although a video graphic adapter and VGA monitor is highly recommended, a monochrome or color graphic adapter will provide acceptable results. An industry standard mouse is required to interface with standard desktop publishing environments such as Microsoft Windows for Aldus Pagemaker or Graphics Environment Manager (GEM) for Xerox Ventura Publisher.

A laser printer is required to produce professional looking documents and should have a minimum of 512 Kbytes of memory. To obtain additional font capabilities, add one or two font cartridges. Font cartridges are easy to use, require no extra memory and do not need to be downloaded into the printer's memory at system startup time. Text is always printed at 300x300 dots per inch on laser printers and additional memory for the laser printer will provide the user with the ability to merge text and graphics on a full page (8.5 x 11) at 300x300 dots per inch resolution.

Choosing software for an entry level system typically depends on standards set by individual companies. Word processing software (i.e. Wordstar 2000, Multimate Advantage II, MS Word, Wordperfect 5.0, Q&A, Hewlett-Packard Executive Memomaker) allows the user to create text files that will be placed in the DTP document. Graphics software (i.e. Hewlett-Packard Gallery Collection, Harvard Presentation Graphics, Lotus Freelance) allow clip art or business graphics to be merged with the text to produce visually exciting documents.

There are many software solutions available that address the desktop publishing market. The best DTP software package at the entry level is Aldus Pagemaker in the Microsoft Windows environment. Since the target task at the entry level is multicolumn reports and simple newsletters, Pagemaker's ability to quickly and easily produce short documents with easy access to font selection make it an excellent choice for a DTP system costing $8,000.00. Aldus Pagemaker has become the product of choice when a user will produce shorter documents, usually less than 100 pages. Ventura Publisher, using GEM system software from Digital Research, Incorporated remains dominant for users that need document processing in excess of 100 pages. Ventura is far more complex to learn and use, especially for a novice or casual DTP user and is therefore not recommended for an entry level system.

However, one of Ventura's most powerful features is the method used to place supported files into a publication. A pointer is used to access a supported input file. This allows files to be updated outside of the Ventura software. Also, if a user updates a file from within Ventura, the change is reflected in the original input file. Aldus Pagemaker 3.0 does not have this capability.

The following chart shows the components of an entry level desktop publishing workstation:

SAMPLE ENTRY LEVEL CONFIGURATION

HARDWARE:	**US LIST PRICE:** (As of 5/89)
Hewlett-Packard Vectra ES 8 megahertz 80286 based CPU 20 megabyte hard disk, VGA adapter 5.25 inch floppy disk drive 640 Kbytes main memory	$2,795.00
MS-DOS 3.3	120.00
VGA monochrome monitor	250.00
Hewlett-Packard Mouse	155.00
Hewlett-Packard Laserjet Series II	2,695.00
Hewlett-Packard Z Font Cartridge	330.00
SOFTWARE:	
Word Processing Software:	
Hewlett-Packard Executive Memomaker	300.00
Desktop Publishing Software:	
Aldus Pagemaker	795.00
Microsoft Windows/286	150.00
Graphics Software:	
Hewlett-Packard Graphics Gallery	495.00
TOTAL:	**$8,085.00**

THE MID-RANGE SOLUTION

DTP users that will be producing extensive, multi-column business reports and newsletters with merged text and graphics require more sophisticated and expensive equipment. The expectations for DTP computer systems that cost in the range of $15,000.00 would include faster document processing, larger hard disk capacity and expanded printing and font capabilities.

A 12 megahertz 80286 or a 16 or 20 megahertz 80386 based processor with two megabytes of LIM 4.0 expanded memory specification will produce a dramatic increase in system performance. An EGA adapter and monitor is the lowest recommended video solution. The same basic requirements outlined in the entry level system (i.e. a parallel printer port, mouse and floppy disk drive) are also required. The minimum hard disk size to handle longer documents and more sophisticated, larger software programs is 40 megabytes.

A laser printer for this system should have a minimum 2.5 megabytes of memory to hold downloaded fonts that allow the user to create (among other things) headlines with no character height (point size) restriction. These "soft" fonts are generated by two popular packages called Type Director, distributed by Hewlett-Packard, and Fontware from Bitstream. Type Director utilizes a font scaling technology that allows users to:

o create fonts in sizes from four to 200 points (72 points is equal to one inch) in 0.5 point increments for your software applications.

o create matching screen fonts for "WSIWYG" (what you see is what you get) applications.

o minimize memory requirements on the personal computers hard disk and the laser printer by allowing the user to create fonts with reduced symbol sets.

o easily manage the soft fonts with a font manager and downloading utility.

o automatically install fonts into desktop publishing applications such as Aldus Pagemaker, Xerox Ventura Publisher and Windows/286.

o easily create font metric files for applications like Microsoft Word.

o use any brand of soft fonts compatible with the Laserjet printer.

Bitstream Fontware is bundled with Aldus Pagemaker 3.0 and comes with four different typefaces that are available with numerous character sets including Windows ANSI, ASCII and HP Roman 8. Printer and screen fonts can be generated ranging in size from six points to 72 points. The Fontware program is menu driven and automatically creates fonts and installs them for use with the Windows program and Pagemaker. The installation kit allows the DTP user to create PCL and Postscript fonts.

If an extensive DTP document containing a variety of fonts will be created, the careful use of soft and cartridge fonts will allow the user to conserve laser memory for graphics. Soft fonts should be used only for larger fonts greater than 30 points. Cartridge fonts should be used as much as possible because they take up virtually no memory on the target printer.

The vehicle that allows graphics such as line art, printed business graphics, photographs (color and black and white), signatures and company logos to be integrated into a DTP document is a scanner. Flatbed scanners scan images on paper that are placed face down on a glass surface. Sheet-fed scanners feed paper over a fixed unit in the body of the machine. Scanners form Cannon, Microtek, Dest, Datacopy and Hewlett-Packard have dominated the scanner market in recent years. Adding a scanner to a desktop publishing workstation will make a user quickly realize how valuable large hard disks are. Scanning software can consume one megabyte of a hard disk and each image, if stored on the hard disk can also consume up to one megabyte of disk space. It is highly recommended to store scanned images off-line on flexible disk media that have a storage capacity of one megabyte or greater.

Until recently, the primary use of optical scanners has been for raster graphics only. Scanners are excellent front-end tools to scan in printed text documents that will be "placed" or integrated into a DTP document as text. Popular optical character recognition (OCR) software available today include ReadRight from OCR Systems, TrueScan from Calera and Omnipage from Caere Corporation. Omnipage software is an advanced OCR application that has the ability to include non-text items in the file that is produced by the software. The ability to retain margin and column settings makes Omnipage a very useful product for desktop publishing. Omnipage's OCR capabilities include:

o the ability to recognize virtually any font in point sizes ranging from 8 to 72, and

o the ability to read typeset, kerned and proportionally spaced text.

ReadRight is an excellent entry level solution for casual OCR users. Readright will not produce acceptable results from scanning typeset documents, poor quality originals created on dot matrix printers or poor quality photocopies of documents. However, Readright supports numerous mono-spaced and proportional spaced fonts and works especially well with the most common typewriter fonts including Courier 10, Letter Gothic, Prestige Elite and Pica.

Software requirements for users of a mid-range DTP installation include a full feature word processor such as Wordperfect 5.0, Wordstar 2000, Multimate Advantage II or Microsoft Word and a sophisticated graphics package for business graphics and clip art. Using the guidelines diskussed in the entry-level DTP solution, a user can choose between Pagemaker and Ventura and achieve excellent results. Software packages that allow the user to create freehand drawings are very useful. PC Paintbrush from ZSoft Corporation and Microsoft Windows Paint are excellent examples of software that allow you to "paint" a line art image. This software produces a high resolution picture that can be scanned in or saved to a file format that is compatible with DTP software.

As a user ventures into the world of mid-range and high-end DTP, it is important to remember that software tools do not guarantee a successful DTP workstation. The integration of the best hardware and software components at each price level will dictate if a user's performance and output expectations will be met.

THE MID-RANGE SOLUTION

HARDWARE:	**US LIST PRICE:** (As of 5/89)
Hewlett-Packard Vectra QS/16	$5,495.00
16 megahertz 80386 based CPU	
40 megabyte hard disk, VGA adapter	
5.25 inch floppy disk drive	
1 Megabyte main memory	
1 Megabyte Expanded Memory (LIM 4.0)	650.00
MS-DOS 3.3	120.00
VGA color monitor	695.00
Hewlett-Packard Mouse	155.00
Hewlett-Packard Laserjet Series II	2,695.00
2 Megabytes Laserjet Memory Board	995.00
Type Director Font Generation Software	225.00
HP Scanjet Plus Scanner and Interface	2,190.00
OCR Systems ReadRight OCR Software	595.00
SOFTWARE:	
Wordprocessing Software:	
WordPerfect 5.0	495.00
Desktop Publishing Software:	
Aldus Pagemaker	795.00
Microsoft Windows/286	150.00
PC Paintbrush	149.00
Graphics Software:	
Hewlett-Packard Graphics Gallery	495.00
TOTAL:	**$15,899.00**

THE HIGH-END SOLUTION

A DTP system capable of generating the highest quality documents such as trade journals, publications for clients, data sheets and price lists, and incorporating merged images must have flexibility on the software side and top performance on the hardware side.

An 80386-based, 20 or 25 megahertz cached memory system with four megabytes of expanded memory and over 100 megabytes of disk storage, along with the requisite parallel port, mouse and both a 5 1/4 and 3 1/2 disk drive will address this market. Of course, premium performance carries a premium price tag compared to the entry-level and mid-range systems.

There are many other hardware options available at this price level. The ability to see an entire 8 1/2 x 11 DTP document without scrolling will aid productivity. Monitors that show two 8 1/2 x 11 pages are available and range in price from $1,800.00 to $2,500.00. Monitors from NEC (Monograph System, 16 inch monochrome screen, 1024x1024 resolution or MultiSync XL, 20 inch, color screen, 1024x768 resolution) or Micro Display Systems (Genius that shows one 8 1/2 x 11 screens at 736 x 1008 resolution or two 8/12 x 11 screens at 1280 x 1024 resolution) are excellent examples of the high end monitor market for desktop publishing power users. Hewlett-Packard offers a 16 or 20 inch color display with VGA (640x480) and multiscanning, high resolution capabilities to 1280x1024. A VGA or EGA solution with a 14 inch monitor is an acceptable and less expensive alternative at this performance level.

There are many printing options when your budget is not limited. A superior solution includes a Hewlett-Packard Laserjet Series II with four megabytes of memory. The Laserjet Series II is the de-facto industry standard laser printer with an installed base far exceeding one million units. The expanded memory can be used to store soft fonts from font generation software or graphics.

At this price level, there are alternatives to using Printer Control Language (PCL) from Hewlett-Packard. PC Publisher Kit ($1,995.00) from Imagen Corporation provides a platform that emulates multiple font cartridges, adds two megabytes of memory to the Laserjet and provides the user with the ability to create over 20 different fonts in any point size. The Publisher Kit also lets you use PCL, HP-GL (Hewlett-Packard's graphic language), Postscript from Adobe Systems and DDL. DDL is Imagen's document description language which has been shown in benchmark tests to be the fastest printer language available for the Laserjet printer.

Another alternative is Jetscript ($2,795.00) from QMS Corporation. Jetscript provides the Laserjet Series II with Adobe Systems Postscript capability by using an interface board in the optional accessory slot on the Laserjet Series II and a full length AT type board with four megabytes of memory. Postscript compatibility is an important enhancement for high-end DTP users and is one of the most popular options at this price level.

Software requirements for a high-end DTP solution are similar to the mid-range level. Full featured word processors and graphics software which produce files that will be incorporated into DTP software are not the most vital elements of a successful DTP workstation. Performance expectations will be met using the fastest hardware and best DTP software solution for your needs - not by debating which word processor works best in the word processing environment.

Aldus Pagemaker and Ventura Publisher can compete successfully in all price and performance categories. There are some DTP software packages that address only the high-end market. Superpage II ($7000.00) from Bestinfo, Incorporated supports 20 professional typesetters (over 2400 dots per inch) and any Postscript device. IBM Interleaf Publisher ($2,495.00) has recently been ported from the mainframe environment to the PC environment. Interleaf Publisher is produced by Interleaf, Incorporated and marketed exclusively by IBM. Interleaf is different from most DTP software because it is designed to act as a word processor, charting program, graphic generation and page formatter package. Interleaf requires 6 megabytes of RAM on your personal computer, over 25 megabytes of hard disk storage and operates only on an 80386 based personal computer. There are add-on products (e.g. Professional Extension, $595.00) to Ventura Publisher that allow it to compete with the immense power and capabilities of Interleaf Publisher.

As a general guideline, choose Pagemaker or Ventura when you plan to use your personal computer for other tasks such as spreadsheets, word processing and data communications. DTP users dedicated only to DTP tasks can learn to function professionally in the high-end software market such as Interleaf. Regardless of the software you choose, it is not possible to have a successful DTP workstation by making a casual commitment to learning DTP software. Be prepared to read the manual!

The following chart represents a sample high-end desktop publishing workstation:

THE HIGH-END SOLUTION

HARDWARE:	**US LIST PRICE:** (As of 5/89)
Hewlett-Packard RS/25C 25 megahertz 80386 based CPU Memory caching VGA adapter card 155 megabyte hard disk, 17ms access speed 5.25 inch floppy disk drive 4 megabytes RAM memory	$13,295.00
MS-DOS 3.3	120.00
VGA color monitor	695.00
Hewlett-Packard Mouse	155.00
Hewlett-Packard Laserjet Series II	2,695.00
QMS Jetscript Accessory Board	2,795.00
HP Scanjet Plus Scanner and Interface	2,190.00
SOFTWARE:	
Word Processing Software: WordPerfect 5.0	495.00
Desktop Publishing Software: Choice of:	
Aldus Pagemaker	795.00
Microsoft Windows/286	150.00
-or-	
Ventura Publisher	895.00
PC Paintbrush	149.00
Graphics Software: Hewlett-Packard Graphics Gallery	495.00
Pagemaker Solution:	**$24,029.00**
Ventura Solution:	**$23,979.00**

III. SAMPLE INSTALLATION SOLUTIONS

Configuring and installing the hardware and software components of a desktop publishing system is a task that requires the user to have a basic understanding of MS-DOS commands and a good knowledge of the equipment that resides under the cover of the target personal computer. Hardware devices such as video monitor, mouse type, memory boards and printers will dictate how DTP software must be installed.

The following is a scenario that describes the steps a DTP user needs to understand in order to successfully install desktop publishing components. A DTP solution incorporating Microsoft Windows/286 Version 2.1 and Aldus Pagemaker 3.0 has been chosen as an illustration to acquaint the reader with the steps necessary to begin desktop publishing.

MICROSOFT WINDOWS/286

Microsoft Windows/286 is a graphical interface that allows a user to work in a "window" environment as an extension of the MS-DOS operating system. Windows/286 is a productivity tool that allows the user to work with different software programs simultaneously and cut and paste data between these applications. The drop-down menus and icons provide a consistent user interface for all Windows applications. The ability to suspend an application with a click of a mouse and switch to another program is a powerful feature of the Windows environment. Windows/286 will work very efficiently on a personal computer with 640 Kbytes of RAM. For increased performance, it is highly recommended to work with one or two megabytes of LIM 4.0 expanded memory and give this memory to the Windows/286 SMARTDrive disk caching program.

The 5 1/4 inch floppy distribution of Windows/286 version 2.1 software is distributed on 13 disks as follows:

Setup disk, Build disk, Displays 1, Displays 2, Fonts 1, Fonts 2 (Font files for printers and graphic adapters), Desktop Applications (Contains the Windows applications), Microsoft Windows Write (Contains the Windows Write program), Additional Drivers, Utilities 1, Utilities 2, Utilities 3, Utilities 4 (Printer device drivers)

The installation program is found on the Setup disk and prompts the user to place the correct disk in the target drive when needed. Windows/286 should take a user no more than 15 minutes to install and will occupy more than 1.5 megabytes of hard disk storage.

The installation program has 20 different screens and asks the user the following questions:

- o What kind of display do you have.
- o What kind of pointing device (mouse) do you have.
- o What additional memory you have, if any.
- o What kind of printer(s) you have, if any.
- o What port your printer is connected to.

In order to install the Windows/286 program successfully, the installation program must be run to completion. The install process is initiated by placing the Windows/286 disk labeled "Setup" in the "A:" disk drive, and with an "A:" prompt, type "SETUP". The installation program will create and place the Windows/286 software in a directory named "C:\WINDOWS". The second screen allows you to change this default directory if desired. If you upgrade your desktop publishing hardware, modify your existing hardware or add new devices, the setup program must be run again to completion.

If you are using a Hewlett-Packard Laserjet Series II as your DTP printer, read the "READMEHP.TXT" file on the Utilities 4 disk. This ASCII file contains valuable information on configuration of your PCL printer. If you are using a Postscript printer, read the READMEPS.TXT file.

Your printer is installed into the Windows environment after completion of the SETUP program but is not operational at this point. In order to use the printer you must configure it with the Windows/286 "Control Panel". Using your mouse, click on the "CONTROL.EXE" file in your Windows subdirectory and pull down the Setup menu. Choose the Connections option and highlight the PCL/Laserjet option. Using the mouse, click on the desired output port and exit the menu by clicking on the "OK" button. Next, select the Printer option (under the Setup command) and make sure the PCL Laserjet option is highlighted. To get to the actual configuration menu, click the OK button. This menu allows you to set various attributes of the Laserjet including model type, RAM memory, default orientation, dots per inch, and hard cartridges. It is possible to select two hard cartridges by holding the shift key when clicking the mouse on your second hard cartridge choice.

Windows/286 Version 2.1 comes with version 3.1 of the HPPCL.DRV printer driver file. Under the Printer menu, this version of the Laserjet driver has a Fonts option below the OK and cancel button. This option will allow easy installation of soft fonts into your Windows WIN.INI file that were created by your font generation software.

The installation program will alter your CONFIG.SYS file and AUTOEXEC.BAT file. The CONFIG.SYS should contain the following entries after installation is complete:

FILES=99
BUFFERS=10

The FILES command specifies a range from eight to 255 file handles that can be opened concurrently. Each handle over eight increases the size of MS-DOS by 48 bytes. Setting FILES=99 will allow enough files to be opened concurrently to operate within a Windows/286 environment successfully and will consume just over 4K of main memory.

The BUFFERS command specifies the number of disk buffers and can range between 1 and 99. Setting BUFFERS to a higher number will increase performance with applications that do random read and writes to the hard disk. As data is read from the hard disk, it is stored in a buffer. The larger the buffer, the higher the chance that data requested by the application will be in a memory buffer and not have to be read from the disk. For applications that perform sequential reads and writes to a disk, no performance gain will be realized by having a larger buffer size. Each additional buffer will increase the resident size of MS-DOS by 512 bytes or more depending on the size of the hard disk. Main memory can be substantially reduced by specifying a large number of buffers. Setting BUFFERS=10 reduces main memory by 5 Kbytes (The Pagemaker 3.0 installation program will increase the BUFFER command to 30, consuming 15 Kbytes of main memory).

If your computer has LIM 4.0 expanded memory, the following line should be added to your CONFIG.SYS file:

DEVICE=SMARTDRV.SYS xxxx /a

Where xxxx = expanded memory size, i.e. 2048 for 2 megabytes of expanded memory. This line must appear after the expanded memory manager device driver has been loaded in the CONFIG.SYS file.

If your computer has extended memory, the following line should be added to your CONFIG.SYS file:

DEVICE=SMARTDRV.SYS (This will give SMARTDrive all the available extended memory in your personal computer.)

SMARTDrive is a disk caching utility that comes with Windows/286 that was designed to work with expanded or extended memory. The amount of time required to read data from your hard disk will be reduced by using the SMARTDrive program. Windows/286 and SMARTDrive compete for expanded memory. SMARTDrive will release expanded memory when requested by Windows/286.

Your updated AUTOEXEC.BAT file will contain an MS-DOS path statement and SET TEMP statement.

```
PATH=C:\WINDOWS
SET TEMP C:\WINDOWS
```

If a program, batch file or command is executed but is not in the default directory, MS-DOS will search the directories specified by the PATH statement. The SET TEMP command specifies the directory to which MS-DOS will write temporary files that are created by application programs.

The WIN.INI file for Windows/286 is similar to the CONFIG.SYS file for MS-DOS. When Windows/286 is executed, the settings in the WIN.INI file are checked. Installation selections and CONTROL.EXE selections dictate the Windows environment parameters.

If you have a minimum of two megabytes of expanded memory in your computer, change the "SWAPDISK=" line in the WIN.INI file to "SWAPDISK=C: /e" to obtain the benefits of swapping main memory to fast expanded memory. This option tells Windows/286 to swap to expanded memory first if available, followed by the drive specified in the AUTOEXEC.BAT "SET TEMP=" command followed by the root directory "C:".

Since Windows/286 is the environment that Aldus Pagemaker 3.0 uses, it is vital that the CONFIG.SYS, AUTOEXEC.BAT, WIN.INI and the Control Panel are all configured properly.

ALDUS PAGEMAKER 3.0

Windows/286 must be properly installed on your computer prior to installing Aldus Pagemaker. As a precaution, it is highly recommended to make backup copies of your CONFIG.SYS, AUTOEXEC.BAT and WIN.INI files prior to installing Pagemaker 3.0. Keep a copy of these files by issuing the following commands from your root MS-DOS directory:

```
COPY CONFIG.SYS CONFIG.SAV
COPY AUTOEXEC.BAT AUTOEXEC.SAV
COPY C:\WINDOWS\WIN.INI C:\WINDOWS\WIN.SAV
```

The 5 1/4 inch version of Aldus Pagemaker 3.0 is distributed on five disks as follows:

Install/Program/Dictionary disk, Getting Started disk, Drivers/Filters/Templates disk, Fontware Installation disk, Fontware Typefaces disk

The installation program is found on the Install/Program/Dictionary disk. Pagemaker requires less than 15 minutes to install and consumes one megabyte of hard disk storage. The installation program has 19 different screens and will complete the following tasks:

o copy all required Pagemaker 3.0 files to "C:\PM"

o copy the desired import filters that determine which file formats can be imported into the Pagemaker 3.0 program

o copy the desired Pagemaker 3.0 templates for standard forms and newsletters to a separate subdirectory

o copy the tutorial files to a separate sub-directory

The installation process is initiated by placing the Install disk in the "A:" drive and with an "A:" prompt, type "INSTALL". The Pagemaker install program will automatically update your CONFIG.SYS file and AUTOEXEC.BAT file. The PATH and TEMP statements will be altered as follows:

```
PATH C:\WINDOWS;C:\PM
SET TEMP=C:\PM
```

The CONFIG.SYS file should be updated as follows for optimum performance:

```
BUFFERS=30
```

Pagemaker performance is dependent on the following factors:

o how much and what type of memory is in your computer

o the access speed of your hard disk

o the CPU speed of the computer

Purchasing LIM 4.0 expanded memory is the best option available to improve the performance of desktop publishing software. Pagemaker 3.0 and Windows/286 requires at least 550 Kbytes of main memory. If you intend to use many different device drivers, memory-resident programs or LAN software, consider creating different versions of your AUTOEXEC.BAT and CONFIG.SYS files. A DTP version of these files will load only the device drivers required for your DTP hardware and software such as an image scanner.

SOFTWARE INSTALLATION TIME REQUIREMENTS

Installation programs common to all components of MS-DOS and DTP software provide the user with an easy method of installing programs. There are no industry standards that dictate the format to initiate these procedures. Most programs will automatically update important system files such as CONFIG.SYS, AUTOEXEC.BAT and WIN.INI. Before using your software, check these files and be sure to re-boot your computer so that any changes will be reflected in your operating environment.

The following chart represents the MS-DOS commands that must be run in order to begin the software installation process:

APPLICATION	MS-DOS COMMAND	disk LABEL
WordPerfect 5.0	Copy *.*	All disks
Windows/286	Setup	Setup
Gallery Collection	Setup Gallery	Setup Master
Pagemaker 3.0	Install	Install
Ventura Publisher	Vpprep	Application disk #1
Scanning Gallery Plus	Sjsetup	Installation
ReadRight	Setup C:\RR	disk 1
	Setup2 C:\RR	disk 2
Type Director	Install	Type Director 1
Bitstream Fontware	Fontware	Installation
QMS Jetscript	Jetinstl	Installation disk

Use the following chart as a general guideline to help set expectations regarding the time that is required to install and fine tune a desktop publishing workstation.

SOFTWARE	ESTIMATED TIME TO INSTALL SOFTWARE (MINUTES)	ESTIMATED TIME TO FINE TUNE SOFTWARE (MINUTES)
Windows/286	15	15
Pagemaker 3.0	15	10
Scanning Gallery Plus	10	0
Wordperfect 5.0	10	10
Gallery Collection	5	0
PC Paintbrush	5	0
ReadRight	5	0
Jetscript	60	0
Soft font generation	60+	5
(Depends on point size and symbol set selected)		
TOTAL:	+/- 3 Hours	+/- 3/4 Hour

IV. PREREQUISITES FOR A SUCCESSFUL IMPLEMENTATION

Installing the software and hardware components of a DTP workstation does not complete the process required for a successful DTP installation. It is vital that the DTP user understand the basic rules of typesetting. DTP software gives a user the tools to create documents but it is up to the user to learn and understand how to use these tools to generate professional looking documents.

DTP classes are available at local universities, computer dealers and computer schools. These classes teach users the basics of the DTP software package. Before attending these one or two day classes, experiment with the software and complete the tutorials that are included with the package.

There are many books available in local bookstores that will teach a user how to use various DTP software packages and how to produce professional looking documents. Two books that present the user with an excellent overview of the principles of basic design, typesetting and page layout techniques are:

<u>Instant Pagemaker</u> (IBM Version 3.0) by Kate Hatsy Thompson, Peter Randall and Steven J. Bennett, $39.95

<u>Instant Ventura Publisher</u>, by Tony Pompili, Kate Hatsy Thompson, Steven J. Bennett, $39.95

These books provide the user with "templates" which are example DTP files designed by a professional. New DTP users can use these templates to produce their first DTP output and experiment as they gain more experience. Use of templates (numerous examples are included with the book and come standard with Pagemaker 3.0 and Ventura Publisher 2.0) allows a beginner to produce professionally designed documents by just adding text and graphics.

The following represents an overview of current literature available in bookstores today:

XEROX VENTURA PUBLISHER:

<u>Mastering Ventura</u>, Matthew Holtz, $22.95

<u>Inside Xerox Ventura Publisher - A Guide To Desktop Publishing</u>, James Cavvoto and Jesse Berst, $27.95

<u>Publishing Power With Ventura</u>, Martha Lubow and Jesse Berst, $27.95

<u>Using Ventura Publisher</u>, Linda Mercer, $24.95

ALDUS PAGEMAKER:

Mastering Pagemaker On The IBM PC, Antonia Jolles, $22.95

Using Pagemaker On The IBM, Diane Burns and S. Venit, $24.95

Using Aldus Pagemaker 3.0, Douglas Kramer and Roger Parker, $27.95

UNDERSTANDING DESKTOP PUBLISHING PRINCIPLES:

The Aldus Guide To Basic Design, Roger C. Parker, $6.95.

Design Principles For Desktop Publishing, Tom Lichty, $19.95

Design For Desktop Publishing, John Miles, $16.95

Looking Good In Print - A Guide To Basic Design For Desktop Publishing, Roger Parker, $23.95

Pocket Pal, Michael Bruno (This handbook is considered the publishers "bible")

HEWLETT-PACKARD LASERJET:

Laserjet Unlimited, Ted Nace and Michael Gardner, $24.95.

Laserjet Companion, The Cobb Group, $24.95.

DTP AND PUBLISHING MAGAZINES

Publish!, PCW Communications Incorporated, San Fransisco, California, monthly

Personal Computing, VNU Business Publications, Incorporated, Hasbrouck Heights, New Jersey, monthly

PC Magazine, Ziff-Davis Publishing Company, New York, New York, monthly

PC Publishing, Hunter Publications, Des Plains, Illinois, monthly

Personal Publishing, Renegade Publications, Itasca, Illinois, monthly

Print: Americas Graphic Design Magazine, RC Publications, New York, New York, Bi-monthly

The Seybold Report On Desktop Publishing, Seybold Publications, Incorporated, Media, Pennsylvania

Many users of desktop publishing software participate in professional organizations that promote sharing of information and best practices. The following represents various users groups that distribute information for the desktop publishing user community.

DESKTOP PUBLISHING USERS GROUPS:

Ventura Publisher Users Group
16160 Caputo Drive
Morgan Hill, California 95037

Aldus Pagemaker Users Group
Aldus Corporation
411 First Avenue South, Suite 200
Seattle, Washington 98104

National Association of Desktop Publishing
P.O. Box 508 Kenmore Station
Boston, Massachusetts 02215-9998

V. FUTURE DIRECTIONS

The introduction, acceptance and versatility of publishing software has changed the methods that companies use to produce typeset documentation. The future of the desktop publishing environment will be influenced by the following factors:

o Faster microprocessors:
Currently running on 32 bit platforms, future software will enjoy the speed of 64 bit microprocessors.

o Higher resolution scanners:
Currently output resolution of 300 dots per inch is standard. The future will see resolutions of 1500 dots per inch or more.

o Faster, less expensive, higher resolution desktop laser printers that will simulate halftones and gray scale
Current output resolution of 300 dots per inch has revolutionized the way companies produce documents. In the future, printer engines will approach typesetting standards and will print 1,200 to 1,500 dots per inch.

o New inkjet technology
Low cost inkjet printers will approach the standards of todays laser printers. Future inkjet printers will produce near-typeset quality color output (300x300 dpi) on plain paper and will support grayscale.

o Enhancements to existing publishing software
Software companies will continue to produce and refine their programs according to industry standard platforms and environments.

o Improvements with Artificial Intelligence
Software that understands the complexities of the printing process will change the way workers complete composition and printing tasks.

Today, a company can successfully implement a desktop publishing workstation as long as expectations are set correctly regarding hardware performance, software ease-of-use, installation procedures and printer language capability. After the hardware and software is installed, the user must learn to to design effective documents.

The future of the desktop publishing industry is exciting and will continue to grow. The best way to experience this technology is to purchase solutions available today for your current needs and eagerly await the technology of tomorrow.

TRADEMARK ACKNOWLEDGEMENT LIST

Ashton-Tate is a registered trademark of Ashton-Tate Corporation.

Lotus and 1-2-3 are U.S. registered trademarks of Lotus Development Corporation.

Micropro is a U.S. registered trademark of Micropro International Corporation.

Microrim is a U.S. registered trademark of Microrim, Incorporated.

Microsoft and MS-DOS are U.S. registerd trademarks of Microsoft Corporation.

Multimate and Multimate Advantage are U.S. registered trademarks of Multimate International Corporation, and Ashton Tate company.

Pagemaker is a U.S. registered trademark of Aldus Corporation.

R:Base 5000 is a U.S. registered trademark of Microrim, Incorporated.

Wordstar is a U.S. registered trademark of Micropro International Corporation.

IMPORTANT: Many of the software and hardware products that appear in this paper are marketed by various vendors under registered trademarks. These trademarks imply ownership of these products and the names of these products. This paper has not placed trademark symbols next to products that are registered and subject to state and federal laws - in no way should the reader imply that these products are not registered trademarks. Please contact the vendor directly to confirm ownership of any trademarks.

Hewlett-Packard disclaims any and all liabilities for and makes no warranties, expressed or implied, with respect to the furnishing, performance, or use of any product described in this publication.

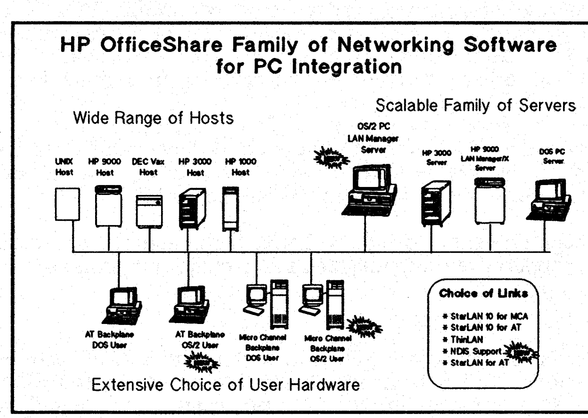
Distribution by Hewlett-Packard of the products or information concerning these products does not constitute endorsement of the product or supplier.

PC Integration and Networking

Steve Martin
Hewlett-Packard Company
3404 East Harmony Road
Fort Collins, Colorado 80525

Personal computers have become a strategic resource across corporate-wide environments, from business-office to manufacturing to engineering departments. As PC's become more predominant, so does the need to successfully integrate them into a corporation's information system; companies need a PC networking strategy that complements their overall system integration requirements.

This paper explores PC networking and integration alternatives. It highlights company benefits of implementing a comprehensive, unified PC networking solution, and reviews HP's standards-based, multi-vendor PC networking strategy. HP products, from PC LAN implementations to PC-mini integration, are discussed in the context of a corporate-wide Cooperative Computing Environment.



Introduction

PC Integration is the total and seamless integration of the services, capacity, data and applications residing on all computers interconnected throughout an entire company. PC users must have easy access to PC and host computing power as well as to PC and host data. Full PC integration provides users with easy and cost-efficient access to all computing power and data that they need to do their jobs effectively.

Personal computers have become common in the workplace because they offer increased productivity with specialized applications, user independence, and desktop power. As a result of technological progress, PCs are now found on the desks of all employees from executives to programmers to clerical personnel. Companies are recognizing that further increases in productivity require increased communication between their desktop PCs and departmental computers and corporate mainframes and now face the challenge of integrating these PCs into the larger information systems of their organizations. The integration of PCs provides opportunities for tremendous productivity, quality and cost benefits from increased communication, network services, and the central administration of assets (including computing hardware, peripherals, applications, data, and technical expertise).

The Benefits of PC Integration

An HP AdvanceNet LAN is the most effective method of achieving PC integration because it enables the PC to become the entry point into an entire corporate computing environment at any level, including the local workgroup, department, site, or corporate processing. Rather than connecting PCs to departmental computers as terminals, or to other PCs using a PC-only LAN with point-to-point connections to a departmental computer, an HP AdvanceNet LAN integrates every PC with all other computers on the network, from other PCs to corporate mainframes.

An HP AdvanceNet LAN for PC integration allows scalable growth from an individual PC all the way up to a corporate mainframe. It enables companies to add users and servers when and where they are needed.

Total PC Integration can provide users with the benefits of lower cost, increased productivity, cooperative computing and multi-vendor integration.

Cost

The price/performance ratio is generally better with a LAN than with point-to-point connections. A LAN provides better file transfer speed with only a little more HP 3000 processing overhead. Also, users don't degrade each others' individual performance because they each have their own processing power. An industry move is on to take advantage of this distributed computing environment by developing distributed applications that minimize data transfer between processes, e.g., concentrated database access. Because the price per PC MIP is coming down faster than that of minicomputers, it is easy and cost effective for users to add more PCs to a network to get more processing power.

A LAN is also the most cost effective method for integrating PCs with HP 3000s, other PCs, and minicomputers. The initial incremental cost of LAN connections compared to point-to-point connections is attractive when considering the additional functionality, connectivity, and scalability a LAN provides. A LAN protects users' investments in new and existing PCs, peripherals, software and data, and paves the way for their use of future LAN functionality including new distributed applications.

The functionality provided over an HP AdvanceNet LAN is superior to point-to-point connections because RS-232 links cannot provide the inter-connectivity, services, and scalability of a LAN.

Productivity

PC Integration increases productivity in several ways - through multi-system and multi-vendor connectivity, through a wider and more appropriate selection of applications and through the use of APIs to develop cooperative computing applications.

HP's LAN Manager offering supports OS/2 servers and both DOS and OS/2 clients. The use of OS/2 allows a number of productivity enhancing capabilities including software access to more than the 640K of memory allowed by DOS and support of multi-tasking capability.

PC users have seamless access to ARPA services, NS services and OfficeShare PC, HP 3000 and HP 9000 (LAN Manager/X) servers without having to reboot. This means increased productivity through multi-system communications with no gateways and without rebooting.

User productivity is also increased through a wider choice of applications , more powerful applications, and through utilization of a variety of APIs including Named Pipes and Mail Slots. APIs allow companies and third parties to develop cooperative computing applications that meet the specific needs of individual companies.

Cooperative Computing

An HP AdvanceNet LAN enables users to develop their own or to use third-party-developed distributed applications across multiple CPUs with one consistent network user interface. This cooperative computing is the next big step in application development to meet specific user needs.

Cooperative computing means that each machine does what is designed for:

- PCs are used for word processing, graphics, application development and decision support
- Hosts are used for centralized applications, database and networking

- and most importantly, programs can run cooperatively between the PCs and the systems on the network for optimal overall system performance

Multi-Vendor Integration

An HP AdvanceNet LAN provides inter-connectivity with shared data and resources residing on multiple CPUs and allows multi-vendor connectivity via terminal access and file transfer. Host systems that can be accessed include: HP 1000, HP 3000, HP 3000, DEC VAX and UNIX systems.

HP solutions are standards-based allowing excellent multi-vendor connectivity. HP LAN Manager provides full TCP/IP support allowing full ARPA services support (FTP, Telnet and Sockets). As a result, users can easily connect to multi-vendor environments in CIM, Engineering or Business Office.

PC Integration Alternatives

This section discusses alternatives in the following areas of PC integration:

- network cabling
- network links (the PC interfaces on that cabling)
- network services software
- application software and
- server selection

Network Cabling

The wiring infrastructure of a building or campus is the foundation upon which a well-conceived network design rests. If this foundation is inadequate, it will be impossible to build an effective, flexible and manageable network solution.

Every business environment has its own unique needs, characteristics and computing automation programs. The network must be versatile to meet the wide range of information needs in your organization. It must be flexible so it can grow as those needs change. The network must provide connections to many vendors' systems to protect user investment, and solve complex communication problems. Users need to be connected together in logical workgroups to share data and resources. Workgroups then must be connected to a site backbone to provide facility-wide communication. The office wiring system must be well defined and limited to a uniform medium to eliminate costly rewiring for computer moves, adds or changes.

Although the cost of network user interface hardware and software has been decreasing significantly, the cost of network cabling has remained relatively constant. Since building wiring has a life span that is two to six times greater than the equipment it connects, the choice of wiring media is one of the most important long-term decisions an MIS manager can make.

HP has developed a complete set of communications wiring guidelines, products, and services, called *HP SiteWire*, to help users with their wiring decisions. HP SiteWire adheres to an open, proven multi-vendor wiring foundation that follows the guidelines of an emerging industry standard, EIA (Electronic Industries Association) TR-41.8. Adherence to standards ensures that the wiring system will provide multi-vendor compatibility and lasting value.

HP's wiring architecture addresses the needs of any physical environment and also provides a way to implement networks in a controlled, step-by-step manner. The wiring architecture is based upon a distributed star topology compatible with existing telecommunication systems - providing easy network cable administration, flexible growth, simplified network management and reduced cost of adding and moving network users.

The architecture includes the use of a thin or thick coaxial backbone cable with sub-nets of unshielded twisted pair or coaxial cabling. With support of unshielded twisted pair wiring, the wiring system that once supported only the phone system can now be used to also support data applications.

There are many benefits to the use of unshielded twisted pair wiring over shielded twisted pair and coaxial cable. Unshielded twisted pair cable costs less and is easier to install. Unshielded cables offer much more flexibility. They are of small cross-section, making installation easier and requiring less space in ducts and satellite closets. They also currently support high-speed data transfer. In addition, the existing unshielded twisted pair voice system in a building may also be able to support data as well as voice communication, virtually eliminating the large cabling cost component of a LAN.

When a user wants to utilize an existing unshielded twisted pair wiring system whose condition is unknown, HP offers a unique service, called *HP WireTest*, which provides users with an evaluation of the suitability of their existing unshielded twisted pair wiring for a StarLAN network.

Network Links

A network link is the interface by which a PC is connected to the network cabling. The network links are sold separately from the services. Network services software can be used over any of the network links, offering an extremely flexible and scalable architecture. Every link can be integrated into a single, enterprise-wide HP AdvanceNet LAN to provide all users with the access and services they need.

HP offers four network links. *HP StarLAN 10* and *StarLAN* allow users to use existing telephone wiring to support their office data communication needs. *HP ThinLAN* is available for those users who have coaxial cable already installed, and the *HP SERIAL Network* link provides a remote, asynchronous (point-to-point) link to network services on an HP 3000.

1. HP StarLAN 10 - StarLAN 10 is a 10 Mbps LAN link, using unshielded twisted pair wiring, for HP Vectra, IBM PC/XT/AT, IBM PS/2 Models 25 and 30 personal computers and for MCA Backplanes, and HP minicomputers. HP StarLAN 10 allows users to use high-speed applications and workstations, such as 80386 processor-based personal computers, without the need to install new building wiring in many cases. It is a flexible network solution for integrating PCs and minicomputers in complete office automation solutions. It is especially useful in business office environments that require a large number of nodes and experience heavy network traffic.

2. HP ThinLAN - HP ThinLAN is a 10 Mbps thin coaxial cable LAN link, supporting HP Vectra, IBM PC/XT/AT, IBM PS/2 Models 25 and 30, and HP Touchscreen PCs, as well as HP minicomputers. It is especially useful where coaxial cable is already installed, or in engineering and manufacturing environments.

3. HP SERIAL Network - The HP SERIAL Network link provides an asynchronous connection to HP 3000 computers for HP Vectra PCs, HP Touchscreen PCs, IBM PC/XT/AT, and IBM PS/2 Models 25, 30, 50, 60 and 80. This connection allows remote PC access to shared peripherals and PC files residing on HP 3000 servers, distributed applications, terminal emulation, and network file transfer (NFT). HP Vectra and IBM PCs can also make this connection via back-to-back HP 2334A multiplexers over an X.25 network.

4. HP StarLAN - HP StarLAN is a 1 Mbps LAN link using unshielded twisted pair wiring. This wiring often already exists, running parallel with the facilities' telephone wiring. The link supports HP Vectra, IBM™ PC/XT™/AT™, IBM PS/2™ Models 25 and 30 personal computers, and the Micro 3000 and HP 3000 Series 37. Personal computers on a StarLAN network can communicate with other PCs and minicomputers that are on StarLAN 10, ThinLAN or ThickLAN networks via a StarLAN bridge.

NDIS Support

HP's LAN Manager products support the NDIS (Network Driver Interface Standard) specification, a growing defacto standard for network interface cards. This allows users to select from a wide variety of HP and third-party network interface cards.

Mixed Link Networks

IBM, AT are registered trademarks of International Business Machines Corporation.

PS/2, XT are trademarks of International Business Machines Corporation.

As specified by HP SiteWire guidelines, ThinLAN or ThickLAN coaxial cable can be used as a backbone for a site-wide LAN. HP StarLAN 10, StarLAN, and ThinLAN sub-networks can be attached to a backbone cable to increase distances, the number of network nodes, and to provide intercommunication for all of the computers on these LANs.

HP's support of all these links provides users with complete flexibility to meet specific and multiple connectivity needs for the least cost.

Networking Software

LAN Manager

In 1989, HP announced new OfficeShare networking products based on Microsoft OS/2 LAN Manager, giving PC users increased capabilities to share applications and resources across MS-DOS, MS-OS/2 and UNIX operating systems in multivendor networks. The new announcements enhance HP's strength in full PC integration into a site-wide (or company-wide) network.

With LAN Manager from HP, PC users have access to a family of servers - OfficeShare DOS servers, LAN Manager OS/2 servers, HP 9000 LAN Manager/X and HP 3000 Business System Plus Servers. PC users also have access to HP NS and ARPA services. These servers and services can be accessed without the need for gateways and with one common user interface.

HP's LAN Manager offering supports integrated NS and ARPA (telnet and ftp) Services by including an industry standard TCP/IP transport. This provides PC users a complete, integrated set of services.

The HP LAN Manager for OS/2 software provides the following capabilities:

1. Peripheral Sharing enabling users to enhance communication, increase productivity, and reduce costs by sharing files and peripheral devices such as discs, printers, and plotters.
2. Complete Microsoft OS/2 support including the ability to use more than 640K memory on clients and servers.
3. A higher-performance, non-dedicated server which takes advantage of large memory support and the protected mode operation of the OS/2 operating system. (This is the feature of OS/2 that allows multi-tasking). The LAN Manager server software runs on a PC running the OS/2 operating system. Additionally, users can access existing OfficeShare and BSP servers and the new LAN Manager/X server.
4. User-based security system with logon and server-based password control, group names and audit trailing.

5. Advanced network administration tools including full screen interface and remote server control (subject to user permission level).
6. User-to-user message facility.
7. Comprehensive print spooler capabilities include routing of print jobs to the first available printer, notification of print job completion and priority control.
8. Remote program execution enabling qualified users to run programs on a remote server.
9. Remote interprocess communication allowing the creation of truly distributed networking application software using LAN Manager APIs such as Named Pipes, Mail Slots and NetBIOS. Open architecture enables customization of features by networking suppliers.

Use of LAN Manager provides users with increased productivity through an industry-standard, high performance, non-dedicated server, increased flexibility through an expanded range of servers, less training required as a result of having one user interface to HP's entire family of servers, easier administration through more powerful network administration tools, secure data through powerful user-based security and protected investment because of backwards compatibility

The new LAN Manager server software for OS/2 joins HP LAN Manager/X (LAN Manager on UNIX) software for UNIX-based HP 9000 computer servers. Together, they allow developers to build integrated OS/2 and UNIX distributed applications, and provide users a scalable growth path from low-end to high-end LAN servers. When combined with the existing OfficeShare DOS server and BSP (Business System Plus) HP 3000 server, the LAN Manager servers (on UNIX and OS/2) complete HP's broad family of servers. HP is the only vendor to have a complete OS/2 and UNIX LAN Manager solution.

The LAN Manager version of OfficeShare is backwards compatible with existing products (existing OfficeShare users are able to talk to LAN Manager servers and LAN Manager users are able to talk to existing OfficeShare servers).

Like all HP AdvanceNet products, OfficeShare is based on the International Standards Organization OSI networking model. OfficeShare provides compatibility with IEEE 802.3 industry standards, and employs de facto standards such as MicrosoftTM Networks, allowing applications written to the MS-NetTM interface to function on the network. This support of industry standards protects your investment and ensures the widest selection of growth options available. The

---

OfficeShare products also provide a single, consistent PC user interface to all network services, providing easy user integration into an enterprise-wide LAN.

Applications

With well-implemented PC Integration, users can run a wide variety of applications from their PCs:

PC Applications - Thousands of PC-Based applications are available from a multitude of vendors to meet a variety of general and specific user needs. Applications are available for both DOS and OS/2 operating systems, depending on the needs of each user. OS/2 applications can be more powerful because OS/2 has released the 640K memory limit and supports multi-tasking. Most of these applications can be used over a network.

Host-based applications - Host-based applications are available from system-vendors, third-party developers or can be developed within the user organization to meet very specific needs. There are also thousands of these host-based (mini-computer and mainframe) applications available to meet almost all needs. Access to these applications using terminal emulation is easily available over a well-implemented PC integration system.

Cooperative Processing - An HP AdvanceNet LAN enables users to develop their own or to use third-party-developed distributed applications across multiple CPUs with one consistent network user interface. This is referred to as cooperative computing and is the next big advancement in bringing more powerful applications to the desktop.

Cooperative processing means that each machine does what is designed for:

- PCs are used for word processing, graphics, application development and decision support.
- Hosts are used for centralized applications, database and networking
- and most importantly, programs can run cooperatively between the PCs and the systems on the network for optimal overall system performance

Application Programming Interfaces (APIs) - In developing their applications, PC Software developers have a choice of APIs. For PC to PC communications they can select NetBIOS or Named Pipes / Mail Slots. For PC to Mini communications they can select NetBIOS, Sockets, HP NetIPC (DOS Only) or Named Pipes / Mail Slots.

As a result, developers can build distributed applications that combine the power of DOS, OS/2 and UNIX operating systems. They can use the LAN Manager APIs to develop customized applications or to access many other applications that use LAN Manager APIs. Because LAN Manager is becoming a defacto industry standard, a significant number of applications are already being written to utilize it. In addition, other APIs are available including NetBIOS, NetIPC (DOS Only) and Berkeley Sockets.

Server alternatives

HP offers as range of servers:

- DOS-Based PC Servers
- OS/2 PC LAN Manager Servers
- HP 3000 (Business Systems Plus) Servers
- HP 9000 LAN Manager/X Servers

Each of these is designed to fit a specific type of user need. If appropriate, a mixed-server environment can be used. This is extremely useful where user needs vary significantly within one organization.

Since HP solutions are scalable, HP users can purchase a solution that fits their current needs and be confident that their investment is protected as their needs change or their organizations grow.

What to do today to achieve PC Integration

PC integration is the next big step in unleashing the power of employees at all levels - it gives users easy access to all of the data and all of the computing power that they need to effectively do their jobs. This section addresses what can be done today to ensure that future developments in PC Integration are available to you.

Focus on standards - Standards are the key to remaining open to new developments in PC Integration. HP is committed to standards. Our strategy is built on adherence to both internationally accepted standards and defacto standards such as SNA. HP also is working to drive standards by working closely with every significant standards organization in the industry to establish standards which will benefit users worldwide. Examples of this participation include: founding member of Open Software Foundation, founding member of COS, driver of the StarLAN 10 standard, Member of X/OPEN, Co-Chair of IEEE POSIX Committee, Member of OSI, etc.

HP supports standards in wiring, networking protocols, networking services and applications. LAN Manager is becoming a defacto standard. This means more software, more connectivity and more choices for users. As a result, users can easily connect to multi-vendor environments in CIM, Engineering or Business Offices. HP also provides methods of integrating non-industry-standard networks, such as Novell, into an industry-standard departmental, site or corporate-wide network.

Select the appropriate server for your needs - HP offers a scalable family of servers to fit the needs of almost all users. A company can select either a PC server, an HP 9000 server or an HP 3000 server that fits their current needs and be confident that their investment is protected as their needs change or their organizations grow. They can also be confident that future developments in PC integration can be incorporated into the systems that they implement today.

Conclusion

PC integration goes beyond just the physical connection of PCs to departmental computers. It is the seamless integration of the services and capacity, data, and applications residing on all computers interconnected throughout an entire company. This integration is the basis for further productivity gains at the desktop.

HP AdvanceNet integrates PC users into scalable, low-cost networks that provide enhanced PC-to-mini integration, multi-vendor communication, and PC-to-PC communications, for increased productivity, improved quality, and lower overall costs.

PC integration is a key part of HP AdvanceNet, Hewlett-Packard's long-term strategy for providing high-quality networking solutions for HP and non-HP computers. This strategy is based on the International Standards Organization OSI networking model and includes compatibility with industry standards, such as IEEE 802.3, and de facto standards such as LAN Manager, to ensure lasting value. HP AdvanceNet signifies a commitment by Hewlett-Packard to provide powerful and easy-to-use networks with a well-supported growth path.

Cooperative Processing: Putting a NewWave Interface In Front Of An HP 3000

By David Johnson

Johnson Computer Software Team Limited

3080 Yonge Street

Toronto, Ontario, Canada M4N 3N1

Telephone: (416) 487 3631

The kind of interface that can be developed on terminals connected to a host computer is inherently limited by the technology. With the advent of graphic interfaces on PCs we became acutely aware of these limitations. In fact, as a software development shop, we have come to realize that the user interface of a system is of fundamental importance. This point of view has led us to our current exploration of putting a NewWave interface in front of HP3000 based applications. That is, the user has a PC connected to the host and interacts with the system through the NewWave office. Our emphasis on the importance of the user interface led us to undertake the developement of applications with the "front end" on a PC and the "backend" on an HP3000.
The interaction with the user, in simple terms, takes place via the PC and the data is, in general, managed in an IMAGE database on the host. What's involved and is it worth it? We believe that the overwhelming acceptance of PCs in the market place is in large part due to the comparative ease of use of PC applications contrasted to traditional mini or mainframe applications. This point of view is further reinforced by the evident acceptance of the MacIntosh. If, on the other hand, you don't believe the user interface is all that important, keep reading - what follows may be a good argument to stay clear of this approach.

What is involved is a lot of hard work and a lot of learning. While we don't yet have enough experience to quantify any results, we believe the resulting systems will be much superior to conventional screen driven ones and that once the learning curve levels off, there may not be much of a penalty in the development effort required.

Of course, the type of application and the intended user is important to consider. A heads down data entry application to be operated by keypunchers might actually suffer from an overly sophisticated interface. However analysis tools intended to be operated by senior management may not be acceptable with anything less than the kind of approach we are talking about.

In general, the approach we have taken is based on the dual premise that PCs (single user, multi-tasking workstations) are best suited for interaction with users, while more sophisticated mini/mainframe or "midframe" host computers are better at storage and retrieval of corporate data. Specifically, we chose the IBM PC running MS-DOS with Windows/NewWave as the workstation. The PC has found undeniable acceptance in the market place and Windows provides enough multi-tasking to satisfy the basic needs. Often all that is required is rapid and convenient context switching. Moreover, Windows is the entry in to Presentation Manager if and when OS/2 catches on. The host computer, of course, means an HP3000 to us. The glue that binds is Cooperative Services.

Put another way, what we are setting out to do is to write the interactive part of an HP3000 system on MS-DOS PCs. Basically this involves MS-DOS PCs (an 80286 or better), Windows, NewWave, "C", and Cooperative Services.

For the user interface we have chosen Windows, and, by extension, NewWave. The first thing we determined was that it would be easier to develop our first application as a Windows application and then start adding to it to make it a NewWave application rather then attempting a full blown NewWave application at the outset. The PC program establishes a connection to the host using Cooperative Services. Access to data on the host is by Cooperative Services intrinsics, either by file intrinsics (FOPEN, FREAD, etc.) or by database calls (DBOPEN, DBFIND, etc.) in almost the same way as a program on the HP3000 would access it. Manipulation of the data is done either directly on the PC or with subroutines on the host (residing in an SL on the HP3000). Further processing on the host can be controlled from the PC by creating remote procedures, or processes, on the host. This approach introduces a variety of new hardware, software and new concepts for system designers, programmers, system coordinators and support functions.

The first task faced by the system designer is to understand the implications of the new interface provided by Windows/NewWave and new system architecture, a PC, a processor in its own right connected to the host. There is an enormous difference between this and a 24 line 80 column ASCII terminal connected serially to an HP3000 port. The interface environment is so rich compared to an ASCII terminal that it takes some time getting used to it.

Just learning the terminology takes a bit of time - the difference between a modal and a modeless dialogue box wasn't too difficult, but we found it took a second reading to understand semi-modal dialogue boxes. Then there's scroll bars, push buttons, radio buttons, child windows, pop-up windows, edit boxes, etc., all of which require time to understand where and how they are used.
It is here that the system designer finds the first major "constraint" on his/her new found freedom in the HP manual "HP NewWave environment User Interface Design Rules".
These are **not** guide-lines, these are rules. The designer must recognize the necessity of having these rules and the value of adhering to them. For a discussion of the importance of adhering to these rules, see the first chapter of the manual.

While these rules impose what is really a discipline, not a limitation, on the system designer, they require time and thought. The designer must be familiar with the Microsoft Windows Application Style Guide and we recommend spending time with the Dialogue Box Editor. At the very least, the system designer should have a PC on hand to experiment with existing applications while reading these documents. We discovered a real advantage in having the system designers familiar with the Dialogue Box Editor; because so much of the user interaction is with dialogue boxes, there is a real opportunity to distribute the development process. That is, the system designers can create the dialogue boxes as part of the system design given to the programmer. The system designer is "forced" to think through more of the details and the programmer is relieved of the somewhat time consuming task of designing and laying out dialogue boxes. Moreover it needn't be either the designer or the programmer who spends a lot of time laying out or tidying up the boxes. Any non-technical person on the development team, such as a technical writer, can be assigned the task of making the boxes conform to the design rules. All that is required is reasonable hand-eye coordination (you must use the mouse), a good knowledge of the interface design rules, and the ability to do mental arithmetic (any placement of items in dialogue boxes are done in dialogue boxes units).

The next challenge we faced was to determine which information should be stored on the PC and which on the host and where it should be processed. The rule of thumb for storage is personal data belongs on the PC and corporate or shared data belongs on the host. We have avoided keeping copies of the host resident data on the PC regardless of how tempting this may seem. The problem of where and even how to process host based information is more difficult. Given an application that reads a file (or a dataset) on the host, displays the information for the user to browse through and update: is it better to start by downloading all the information to memory or a file on the PC and then upload when the user is finished? Or is it better to read from the host whenever information is required and update to the host whenever the user makes a change? The answer is usually somewhere in between and depends upon the typical volume of data, volatility of the data (i.e. are the users going to make a lot of changes to what they get) and the type of the connection between the PC and the host. One thing to note is that while Cooperative Services provides all the IMAGE intrinsics, it does not provide all the file intrinsics. A sorely missed one is FPOINT.

Another thing to realize is that in the non pre-emptive multi-tasking environment of Windows, once the PC program issues an FREAD (or any other file or IMAGE intrinsic) all processing on the PC stops until the read is complete. Our experience thus far is that there are significant overheads in terms of CPU usage on the host when processing Cooperative Services calls. It is, therefore, desirable to structure the application so calls to the host are associated with a change in context on the screen rather than when a user is working in a Window. Or they should be concentrated so that the pauses to collect data from the host occur as infrequently as possible. If the application needs to move a relatively large number of records from the host to the PC, it is better to write a simple procedure on the host to collect a number of records and pass them to the PC with a single Cooperative Services call.

Programmers used to developing code on the HP3000 also have a lot of new things to learn. MS-DOS is not nearly as rich and robust as MPE and can be frustrating. Unless their PCs are on a LAN, the programmers even have to remember to do their own backup. However, while switching from MPE to any other operating system means learning a whole new set of idiosyncrasies MS-DOS hardly presents a major challenge.

Switching to a new language (we use Pascal on the HP3000) is a greater challenge but should by no means daunt a competent programmer. To develop the kind of applications we are talking about, the only language to really consider using on the PC is "C". In general this will thrill programmers used to working in other languages on the HP3000. Objectively, it means not only learning a new language but also a new development environment - compiler, editors, libraries, etc. Careful thought should be given to which editor is chosen. Subjectively, it means being aware of the need for self-discipline in writing code. While Pascal, by its nature, forces a programmer to write "structured programs", "C" imposes no such discipline. As one "C" compiler manual admonishes "with freedom comes responsibility". This leads to one of the biggest changes with which developers have to cope. On the HP3000, programmers can do a lot of sharing either at the source code level, e.g., with include files, or at the object code level via SLs. Source code, data dictionaries are all centrally located. However when more than one programmer goes to work on a PC the whole problem of compatibility and consistency becomes an extremely important one to deal with before too much actual programming takes place.

The big pay off is programmer productivity. The development environment is generally much better than that provided by the HP3000 and it is fast. Lavish use of the CPU to recompile a program to add debugging statements is virtually free; this is not something you do without careful thought when a large Pascal compile can take 15 minutes on a series 37. The difference can be a few seconds versus 15 minutes. Another pay off, while more difficult to quantify and very important to a shop like ours, is the issue of keeping programmers sufficiently challenged. Give programmers their own PCs and the "C" language to work with to develop software that is really exciting. There will be a very positive effect on productivity and tenure.

Cooperative Services is the next technical challenge, but for experienced HP3000 programmers it is the least formidable of all the new pieces. There are some surprises beyond the aforementioned dearth of intrinsics. Debugging RPCs is one that brought us up short: the first time a remote procedure fails one is left with a PC no longer connected to the host! No error messages, no indications of what happened, just a PC that must be rebooted. One method we use is to do TELLOPS to the console for debugging messages. Another is to write drivers on the HP3000 and do the debugging all on the host.

Error handling in general in RPCs requires a little more work then in "classical" applications; one cannot just blurt out a DBEXPLAIN to $STDLIST. All in all, pretty straight forward stuff although it would probably be the most challenging piece for programmers used to developing PC applications and unfamiliar with the HP3000.

The really big challenges are Windows and NewWave. There are over 450 Windows calls which gives an accurate indication of the size of the learning exercise. The books on Windows programming - all of which seem to be 1-2 inches thick - all admit it is difficult. (The book we relied on the most, based on HP recommendation at the NewWave training course, is "Programming Windows" by Charles Petzold.)

In learning to program Windows applications, everything is a challenge. Windows programs are all message driven so they require a different structure from classic programs on the HP3000. Just to get the first application running, one needs, in addition to source code, a resource file, a make file, an icon file, a header file and a module definition file. It is also necessary to learn to use the icon editor and dialogue box editor and to understand the compiler libraries.

Planning for "hot links", agent interaction, computer based training and context sensitive help in New Wave requires careful structural planning. There is a lengthy discussion of this topic in the NewWave manual.

Debugging Windows programs can be formidable. The first time a program stops and all that can be done is reboot the system, one realizes MPE has certain advantages over MS-DOS. One cannot do a break and look at the process with a monitoring utility - one just stares at a motionless screen. There are debugging utilities such as SPY and SYMDEB , and, although we have an extra monitor hooked up to our PCs to use them, we've relied mainly on inserting debugging messages and staring at the code.

Familiarity with and confidence in Windows are the two most important assets a programmer can develop. Once over the initial hurdles, application development can be quite expeditious. Programming of on-line applications on the HP3000 generally requires a lot of effort and attention to collecting information from the user, such as "enter the customer number", providing a list of valid customer numbers, confirming by displaying the name, etc. Windows provides a set of standard features for this type of interaction and is far more effective than what can be done with a screen on the host.

Thus far most of our efforts have been devoted to developing Windows applications, not full blown NewWave ones. Our strategy has been to develop our first application without the additional overheads of NewWave and then, once this was proven feasible, to stage in the NewWave functionality. This is a reasonable thing to do since there is a continuum to making an application a NewWave application from simply encapsulating a Windows application to adding all the possible features such as context sensitive help and the hooks for hot links. It also has some advantage of spreading the learning curve. NewWave functionality and programming add just that much more on top of what the system designer has to learn. The estimates we have heard suggest that there is approximately 30% more effort required to develop a full NewWave application over and above what it takes for a Windows application.

Still there are some advantages that should make this incremental effort worthwhile. Context sensitive help and the computer based training can, in certain environments, more than pay for themselves. What really appealed to us was finding NewWave could assume some significant responsibilities where we had previously devoted a lot of development resource in screen based applications. For example, in some of our software packages, users have the facility to request jobs to run overnight. The request often involves providing many parameters and the user may require several very similar jobs. On the HP3000, we built special mechanisms to allow the user to create, file, copy, edit and delete report requests. If this functionality is moved to the PC as a NewWave program, the programming effort is significantly less to provide the user with the ability to edit the parameters and let NewWave do the filing, copying, naming etc. in a manner the user can grasp at a much more intuitive level.

While we haven't got there yet, we suspect the use of the agent will open up a whole new set of possibilities under the control of the user.

Overall, we judge our investment of time, energy and money in developing software using this approach to be worthwhile. It represents a major investment, particularly for a small shop like ours, but the extremely positive user acceptance points to future successes. It also appears that once the initial investment is made, there is little penalty to be paid in terms of development effort, particularly when increases in functionality are factored into the equation.

More and more people have PCs on their desks and are becoming accustomed to this style of interface. Accessing host data using terminal emulation software will become less and less acceptable. Putting a Windows/NewWave style interface in front of the host may, in fact, be the only way to meet continually evolving user expectations.

The HP Vectra
Glaxo Pharmaceuticals Workhorse in Production

R A Pearson
Systems Manager, Information Management Division
Glaxo Pharmaceuticals Limited
891/995 Greenford Road
Greenford, Middlesex
England. UB6 0HE

Telephone: 01 422 3434

INTRODUCTION

Glaxo Pharmaceuticals is a subsidiary of the world's second largest pharmaceutical company, GLAXO. It is responsible for manufacturing and marketing in the U.K., and for manufacturing products for certain overseas markets. Some three years ago, three major manufacturing centres were authorised and modern computing facilities were requested. This paper will concentrate on the technical and managerial aspects of progressing these projects and on the reasons for building the system we did. It will not endeavour to justify the system itself although some preamble is provided to give some contextual framework.

OUTLINE OF REQUIREMENTS

The standard personnel, accounting, mailing, etc. systems were supplied as packages from the Information Management Division. The main new requirement was for assistance in the actual production areas - where tablets, vials, and inhaled products are manufactured and packaged.

Good Pharmaceutical Manufacturing Practice (GPMP) requires that detailed records be kept of the those events in production that impact on product quality. In the past several pages of records have been kept for every production batch. There were four major reasons why a move to electronic records was desirable:-

1. Paper records are labour intensive to compile,

2. Once collected paper records are difficult to recall and search through,

3. For a variety of purposes certain aspects of the production records are needed for use in the company data bases and keying the data is costly,

4. Paper is looked upon as a contaminating (dirty) substance to have in a clean environment.

The customer stated that such systems should be capable of:-

1. Recording resource usage - labour, materials, machines,

2. Recording line events - automatically or by manual means,

3. Recording exception conditions - automatically or by manual means,

4. Making use of data already in electronic form, including data already available in HP3000 databases,

5. Having, for 90% of transactions, less than a one second response,

6. Interfacing to a variety of devices such as barcode and magnetic stripe readers, special displays, programmable logic controllers (PLCs), counters, and allowing digital and analogue signals to provide input and output,

7. Being easily and quickly extended without too much cost and hassle,

8. Being robust to the extent that there are few breakdowns and in the event of a breakdown loss of production is kept to a minimum,

9. Allowing maximum flexibility of operation in manufacturing,

10. Maintain data integrity across the company as a whole.

In other words the customer wanted everything! And why not if he is paying the bill!

OPTIONS CONSIDERED

After mapping the logic of the system with data flow diagrams, and getting some idea from our customers of the likely volumes of data, two approaches presented themselves:-

1. To have the traditional central computer with terminals,

2. To have distributed computer power making use of the P.C. which, at that time, was coming into its own and down in price!

Central Computer:

This was the obvious choice - we had built many systems in this way and our customers were used to them. However, there were always the grumbles - the computer is going slow; the computer is down again; why can't we have those graphics we have seen on Tomorrow's World? Our development staff too had their moans - you have to wait an hour to get a compilation done!. If a Central Computer meant just one computer that was attractive, but experience showed us that we always needed more than one - we have currently 17 HP3000s on four sites throughout England exchanging and sharing data over fast communication links. There were too the problems of interfacing sensors, counters, instruments, PLCs, etc.. This would certainly require at least one more computer. HP suggested the HP9000, and this was new territory for us.

Distributed Computers:

Just a tentative idea initially, but soon the attractions became evident. Cost savings (about +80,000 per manufacturing centre), robustness, extendibility, resilience, customer acceptance - in fact this approach could meet most of the requirements given to us. There was a question of response but it was considered that a good design philosophy could overcome the problems here, and a fast response for machine control purposes could and should be met with PLCs. After some initial sizing and experiments to establish feasibility and work involved, a decision was made to go ahead.

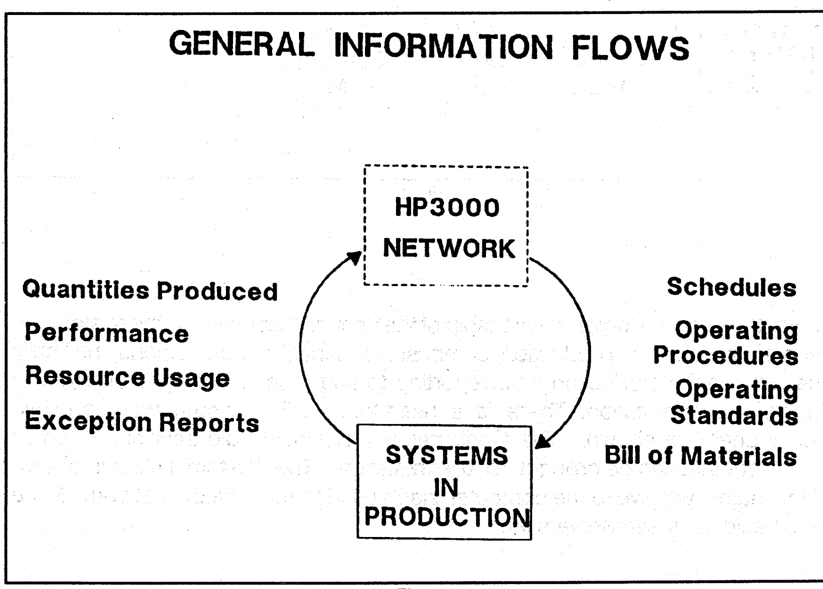


Fig 1

BRIEF DESCRIPTION OF THE SYSTEM

Fig. 1 shows Information Flows from established company databases resident on an HP3000 network to the systems in production. All the information required to make the products scheduled for a series of workstations is collected together and transmitted. Subsequently when the products have been made all data collected is sent up to the HP3000 for batch/sleeper programs to update the company's databases. In general, this routine is followed once a day but is sometimes more or less frequent depending on requirements.

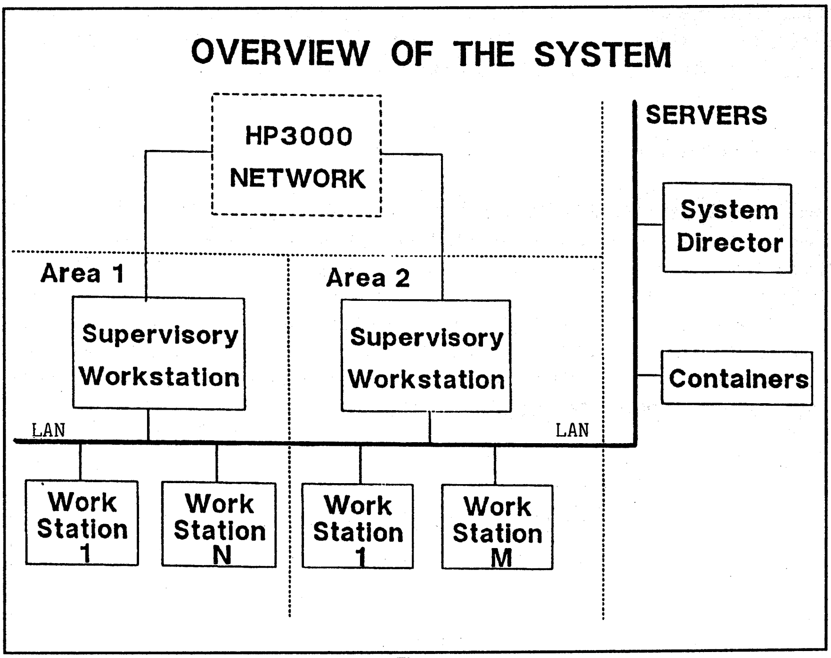


Fig 2

In Fig. 2 we see the logical relationship of the Vectras that make up the system. In general, an area of production comprises a series of workstations, normally one for each production unit, reporting to and receiving instructions from a Supervisory workstation. There is a need too, for Server computers, and two typical ones are shown. The Container Server would hold information on all containers that can be brought to a workstation. The System Director shown is the name we give to the computer that is used to manage the network. More will be said of these Servers later.

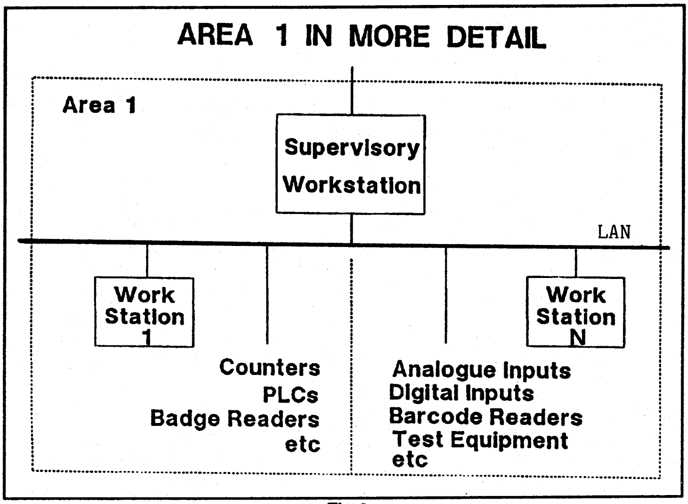


Fig 3

A little more on hardware can be seen in Fig. 3. A variety of input/output devices are used in production and a sample collection of such devices is shown. These are connected directly into the network. Typically the barcode reader is used to identify containers of materials being brought to a production line. An increasing important device is the Programmable Logic Controller (PLC) that is frequently supplied with a machine as part of its control mechanism. This device will have its controlling parameters set depending on the product being made. Thus there is the possibility of sending data to it automatically. In a similar way it can report faults to the outside world.

A few words about the LAN itself. It consists of up to 250 nodes connected by up to 3 Km of co-axial cable. Computers are attached via the serial port, and there can be up to 13 computers per node. Analogue and digital signals are interfaced using an appropriate card. A mix of cards can be put in the node according to requirements. Each serial commmunications card has a buffer of 1016 characters. Transmission over the co-axial cable is at 375 Kbaud.

Fig. 4 shows the information flow from the HP3000 to the various workstations - the data travels via the Supervisory workstation which retains all the information for the computers reporting to it. Thus the individual workstation "knows" what it has to do; the Supervisory workstation "knows" what all its sub-ordinates have to do. Note too, devices are capable of receiving data.

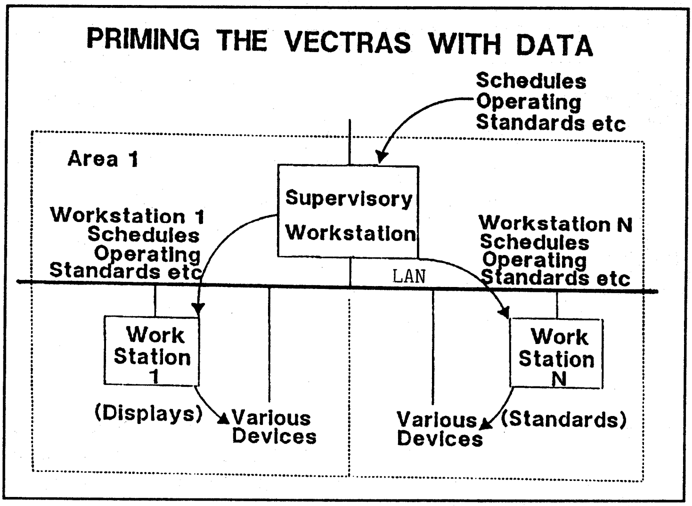


Fig 4

Data is collected as illustrated in Fig. 5; devices can talk with the workstations; the workstations talk to their supervisor; and the supervisor may talk occasionally to the company databases. All this in virtual real time. In effect, all data captured by the workstation is transmitted to its Supervisory workstation. This is useful for recovery in the unlikely event of a Vectra failure. Fig. 6 shows that if Workstation 1 had failed and was replaced by a working Vectra, all the records up to the point of failure were stored in the Supervisory Vectra and can be transmitted to the new/repaired computer. Our strategy is to have a working computer available ready for substitution but it has not been necessary yet to invoke this procedure.

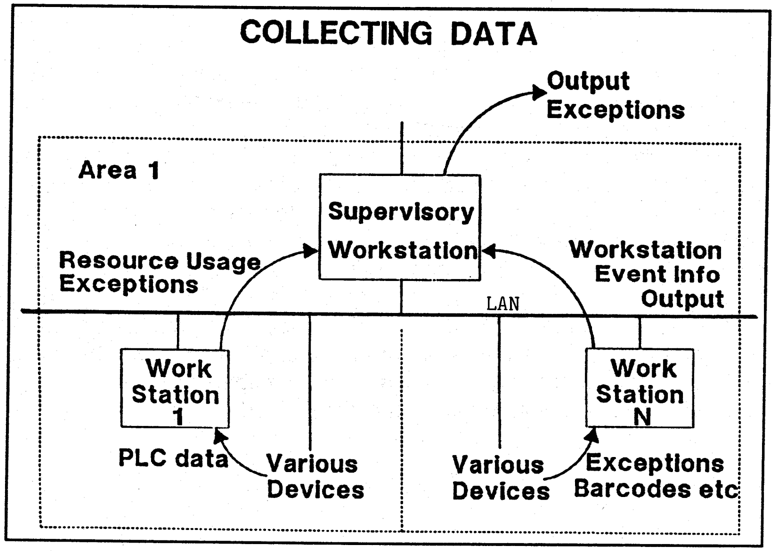

COLLECTING DATA
Output
Exceptions
Area 1
Supervisory
Workstation
Resource Usage
Exceptions
Workstation
Event Info
Output
LAN
Work
Station
1
Work
Station
N
PLC data
Various
Devices
Various
Devices
Exceptions
Barcodes etc

Fig 5

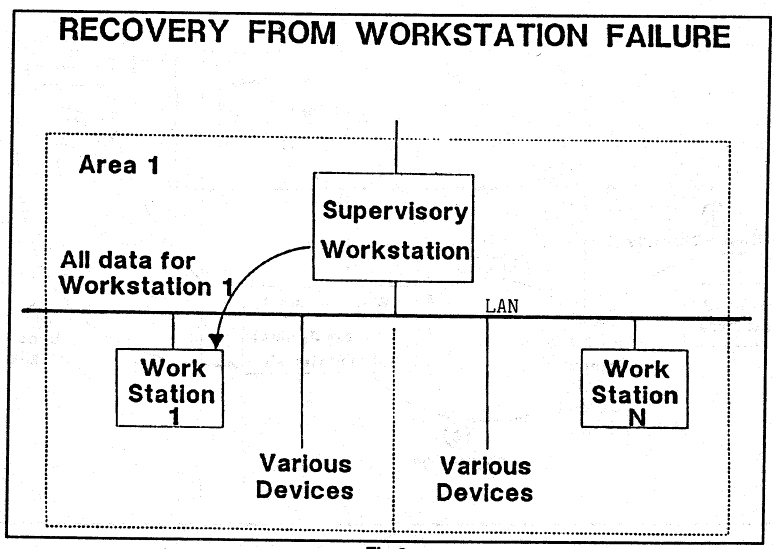

RECOVERY FROM WORKSTATION FAILURE
Area 1
Supervisory
Workstation
All data for
Workstation 1
LAN
Work
Station
1
Work
Station
N
Various
Devices
Various
Devices

Fig 6

To illustrate how the data flows in practice, let us examine the transaction in Fig. 7 - not quite typical, as this one is more complex than most. It is required to identify materials needed for the production of a product. The workstation computer holds the Bill of Materials (the formulation); the container Vectra holds the information on all containers in the manufacturing centre. Typically there are the five steps as shown. A man will read the barcode of the container arriving at the point of production with a barcode reader attached to a small hand-held portable computer. The computer is then connected into the network at the nearest convenient point and the barcode number is sent to the workstation. This then requests the container computer for information on this barcode (container) and on receiving the reply will check the validity of the materials against the Bill of Materials it holds for the product being manufactured. The workstation will then send the "sentence" (material is OK/not OK) to the portable computer for the man to read and take the appropriate action. Such a transaction involves three computers and the network.

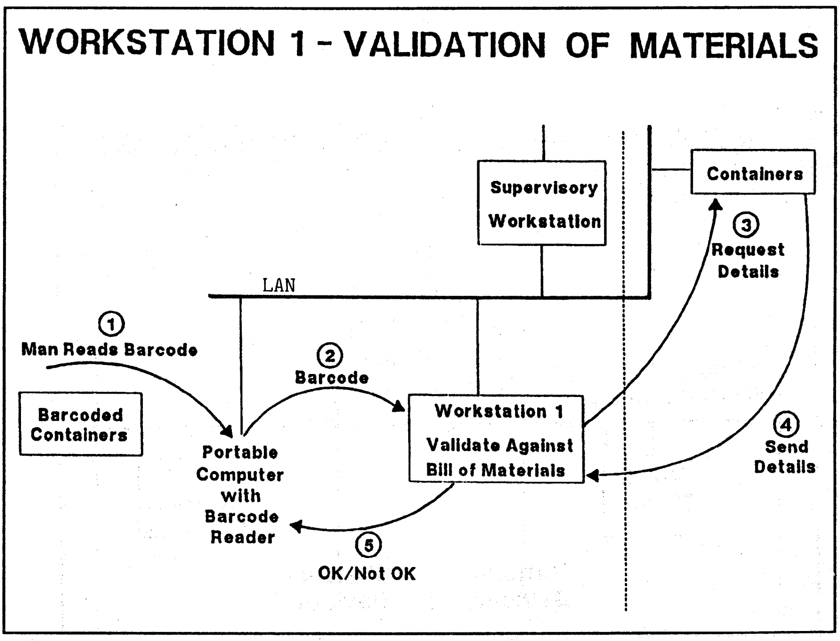


Fig 7

Backup and Recovery Strategies:

Our experience had shown that, by a long way, the most vulnerable part of a P.C. is its disc drive, and so we directed our attention to this. Four possible strategies were considered:

1. Keep a log of transactions on the P.C., and recover by running the log file against the dumped file. As the log file and (for speed of recovery) the dumped file would be on the same drive as the files themselves, all files would be vulnerable when the drive failed. This strategy is not used.

2. Have twin disc drives on the same Vectra, and "simultaneously" write to both files. The system can continue working by reading the non-faulty drive. The faulty drive is replaced as soon as convenient. This approach is used where there are large files to recover, or where the extra cost could be justified. The container computer was such a candidate, as recovery over a network would not be quick enough, and it could slow down other users of the system.

3. Having battery backed up RAM. These are relatively expensive and do not have a large capacity. We do use this in one critical area - in fact we use two which are written to "simultaneously" as in example 2 above.

4. Sending a copy of all data collected over the network to another computer, and recover by sending the data back again. This is more complex to put in place, but has the benefit of having the data on two entirely different computers. This method is frequently used as it can, in addition, be used to communicate to other users of the network, as in the supervisory/workstation scenario already talked about.

In the event of a Power Failure, recovery is as above although certain applications, like the System Director described below, do have a battery backup unit which allows us to close down the system gracefully if the Power Failure is likely to be of an extended duration. The network and the application programs are designed to automatically recover after a Power Failure.

The System Director

At an early stage of development it was recognised that it was necessary to have a means of managing the network. When there was a Vectra or network failure this had to be known about quickly and good diagnostics to pin-point the problem were essential. The System Director is a Vectra, like any of the others connected to the network, providing this service. It is resident in the HP3000 computer room which is manned 24 hours a day and 6 days a week. Any network changes can be entered into the system via the System Director; problems detected are automatically time-stamped and logged by the System Director and subsequently sent to an HP3000 database. This is an important feature as far as our industry is concerned - this way we can quickly and accurately associate the "state" of the networked system with the products we were making at the time.

As the programs running on workstations can affect the quality of our products it is essential to know and record the version of every program. Our internal standard says that the first thing a program will do is to send it's version number to the System Director. The System Director logs this automatically for subsequently incorporating into the HP3000 database.

The System Director has within it a polling schedule for every Vectra on the network, (and for other devices too) and will send a message to the Vectras according to the schedule. If the Vectra does not respond within a specified time, (normally one second) the System Director will alarm the operators.

From the System Director it is possible to "down" and subsequently "up" a part of the network which may be required for maintenance or for upgrade. As mentioned above, complete event logs are automatically produced. Similarly any undeliverable messages are logged to the System Director.

At least once a day the System Director will send the date/time to all the Vectras it knows about, so that all Vectras will tell the same time within about a second.

PROBLEMS ENCOUNTERED WITH SOLUTIONS

When we started building the system the HP Vectra was not around and so we were using the HP150. I will grit my teeth and say no more about that.

The programming staff seemed to have forgotten some basic data processing principles. When a file (or record) is sent from A to B there should be some check for correctness of transmission - e.g. checksums; record counts. Education was the answer.

The system should respond to the customer needs. Programmers have a tendency to look at the technical correctness rather than concentrate on the waiting customer. Programs should be made to respond to the customer first, and housekeeping should be a second priority. In one particular case, staff using ID cards to identify themselves were getting response times in excess of 10 seconds. After re-writing a module of the program response times were reduced to less than a second, our design goal, - even at peak times.

One possible difficulty known from the outset was the lack of a multi-tasking Operating System. The production workstation can get information from the keyboard or from its serial communications port at any time and in particular at the same time. It must serve both and not lose any messages. It was decided that top-priority should be given to servicing the communications port, so programs are written such that the port is always polled. The 1K buffer capability of the node provides sufficient leaway here but as an extra precaution the sending Vectra always asks the receiving Vectra if it is in a position to receive a message. Using this strategy no problems have been experienced in losing messages, and response has been within our target. Although there are multi-tasking Operating Systems available today, the cost to implement on a Vectra is considered too high - unless somebody out there knows differently!

A major problem was documentation - some programmers think that it is completely unnecessary! Potentially there are a large variety of messages that can be sent to any machine - which in effect means sent to any program. The uses and formats of the messages needed to be held and communicated to the programming teams. To do this we invented the term "Transaction Map" which in effect describes for a particular transaction which programs were involved in the transaction and which programs they "spoke" to, and "how they spoke" to each other. Fig. 8 shows the Transaction Map for receiving Materials onto a production Line which we have already discussed. These are very like the "Entity Life Histories" that can be found in some CASE products.

TRANSACTION MAP

Receiving Materials onto a Production Line

Sequence Number	From Device Role	To Device Role	Mess Code	Description
10	LMON	PBCR	AE	Enquiry requesting a reply
20	PBCR	LMON	AF	Reply to enquiry
30	PBCR	LMON	LH	Container barcode from PBCR
40	LMON	CTST	BE	Request for container data
50	CTST	LMON	BI	All data about container
50	CTST	LMON	B!	Container system err message
60	LMON	CTST	BR	Delete container local/backup
70	LMON	PBCR	LV	Container OK/not OK message
70	LMON	PBCR	LF	Cannot access container file
80	LMON	LSUP	DU	Material usage

Device Role	LMON	Line Monitor Computer (Workstation)
Programs	T560 T565	
Device Role	PBCR	Portable Computer with Barcode Reader
Programs	T406	
Device Role	CTST	Container Store Computer
Programs	T403	
Device Role	LSUP	Line Supervisor Computer (Supervisory Workstation)
Programs	T564	

Fig 8

Two different programs may have the capability of carrying out the same transaction - for example, in Fig. 8 program T560 and T565 are used in different parts of the factory but both are capable of receiving materials onto a production line. Their "role" in this respect is the same, so our documentation must be capable of reflecting this.

We had problems associated with picking up data from sensors. The input was supposed to be +24v +/- 10v; in reality there were frequent spikes of +100v and occasional ones of +300v. This generated noise on the LAN causing throughput degradation on the LAN although the system did continue to work. Suppressors were fitted, but our engineers recognised they could have been a bit more clever as regards the wiring. There were other problems too with respect to other devices but it would take another paper to do these justice. However, all's well that ends well!

LESSONS WE HAVE LEARNED

Documentation would certainly be up front today. The concept of "Transaction Maps" we now know and understand. Mapping the basic functions (like Receiving Materials onto a Production Line as shown above) enables one to design the system so that you only need to decide at the implementation stage where to run the function. This will be especially true as and when we move to a viable multi-tasking environment for the P.C.. For example, all the functions/programs could run on one machine, or run on several machines as we have done. For both implementations the documentation would be the same; for individual implementations it is only the addressing of messages that changes.

The various programs were developed on a multi-site basis. Good "up front" documentation would have helped here, but if there was a choice, a central development team would be more effective by creating the "learned and cohesive" unit that is required.

We have encountered more compiler errors than we anticipated. Things are getting better in this area now we have moved to Microsoft Pascal Version 4.0.

It would have been worthwhile putting more effort into the use of data generators early on. We had written some programs to simulate responses to messages and to generate unsolicited messages, but these could have been made more useful thereby saving coding time.

A reliable network is essential. Initially we had problems with the network and as we had already found compiler errors we didn't know where to look for the errors.

If you are using sensors, do talk to your engineers - there is nothing better than talking, reviewing, meeting, and talking again. It will save you time and money in the long run.

We chose Microsoft Pascal. Today we would choose C++.

AND THE RESULTS - HOW SUCCESSFUL HAVE WE BEEN?

Do turn up at session 3601 and find out.

TELEPHONE TRANSACTION PROCESSING

David W. Stover, CDP
Telephone Employees Credit Union
123 South Marengo Avenue
Post Office Box 7058
Pasadena, CA 91109-7058
(818) 795-8181

SUMMARY

Telephone Transaction Processing is not a new concept; the technology has been around for a long time. But until the application was placed on the Personal Computer, it was often too expensive for all but the largest companies. This paper discusses the approach taken by Telephone Employees Credit Union (TECU) to develop a system that would be feature rich and yet cost effective, improve member (customer) service, be "easy to do business with", and utilize existing hardware and staff expertise. At the presentation, transparencies will be shown (and included in the handouts) to more fully explain the PC integration to the host and PBX, and a phone call will be placed to demonstrate how the system was designed to respond to both new and experienced users.

TELLER#PHONE

Recognizing the Need

Prior to a search for an Automated Voice Response System (AVRS), TECU projected that the growing popularity of doing banking business by phone would require an increase in personnel from the then 26 live telephone operators (we call them Information Specialists) to more than 50 in less than five years. These highly trained Information Specialists provide members with a wide range of information about TECU products and services, information specific to a member's own account, and many banking transactions over the phone. Examples of transactions are: making an advance from a line-of-credit to a checking account, making a payment on a loan, verifying account balances and activity such as when deposits were received and checks cleared, and sending a check to the member. This service has received much praise from members for being fast, efficient, and courteous. However, because the service is very labor intensive, management sought to automate the "easy" transactions (such as balance inquiries) and thus delay or preclude the hiring of additional Information Specialists, and provide extended hours of service.

Defining the Requirements

After exploring the many features of several Automated Voice Response Systems with several vendors, TECU management prepared a Request for Bid which specified the requirements, including the following:

1) Allow members to call a local branch number and access a centrally located AVRS (via foreign exchange network).

2) Allow members to obtain a wide range of information, including account, loan/saving rates, marketing, etc.

3) Allow members to initiate many on-line real-time inquiries and monetary transactions.

4) Allow a switch-hook transfer out of the AVRS to an Information Specialist upon user request or error.

5) Allow switch-hook transfers into AVRS from anywhere within TECU's 10 branches or headquarters office.

6) Make the AVRS "transparent" to non-user members so that those who prefer talking to a live person are not aware of the AVRU's presence.

7) Keep AVRS interaction with the mainframe CIF data bases to a minimum.

8) Provide extended service hours, seven days per week.

Selecting a Vendor

InterVoice, a company headquartered in Texas, was selected as the vendor because, among other things, their AVRS was price/performance competitive, was PC-based with a modular growth path, included professionally recorded voice words and phrases, met all of TECU's requirements, and they had a long list of satisfied customers.

Designing the Dialogue

Perhaps the most difficult part of the entire AVRS project was designing the user specification, which included the dialogue. This is the detailed document which describes exactly how the unit will interact with a member, all the menu options, the error codes, etc. It is a difficult and tedious task, but if done well, makes the difference between just a mediocre system or one that is user-friendly and efficient. The best approach is top-down, or looking at all of the possible transactions as being a part of a logical group of menus. Sub-menu layers must be limited, else the uninitiated user will become totally lost in the maze; if frustrated by the system, people will tend not to use it.

One of the features which we feel is important to the success of our system is the ease of access for the uninitiated caller. Rather than requiring a separate number be dialed to reach the AVRS, we decided to place all of our incoming calls from outlying branches directly into the AVRS (with the use of foreign exchange lines into a Northern Telecom SL-1 PBX). Once there, the caller hears the message "This is the Credit Union, please stay on the line..." followed by a 3-second pause, after which the call is transferred to the live operator queue. We then teach the caller that if they wish to use the AVRS, then to press the "#" key on their telephone key-pad during or just after this initial message, which aborts the impending transfer and instead puts them into the AVRS main menu. With no new phone number for members to learn (or for us to publish), the usage of the system was an overnight success. And, for the caller with a rotary dial or someone who would never use such a system, they are not offended by being asked to press a certain key to speak with a live operator.

The Gradual "roll-out"

After a pilot program involving several hundred members, we were ready to introduce the system to all of our 90,000 members. However, we discovered during the pilot that first-time users tend to "try out" every possible transaction, thus causing a very long holding-time per call. Non-users being transferred to live operators would free the ports quickly; but new users could tie ports up for long periods, thus invalidating all those nice queuing theory models. Rather than adding a large amount of capacity that would later go unused, we decided to introduce the AVRS to members slowly to avoid a big rush of new users.

We introduced the service at the rate of 5,000 members per week, thus requiring about 4 months to notify the entire membership. This seemed to work well, giving us a good mix of new and experienced users, allowing us to provide good service and seldom running out of ports. Introductory marketing kits were mailed to each member, explaining how to "get into" the AVRS, and about menus, passwords, and transactions.

One Year Later

As the AVRS processes more and more of the mundane inquiries and simple transactions, live operator time becomes more readily available to work on difficult tasks such as customer problems, loans by phone, and innovative cross-selling. There has been no growth in the number of live operator transactions, and no increase in the number of employees in this area. However, the AVRS now processes an average of 35,000 transactions per week, and has been expanded from the original 8 ports to 24 ports. The AVRS has helped the Credit Union provide better member service, to be perceived as an innovator, and to maintain and build our market share in a competitive banking environment.

Sharing resources among mini and micro computers

Introduction

Today we are witnessing important changes in computer tecnology, not only in computer size, today's computers are smaller than five year ago, changes also given in the power of the computer itself, the micro-computer power has increased so much that words like *personal computer* or *micro-computer* dont fit with the real capability of the computer.

The first of these changes was given in 1984 with the first 16 bits based computer , PC AT model , of course the software running in this computer couldn't complety use the hardware capabilities, but little by litttle the software running in a PC AT or 80286-based computer was taking more control of the hardware capabilities.

Today the problem is not complety solved but most software can use all the address mode of a 80286 process.

The advantages of this processor are : more address capability given a major control in memory and hard drives.

Processors like 80386 begin to be used in the computer world today, without a real software that can take all the procesor advantages. In 2 or 3 years all the software will be capable of using more than 16 Mbytes in RAM and hard disc with more space.

If we talk about speed in a 386 based micro-computers the panorama is :

* Capabilities in memory management that can compete with the smallest in some mini-computers families

* Speed enough to be used as servers in LANs

* Hard disc to storage all the lan users information

When we make a connection of several micro-computers in a LAN some minicomputer´s features can be used without lacking velocity.

A mini-computer connected with micro-computer 80286 based and 80386 based can work linked in two ways :

1.- Micro-computers as mini-computer´s workstation

2.- A LAN connected with the mini-computer using the lan serves as a font-end to the mini-computer.

It´s clear that we win power and performance with the conecction because the capabilities of process in each kind of computer is enhaced with the other end.

This is today´s panorama in the computer arena when we talk about connectivity.

USER VIEW

When we present this panorama the user will make two important questions that have to be solved before making any connection.

There is an answer to be made to our computer's experts in connectivity :

What I can do if I have a mini-computer and my users wants to share resouces among their work-station (PC based) and the minicomputer ?

AND

What kind of resources can they share ?

The first answer, the easy one, in this case is : Basicly we can share files, printers, disc drives and any other hardware connected to the server or connected to the mini.

The second answer, the hard one, is : We can also share information between the equipment, taking advantage of each kind of equipment.

From the micro-computers the resources to be shared are :

* Monitor
* Keybord
* Drives, hard drive and floppy dirive
* RAM memory
* Microcomputer CPU
* Microcomputer co-procesor if it's present.
* Another micro-computer hardware just like mouses and scanners .

From minicomputers the resources to be shared are :

* Hard drive
* RAM memory
* Mini - computer CPU
* Tape unit
* High - speed printers and plotters
* Huge data bases

Working with these two equipment linked together the computer process and performance in all the system can be enhanced and made much eficcient than of a minicomputer with dumb terminal or a micro-computers stands alone.

The advantage of sharing resources cannot finish with the computer's hardware, if we only share hardware and not software we are not winning anything, because our intelligent terminal, in this case a micro-computer, is working in the same way that a dumb terminal, with the clear lose of power in the micro-computer.

To make the subjet cleaver I´m going to show the process in a dumb terminal connected with a mini-computer and the process of a micro-computer linked with a mini-computer :

The dumb terminal's behavior is in this order of process :

a) In the terminal the user writes a command

b) The command is sent to the computer, letter by letter, the user gets a background answer given by the computer in each letter displayed in the terminal

The CPU in the minicomputer is interrupted with each keystroke given by the user making an overhead time in the process.

c) When the user hits [RETURN] the computer process´s the command

d) The user will wait until the computer processes this command and sends the answer to the user.

The next sketch is a ilustration of this process :

With an intelligent terminal the process can change substancially in the next order :

a) In the terminal the user writes a command

b) The command begins to be formed letter by letter in the micro-computer

c) When the user hits [RETURN] the micro-computer processes the command, and if it is a valid command it is transmmited to the computer, if it is not a valid command the user is informed without disturbing the main processor.

d) When the computer processes the command the user gets an answer.

In the next sketch we can see the difference in the process

In bussiness aplications there is an existance of too many little tasks one just like fields validation or display management or keyboard atendence, all these tasks can be done by an intelligent terminal, letting to the minicomputer free of all this task.

If the main processor can be used without wasting overhead time attending dumb terminals , the processor wins time to use in the task involved with the process of information.

This team-work style is of great advantage in offices were the main work is based on huge data-bases shared by all the users, usually in this type of office the user has an overhead time when the mini-computr attend each keystroke given by the user or refreshes the monitor.

A solution to win time and performance is replacing all the dumb terminals in the system for mini-computers.

At once the user will have more computer facilitied sharing hadware.

The next step is developing applications that share the process time in each computer.

INTERNAL WORK

POWERHOUSE is a platform to develop applications where the process time is shared in each computer.

Micro-powerhouse is a subset of powerhouse running in a HP 3000, with this subset we have all the tools needed to make software linked in this two kinds of computers.

The software running in the workstation (80286 based microcomputer) will make all the validations in fields, when the user finishes with the transaction the workstation sends the transaction to the minicomputer and the user is free to make another transaction or wait for the answer.

The steps to make this link are :

a) The form is captured in the worksation

b) All data is transmmited to the mini-computer

c) The transaction is processed by the mini-computer

d) An answer is given to the user when the processor finishes with the transaction.

All this steps are invisible to the user given a better time of processing.

The workstation task are :

* Validate fields
* Windowing
* Menu management
* Hardware administration like keyboard, monitor, mouse
* Acess to a local printer using it as a slave printer.

To the user the process will be real fast because it is done in the workstation CPU until the moment to change the data base.

When the user is filling a form to process data in the minicomputer, the proccess involucred with the form by itself is done in the workstation faster that making all this process in the minicomputers.

In a workstation filling a form with 20 fields can be done in 3 min. Filling the same form working with a dumb terminal can take as long as 10 min.

The mini computer is used to keep data integrity in huge data bases accepting all the transaction from the user.

The mini-computer task are :

a) Transaction vality

b) Data base integrity

c) Share hardware resources

d) Data security

Putting togheter this team we will gain time and obtain an excellent performance in the system

Connections

At the Universidad Anahuac and other schools related to the university we have 2 types of instalations :

1) An HP 3000 with workstation, 80286 microcomputers, linked like terminals and running powerhouse applications.

2) A star-lan using the server like a front-end to the HP 3000

In this mode the worksation is running an inteligent terminal emulation with capabilities of printing and saving data in the workstation .

Using an environment like WINDOWS we can obtain a good performance in the system, OS/2 is our next try in connection to use the multi-task capability in the workstation.

Distributed Processing: Getting Users and PC's into the Act

Don Grady
Infocentre
One Penn Plaza
Suite 1620
New York, NY 10119
(212) 563-2145

While speaking at a local users group meeting in upstate New York, I asked members of the audience what they used their PC's for. One member raised his hand and growled, "We use them to heat the room." The audience laughed and several people nodded their heads in agreement.

It seems that the demand for personalized computing has, in the minds of the mini computer community, exceeded its usefulness. Other than spreadsheets, word processing and DBase-type applications, many PC's are used for little more than terminal emulators. (A user once asked me what kind of configuration I would use if I wanted to "network" individuals into a "centralized server" that would allow "shared accessing and updating" of information. I suggested he hang a few terminals off of an HP3000.) The point is, once we get past the jargon, PCs are still quite often used as terminals or standalone processors. And MIS managers are justifiably angry that DP dollars have been spent on a few high priced "personal" computers rather than on resources which could benefit the general processing community.

The problem is not that PC's aren't valuable to us, or that we have no use for them. Our frustration is, in part, caused by our acceptance of the PC, or at least the PC's power. The implicit question is "Why can't we use all of this computing power to create a more efficient processing environment?" Our focus will be on three ways in which we may answer this question:

- allowing users to download data from minicomputer to PC
- porting standalone applications between minicomputer and PC
- distributing processing across minicomputer and PC

I. PC Downloading -

A. The First Stage -- Transferring Files

When PC's began to gain increased acceptance in the early 1980s, the need to share data among machines became apparent. All PC's allowed users to port data from PC to PC via floppy disc, but data stored on mini or mainframe discs--data generated by the day-to-day activity of running the business--was inaccessible. Programs were eventually developed that allowed PC's to link to the host and transfer files up (to host) or down (to local PC). This was great for data stored in flat file format, but other than word processing files or source programs, most information was inaccessible to users since it was locked safely away in a privileged file of an IMAGE dataset. To get at data, a program had to be written that would open and read a dataset, create an MPE file, and then download it to a PC. (Uploading required the reverse process: reading an uploaded file and programaticaly "putting" or updating records in a dataset. Since users could not, for the most part, write these programs, and since programmers were unable to respond to the volume of requests from each user, downloading and uploading were limited to those processes which were repetitive and clearly defined (such as extracting data to be merged with word processing programs.)

To become valuable to the PC user, a program would have to be written to allow flexibility of extraction. The programmer had to be taken out of the loop so that users could freely manipulate data and customize the files that he would ultimately download to his PC.

B. Ad hoc Report Writers to the rescue

Several utilities--Query being the most common--were available to users as a means of accessing data. But although data could be manipulated, there were several problems: a user had to have a good understanding of how IMAGE stored information so that appropriate sets could be joined and read efficiently; he had to know what data was located in what fields, sometimes needing to inquire within a field (as in arrays or compound items) and most importantly, he had to devote time to write and debug code. Finally, even if the utility provided the ability to create records in an MPE-like format, that format had to be further modified so that it would be acceptable to the PC's program. Even experienced programmers found it an effort to customize file formats for PC programs. To successfully allow users to download data, a report writer must be friendly enough to perform complex database functions simply and understandably, it must recognize differences in PC formats, it must not rely on the user to write code, and finally, it should download the data automatically as part of the file creation process.

The solution to the above problem requires three components:

1) a sophisticated dictionary for database redefining
2) a flexible reporting facility for accessing data
3) a download utility

The primary purpose of a dictionary in this case is to redefine the database so that it is meaningful to the user. This requires several levels of redefinition:

- At the item level aliases should be assigned to items so that they are more meaningful to the users.

- Sub items for compound fields or arrays should be assigned so that each one appears to the user, as a separate item rather than as a position or occurrence within an item.

- Temporary variables and macros can be added so that repetitious or complex programming logic can be stored in the dictionary.

- At the set level, aliases can be assigned for sets or files to make them more meaningful to the user.

- At the group level, sets, files, or databases can be joined so that a user is presented with a concise listing of items that are extracted in the most efficient manner.

At the reporting level, users need to be able to create downloadable data in much the same manner as they would generate reports. The primary difference is the format of the output which is governed by the PC program. The download utility must recognize the required delimiters and input them automatically, and since these delimiters are different from program to program, users must be able to customize them. Users must also be able to do programatic functions such as creating temporary variables, sorting, control break processing, and conditional logic. For the programmer to be taken out of the loop, it is important that users be able to do the kinds of processing that programmers do.

Finally, a download utility or process should be incorporated into the software. Parameters in the host program establish the download utility to be run and automatically transfer the data from one disc to another.

An example of how the above process might work is as follows:

> A sales manager would like to compute the commission for his sales reps which he does at the end of every month on a spread sheet program. Using an emulation package on his PC, he logs on to the HP3000 and runs his ad hoc report writer and generates a report, sorted by sales rep, that displays the products he sold, the price, an extension and total. Once satisfied that he has the desired data, he generates the report in a delimited format that is automatically downloaded to his PC. He may then massage the data (add bonuses, subtract exclusions, perform "what if" calculations) with processes more easily done on the PC. Without a programmer's assistance, the user has customized data for his PC and done so without having to learn a new language or software package. And although the PC and HP are performing very different processes, they are "networked" in a loose sense of the word since they are sharing common files and similar data.

II. Portable Applications

In the previously mentioned example, a file is transferred from one disc to another so that common data can be shared by different programming processes: one machine performs a batch process from a database, the other performs a spreadsheet function. Suppose, however, we needed to perform <u>similar</u> functions on <u>either</u> machine. Specifically, not simply transferring a data file, but an application: DBMS and program code. We could provide similar benefits as in the previous example--familiar programs on either machine, ease of use, common data--but now the PC is <u>processing</u> like the minicomputer. In short, processes that were previously only performed on the HP can now be offloaded to other CPU's. This can be beneficial in two ways: saving resources while developing applications; saving resources while running applications.

In the first case, those shops that develop and process on the same machine know the effect of running database utilities and compilers during the working day. Performance in minimized. Those of us lucky enough to work in environments with separate development machines realize the performance benefits to both users and developers by splitting the tasks. To further optimize the process, we can use each PC as an individual development station dedicating all of its resources to compiling and testing databases and source code. Completed applications can then be ported up to the HP and compiled/created once to put it into production.

More important are the resources that can be saved by moving the production process to the PC. If we can perform minicomputer processes on the PC, we buy ourselves more computing resources for the other HP 3000 users and dedicate all 640K or more of our PC to one process. The performance increases at the PC level are dramatic.

To offload processing in this way is currently being done and should be optimized by looking for the following:

1) Compatible Database Management System. Developers should never have to do duplicate work. Once a DBMS is developed on either mini or PC it should be portable to the other machine without the need for modification.

2) Powerful DBMS - Relational DBMS's on the PC do not provide fast retrieval or allow for complex IMAGE data handling; IMAGE provides for complex applications but does not allow for friendly relational access. Ideally we would want the power of IMAGE with the relational features of popular PC Databases. (see Speedbase from Infocentre for the best-of-both-worlds in PC databases).

3) Compatible source code. As we said, duplicate programming should be avoided. Source code that can run, unchanged, under MS DOS or MPE is a must.

Applications for this type of processing are quite common: sales reps who maintain a subset of corporate data -- customers and prospects for their specific area -- on a portable PC; point of sale systems that emulate corporate order processing at small, single-user branches; manufacturing plants that collect labor data on the shop floor away from the host's location. Building on our previous example of the sales manager, we can now have a scenario where a similar process is run to generate commission data, except that now the processing is done on the PC rather than the HP3000. The end result is that where once the HP did most of the running and then relayed the data to the PC, the PC can now run two legs of the race and allow the HP to perform its functions faster and with less effort.

III. Distributing Processing Across PC's

Up to this point, we have been dealing with either-or situations: either the HP is processing or the PC is processing. In a true distributed environment, this is not the most desirable situation. Optimally, we would like processing to be a cooperative effort with the computers controlling their resources without user intervention.

Instead of being like a race where one runner runs a leg than stops while the next one runs, cooperative processing is like racing a two-seater bicycle. Sometimes one pedals harder than the other but the overall work is shared and the end result is faster performance.

An example of a cooperative environment would be the following: An order entry clerk places orders on a PC which has a subset of customers stored on its hard disc. The entire customer database is located on a disc controlled by the HP3000. When placing an order, the PC looks first to the local database for the customer record. Not finding it, the HP3000 database is searched, the record found, and then retrieved and stored on the PC for subsequent orders.

What has happened here? The processing of the order -- "putting" the order header, calculating the order number initializing fields, checking edit masks--is done on the PC. The DBGET is done on the HP3000. Transparent to the user, the program has decided where to get the data. (And it just so happens, that in this example the PC is pedalling harder!) This is the optimum situation. Many installations have substantially more aggregate processing power with their PC's than on their HP3000. By tapping that power, we distribute the workload and create a more efficient environment across all machines.

Distributed Processing Concepts

How do we go about establishing this kind of environment?

[One solution is Infocentre's Speednet product. Speednet is a unique program that allows IMAGE on the HP3000 to process concurrently with Speedbase (IMAGE-like PC Database). The following concepts are derived from Speednet.]

- Establishing sessions.

To begin, a session must be established on the HP3000. This may be done in two ways: Session mode and dedicated mode. In session mode, the user runs a terminal emulation package and logs on to the HP3000. In this case, users control access to the HP3000. In dedicated mode a system manager streams a job which establishes the network by dedicating ports to the process.

- Directing traffic.

Control of where the data comes from (PC or HP) is done in the PC's database. A line of code in the schema specifies whether data is Remote or Local. No coding is required at the applications source code level. Because the network is established in the schema code, the network is independent of any one programming language and it may be used with existing HP3000 IMAGE databases without changing these databases.

- Aging Data - Static vs Dynamic Data

Just as disc caching stores frequently-used data closer to he CPU, data in the network may be stored and aged on the PC for specified lengths of time to facilitate access rather than forcing an IO to the HP3000 for each record. Database developers may wish to redesign their PC databases so that some items which change frequently (dynamic) are accessed from the HP3000 in a real time manner, while data that changes infrequently (static) is aged on the PC for a specified period. The developers can minimize IO's to the HP by aging as many fields as possible and accessing dynamic fields only when necessary.

Uploading/Downloading/Updating

In this distributed environment, downloading and uploading is done more smoothly and efficiently. Unlike the two-step process described in section I, where one program is used to extract and format a file, another is used to download it, downloading (and uploading!) is done in one step in much the same way as a program might move data from one set to another on the HP3000. The only difference is that one set is defined as local in our schema and resides on the PC, the other as remote and resides on the HP. When creating a record in a remote dataset, we are, in effect, uploading; when reading a remote set and writing to a local one, we are downloading.

Similarly, updating is done by moving a value to a field specified as REMOTE.

To illustrate with an example, we'll refer to our sales manager who has adjusted his salesman's commission records on his PC locally and now would like to write a summary record in his commission file and also update the balance of total sales for each salesman. Both commission and salesman file are located on the host. One program would read the PC data, summarize it, and finally upload to and update the datasets specified as remote. Programmatically, this is a typical process. The uniqueness lies in the fact that it occurs across different discs with different CPUs.

Conclusion

We began by describing how disgruntled many MIS people are over the underutilization of PC's. Where should the MIS professional take his company? The answer, obviously, is that he should take them where they need to go. But more importantly, he should have the vision to invest in solutions that will take them where they want to be. Today's PC solution should be compatible with large mini/mainframe systems. Portability of data and applications rather than specific product features are the criteria by which to judge a software selection. And although today's PC's may be "heating the room," the technology exists to integrate them with the HP3000, initially with report writers that download, but ultimately, with other PC's that will share the processing load. This is the goal for the future.

At a seminar in a predominately rural area, I was being severely tested by an attendee with some very detailed questions on distributed processing. I assumed some complex network was being used by which a myriad of PC's were dynamically processing across multiple platforms and operating systems. I asked, "What is your current configuration?" "Well," he answered, "we've just installed Reflections on our Vectra...." Exasperated, I asked why, given his simple environment, he was so interested in such highly progressive technology. "Son," he said, "out here we plant seeds, not flowers." And that succinctly expresses the focus of today's MIS manager: planting software solutions that will allow us to reap benefits with today's technology but that will also grow into solutions for tomorrow. In evaluating software, we should concentrate on solutions that integrate PC's and minicomputer. Anything less simply has no future.

Enhancing System Resources by using Personal Computers

Leroy Friesenhahn
Blanket Resources
12337 Jones Road
Suite 408
Houston, Texas 77070
(713) 469-0869

The personal computer (PC) has touched our lives in many ways since its introduction in the early 1970s. With the power of the PC rivaling the HP3000 Series II and III of yesterday and the availablity of "user friendly software". the PC stands ready to improve our personal and organizational productivity in numerous ways.

In the past. improving system performance was approached in one of two ways: Purchase a larger computer or eliminate the number of users using the HP3000 system at a given time. The first alternative proves to be very costly and one that many organizations cannot afford. The second option, reducing the number of users using the system at a given time. may cost less to implement. but usually cost the company more in poor information processing. A better method of improving system performance is to transfer task from the HP3000 to the PC. This method eliminates the number of active users using the system at a given time. thereby improving system performance and users productivity.

With the CRT firmly established as the accepted tool thru which the company's information is viewed and processed, the PC is a better alternative with many enhancements not found on standard terminals. The price of a PC is very competitive with that of a new block mode terminal.

A terminal emulation program, such as Reflection from Walker. Richer & Quinn, enables a PC to function as a sophisticated terminal with many features not found on block mode terminals. These features can reduce the load placed on the system and thereby improve system performance.

Type Ahead is a feature of Reflection that provides the user with the capability to type in a series of replies before the system is ready to receive them. The keystrokes are stored within the PC until the HP3000 is ready to receive them. As the system request additional input, the PC responds with the next response. This allows the user to continue with other tasks while waiting for the system to complete its assigned task.

The PC, thru the terminal emulation, can store its display memory. Transferring the information from display memory to a PC file allows the user to sign off from the HP3000 computer and view the information with a simple PC editor. This feature allows information listed to the screen to be reviewed without the necessity of the HP3000 computer being on line. The file can be printed with a printer attached to the PC. Sensitive information can be printed without sending it to the system printer. Remote site can save considerable money on long distance charges by displaying the informationto the PC screen, disconnecting from the main HP3000. and saving the information to a PC file for viewing throughout the day. Once the information has been saved to a PC disk file, it can be rewritten to the screen for viewing. From within Reflection after the information has been displayed. the procedure for saving the PC display memory is as follows:

Press Alt Y (Display the Command Line Prompt)
SAVMEM (Saves display memory to file SAVMEM)

The information has been transferred from the display to a file called SAVMEM.

File Transfer is another important function provided with the terminal emulation software. File Transfer allows for ASCII or Binary files to be copied to or from the PC. This process allows programs on the PC to view, manipulate, and print reports from data stored on the HP3000 computer. PC editors, word processing or spreadsheet software can use this data to produce documents without burdening the HP3000 computer. This eliminates the need for the main computer to execute word processing, spreadsheet, or additional report programs which reduce system requirements. From within Reflection, the following commands are used in executing a file transfer.

Press Alt Y (To display the command prompt)

RECEIVE C:MYFILE.TXT FROM MAINFIL.PUB
(To receive a file from the Host)

File MAINFIL.PUB on the HP3000 is transferred to the PC and named MYFILE.TXT

or

SEND C:MYFILE.TXT TO MAINFIL.PUB
(To transmit a PC file to the Host)
File MYFILE.TXT on the PC is transferred to the HP3000 and named MAINFIL in group PUB

With more information being stored on the PC, a means of recovering the data should a PC catastrophe occur is essential. **Backing up the PC** to the HP3000 is a perfect solution. This operation eliminates the need for expensive PC tape backup units, or enormous quantities of floppy diskettes. The operation can be set to perform automatically at night. This saves the user time and data. Most users would rather take a chance that their PC system will not fail than spend valuable time during the day to backup up their hard drive. The following is a simple program used to backup a PC to the HP3000 during non-working hours. The user is only required to type in "BACKUP" at the "C" prompt before leaving for the day. The PC will wait until a predetermined time and begin the backup process. The program is stored in a file called "HPSAVE.BAT".

File: HPSAVE.BAT

```
R1 MONO.CFG BACKUP.CMD
```

The batch file invokes the backup command file

File: BACKUP.CMD

```
WAIT UNTIL 2300          :start backup at 11:00pm
 PTRANSMIT ""              ;get colon prompt
TRANSMIT "HELLO MANAGER.PC^M"  :log on
 WAIT FOR ":^Q"            :wait for go ahead
 BACKUP C: *.* /S /C       :backup command
 IF ERROR-CODE > 0          ;report error
   TRANSMIT "BYE^M"
   DISPLAY "^[H^[JError in Backup. "
   STOP
ENDIF
TRANSMIT "BYE M"           :ok, sign off
    EXIT
```

By using a PC with a terminal emulation program, a PC program and a HP3000 application can be executed nearly simultaneously. Flipping between sessions is as simple as pressing the Alt-Right Shift Key.

Developing new or maintaining existing software is an excellent example of using the PC and HP3000 together. Begin by starting a session on the HP3000. Using the file transfer facility, the source code is down loaded to the PC. A PC editor is used to modify the code. After editing, the program is transferred back to the HP3000. The program is then compiled on the HP3000. If errors occur during the compile, the program can be edited on the PC and the errors viewed on the HP3000 by switching between the PC editor and HP3000 session with the simple pressing of two keys. The HP3000 no longer supports the overhead of an editor and the programmer can continue to be productive while the compile or test is being performed on the HP3000 computer. During the time the programmer is making changes to the program, the connection between the PC and the HP3000 can be dropped. This saves on telephone charges if a dial up line is being used.

The PC is capable of exchanging information with the HP3000 system but only thru ASCII or Binary file transfers. The majority of information stored on the system is in anIMAGE database. The following is a procedure written to transfer information from an image database through query to an ASCII file. Down load the file to a PC for use by a PC spreadsheet program. The user executes the file called GETDATA which calls the file GETINFO.CMD through Reflection. This procedure sends the commands to invoke a query procedure to retrieve information from an image database and store it in a delimited file. HP's Editor is used to change the "?" delimited to a "," delimited file for use by popular spreadsheet programs.

File: GETDATA.BAT

```
R1 MONO.CFG GETPAY1.CMD  ;invokes the file GETPAY1.CMD
```

File: GETPAY1.CMD

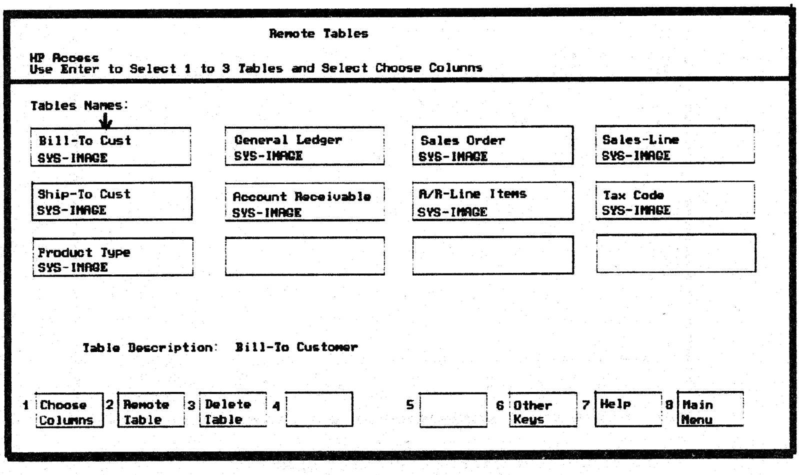
```
QUIET COMMAND ON
PTRANSMIT ""                    :gets colon prompt
TRANSMIT "HELLO ART.MANMAN^M"   ;logs on HP3000
WAIT ": ^Q"                     ;waits for reply
TRANSMIT "PURGE QSLIST^M"       ;purge existing file
WAIT FOR ":"
TRANSMIT "FILE QSLIST;DEV=DISC;REC=-120,,F,ASCII;NOCCTL^M"
WAIT FOR ":"
TRANSMIT "RUN QUERY.PUB.SYS ^M"  ;run query
WAIT FOR " "
PTRANSMIT "DEFINE"
WAIT FOR " "
PTRANSMIT "MANDB.DATABASE"
WAIT FOR " "
PTRANSMIT "ASK"
WAIT FOR " "
PTRANSMIT "5"
WAIT FOR " "
PTRANSMIT "IM"
WAIT FOR " "
PTRANSMIT "DATA.SPECIAL"
WAIT FOR " "
PTRANSMIT "TERM"
WAIT FOR " "
TRANSMIT "XEQ IM1.SPECIAL^M"    ;execute query procedure
```

```
WAIT FOR ":"
TRANSMIT "EDITOR ^M"          :invokes HP editor
WAIT FOR "/"
TRANSMIT "T QSLIST.SPECIAL ^M"
WAIT FOR "/"
TRANSMIT "CHANGEQ #?# TO #""# IN ALL ^M" ;change ? to "
WAIT FOR "/"
TRANSMIT "CHANGEQ #*# TO # # IN ALL ^M"
WAIT FOR "/"
TRANSMIT "K ^M"               ;save file
WAIT FOR "OLD?"
TRANSMIT "Y ^M"
WAIT FOR "/"
TRANSMIT "EXIT ^M"
WAIT FOR ":"          ;after reply transfer file
RECEIVE "C: ACCTNG TRAN.PRN" FROM "QSLIST.SPECIAL" ASCII
TRANSMIT "BYE ^M"
QUIET COMMAND OFF
EXIT
```

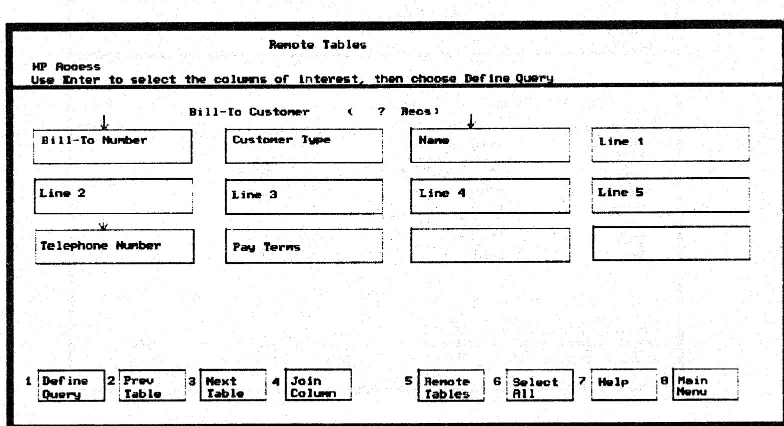
Query Procedure File: IM1

```
FIND ITNO NE " "
OUTPUT = LP
REPORT NOPAGE
D1,"?",1              ;surround fields with ?
D1,ITNO,19
D1,"? ?",22
D1,DESC,52
D1,"? ",54
D1,LPPLLH,65,E1
E1,"ZZZZZZ.99"
END
```

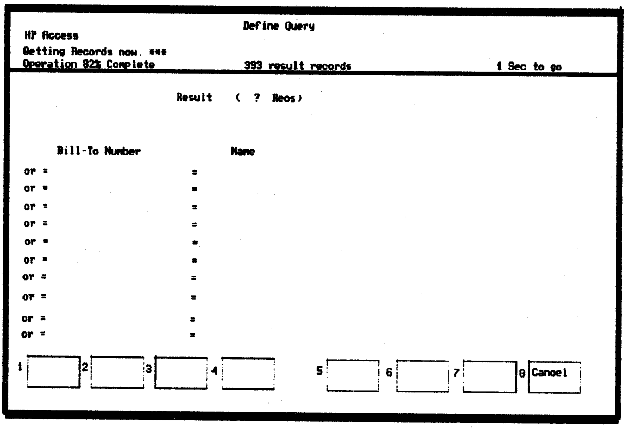
Although the above process will save a system's resources and transfer data from an image database to a file on a PC, a product called "Information Access" from HP is available to perform the tedious task of developing a Query procedure for down loading data into many different PC formats. The following highlights the process of selecting, retrieving, down loading information.



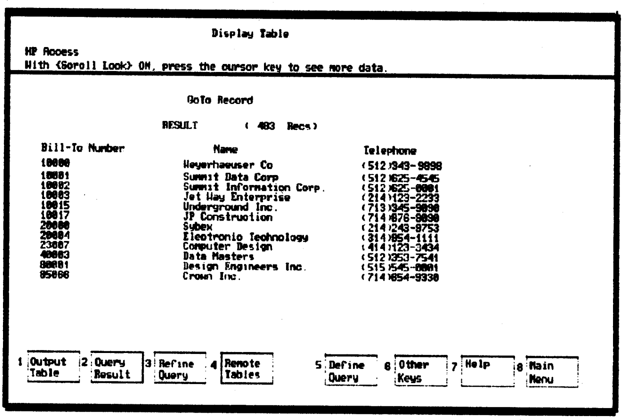
After logging on to the HP3000 and executing Information Access select the desired dataset.



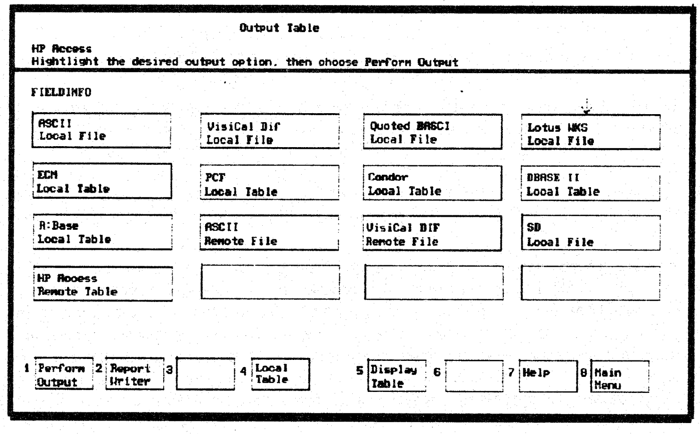
Select the desired dataset followed by the fields from the selected dataset.



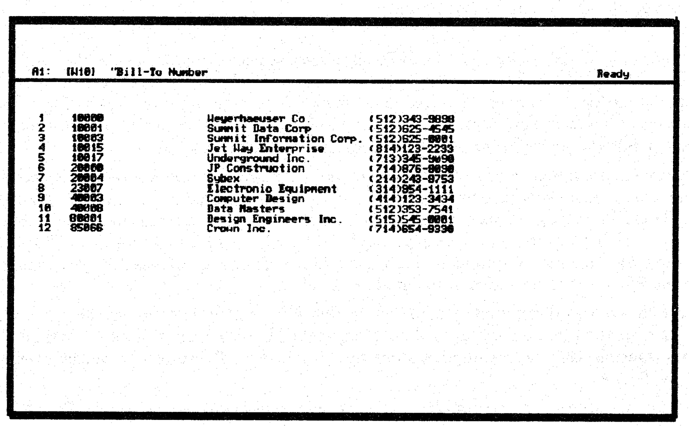
Gather the selected records for further proccessing.



Display the records selected. Additional selecting can be performed.



Select Output File Format



Load Data into a PC spreadsheet

Users of Electronic Mail can improve system performance by creating documents on the PC and then transferring them to the HP3000 for mailing. Advance Mail from HP performs this task well. Word processing can be done much more efficiently on a PC than on the HP3000. Documents can be transferred to the company's computer for use by other individuals within the organization.

Additional work is being done in moving programs from the HP3000 to the PC. A product called Process to Process Link (PPL) from Walker, Richter & Quinn Inc. addresses this approach. PPL is a tool designed to assist programmers in writing applications on the PC while using data on the HP3000. This process uses a device driver on the PC to communicate with the HP3000 thru IPC files. Reflection is used to established and handle communications between the HP3000 and the PC. This approach allow the PC to handle the screen handling and calculation portion of applications while maintaining the data on the HP3000 for all users to access.

The latest technique for improving performance of the HP3000 computer system is the use of 4GL languages. One of these products is called Synergist from Gateway Systems Corporation. Synergist removes many of the tasks from the central computer and places them on the PC. Users of terminals executing in character mode are constantly calling upon the system to service their requests. Requests consist of getting and saving each character entered, interpreting input strings, executing application code, and maintaining database information. Programs using VPLUS for block mode display uses system resources in painting the screen, moving the cursor and reading the input data for processing. Synergist's applications remove these data handling tasks from the HP3000 to the PC. All screens and execution code, developed with the Synergst, reside on the PC and are executed by the PC. The HP3000 functions as a file server handling request for data from the PC and updating the database with information transmitted from the PC. If the database being accessed by the user is small enough or is exclusively used by an individual, it may be stored on the PC. An examination of a typical user session illustrates the advantages of this approach.

A user logs on to the Host from the PC. The Synergist application is started by the user. The program verifies with the Host that the program being executed on the PC is the current version. The date stamp of the code residing on the Host is compared to the date stamp of the program on the PC. If the program on the Host indicates a later version, the program is transferred from the Host to the PC. This insures that program updates are applied without requiring a copy to be manually placed on the PC.

The Synergist application begins execution on the PC. All screens and program codes are stored on the PC. Program execution is controlled by the PC. When information stored on the HP3000 is needed, a request is sent to the Host. The Host accepts the request and searches for the requested information. The information or an error code, if the request can not be successfully completed, is returned to the PC. The PC will process the information and continue to execute, generating additional requests as needed. Information added or updated on the PC is sent to be added or updated on the Host.

To reduce the overhead required by the HP3000, only information that is requested is transmitted to the PC and only information that is changed and updated on the HP3000 is transmitted from the PC to the Host. Keeping data transmission to a minimum between the Host and the PC results in excellent response even on dial up telephone lines.

Except for database creation and application testing, all programming is done on the PC. Applications can be designed to execute as stand alone programs on the PC, stand alone data collection programs with periodic transfer to the Host, or programs that run interactively with the HP3000. Synergist applications give the appearance of a block mode display. VPLUS or block mode applications only verify the data on the screen after an enter has been pressed. Data entry errors are not highlighted as the data is entered. With this approach, the user is required to tab back to the fields in error and correct it before trying again. Synergist's applications allow for verification of data as it is entered at each field giving the user instant notification of an error. The speed of a Synergist application is based primarily on the speed of the PC. The Host is only involved on initial program execution and database requests, thereby reducing even more overhead in current applications.

With the introduction of 386 and 486 microprocessors, PCs will provide additional ways to improve system performance without large investment in central computer hardware. New methods of linking the HP3000 with PC are only beginning to be developed. These new methods will increase productivity in a cost effective manner.

Acknowledgements

HP is a registered trademark of Hewlett-Packard Company. Image, Advanced Mail, and Information Access are registered trademarks of Hewlett-Packard also.

SYNERGIST is a registered trademark of Gateway Systems Corporation.

Reflection and PPL are registered trademark of Walker Richter and Quin, Inc.

PCs, Making Users More Productive While Reducing Mainframe Load

Russell T. Bradford

Bradford Business Systems, Inc.
25301 Cabot Road, Suite 201
Laguna Hills, CA 92653, (714) 859-4428

Much has been said and written concerning the use and virtues of PCs in the past decade. While millions of PCs have been sold, they tend to be terribly under-utilized and are yet to approach their full potential. The majority of PCs today are either being used for word processing, spreadsheets, or are in the hands of users that perform such mundane tasks as balancing their checkbooks, preparing taxes, and filing recipes. An increasing number are also becoming the entree into computers for small businesses. However, PCs have yet to take hold in the area of cooperative processing, where the largest potential use lies.

Cooperative processing is the sharing of the computational load between two or more processors, with each performing the tasks at which it is best suited. A cooperative application such as order entry would allow a PC to handle the capture of the order information. This allows a larger host processor to provide customer information, check inventory, and process the order once it has been submitted. In a cooperative environment such as this, the flow of information might be: a) Customer phones in order. b) Order processing person enters the customer name into their PC which in turn checks a local database of customer names. This provides the order processor with the customer number and an address. All the other customer information such as account balance, order history, etc. remains on the host system. c) The line items are entered, with the PC verifying the product entries against a local data base that contains little more than part numbers, units, and a description. The full inventory remains on the host system. d) Once all the line items are entered into the PC, the order is stored in a local PC data base for later transmittal to the host system as a batch with all other orders captured during this cycle. For orders that require an immediate status on availability, the order could optionally be transmitted to the host immediately with the status of each line item being returned.

In such an environment, the host processor becomes primarily a batch processor and at times a file server for the PC. The PC can be highly responsive while providing a high degree of flexibility. As part of this flexibility, we have achieved full off line capability for each PC data entry station, so that the stations can be disbursed to smaller offices at little, if any additional cost per office. Another advantage to using PCs as front ends is the ability to program them in their entirety, allowing for 100% validation of each keystroke as it is pressed. This serves to increase productivity by reducing error rate.

Why Cooperative Processing?

The advantages are simple. a) By implementing cooperative processing, you can reduce the load on your main computer system by virtually removing the interactive processing. b) Order entry and other user intensive applications like spreadsheets, graphics, word processing, and program development receive the benefits of the more sophisticated PC environment. c) Speed. PCs can often handle interactive applications far faster than most shared central systems since there is no contention for resources on a PC. d) Applications can be readily decentralized, allowing the computing to follow people wherever they go. e) Growth. Without cooperative processing, we need to evaluate the impact of adding another terminal to our system in terms of machine load, port capacity, and response time. With cooperative processing, the evaluation process comes down to the rather simple calculation of whether the current system can handle the increased data and whether there is enough time left in the processing cycle to upload the additional data from the new station(s).

How?

The "how" of cooperative processing turns into several questions. How do I get started? How do I know what to distribute? How do I implement it?

Getting started is relatively simple. First, take small steps and start with the easiest items such as word processing and text editing (program development).

Word processing comes in all shapes and flavors and simply pushing a secretary's terminal off her desk and replacing it with a PC just won't do. There are the minor matters of compatibility, training and features. Many host based word processing systems have features that just aren't available on their PC counterparts. For example, procedures can be created to allow reading and writing of Image data bases and keyed files for complex merging operations such as quotations, dunning letters, and other items that require the use of existing data. Providing this on the PC level would be a large chore as it would require much in the way of custom programming. In cases such as this, a hybrid approach might be required, using the PC for typing and editing the text of the documents that need to access the data files and leaving the actual processing to the host system. Documents that needn't access data such as letters and reports could be handled in their entirety on the PC.

Text editing and program development can be quite a load on a host system, especially if the tools used are not terribly efficient. With a few new products, the programmer and others can perform all text editing on a PC, completely removing this load from the host. While many don't perceive text editing as much of a load, the facts from hundreds of performance tests indicate otherwise. Text editing is very disk intensive, especially for large source files and doubly so if the text editor isn't too concerned with how much disk I/O is performed. In the course of a day, a programmer might edit as many as ten source files, five times each. If these source files are each only 2,000 lines long and he scans through them looking for data five times each, we would have 10 X ((2000 X 5 X 2) + (2000 X 5)) or 300,000 I/Os per day, let alone the amounts of CPU that get chewed up during this process. Many programmers impose a heavier load than this and if you multiply this figure by the number of programmers, the results become quite significant indeed.

Using a PC based editor that automatically retrieves files off the host and puts them back without involving the programmer will actually increase programmer productivity by allowing the editing sessions themselves to go much faster. This will make up completely for any additional delay imposed by a download. Equally important, new products mask the download by performing much of the transfer in the background while the user is already editing the first part of the file. The upload requires that only the changes be sent back to the host, so even this aspect of the transfer becomes a rather insignificant part of the whole process.

An additional benefit of off loading programmers to PCs is to get them started towards a better understanding of cooperative processing. By using cooperative processing on a day to day basis, programmers will become acquainted with it and later will be better able to take steps to further integrate PCs into their computing environment. In other words, programmers will become proficient with PCs through use. Additionally, programmers will become familiar with a primary tool, the text editor, which they will need in creating PC applications that serve as the basis for the PC end of the cooperative processed applications.

Once you have off loaded word processing and program development, other areas such as graphics and spreadsheets can be approached. Both graphics programs and spreadsheet programs are rarely autonomous in that they often get their data from other files or databases. In the process of off loading these programs, you will need to formulate new schemes for handling the extraction and download of this information in a form that is palatable for the receiving application. There are several vendors that offer a variety of solutions to this task which might just turn this procedure into little more than purchasing the right software packages and providing some new training to the users.

Training

Training is far too often overlooked in the process of making the transition to PCs and as such, is often a key reason for failed or stalled attempts to implement cooperative processing. The steps we have mentioned so far are the easy ones, yet in the areas of word processing, spreadsheets and graphics, often the users are either clerical or computer illiterate executives who find the prospect of changing tasks more than a little disquieting. If these early steps toward cooperative processing fail, its true potential will never be realized so training on these early, relatively easy tasks is all the more important. Selecting PC applications that require less training than others is no substitute for actually doing the training, yet this can serve to cut the training time dramatically.

Determining Which Applications Should Cooperate

What we are referring to is breaking an application into two distinct pieces; data capture and processing. The applications that are best suited to cooperative processing are those that have complex data entry requirements. Our order entry example is ideal in that entering an order is rarely as simple as responding to a few questions. Moreover, it is a complex interaction of entry and validation. Each field entered by the operator must pass several checks, many of which require a degree of look up in a file or table. When the volume of these look up tables and files can be reduced to a size that is reasonable in terms of both cost and volume to fit on a PC, your application is a candidate for cooperative processing.

The Goals of Cooperative Processing

Of course we wish to off load the host system, but at the same time we wish to increase productivity and speedup the transaction which provides better service to the customer and maximizes the investment in personnel. As part of the goal of increased productivity, we would create a consistent and easy to use interface that would flow from one application to another. Our overall objective would be to make every application on our computer work in a fashion similar to every other application. This provides an easy transition from one to the next, allowing a user to move from word processing, to text editing, to data entry, to spreadsheets, with a minimum amount of training required.

Consistent User Interface

We mentioned a consistent user interface as one of the goals. This interface becomes the window to all computing that our users see. Simply stating that a PC should be the interface isn't enough, rather a standard method of performing all forms of entry must be chosen. It appears that Windows in various forms,has become the common interface that will be available on just about all hardware, including micros, UNIX based systems, DEC Vax Systems, and most if not all new IBM systems. A notable exception to date is Hewlett-Packard who as of this writing, hasn't announced plans to provide X-Windows on the HP-3000, although they do provide X- Windows on the HP-9000 systems.

Windows

Windows in general is an interface concept which has been around for quite some time. Most people have seen and possibly used an Apple Macintosh, which uses windows as its only form of interface with the user. This interface is the primary reason for the remarkable success of the Apple Macintosh and can also prove successful for your application.

Windows provides a consistent means of dealing with various forms of text and data entry. Much as we have become accustomed to filling out printed forms by filling in blanks, checking boxes and choosing from lists, windows provides similar electronic forms. Additionally, windows provides a means by which a user can quickly scan lists of information and choose from that list simply by pointing to the item or items desired.

Windows are logical and intuitive. From looking at a screen filled with "widgets" that allow one to choose from lists, select items, check boxes and fill in blanks, a user can quickly become familiar with a new application once the initial windowing concepts have been learned.

Windows are popping up everywhere. On the IBM-PC running DOS, Microsoft Windows is available. Under OS2, Presentation Manager is available and on virtually every Unix based computer system and a few proprietary operating systems, X-Windows or a derivative is available. What this means is that after years of competing at the expense of the user, hardware vendors have cooperated to the extent of providing a mechanism for allowing applications to behave in a friendly manner with a common look and feel that is basically independent of hardware. Windows is truly the wave of the future.

On the other hand, the down side to using windows is the fairly substantial cost of purchasing a windows development kit. Moreover, programming windows is not only very different than writing a vanilla flavored Cobol application, but it is also more difficult and requires new training for all but the truly brave. The results are usually worth the effort, but these points must be considered before taking the plunge into windows.

OK. So we plan to use some form of windowing software to front end our PC portion of the application. Now what do we do? First, sit down with your users and determine what they need and want. Find out where errors are currently being introduced and rectify existing problem areas. Next, show a few key users what "widgets" look like and let them design their own input forms, with your help. After all, they are the ones who will be using them.

You still have a few decisions to make. You will want to standardize on a data base system that runs on the PC. A few outfits offer an Image look alike of sorts and perhaps this will do. You also have powerful data bases such as Oracle which run on both the HP-3000 and PCs and this may be an alternative. Other data base systems such as DBASE, RBASE, and Paradox also provide a high degree of capability which could readily serve your needs. Many of these systems are really fourth generation languages which might allow you to write the entire application within them. Not all of these packages provide a windows interface, although it would be anticipated that they will make that move in the near future.

You also need to decide which operating system you want to use on the PCs. Your basic choices are DOS, OS/2 and Unix. All have their advantages and disadvantages. Unix is a multi user system and most likely wouldn't be used unless you have Unix on your host and wanted compatibility or plan to connect multiple workstations to the PC. OS2 is a very powerful operating system which allows for multi tasking and concurrent processing. OS2 comes standard with Presentation Manager and stands to play a prominent role in IBM's future plans. OS2 is somewhat more expensive than DOS and requires a far more powerful computer on which to run. DOS will be around for awhile and will most likely serve the needs of most users, although it lacks much of the flexibility that OS2 provides.

The OS2 advantage might come into practical application in situations where a user would be required to jump from one application to another, without the need to close down one and start the other. Although this capability exists using MS Windows under DOS, the OS2 system was originally designed with this capability in mind and would offer a larger degree of flexibility.

Once the pieces have been chosen, next comes the task of dividing your application into its host and PC parts. The ideal solution would normally be to have all data entry functions handled autonomously by the PC, allowing the host to gather the data one or more times a day by polling the PC. Some applications would require interaction with the host, allowing the PC to do most of the work, while still using the host for portions that can't be satisfied on the PC. In our order entry example, we might have the capability to report the delivery times and availability of items from inventory which might require a host inquiry to retrieve the most current information, while placing a hold on those items the customer is requesting. This normally would be the exception, not the rule.

You will now create the necessary data base structure. Don't forget to set up procedures for updating your PC resident database from the host as information changes, probably on a daily or weekly basis, and create the data entry application to create one or more data files to be uploaded to the host for batch processing.

Tools

A multitude of tools are available for creating such an application. Our order entry application might require a text editor, a file transfer program, a data base system, and possibly some type of data extraction system for the host files, among other facilities that may be necessary. To drop a few names of products that you may wish to investigate:

SpeedEdit	Full Screen Editor for DOS, OS2, UNIX, & MPE
Reflection	Terminal emulator, file transfer
Sessions	Terminal emulator, file transfer
Advancelink	Terminal emulator, file transfer
SpeedWare	4th GL for MPE & PCs
Powerhouse	4th GL for MPE & PCs
Oracle	Data Base System for MPE, UNIX, and PCs
DBASE-IV	Data base Systems for PCs
HP Coop Services	Cooperative Services for the HP-3000
HP NewWave	Windowing System for PCs
Data Express	Data extraction system
...	

Integrating these products can be straight forward or terribly frustrating, depending on your application and approach. Getting programmers used to dealing with cooperative processing is a first step. Moving clerical users off-line as much as possible is another early step. Organizing your needs, wants, and priorities is essential.

Do the Benefits Outweigh the Costs and Headaches?

Like most things, the question of value is relative. The benefits of cooperative processing are many and persuasive, but they come with a cost. Initially, you spend a bunch on hardware in the form of PCs, and ramp up time for programmers will be costly in terms of training and the initial reduction in productivity. In the long haul however, costs may actually lean in favor of cooperative processing. It would seem likely that upgrades to minis and mainframes would be fewer and further between. Software most likely would need less maintenance in terms of the human interface, since that is handled at arms length by the windowing system. As new devices and peripherals become available, the windowing software will handle it, and the application should be relatively isolated. User training time will be substantially reduced, and hopefully, productivity of the end users would increase dramatically. Utilization of modern software offerings may just be practical for the average user, allowing less skilled personnel to perform functions they otherwise might shy away from. A whole new world of spreadsheets, word processing, electronic filing, electronic mail, calendar scheduling, and much more would open up to just about any PC user. This becomes true since the interface to all of these products would be identical, allowing users to choose from drop down menus, push buttons, check boxes, etc.

Summary

Cooperative processing promises to off load host system work to machines which are better suited to the task of interfacing with humans. The user stands to benefit from increased productivity, reduced learning times, and a more appealing interaction with the computing environment. Much stands to be gained in destroying the barrier perceived between the user and the machine by a more inviting and intuitive user interface. The differences between various computer manufacturers will be minimized, thus opening employment opportunities for both employer and employee. Additionally, this masking of differences in hardware will also give the purchasers of computer systems more options when purchasing new systems and might promise to drive hardware costs down due to increased competition.

Cooperative processing isn't cheap and isn't entirely simple to implement, but the benefits seem to indicate that cooperative processing is the direction computing is headed, and soon.

DATA SYNCHRONIZATION

Larry Kingsbury
Gateway Systems Corporation
2400 Science Parkway
Okemos, Michigan 48864
(517)349-7740

INTRODUCTION

Thanks to recent advances in the use of distributed databases, a single piece of data may now exist in databases on more than one computer. Similar or identical databases may exist on a local mainframe computer, a departmental workstation, desk top PCs, portable computers, or any combination of these machines.

For example, a sales force application stores information in a mini- or mainframe computer database at the company's home office. Company sales representatives store similar data in the databases of their laptop computers. On a regular basis, perhaps each evening, the sales reps dial into the home office database. The host computer downloads new information, like product pricing changes, to the laptop computer. During the same interchange, the sales rep uploads data entered that day (such as new customer orders), to the host computer.

Since multiple users may be viewing or modifying copies of the same data, eventually, those changes need to be synchronized. Synchronizing data consists of two processes. First, you must reconcile changes made to different copies of the data in one copy of the data (usually the owner). Second, you need to re-distribute the newly-reconciled data to all machines that use it.

The challenge in a distributed database environment therefore is to reconcile the information at all locations, and to distribute the most recent version of the data to all machines. There are many ways to accomplish this process, each of which has pros and cons for both the programmer and the user. This paper discusses the range of synchronization possibilities and analyzes the advantages and disadvantages of each. It also discusses the frequency with which reconciliation and re-distribution should occur.

Distributed database systems commonly involve a host mainframe or mini-computer connected (continually or intermittently) to several PCs. Most of the examples in this paper describe a host and PC distribution system.

DATA OWNERSHIP

Before your organization can begin to implement a distributed database system, they must make some preliminary decisions. They must consider the following questions.

- Who will use the data?
- Will users need to change the data, or only view it?
- On which computers will the users need to access the data?

Based on the answers to these questions, your company can determine which computer should "own" the data. The computer that owns the data is the primary source of the information and the machine on which you reconcile the data.

SINGLE OWNER OF A DATABASE

A distributed database system in which only one computer owns the information is the easiest to implement and maintain. In such a system, you designate one of the computers on which a database resides as the owner of the data. When users want to access that same data on a different computer, they make a copy of the owner database and use it on their computer(s). The copy of the data is retrieve-only; the only computer on which users may change the data is the one that owns the data. This method is well-suited for data which does not often change, such as a mailing list.

The primary advantage of a system in which only one computer owns the data is that it is simple to maintain. To synchronize data on all the machines that are using it, you simply re-copy the data from the machine that owns it to the other machine(s). Although machines that do not own the data may not change it, in many cases the ability to update is not necessary.

MULTIPLE OWNERS WITHIN A DATABASE

It is also possible to have multiple machines each own different <u>portions</u> of a single database. (It is not possible to have more than one owner for a single piece of data.) You can divide the data at the file, record, or field level. You still may change information only on the computer that owns it. However, each user may change their individual data. This variation is useful when each user maintains a certain portion of a file or database. For example, 10 sales representatives are each assigned a territory made up of 5 states. The customer records for all clients in Michigan, Illinois, Indiana, Ohio, and Wisconsin would be owned by the laptop computer used by the sales person who represents that territory.

A multiple ownership system allows more flexibility than a single ownership system, because more than one user can modify data. You synchronize a multiple owner system like a single ownership system: simply re-copy the data from the machine that owns it to the other machines that need it (for example, from host to PC). This process is more complex however in a multiple ownership setting.

There are disadvantages to multiple ownership of data. Machines that do not own the data are still not able to change the data. Additionally, record deletes may create a dilemma. For example, a problem could occur when one machine owns the customer records, and another machine owns order records. If you delete a customer record, there is no way to delete automatically any corresponding order records.

CHANGING DATA AT MULTIPLE LOCATIONS

A real disadvantage of both single and multiple ownership of data as described above is that you may have several users who all need to be able to change the same data. For example, you might want all of your sales reps to be able to take orders for a product and modify the quantity on hand accordingly.

It is possible to allow multiple users to update data. Users on machines which do not own the data modify their copy of the information. Users send the changes to the owner machine, and the owner machine either accepts or rejects the change. If the owner machine rejects a modification, appropriate action, such as notifying the user who submitted it, must be taken. The owner machine updates its data according to changes that it accepts. The newly-reconciled data must then be re-distributed from the owner to all other users.

Of course, the advantage of this type of system is that all users can change the data. The primary disadvantage is that you must develop a method of handling (or backing out of) changes that you cannot reconciled.

FREQUENCY OF RECONCILIATION AND DISTRIBUTION

No matter what type of ownership your company determines to use, you must also address the question of how often to reconcile and re-distribute the data. Reconciliation and re-distribution need not occur at the same time.

For example, a real estate company has several branch offices in one state. Both the branch offices and the main office use the same listings database. Each evening, the branch office PCs dial into the main office's host computer. The branch offices upload any changes they have made in the listings database to the main office computer. Then the PCs disconnect. The host computer reconciles the changes. Once the host computer makes the changes received from <u>all</u> branch offices, it re-dials each branch office, and downloads the newly-reconciled data to each PC. In this way, each branch office has the most current information available for the entire state.

As with the decision about data ownership, the frequency of reconciliation and distribution will hinge on an individual site's needs. Depending on the data and how you use it, your organization may choose one of the frequencies described below.

1. If the information is extremely static, users may never need to get a new copy of the data. This approach is used typically with historical data, but may also work in other situations.

2. If the information is not likely to change, a company might decide to let users get a new copy of the data when they think they need it.

3. If the information changes at predictable intervals, users may get a new copy of the data periodically, when they expect changes have occurred. This might be yearly (e.g., salary figures), monthly (accounts receivable), weekly, or daily.

4. Periodic updates of the data are also possible. This is similar to the method described above (number three). However, instead of the user obtaining a new copy of all information, only those pieces of data that have changed are downloaded. Here again, you must remember that a deleted record may have other records dependent on it that must be updated or deleted correspondingly.

5. There are software products available that will alert all users that a change has occurred to a piece of data as soon as it happens. (Hewlett Packard offers a product like this, called Silhouette.) This method, however, assumes that all users are connected all the time. What if a user is not connected when changes occur (e.g., they've gone on vacation or are out sick for a day)? What if the link is down? Your system must record all changes so that absent users can incorporate them upon their return.

The frequencies described above refer to users dialing in or in some other way connecting to another machine to receive revised data. However, you can make this process occur automatically. You can mark the data with a timestamp indicating when it was distributed to the machine. When a user initiates an application that accesses the data, the program checks the timestamp. If the timestamp shows that the data has expired (i.e., exceeded the frequency of distribution that you have selected), the program dials the host computer, and new data is downloaded. This could occur every week, every day, or every hour -- whatever frequency is correct for your organization's needs.

LEVELS OF DATA RECONCILIATION AND DISTRIBUTION

In addition to selecting a frequency for synchronization, an organization must choose at what level to perform reconciliation and distribution. The following levels are all possible choices:

- database
- file/set/table
- record/row
- field/column

The smaller the portion of data you distribute, the more complex the procedure is apt to be. However, it may take fewer resources to perform the distribution, since there's apt to be less data involved.

Database-level distribution is a fairly easy operation to perform. If a user has made a change anywhere within the database, the entire set of files is simply re-distributed to all machines, following reconciliation.

File-level distribution is slightly more complicated. Files which have had changes are simply re-distributed to all users following reconciliation. However, there may be dependencies between files that direct that other files also need to be re-distributed. If, for example, a change has occurred to a detail file, both the detail and its master may need to be re-distributed.

Record-level distribution is the next degree of complexity. For each record, you must keep some indication of whether a change has occurred. This may be just a flag or a timestamp. Again though, deleted records may require special consideration. If your company uses other programs which will not recognize the delete flag, it may be necessary to maintain deleted records in a separate file.

Field-level distribution is similar to record-level distribution. However, instead of keeping a history for each record, you must note whether each field has changed. Maintaining a flag or timestamp for each field may require more than a reasonable amount of space. In many instances it is preferable to maintain an image of the entire record as it appeared when the most recent distribution occurred. When it's time to perform the next distribution, the old record image is compared to the current record. Any fields that have been modified are then re-distributed.

COMBINING SYNCHRONIZATION METHODS

It is very likely that your organization will use a combination of the techniques described in this paper to synchronize your distributed database system. For example, your may synchronize different databases, files, records, or even fields at different frequencies. Likewise, not all data may need the same level of synchronization. And, you may choose to reconcile data at one level, but re-distribute it at another level.

Take for example the sales force application described at the beginning of this paper. This application maintains information on the mini-computer at the company's home office. Sales representatives store similar data in the databases of their laptop computers.

Both the host and laptop computers maintain customer records in the CUSTOMER file. The host owns most CUSTOMER fields, but both the host and the PC use them. The PC owns a few fields (customer contact, comments about most recent sales call, etc.). Each evening, when the sales rep dials into the home office database, the PC uploads changes to the host-owned data to the host. The host reconciles those changes with its database. The CUSTOMER fields that are owned by the PC are never used on the host, but they are archived on there. Therefore, any CUSTOMER fields owned by the PC that have changed are also transferred to the host. This is an example of a file whose ownership is shared. The data is reconciled and distributed at the field level, on a daily basis.

Once a sale has been completed, the sales record is transferred from the CURRENT SALES file to the SALES HISTORY file. The host owns the entire SALES HISTORY file. If sales reps need to review information in the SALES HISTORY file, they download the file to the laptop. They have retrieve-only access to the file. This portion of the sales force application is an example of single ownership of a database. Distribution occurs at the file level, when the user (the sales rep) deems it necessary. Reconciliation is not required.

CONCLUSION

The examples above are just two ways in which a single application could use multiple levels of reconciliation and distribution, with synchronization occurring at varying frequencies. The whole process easily can become complex when you must employ several techniques within one system. It is this complexity that makes synchronizing distributed databases a challenge.

DISTRIBUTING APPLICATION PROCESSING

Charles E. Warzecha
Gateway Systems Corporation
2400 Science Parkway
Okemos, Michigan 48864
(517)349-7740

INTRODUCTION

In 1986, Dataquest completed a study that predicted that by 1989, PCs would comprise approximately 43 percent of all business office automation devices (keeping in mind that 21 percent of all offices are not automated).

This prediction has become fact. In 1988, Hewlett Packard reported that business purchases of personal computers in the United States had increased steadily from 1985 to 1987 (see figure 1). Furthermore, in 1988, 90% of all CPU processing (measured in millions of instructions per second -- MIPS), occurred on personal computers and workstations.[1]

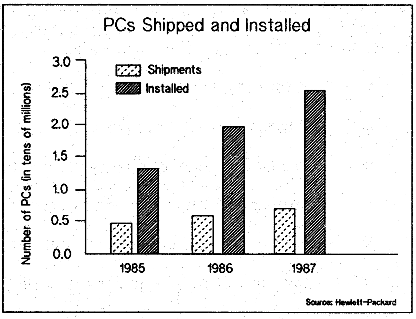


Figure 1

Personal computers provide much more cost-effective processing than a mini or mainframe computer can, which may be one reason for the increase in their use. Processing on a host computer can actually be 30 to 100 times more expensive than on a PC.

As personal computers become increasingly common, businesses face new data processing challenges and decisions. One such challenge is to effectively utilize a mixture of PCs, terminals, and mini or mainframe computers.

In meeting this challenge, data processing managers must decide whether it would be to their advantage to offload the processing of business applications from the mini or mainframe computer (the "host") to personal computers. If you do decide to distribute application processing, a second question must be considered -- should you also distribute the application database? You must decide if you want the database to reside: 1.) on the PC along with the application, 2.) on the host computer, or 3.) on both the host computer and the PC.

[1] George Schussel, Database and Cooperative Processing Symposium, Spring, 1989.

The decisions about where to process applications and where to maintain the database must be made in tandem. Both decisions must be based on where and how you will use the data, and the decisions are dependent on each other.

This paper discusses the advantages of distributing your application processing and databases. It examines the process of developing a distributed application, and talks about methods of distributing the application. Finally, it describes the ways in which you can distribute your database.

DISTRIBUTING APPLICATION PROCESSING AND DATABASES -- THE GOAL

There are many arguments for choosing to distribute application processing and databases. Some or all of the reasons may apply to your company. Your goals may include one or more of the following:

- Reducing the cost of processing transactions.
- Improving both programmer and end-user productivity.
- Providing your company with strategic flexibility. The ability to port applications to all current platforms and also to future planned or unplanned platforms can give that flexibility.
- Producing a high-quality user interface.
- Implementing on-line transaction processing.
- Improving application performance.
- Providing consistency of presentation and use in your applications, regardless of the machine on which they are processed.
- Accessing programs and data that vary according to user and therefore must be stored locally.
- Being able to easily port an application from one machine to another, without extensive coding changes, thus increasing programmer productivity.
- Making full use of PC power.
- Offloading the processing burden of your host computer.
- Achieving Standard Application Architecture (SAA). In an SAA environment, the goal is to implement applications which utilize a common user-access interface, support common communications, are built with a common programming interface, and run in one or more environments.

DISTRIBUTING APPLICATION PROCESSING TO PERSONAL COMPUTERS

Advantages

The importance of fully utilizing your personal computers cannot be overemphasized, for many reasons. The most important of these reasons is that PCs are an economical source of processing power. In an article in Datamation, Dr. George Schussel, a specialist in software productivity tools, said:

> *"The cost of processing logic on workstations is only about one tenth of the cost of executing the same instructions on a minicomputer. Likewise, processing an application on a minicomputer costs only one third of what it costs on a mainframe."*[2]

One measure of processing power is the number of instructions executed by a computer device in a given period of time. This is called MIPS -- millions of instructions per second. Dividing the cost of a device by its MIPS rating provides a measure of its cost effectiveness.

In a 1987 study, Digital Consulting Incorporated compared the cost-per-MIPS of PCs, micro-computers, mini-computers, and mainframes. Figure 2 shows their results, demonstrating that PCs provide the greatest processing power for the least amount of money.

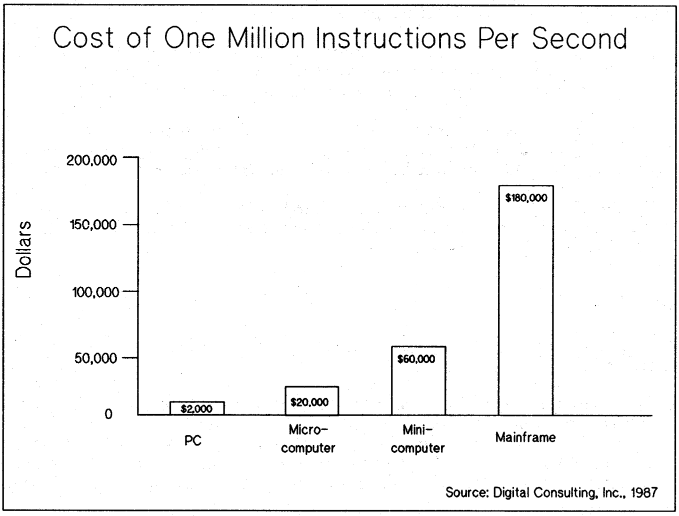


Figure 2

---

[2]George Schussel, "Application Development," *Datamation*, November 16, 1987, p. G-19.

Because PCs are the least expensive way to obtain processing power, it follows that they are well-suited for performing your most CPU-intensive tasks. These tasks include:

- Calculation processing
- Performing logic
- Editing the user's field entry
- Screen presentation
- Windowing
- Cursor movement

Processing applications on the PC also provides many benefits for your users. For example, most users are more comfortable with an easy-to-use PC interface than they are with the look and feel of a host-resident application. The PC interface is usually more attractive, and you can provide users with many special features like windowing, function keys, etc.

Besides the appearance of PC-resident applications, users will also appreciate being able to access popular tools such as word processing and spreadsheets.

Users will also find that applications on the PC are faster to use. Because only one user is accessing the CPU, processing time is faster than on the host. Additionally, little or no datacom is needed, which speeds up response time. Quicker processing and speedy responses lead to increased user efficiency and productivity.

It's likely that each of your end users will need access to materials that are customized for their individual purposes. This could be:

- Database information. End users may need to maintain information that only they will find useful, or that they want to remain confidential. For example, a sales representative might keep notes about the results of each sales call.

- Software packages. End users may also need to use an collection of software specifically suited to their job function -- spreadsheets, payroll packages, word processing, or other business applications for example.

It is beneficial to maintain this individualized data and software locally, on the user's PC. You won't waste valuable host storage space storing information that will only be utilized by one or two users.

Disadvantages

Of course, there are disadvantages to processing an application on a personal computer too.

To begin with, users need to share information. For example, an order entry department and an accounting department will both need to use data about customers. The danger in using PCs is that it is all too easy for each department (or even each machine) to use a different storage format or database. When different departments access the data using different tools, they may no longer be able to share the data.

Database managers invariably prefer the security and integrity of a centralized database to individual PC databases. On a host computer you can deny access to for unauthorized personnel at the application level, the file level, the field level, or by terminal. This is generally not available on the PC.

Another disadvantage is that high performance database management systems are not available for the personal computer. Although larger PC disks (e.g., 300 megabytes) are becoming increasingly common, PC system performance may be very slow when the database is very large. Additionally, it is very difficult to perform routine maintenance on a disk that size. Even a task as simple as a routine back-up can become a complex issue.

DEVELOPING AN APPLICATION THAT WILL BE PROCESSED ON THE PC

When you are ready to develop a distributed processing application, it is only logical that you should build your it on the PC, since that is where it will be processed.

Developing on the PC has all the same advantages that processing on the PC has -- cheap processing power that is well-suited for CPU-intensive tasks, speed, offloading of the host computer, access to PC tools, etc. Additionally, when you develop an application on the machine where it will run, you get an accurate idea of how it will look to your end users.

One important consideration when creating a distributed processing application is what development tool to use. You might choose a 3GL, 4GL, CASE tool, etc. Any of these tools can be employed successfully, but keep in mind that a distributed application needs to be able to communicate with both the host and personal computers. Before you can even begin to coordinate data requests, you will need to be sure that you can establish a protocol and perform the handshake between the PC and the host computer.

If you are using a 3GL, some companies (including Hewlett Packard, for the HP 3000) provide datacom routines that you can incorporate into your programs to perform the necessary communications.

A 4GL or CASE tool is more apt to have built in "hooks" for performing datacom. The important thing to remember with these tools is that some are really intended to be used for building distributed applications, and some are not.

With either a 3GL, a 4GL, or a CASE tool, you might also choose to use a tool such Walker, Richer and Quinn's Process to Process Link (PPL) to establish data communications between the host and personal computers.

Many other differences exist among these advanced tools as well. For example, some may include a proprietary database, others do not. Be sure to check out an advanced development tool completely to be sure that it fits your needs.

DISTRIBUTING THE APPLICATION

The biggest challenge in a distributed application environment is the actual distribution process. This process must automatically reconcile the information at all locations, and distribute the most recent version of the data to all machines.

The distribution process should occur from a centralized distribution point, to ensure security and consistency. Generally, the distribution point should be on the host computer, to provide maximum protection. Workstations may connect to the host directly on an asynchronous line or via a modem. When a user initiates an application on the PC, distribution, if necessary, should occur automatically.

The distribution process must be able to discern what pieces of the application exist on the PC, and what pieces need to be downloaded. For example, if a PC has an out-of-date version of one form, that form should be downloaded. If the PC has nothing, than everything should be downloaded.

The distribution process should also include a security system that will ensure that users and machines can access only those portions of the application that they are authorized to use. Users should be authorized based both on their identity and the PC they are using. You should be able to limit user access at the form level, and on a field by field basis if necessary.

Differing host and PC data formats can complicate the distribution process. During distribution you may need to convert the data from one format to another.

Your organization must choose at what level to perform distribution. The following levels are all possible choices:

- database
- file/set/table
- record/row
- field/column

Often, the smaller the portion of data you distribute, the more complex the procedure is apt to be. At the same time, it may take fewer resources to distribute small portions of information, since there's less data involved.

When you distribute at the database-level, you simply re-distribute the entire set of files any time a user makes a change anywhere within the database.

When you distribute at the file-level, only those files which have been changed are re-distributed to all users. There may be dependencies between files that direct that other files also need to be re-distributed. If, for example, a change has occurred to a detail file, both the detail and its master may need to be re-distributed.

When you distribute at the record-level, you must maintain some indication of whether a change has occurred for each record. This may be simply a flag or a timestamp. When the distribution occurs, you check the flag or timestamp, and send out only those records that have changed.

When you distribute at the field-level, you must track whether <u>each field</u> has changed. Again, this might be a flag or a timestamp. If you find that maintaining a flag or timestamp for each field uses too much space, you might instead keep a copy of the entire record as it appeared when the most recent distribution occurred. When it's time to perform the next distribution, compare the copy of the old record to the current record, and re-distribute only those fields that do not match.

DATABASE LOCATION

There are many database distribution configurations to choose from, depending on your particular needs. You can:

1. Maintain your database on the host computer, while processing applications on PCs.

2. Maintain the database on PCs (networked or stand-alone), and process applications there also.

3. Maintain similar or identical databases on PCs and on the host computer. Users will run the application on PCs, accessing the PC databases. At pre-determined times, or on an as-needed basis, users connect their PCs to the host, and the two databases are synchronized.

4. Maintain databases on both the host and the personal computers. A single application can access both databases.

It is quite possible that a business might use more than one of the configurations described above. In fact, a single organization could use all four of the configurations. An example of such an organization is a company that has a main office on the west coast, a telemarketing office on the east coast, and sales representative offices in several states.

At the main office, this sample company uses cooperative processing (configuration number one from the list above). Account managers, working on their desktop PCs, are able to access information about all of the company's clients. They are also able to audit account activity for the sales reps.

The sales reps use stand-alone PC processing (configuration number two, above). Each representative has a laptop computer on which he or she maintains information about clients. Some of this information is maintained only by the sales rep -- notes about sales calls, contact names, etc. Some of the information, like current billing status, is also maintained on the host computer at the company's main office. On a regular basis (perhaps each night), the sales rep dials into the main office computer (configuration number three from the list above). While the two computers are connected, any new host data about clients is downloaded to the sales rep's PC. Information from the sales rep's PC (sales made that day, etc.) may be uploaded to the host computer. The sales representative may connect to the main office computer at non-scheduled times also, e.g. to check current prices as maintained on the host.

Sales representatives might also use configuration number four (a single application that accesses both host and PC databases). While making a sales call, the rep might wish to check inventory levels for a product the client wants to purchase. The sales rep dials into the home office computer, and runs an application that checks inventory levels. That application might also check any orders that the sales rep has entered in the PC database, and deduct orders for the product in question from the total inventory, before returning information about quantity available.

CONCLUSION

Choosing to distribute application processing and databases dictates that your company must also make many related decisions. You must determine what tool to use for developing distributed applications, and at what levels to provide security and perform the distribution. Finally, you must select the appropriate location for the database(s).

The process of distributing applications and their databases can be very complex, and must be tailored to meet the needs of your company. However, distributed processing has many compensatory advantages. These include increasing productivity, offloading the processing burden from your host computer to personal computers, optimizing PCs as a resource, and providing your company with increased strategic flexibility.

PUSH-BUTTON REPORTING TOWARD A PAPER-LESS BUSINESS SUPPORT ENVIRONMENT

Garry S. Orsolini
Hewlett-Packard
8010 Foothills Boulevard
Roseville, CA 95678-6502

This paper will describe how the Information Systems Group (ISG) of Hewlett Packard has developed push-button integrated business solutions. ISG developed this Executive Insight capability out of their own need for critical business information that is timely, easy to access, and easy to understand. The solution was designed with a focus on information presentation and information access. The information presentation consists of graphical metric displays of key business data, delivered daily to management's personal computers, invoked with the push of a button. The information access component provides business and support analysts with immediate ad hoc PC access to all detail data, from which the presentation summary graphics are comprised.

ISG develops and sells office system software that integrates the personal computer with the HP3000 mini-computing environment. ISG is comprised of three divisions in the United States and one division in Great Britain. The ISG executive management team is focused on a common goal: to provide customers with fully integrated solutions to their information needs.

Bob Frankenberg, General Manager of ISG, had determined that order volume was the group's single key indicator of success. Additionally, the ability to analyze order data quickly was critical to the successful management of the business. The following chart summarizes the ISG business goals and critical success factors:

Hewlett-Packard Information Systems Group

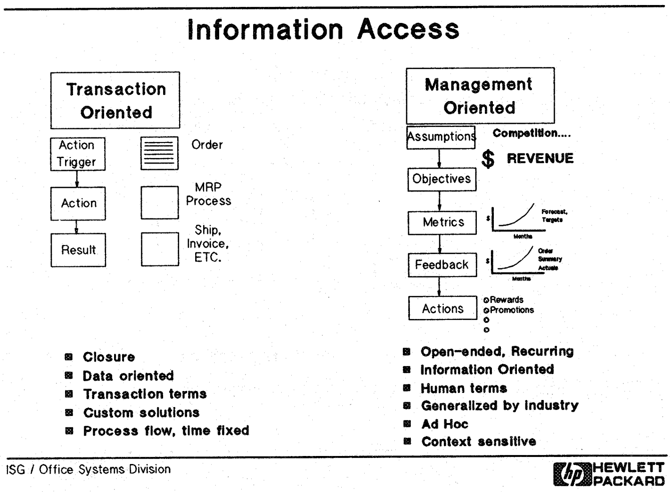
Business Goals	Critical Success Factors ("What must go right")
■ Improve order performance on current products	■ Excellent people motivated to succeed
■ Make NewWave successful	■ Investments in the right applications
■ Make MSS market segment successful	■ Produce quality software and sell in high volume
■ Improve the workplace	■ Manage costs

Prior to the development of the Executive Insight integrated solution, ISG order reporting was done via a batch system where printed reports were distributed three days following the close of the prior month's activities. Throughout the month, periodic verbal updates were given to the management team via the phone. The major deficiencies in this system were:

* order information was not organized or focused on key business issues.
* the information provided was not timely.
* the decision maker could not review and analyze the order data easily.
* the distribution of information was awkward.

Sensing that the existing batch reporting system was inadequate, Bob requested that a better solution be designed. If a system's solution could not be implemented quickly, he was prepared to place an additional business analyst in three of the four ISG operating divisions. A major deficiency in the existing order system was its inability to easily track orders for ISG's software products that had been localized into the native language of a country. Products were being localized, in mass, costing on average $100,000. Localization was being done without the ability to monitor returns (orders) on each localization investment. This meant that key business decisions were being made based upon opinion, not data/information. Thus, without accurate, distributed, world-wide order data, ISG management was spending considerable time and money traveling to each of the four operating divisions. The management team was concerned that considerable time was consumed debating order data and its validity.

To address the above needs, Executive Insight was developed initially as the Business Information Center (BIC). The BIC system was constructed as a prototype to quickly test the viability of building a value-added Turbo Image database leveraged from the existing batch transaction-based order system. The original idea was to build a business support system that was optimized to meet the needs of the ISG management team and ISG business analysts. Instead of being data or transaction oriented, BIC was designed to be information oriented by representing the data in human terms. Typically, codes and abbreviations are used heavily in transaction-based systems. Within BIC, subject oriented literals were included along with their respective codes. Additionally, wherever possible, BIC was designed to provide an easy vehicle for extracting and converting business information into graphs. The following illustration highlights the key differences between transaction-based and information oriented systems:



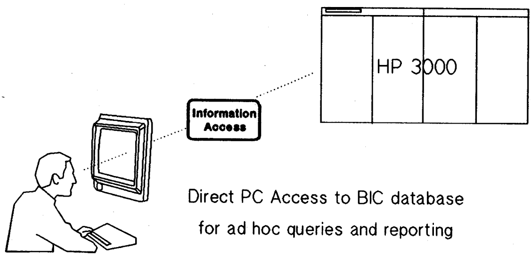
The heart of BIC is the Business-Master. The Business-Master is simply a matrix or table (Image Master Dataset) used for associating business attributes to each of the ISG software products. The following table gives an example of a few of the business attributes:

Product Number	Product Name	Business Group	Oper. System	Local Language
37859A	HPDesk	Communication	MPE	English
36525B	Gallery	Graphics	DOS	English
37859B	HPDesk	Communication	MPE	German
32556A	Advmail	Communication	DOS	French
32323A	Info Access	Data Access	DOS	Dutch
32445C	Symphony	Spreadsheet	DOS	English

With the above table, the BIC value-added environment can facilitate PC ad hoc requests for information organized along the business need.

BIC is information oriented, aligned with business needs:

* product categories and attributes
* customer profiles and SIC information
* geography - office, area, region, and field operation



Direct PC Access to BIC database
for ad hoc queries and reporting

ISG / Office Systems Division

HEWLETT PACKARD

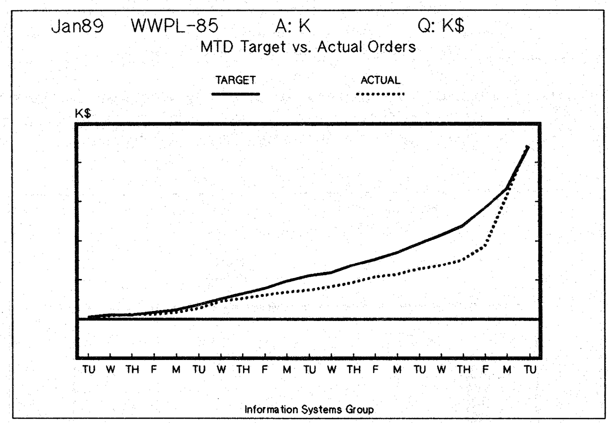
For example, if the analyst wanted to know all orders for HPDESK, they could simply use the HPDESK product name in their request without having to know product numbers (in fact there are over 150 different product numbers and options associated with HPDESK). Another example would be to use the localized language attribute to categorize orders by each language version sold. Thus, it is easy to see that with matrix data, an analysis can be done on multiple dimensions.

It is through the Business-Master and other value-added tables that BIC is able to deliver management oriented, human like information to the knowledge worker and management team. Most of the transaction-based data is represented by codes and is not generally decipherable to a user without a user-guide. For example, the office code "1036" is a key data element in the detail order record of the transaction-based system. The BIC system carries the code "1036" as well as the information that "1036" is the **Orlando** sales office in the **Atlanta** area, located in the **Southern** Region of the **United States** field operation.

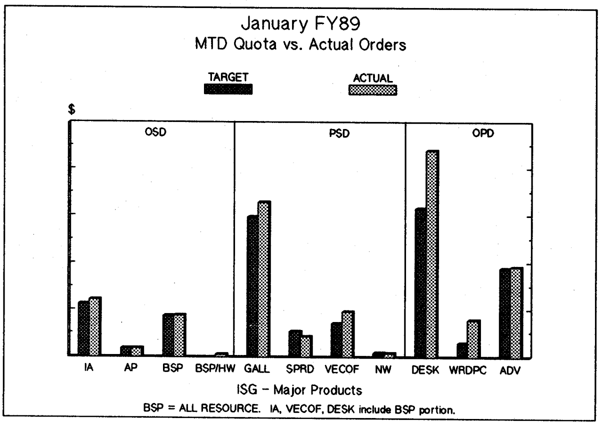
The BIC system demonstrates one way in which ISG leveraged their investment in existing transaction-based systems. The result is a value-added, information oriented, business support system. BIC provides ad hoc detailed information, as described above, as well as summarized graphic metric displays, available daily on the PC at the push of a button.

With the subject or information oriented BIC database in place, the next logical step was to develop an automated process for summarizing the data and converting the actuals (dollars) along with targets (quota) into time series, bar, and pie charts. At the time of development it was decided that the graphs would be pre-processed daily, based upon what the management team wanted to see. This would keep the solution on management's desk as "simple as pushing a single button." This is essentially a merger of our 20th century vending machine mentality with information as the commodity ... and thus a possible explanation for its popularity. Accordingly, it might be paraphrased as, "I've paid my money, I've pressed the button, and now (and I mean immediately), I want the goods!"

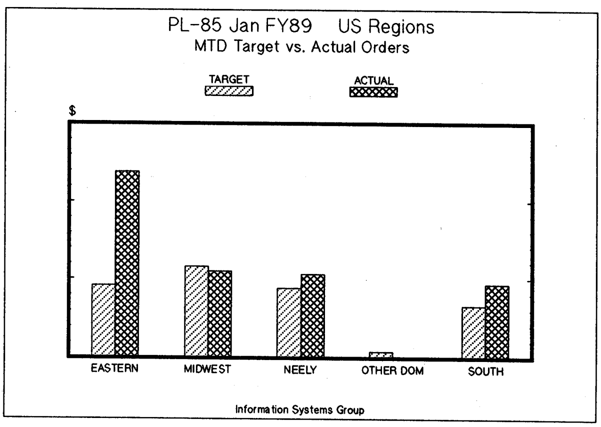
Executive Insight graphic reporting enables the ISG management team to examine daily how the ISG office system software is selling world-wide. This is accomplished simply by glancing at a series of line, bar, and pie charts. Total ISG world-wide orders are first displayed, followed by each of the four operating divisions.



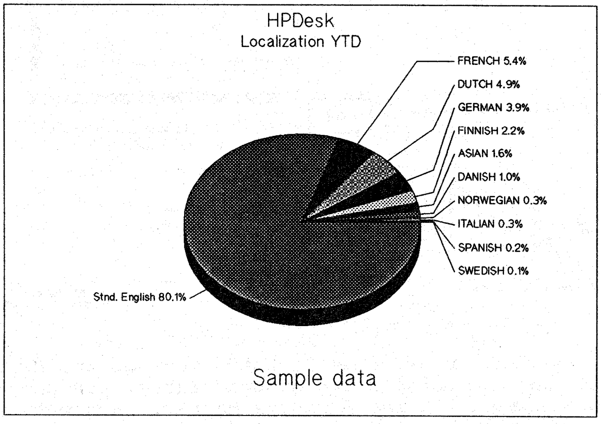
The next graph shows how each of the ISG major dollar contributing products is doing on a daily basis, actual dollars plotted against target.



A geographical perspective shows how each region is performing (daily actual against target), selling office system software throughout the United States, Europe, and Intercon.



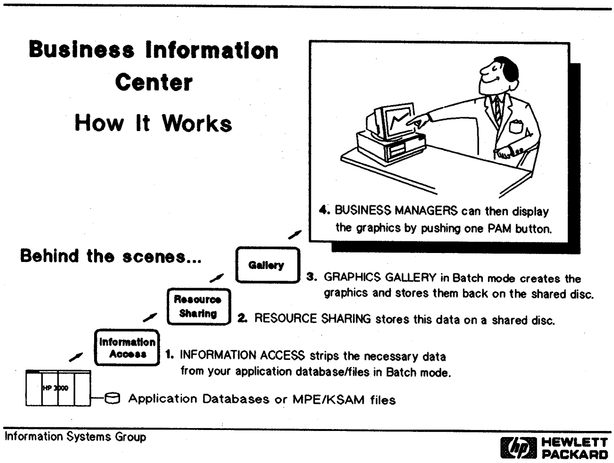
Additional graphs give the ISG management team a perspective of the proportionate share of orders for localized language versions of ISG products. This is done via the local language business attribute that is established through the Business-Master relationship. In the following pie chart, the HPDESK Manager product group is illustrated by language version sales:



The above Executive Insight order graphs are made available at "the push of a button," on a daily basis, to over 150 ISG managers and business analysts throughout the world.

Now we can look in more detail at how Executive Insight provides push-button reporting on a daily basis. As indicated above, BIC as a business support solution, is comprised of a Turbo Image database containing the Business-Master and additional value-added matrix data. A single detail data set (with associated keyed automatic masters) contains monthly and year-to-date dollar and unit data by product, customer, geography, channel, and many other dimensions. The order data is detail oriented, and summarized by dollars and units when the product, customer, and geography are unique. This technique facilitated the inclusion of detailed data in a summary information database. When the value-added matrix data of the Business Master is linked to the summarized detail order records, the result is quick and easy answers to business questions concerning ISG software orders.

The process for generating the Executive Insight graphs is completely automated. Once the daily detail ISG order records have been loaded into the BIC database, the following three steps are executed:

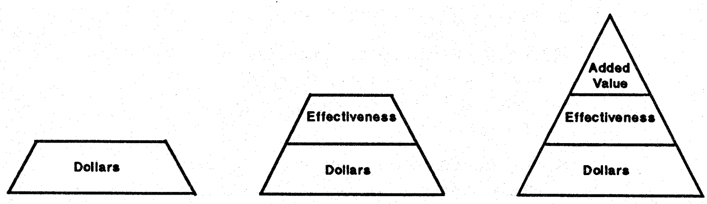


Information Access is executed in batch mode on the HP3000 and the data is easily stripped and grouped based upon the business attributes assigned via the Business-Master. Each cut of the data is saved in either ASCII or DIF format on the HP3000 in MPE format. The second step is to convert the MPE data files to DOS data files in the format filename.GPD. This can be done in the same batch JCL using the DiscManager COPY command within Resource Sharing. Once the newly stripped data files have been stored on the virtual disc in DOS format using DiscManager, the third step that creates the graphs is initiated. The creation of the graphs is accomplished by running Charting Gallery in batch mode on the PC using command files. The command file interface to Gallery 3.0 allows you to merge data files (.GPD) with chart files (.GPH) and save the graph as a picture file (.GAL). Most all the interactive features of Charting Gallery have been replicated in batch/command mode. Especially useful is the ability to update the title, subtitle, footnote, and X/Y axis within the Charting Gallery command files. As each graph is created, it is saved on the virtual disc for access by anyone connected to the local area network. The above three steps are fully automated and execute in succession across the HP3000 and PC environments on a daily cycle. Next (as illustrated in the 4th and last step above), the management team can review the Executive Insight graphs at anytime by simply pressing a PAM label entitled "ISG Order Graphs". The PAM label will invoke a DOS .BAT file that executes Drawing Gallery with an "info" string that links to the command file containing the file names of the graphs to be displayed.

Page-Down and Page-Up will allow you to review the graphs in forward and reverse order respectively. The ESC key is used to exit the review process.

To determine the overall impact that Executive Insight had on the Information Systems Group, a joint study was conducted by Hewlett Packard and Arthur Young. The following illustration categories the benefits achieved through BIC in terms of dollars saved or costs avoided, increased effectiveness or productivity enhancements, and added-value to the organization:

BIC Benefits Analysis



The Dollar Benefits

- Personnel cost avoidance for professionals saving $300K
- Clean up localization cost savings $1M
- Travel expense reduction of $45K

The Effectiveness Benefits

- Total productivity savings: $453,600.00
- Improved timeliness of data... daily vs. weekly or monthly
- Improved accuracy in forecasting
- Fewer order errors

The Added Value Benefits

- Improve teamwork and cohesion of workgroups and divisions
- Improved problem resolution
- Improved responsiveness to market situations
- Ability to identify market trends and invest in the right applications
- Allows users to better understand the business

The Executive Insight BIC system has enabled ISG to save $1.345 million in operating costs and gain an additional $453,000 in productivity savings annually. In the less tangible area of added value or behavioral enhancements, BIC has played an interesting role within ISG. Starting with the fact that order volume is ISG single key indicator of success. It is extremely important to make order performance visible to the entire organization. Displaying daily order graphs on the desks of management, business analysts, engineers, and marketing analysts, is an effective way of communicating the "rules of the game". And more importantly, each day the ISG team knows what the score is.

Executive Insight reporting has provided an integrated business support environment for tracking and analyzing sales of world-wide office system software. A few of the major aspects of its success are listed below:

BIC
Attributes of Success

- The system is tied to ISG's critical business needs and objectives
- Management support
- Timely and accurate business data
- The BIC output relates to how ISG does business and makes for easy access and enhanced understanding of data
- Push Button Reporting, Ad-hoc inquiry, automated reporting

In the final analysis, Executive Insight push-button reporting, is simply one way of focusing the organization on those issues of critical importance to ISG. It is an integrated tool or vehicle for communicating to all members of the organization, where we have been, where we are today, and where we need to be tomorrow in order to realize our business goals and objectives.

FUTURE DIRECTIONS FOR HP'S TERMINAL EMULATION STRATEGY

Nicola S. Cowburn

Hewlett-Packard
Office Productivity Division
Nine Mile Ride
Wokingham, Berkshire RG11 3LL
England

INTRODUCTION

While the trend is towards more computer power on the desk top, and local information processing, and towards more architectures like Hewlett-Packard's Distributed Application Architecture (DAA), centralized databases will continue to play a significant role, requiring some form of terminal access. Booking systems such as those used by national and international airlines and car rental companies are good examples of a requirement that cannot be fully met by Local Area Network (LAN) connected workstations. Organizations will continue to maintain centralized data resources, storing data within purely host driven applications. The personal computer will increasingly be used for the manipulation and presentation of data. Terminal emulation applications will be used as servant applications, acting as a tube between the host computer and the PC, through which data can be transferred. Terminal emulation will therefore remain an important component of the integrated office for the rest of this decade, and well into the next.

Terminal emulation will be an important tool for the success of Hewlett-Packard's Co-operative Computing Environment. As organizations move towards CCE and DAA they will still need to provide user access to their existing HP systems and applications, as well as those of other non-conformant suppliers. They will also continue to want access to independent information services, like the Dow Jones share index, which will not initially offer their services in a DAA compatible form.

HP's terminal emulation strategy is derived from its Organization Communication vision, which focuses on providing:

effective, efficient communication, independent of location;

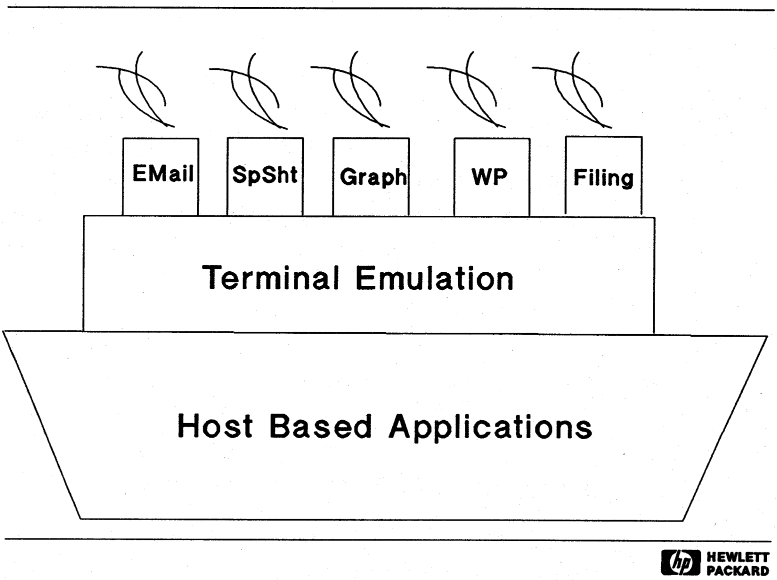
systems built on a solid base of agreed standards such as the OSI standards, X.25, Unix etc;

a consistent window to the world.

HOW DOES TERMINAL EMULATION FIT INTO THE BIG PICTURE?

The concept of "building blocks" provides a suitable analogy for the type of information software products that HP expects will be required by target customers in the next decade.

The need for customized unstructured information systems is being driven forward by end users' demands for access to more information sources or information warehouses and more individual information presentation/analysis profiles. This need will be met by modular software "building bricks" based on industry standard connection interfaces, which allow solutions to more closely match individual knowledge workers' requirements; and they will be relatively simple to put together without excessive custom "glue". Generic product modules, such as Terminal Emulation, Electronic Mail, Filing, Graphics and Spreadsheets, will be the foundation building bricks for such systems. These will then be marketed as individual packs, or in collections of solution packages, which best meet the needs of individual departments.



These generic bricks will also be integrated by third parties into highly customized software solutions which will be designed to meet the information processing needs of vertical markets. In these applications, the software building bricks will, in many cases, be transparent to the end user. This highlights the importance of NewWave and a NewWave terminal emulation solution in providing the transparency, the ease of use and the automation that is a prerequisite to this packaging strategy.

WHAT IS HP'S TERMINAL EMULATION STRATEGY?

HP's terminal emulation strategy encompasses the spirit of the Organization Communications vision and can be expressed in the following way:

> *HP intends to become a world leading supplier of terminal emulation software.*

A breakdown of this mission statement into key phrases helps to understand its full meaning:

"... World" - HP will serve the world market, offering local language versions of terminal emulation products, which meet local country legal and methodology requirements and capitalizing on its world-wide channels of distribution. HP's goal is to satisfy the data communication needs of its target markets through a tighter PC/host integration solution.

"Leading Supplier" - As a leader, HP will set standards in terminal emulation. HP is a key industry player in the development, adoption and application of new technology; either through internal development or through partnerships with third party vendors. HP's terminal emulation goal is to take full advantage of these technologies, and to provide tighter integration both within and between technologies.

"Terminal Emulation Software" - HP aims to supply terminal emulation, file transfer and command language capabilities in the form of products which run on industry standard platforms such as MS-DOS, OS/2 and Unix/X, and which allow connections to all the major vendors' host computers.

The key elements of HP's terminal emulation strategy are as follows:

1. Multi-Vendor Connectivity

Multiple End User Platforms

MS-DOS

Non-Windows

The market for MS-DOS terminal emulation has been the largest sector of the terminal emulation market over the last five to ten years, and will continue to be so for some years to come. HP's strategy is to continue to support customers with investments on this platform.

HP NewWave

HP NewWave is a PC software environment which currently runs together with Microsoft Windows/286 and MS-DOS, and in the future will be available on the OS/2 and Unix platforms. It is not an application or an operating system, but it extends the capabilities of both. HP NewWave provides a consistent graphical user interface and adds object management and system wide services to personal computers.

HP is heavily committed to NewWave, and the terminal emulators offered by HP for use in this environment will take full advantage of its object orientated, task and agent facilities.

Microsoft Windows

Microsoft Windows has become accepted as the standard user interface for the IBM PC and compatibles. The use of terminal emulation under the Microsoft Windows environment offers a number of exciting opportunities for increasing the usability of the system and in turn the productivity of the user.

The benefits of using MS-Windows based terminal emulation include the use of copy/cut and paste, which allows the user to quickly and easily include an HP DeskManager electronic mail message, for example, into a Windows-based word processing document simply by copying/cutting to the clipboard then pasting it, or vice versa. It would also be possible using a terminal emulator to run several host applications on different host computers simultaneously, each viewed through a separate emulator window on the user's screen. Background processing is a natural by-product of operating in a windowing environment; files could be transferred in background whilst the user works on another PC or host based application.

HP's strategy is to continue to develop terminal emulation solutions within the Microsoft Windows environment.

OS/2

OS/2 is expected to supercede MS-DOS as one of the dominant operating systems on the desktop. HP's strategy of providing a common user interface and functionality across all the popular platforms will ease the migration of users from Microsoft Windows.

X/UNIX

Unix, in combination with the X Windows graphical user interface (GUI) is another highly acclaimed end user platform of the future.

X Windows is a platform which is now endorsed by most of the major computer vendors in the market-place today, and commercial implementations are just beginning to emerge. Perhaps of more significance is the announcement by the Open Software Foundation (OSF) that it has adopted X Windows technology as the basis for the user environment component (UEC) of its open operating environment. In addition, DEC's commitment to DEC Windows and the signing up of independent software vendors to undertake development on this platform will probably increase the rate of diffusion of X technology into the market. As the cost per X workstation decreases, the commercial adoption of X technology will, no doubt increase.

HP will be represented in this multi-vendor, multi-platform environment, and its terminal emulation strategy will extend to incorporate support for the Unix and X Windows platforms.

APPLE MACINTOSH

While the Apple Macintosh might not be considered an industry standard, and support for this platform is not HP strategic, HP does recognize the need for some of its customers to connect the Apple Macintosh into their wider computer systems.

HP will facilitate this integration by extending its terminal emulation strategy to incorporate support for the Apple Macintosh workstation.

Multiple Hosts

HP is committed to providing a Co-operative Computing Environment (CCE) in which a user at a single workstation can easily access the applications and data he/she needs, wherever it is located.

HP's terminal emulation strategy supports the CCE concept by allowing access to a variety of HP and non-HP computers (for example HP 3000, HP 9000, HP 1000, DEC Vax) as well as to public information services such as Dow Jones and Compuserve.

Multiple Connections

HP's customers employ a wide range of data communications technology, and the terminal emulation strategy is therefore to allow workstation to host connections through at least the following:

Serial/RS-232/422
X.25
Telnet
HP Local Area Networks
Other Vendor's Local Area Networks

2. Consistent Graphical User Interface (GUI) Across Platforms

Whether your workstation of choice is an IBM compatible PC, Apple Macintosh, Unix or X workstation, HP's strategy is to provide a consistent window to the world.

Until recently, applications developers have been provided with virtually a single target in the form of MS-DOS running on IBM compatible personal computers. However, when looking at the field of emerging windowing technologies there are far more opportunities to be considered. For the IBM PC compatible platforms there are Microsoft Windows, Hewlett-Packard's NewWave and Microsoft/IBM's Presentation Manager. In addition, business use of the windowing Apple Macintosh, is increasing, and interest in applications developed under X Windows is rising.

A growing number of PC users, seeking greater ease of use, increased productivity, lower training costs and increased cost effectiveness from their personal computers, are now looking to graphically driven windowing environments to meet their needs. These users typically expect to have a terminal emulation solution which not only operates in, but which aesthetically matches these environments.

HP is committed to providing terminal emulation solutions which support all of the aforementioned windowing environments and which provide a consistent windowing interface across platforms. The major contribution made by such consistency is to minimise support and retraining costs, and to reduce user error through the provision of intuitive and easy to use terminal emulation, consistent across all platforms.

In addition, support of industry standard platforms, combined with a consistent user interface across those platforms, will serve as an aid to migration from host computer to host computer, or from workstation to workstation, thus reinforcing the concept of a co-operative computing environment.

Leadership for HP will be achieved through leadership in the market for the user interface to all forms of electronic communication.

3. Tighter Integration Between the Workstation of Choice and the Host

By taking advantage of the capabilities provided by such windowing environments, a new generation of more powerful terminal emulators can be offered. Integration will be facilitated by the provision and use of powerful command languages, inter-program communication and also, within the HP NewWave environment, the HP NewWave agent facility. The terminal emulators of the future will make the exchange of data between applications much tighter and simpler than it is with those currently available.

Another powerful integration facility which HP intends to provide with future terminal emulation tools is known as Dynamic Data Exchange (DDE) and will enable the user to control any given procedure from any one of a number of applications.

Example 1

Scenario: Kathy is a product manager who receives an electronic mail message in her intray on the first morning of every month, which shows a summary of the previous month's orders. The order figures have been extracted from a database using Information Access, and are then sent on to Kathy using a batch job set up by her MIS department. Kathy has no need to know how the figures were collated or where the figures have come from. Kathy must then issue an Order Summary Report to her colleagues and management within two working days of the end of each month.

Kathy needs to incorporate the month summary data from the HP DeskManager message into the Order Summary Report as quickly as possible, without having to re-enter the figures, so that she can distribute the Report.

Solution: Using a windowing terminal emulator, the menu items offered could be customized to include options to "copy item to desktop", "move message to MS-Word", or "copy item/message". Kathy could select such a command file defined option simply by double clicking on the menu item. The command file would automatically start MS-Word (which appears as an icon on the screen), open the DDE channel, start logging to DDE, read the item, stop logging to DDE and then close the DDE channel. Kathy could then edit the MS-Word quickly and easily, enabling her to distribute the report in a timely fashion.

In this example, Kathy has been able to transparently integrate host based data into a PC word processing application, achieving inter-program communication by means of the terminal emulator's sophisticated command language to control the data exchange.

Example 2:

Scenario: Barbara is secretary to a Personnel Manager, and uses HP DeskManager to mail documents and messages to employees around the company. Her word processor of choice is Microsoft Write. Barbara would like to create an MS-Write document relating to commission structures, and then send it to all members of a pre-defined sales force distribution list.

Solution: MIS could write a number of script files to modify a particular function within HP DeskManager. From within HP DeskManager, Barbara could then type "create" and would be prompted with a number of format options, one of which could be MS-Write. On completion of the MS-Write document, HP DeskManager would initiate a hostcontrol script which would take control of the terminal emulator. The terminal emulator would then be used as an agent to execute the HP DeskManager script. The script may use a facility such as DDE to define the parameters and send the MS-Write document on to all those on the distribution list. This process would, of course, be transparent to the user.

In this example, Barbara has been able to create and mail a PC word processing document from within a host based application using a simple three step procedure (initiate creation, create document, request mailing).

Example 3:

Scenario: John is an Accountant who uses Microsoft Excel to manipulate and work on his company's sales and expense related data. John must access sales data on a host database and incorporate these into his spreadsheets before he can make calculations and decisions based on the latest available data.

Solution: Excel's macro language could be used to define extra menu entries, such as "Sales". Once an Excel macro had been written for John by his MIS department, he would be able to start Excel and double click on "Sales" if he wanted to update his spreadsheet with the latest available sales data. The macro would display a dialogue box in which John could enter the product numbers and regions that he wanted to incorporate into his spreadsheet. Excel would then become iconized as the macro continued in background mode. The terminal emulator would appear as an icon at the bottom of the screen as Excel's macro started it for the purpose of accessing the database. Through DDE, Excel would control the procedure followed by the terminal emulator. As soon as the sales figures had been extracted from the database and downloaded into Excel, a dialogue box would appear to flag completion of the data exchange. John could then double click on the "OK" box as soon as he wanted to review or edit the document. During background execution of the Excel macro, John would be free to work on any other application in a different window.

In this example, John was able to retrieve the latest sales figures from the database from within his spreadsheet without knowing where they were stored or how to access them using the retrieval application (for example, Information Access).

HP NewWave allows even greater integration and automation.

In Kathy's case, the HP NewWave agent could be programmed to transfer the monthly sales summary figures from the HP DeskManager message to the Monthly Order Summary Report on the first day of the subsequent month. Kathy would be able to edit the report and distribute it without first having to initiate the data transfer. If the report did not require editing, the Agent's task could be taken one step further, including the automatic mailing or printing of the report.

In John's case, the HP NewWave agent could be programmed to transfer the latest sales figures from the database on the host computer to the Excel spreadsheet every night, overnight, so that every morning his spreadsheet would reflect the latest position against quota, and against meeting the company's financial objectives.

Users working within the HP NewWave environment could use the HP NewWave Agent facility to record each step of a process and perform a task, in conjunction with the terminal emulator or any other host based or PC application, at scheduled times. The ability to record agent tasks enables even the most unsophisticated user of HP NewWave to automate many co-operative host/workstation procedures, without the assistance of MIS/DP staff.

In short, HP's strategy is based around tightening PC/host integration with the objective of encouraging and facilitating a Co-operative computing environment (CCE) in which there is automation of transparent data interchange and exchange between different applications on different host computers and workstations, with little or no intervention by the user.

4. Supportability

Of key importance to users of PC/host applications is the need to reduce costs and increase the individual's productivity both individually and as a member of the workgroup.

Inherent in the technologies and platforms and user interfaces already discussed is an intuitive quality, which enables users to quickly and easily learn and use any products developed under those applications. Whether an application is developed by Microsoft, DEC, IBM, HP or any third party software vendor, the learning curve of the user working in the Windowing environments will be much shorter than ever before.

The implications of this are encouraging in terms of both user productivity and supportability. Users will require minimum training and on-going support of their day to day activities.

In addition, a number of companies are adopting Microsoft Windows as an interim stepping stone towards OS/2. HP's strategy of consistency of interface will go a long way towards easing the migration of users from host to host, workstation to workstation, and platform to platform.

SUMMARY

HP is committed to the development and extension of its terminal emulation strategy to meet the needs of users in a Co-operative Computing Environment (CCE). Customer organizations turning to CCE and DAA will still need to access their existing HP systems and applications as well as those of other suppliers. Terminal emulation will therefore remain as an important tool for the support of CCE, which will be facilitated through the support of multiple end user platforms, hosts and connections.

Many hybrid applications can be developed through the use of purpose built applications such as HP NewWave Mail for HP DeskManager integration, Information Access for database queries under Agent control, and Business Systems Plus (BSP) for automatic updates of workstation software. However, a terminal emulator with these capabilities provides a general purpose tool for system builders, not requiring low level programming expertise of environments like HP NewWave, and not restricted to particular host computer systems or network connection types. Complex hybrids can be developed quickly to meet the specific needs of individual users and work groups, so that the terminal emulator becomes a customizable pipe between the central host computer and the workstation.

Support of a variety of end user platforms, combined with consistency of interface across those platforms, will serve as an aid to migration from host computer to host computer, and from workstation to workstation, which supports the concept of CCE and has worthwhile implications for supportability and cost reduction.

Operating System/2^{tm}, OS/2 and Presentation Manager^{tm} are trademarks of Microsoft Corporation and International Business Machines Corporation.

Microsoft^{R}, MS-DOS^{R} and Windows^{R} are U.S. registered trademarks of Microsoft Corporation.

The X Window System™ is a trademark of Massachusetts Institute of Technology.

UNIX® is a registered trademark of AT&T in the U.S. and other countries.

Macintosh is a registered trademark of Apple Computer, Inc.

PC Software Distribution in a Network Environment

Jon Bostrom
Hewlett-Packard
8010 Foothills Blvd.
Roseville, CA 95678

Managing application software on Personal Computers is one of the biggest challenges facing system administrators today. Most administrators are used to working with a central computer system that contains all of their application software. This makes installing new software, updating existing software and maintaining version control relatively easy. Personal Computers have added a level of complication by creating an environment in which tools that are being used to run the business cannot be controlled. PC users like the power and the flexibility that PCs have to offer, however they usually do not have the expertise or initiative to manage their PC as a business asset. The result is an administrator with the time consuming task of walking from PC to PC and installing software each time a new application is acquired or an existing application requires updating.

Business System Plus (BSP) addresses this problem by automatically installing and updating the BSP PC applications directly from the HP 3000. Both the administrator and the PC user benefit from this approach because it builds a more stable environment, provides the user with the correct software when it is needed, and allows the administrator to more effectively utilize their time.

The PC applications that are currently distributed by BSP focus on electronic mail, PC and HP 3000 database access, simple word processing and spreadsheet functions. If your business requires additional functionality you may decide to acquire additional PC software. This paper will describe the techniques that will allow you to design your own software distribution process.

Requirements

Software Distribution requires that Business System Plus be installed on the HP 3000 and PCs on your Local Area Network.

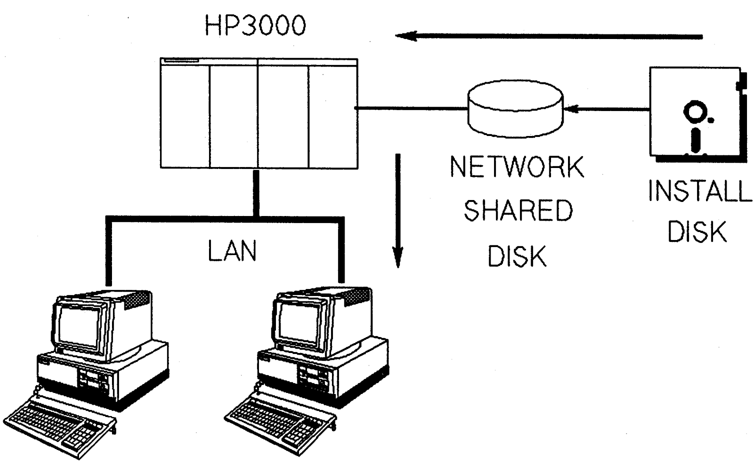
Technical overview

BSP allows the customer to create shared discs on the HP 3000. These discs are viewed by MS-DOS as additional disc drives on the PC. DOS files can be copied to or from these discs by anyone on the network just as if they were hard or flexible discs. Installation of PC software is normally done using the flexible discs provided by the software manufacturer. Some type of installation process is usually provided with the software to facilitate copying the files from the flexible discs to the users hard disc. The Software Distribution concept uses the shared disc of the HP 3000 instead of the source disc provided by the manufacturer as the installation vehicle.

Basic Process Flow

1) The system administrator loads the PC application files from the original flexible discs to the HP 3000 shared disc.

2) A batch file is started on the PC that makes a connection to the correct shared disc and installs the software.



The Distribution Process

This paper will give you some basic information about application distribution. It will describe several techniques that will help you distribute your applications and point out the potential problems along the way.

There are two basic types of distribution processes. One uses the installation process supplied by the manufacturer. The other uses standard MS-DOS file copying commands. Each type has pros and cons involved in setting up a solid distribution process.

Creating a very simple distribution process with a friendly application is quite easy. Creating a distribution process that distributes multiple applications with version control, configuration information and PAM installation for friendly applications is tougher but still reasonable . To accomplish the same task with unfriendly applications is still more difficult. Creating a distribution process with an ugly application is extremely difficult and beyond the scope of this paper.

Applications defined

Friendly Application

A friendly application is one that will install neatly into a set of subdirectories, doesn't modify system files like CONFIG.SYS or AUTOEXEC.BAT, doesn't have hidden files or copy protection, and doesn't require specific configuration information. There are a number of these around and it is quite simple to design a distribution process for them.

Unfriendly Application

There are many symptoms of unfriendly applications. Here are a few: The manufacturer's install process will not work from a shared disc. The application requires modification of CONFIG.SYS or AUTOEXEC.BAT. The application uses hidden files. (hidden files are harder to copy). The application requires specific configuration information about a user's PC, e.g., monitor type, system type, printer type etc. Unfriendly applications can be distributed but they require additional thought and preparation to be successfully implemented.

Ugly Application

Copy protection is the most common cause of an ugly application. There are a number of installation processes that do strange things to your PC hard disc or to the installation source disc, this could prohibit a successful distribution process. If you have this type of application you might want to contact the vendor about a right to copy version with site licensing that removes the copy protection.

Right To Copy

********* WARNING ********

The intent of this paper is to explain useful techniques not to encourage unauthorized software use. Hewlett-Packard does not and can not authorize you to copy any material without the specific permission of the manufacturer.

********* WARNING *********

Choosing a Distribution Process

There are several types of distribution processes:

Type 1 uses the manufacturer's installation process.

Type 2 and 3 use the MS-DOS copy function.

All processes use HP 3000 shared disc and some additional MS-DOS batch files.

Type 1

This type follows the installation process of the manufacturer but uses shared disc on the HP 3000 instead of flexible discs to install the application. To implement this type of distribution process the administrator creates a shared disc on the HP 3000 and copies the manufacturers install disc "as-is" to the shared disc. To install, the PC user makes a connection to the shared disc and follows the manufacturer's installation installation and configuration just as if they were installing from a flexible disc.

This type of distribution provides a clean, standard installation. The possible problems with type 1 are.

The administrator loses some control during the install process because the manufacturer's install process takes over.

If you are distributing several applications, the "look and feel" of each installation process will be different.

The manufacturer's install process may not work from shared disc.

The user must perform whatever configuration needs to be done and must reply to the install process to such questions as.
"What is you monitor type?"
"What kind of printer do you have?"
"Should I modify your CONFIG.SYS?"

Type 2

The type 2 and type 3 processes differ from type 1 in that they fully install and configure the application on 1 PC and then use the file sharing capability of BSP to copy the "fully installed" application to the PC user's hard disc.

To implement type 2, the administrator goes through the normal manufacturer's installation process but defines the destination of the new software as the HP 3000 shared disc instead of their hard disc. Once the application is fully installed on the shared disc, the administrator will follow the manufacturer supplied instructions to configure the application for basic printer and monitor information (i.e., laserjet on LPT1 and EGA monitor). There may be several types of basic configurations that need to be supported. In this case the administrator can install the application again to a second shared disc and create a second configuration.

Once the configuration is complete the administrator must create a DOS batch file that will perform the actual distribution. This batch file will build the directory structure on the user's PC and use the MS-DOS copy function to move the files from shared disc to the users hard disc. Once the distribution batch file is ready, any PC user on the network can start it and receive the applications already configured and ready to run.

Type 3

This type of process is very similar to type 2 but provides for the case in which the manufacturer install process will not install to a shared disc. In this case the administrator follows the install directions of the manufacturer and installs and configures the software on their PC hard disc.

When the application is fully installed and configured on the hard disc the administrator will create an HP 3000 shared disc, create a subdirectory structure on the shared disc that is identical to that created by the manufacturer's install process on the hard disc, and then copy the files from hard disc to shared disc. If the administrator needs to support several basic configurations they can repeat the installation process for each basic configuration.

Once the application is copied to the HP 3000 shared disc the distribution process is the same as type 2.

This type of distribution is slightly more work to set up but it allows the administrator more control in the distribution process. It also provides a common look to the process no matter what applications are installed. The PC user is generally freed from performing the basic configuration and having to interact with a manufacturer's install process. The drawbacks to the type 2 and 3 process are:

Less flexibility in the configuration process.

Lack of a "pretty install front end".

Any changes that would be made to system files by the normal install process, must instead be made by the distribution batch file.

Example of a Type 1 process

The steps that follow will create a very simple distribution process. Following this discussion, will be several ways to further automate and enhance the process.

Type 1 Setup

Scenario You have just purchased "Super-Buck" from the Boy-Oh-Boy software company. This application comes on two 5-1/4" discs. The discs look like this:

```
Disc 1      Root\
                \SB.EXE
                \SB.OVL
                \SB.ASM
                \SB..n
                \INSTALL.BAT
                \SBDISC.ONE
                \DATA-DIR\
                         \SBDATA.ONE
                         \SBDATA.TWO
                         \SBDATA..n

Disc 2      Root\
                \SB.KKL
                \SB.DDM
                \SB.QRS
                \SB..n
                \SBDISC.TWO
                \ADDON-DIR\
                         \ADD.ONE
                         \ADD.TWO
                         \ADD..n
```

Note the files on each disc named SBDISC.ONE and SBDISC.TWO. These files are used by the application to insure that the correct discs are present during installation. This procedure lends itself well to Software Distribution. Not all processes will be quite as friendly. You may find that type 1 will not work with a particular application, If so, try type 2 or 3.

The supplied installation instructions for "Super-Buck" read as follows:

1) Place the supplied disc in drive A:
2) Change to the A: drive with the dos command A:
3) At the A:> prompt type INSTALL ?: where ?= the letter of the drive you wish to have the application installed on.

Steps to Create a Distribution Process

1) Create an account on the HP 3000 called DISTRIB user= MGR

2) Create a group called SUPBUCK

3) Using RESMGR, the HP 3000 Resource Sharing admin utility, create a shared disc in the PUB group of the new account and share it with the shortname DISTRIB (If you are not familiar with RESMGR please consult your Business System Plus Admin Guide P/N 32510-90003). Create another shared disc in the SUPBUCK group and share it with the shortname SUPBUCK.

4) Make sure you have the server configured on your PC and then load the LAN software on your PC (USRLOAD).

5) In the directory where your LAN software resides, type

```
USE X: \\(your server name)\DISTRIB
USE Y: \\(your server name)\SUPBUCK
```

NOTE: Make sure your CONFIG.SYS has the line LASTDRIVE=Z

This will connect your X: drive to the shared disc DISTRIB and your Y: drive to the shared disc SUPBUCK.

6) Place the manufacturer's disc in drive A:

7) Look at the directory structure on the supplied disc with a utility such as the MS-DOS supplied utility TREE.COM so you can duplicate it on the shared disc.

8) Point to the SUPBUCK shared disc	Y:
9) Copy the files from the flexible disc	COPY A:*.*
10) Create the subdirectory structure	MD \DATA-DIR
11) Change to the subdirectory	CD \DATA-DIR
12) Copy the files from the flexible disc	COPY A:\DATA-DIR*.*

13) Repeat the process for DISC 2.

NOTE: If you have version 3.2 or greater of MS-DOS steps, 6/11 can be replaced by the following

6) Point to the SUPBUCK shared disc Y:

7) Copy the directory structure and the files XCOPY A: Y: /S /E

8) Repeat the process for DISC 2.

NOTE: This type of process will not work if the same file name exists in the same directory on several discs. If you run into this situation you might try a type 2 or type 3 process.

PC User Steps to Install an Application

Your application is now ready for distribution. The PC user must perform the following steps (the user must have the LAN loaded and have this server configured).

1) Connect to shared disc USE A: \\(your server name)\SUPBUCK

NOTE: I have used the A: drive to point to shared disc so that the install process matches the manufacture supplied documentation. It is acceptable to redefine a flexible disc designator as a LAN shared disc.

2) Point to shared disc A:

3) Start manufacturer's install INSTALL C:

The manufacturer's install process will now install the application.

4) Delete connection to shared disc USE A: /D

Automating the PC User Task

These user steps could be automated by creating a batch file for your users called C:\startdis.bat. This batch file would contain the same commands that the user would type. You could then create a PAM label called Software Distribution. The PAM label would point to C:\STARTDIS.BAT.

C:\STARTDIS.BAT

```
ECHO OFF
CLS
ECHO **** Distribution Process Starting ***
CD \(your LAN directory)
USE A: \\(your server name)\SUPBUCK
A:
CD \
COMMAND /C A:\INSTALL C:
CD \(your LAN directory)
USE A: /D
CLS
ECHO ***** Distribution Process Complete ***
```

NOTE: In line 5 of the batch file the A: drive is assigned to the shared disc SUPBUCK. The A: designator is used in case the install process is hard coded to the A: drive.

NOTE: In line 7 of the batch file the COMMAND instruction with the /C parameter allows you to use a second process as a subroutine. When the second process terminates, it returns control to the first batch file. (in MS-DOS 3.3 you can use the CALL instruction).

Enhancing The Process (Download Control)

We need to be able to insure that an application is loaded only once to any PC. To accomplish this we will create a new subdirectory on each users PC called \HISTORY. This subdirectory will start off with one file in it called HISTORY.BSP. HISTORY.BSP will contain an ASCII string with a brief explanation of the purpose of the subdirectory.

NOTE: Remember your X: drive is currently pointing to the shared disc DISTRIB

Here is an example of the master history file:

X:\HISTORY.BSP

```
*-----------------------------------------------------------------*
*                   BSP DISTRIBUTION HISTORY FILE                 *
*                                                                 *
* The HISTORY subdirectory on your hard disc is user by the       *
* System Administrator to control the distribution of new         *
* software to your PC.                                            *
*                                                                 *
* !!!!  DO NOT MAKE ANY CHANGES TO FILES IN THIS DIRECTORY !!!!   *
*                                                                 *
*                  Call xxxx for questions                        *
*-----------------------------------------------------------------*
```

The install process will check to see if the X:\HISTORY.BSP file exists. If the file does exist the HISTORY subdirectory is OK and the installation can continue. Each time the BSP Administrator creates a new distribution process a new file will be added to the HISTORY subdirectory to uniquely identify the new application. A date code can be used to name this file, it will be placed in the history subdirectory by the install process.

Here is an example of the application history file:

X:\11-20-88.BSP

```
*-----------------------------------------------------------------*
*                     BSP Distribution file                       *
*                         11-20-88.BSP                            *
*  This file was created by the install process that distributed *
*  SuperBuck(R) version X.01.02.                                  *
*                                                                 *
* This distribution process was initiated on November 20th 1988  *
* by the BSP Administrator John Doe.                              *
*                                                                 *
*  Please contact the MIS dept at xxxx for any questions.         *
*-----------------------------------------------------------------*
```

The distribution batch file can check for the existence of this file with the command:

```
IF EXIST C:\HISTORY\11-20-88.BSP GOTO EXIT
```

If the file is present on the user's hard disc, the batch file will terminate. If the history file is not present on the disc the batch file will continue with the installation and at the end of the installation will copy the file 11-20-88.BSP to the C:\HISTORY subdirectory.

Take a look at the batch file created in the previous example with the addition of this control logic.

NOTE: changed or new lines are marked with an asterisk "*"

C:\STARTDIS.BAT

```
 ECHO OFF
 CLS
 ECHO **** Distribution Process Starting ***
 CD \(your LAN directory)
*USE X: \\(your server name)\DISTRIB
 CD \
*COMMAND /C X:\DISTRIB.BAT
 CD \(your LAN directory)
*USE X: /D
 CLS
 ECHO ***** Distribution Process Complete ***
```

NOTE: In line 7 instead of starting the install process directly, a batch file is started from the DISTRIB shared disc. This is done to allow more control in supporting multiple applications and installation verification. Here is that file:

X:\DISTRIB.BAT

```
 ECHO OFF
 C:
 CD \
*IF EXIST C:\HISTORY\11-20-88.BSP GOTO EXIT
*CD \(your LAN directory)
*USE A: \\(your server name)\SUPBUCK
 A:
 COMMAND /C INSTALL C:
 C:
*CD \HISTORY
*COPY X:\11-20-88.BSP
*CD \(your LAN directory)
*USE A: /D
*:EXIT
```

NOTE: The USE for the A: drive has been moved into this file so that the STARTDIS.BAT file on the user's PC can be as generic as possible. This will make it easier to support multiple applications.

If the user's PC does not have a HISTORY subdirectory, add the logic to the batch file to make sure that the user has the HISTORY subdir in place.

X:\DISTRIB.BAT

```
 ECHO OFF
*C:
*CD \
*IF EXIST C:\HISTORY\HISTORY.BSP GOTO START
*MD HISTORY
*CD \HISTORY
*COPY X:\HISTORY.BSP
*ATTRIB HISTORY.BSP +R
*:START
 C:
 CD \
 IF EXIST C:\HISTORY\11-20-88.BSP GOTO EXIT
 CD \(your LAN directory)
 USE A: \\(your server name)\SUPBUCK
 A:
 COMMAND /C INSTALL C:
 C:
 CD \HISTORY
 COPY X:\11-20-88.BSP
 CD \(your LAN directory)
 USE A: /D
 :EXIT
```

Notice that the batch file checks for the existence of the history file. If the file does not exist, it will be copied from the shared disc and then set to read only so that the user can not accidentally erase it. Once the HISTORY file is on the hard disc the install can continue.

Automate the Process

The next enhancement is the full automation of the process so that the user does not need to start a batch file or press a PAM button to start the process.

As part of the PC start-up process, a file called C:\AUTOEXEC.BAT is run by MS-DOS. This file can contain any valid MS-DOS batch commands. The Administrator can add commands to the autoexec.bat file so that each time a user starts their PC it will check for new software to download. If you add this to a user's PC it will add about 15 seconds to their start-up time if there are no applications to download. If there are applications to download, the time will depend on the size of the application.

C:\AUTOEXEC.BAT

```
  ECHO OFF
  PATH C:\;C:\UTIL;C:\DOS;C:\WP
  PROMPT $p$g
  CD \(your LAN directory)
  USRLOAD /A
  CD \
 *\STARTDIS.BAT
```

Creating a PAM Label

To add a PAM label to a downloaded application use the program that is supplied with PAM. MNGEPAM.EXE can be called from the DISTRIB.BAT file to create a PAM label. The following lines are an example of the process:

```
  C:
  CD \
  MNGEPAM ADD SUPER\BUCK C:\SUPBUCK SBUCK.EXE
```

The syntax for MNGEPAM.EXE is

MNGPAM "action" "label word 1"\"label word 2" "path" "run command"

Adding Another Application (type 2)

This next section will discuss the changes necessary to distribute more than 1 application.

With one application set up for distribution your HP 3000 will have the following shared discs

DISTRIB	used to store distribution control information
SUPBUCK	used to sore the install files for Super Buck

Add a type 2 application called GWhiz to the distribution process.

GWHIZ comes on two flexible discs and has the following installation instructions:

Place disc 1 in the A: drive, change to the A: drive and type SETUP followed by the drive designator and path to install in ie. SETUP C:\GWHIZ

GWHIZ SETUP will create the directory if it does not exist and will also create a subdirectory called GDATA. GWHIZ will then install into that disc and prompt you for the second disc.

Steps to Modify the Distribution Process for a Second Application

1) Create a group called GWHIZ in the account DISTRIB.

2) Using RESMGR create a shared disc in the GWHIZ group of the DISTRIB account and share it with the shortname GWHIZ (If you are not familiar with RESMGR consult your BSP Admin Guide).

3) Make sure you have the server configured on your PC and then load the LAN software on your PC (USRLOAD).

4) In the directory where your LAN software resides, type:

```
USE X: \\(your server name)\DISTRIB
USE Y: \\(your server name)\GWHIZ
```

This will connect your X: drive to the shared disc DISTRIB and your Y: drive to the shared disc GWHIZ.

5) Install the Desired application to your Y: disc by following the instructions provided by the manufacturer:

SETUP Y:\GWHIZ

6) Note the directory structure in which the application was installed on the Y: drive

```
\GWHIZ\
      \GFILE1
      \GFILE2
      \GFILE..N
      \GDATA\
            \GDFILE1
            \GDFILE2
            \GDFILE...N
```

NOTE:If the install process made changes to any other files on your disc such as CONFIG.SYS or AUTOEXEC.BAT, update those files on your user's PCs. Make backup copies of both of these files before installing the application and compare them after installation to verify any changes.

7) Create a history file for this application similar to the one created for Super-Buck but with a different date code.

Here is an example of the application history file

X:\11-21-88.BSP

```
*---------------------------------------------------------------*
*                  BSP Distribution file                         *
*                      11-21-88.BSP                              *
*  This file was created by the install process that distributed *
*  GWHIZ(R) version B.99.99                                      *
*                                                                *
* This distribution process was initiated on November 21th 1988  *
* by the BSP Administrator John Doe.                             *
*                                                                *
*  Please contact the MIS dept at 4430 for any questions.        *
*---------------------------------------------------------------*
```

8) Next modify the X:\DISTRIB.BAT file to support this second application.

X:\DISTRIB.BAT

```
 ECHO OFF
 C:
 CD \
 IF EXIST C:\HISTORY\HISTORY.BSP GOTO START
 MD HISTORY
 CD \HISTORY
 COPY X:\HISTORY.BSP
 ATTRIB HISTORY.BSP +R
 :START
*:APPLIC_1
 C:
 CD \
*IF EXIST C:\HISTORY\11-20-88.BSP GOTO APPLIC_2
 CD \(your LAN directory)
 USE A: \\(your server name)\SUPBUCK
 A:
 COMMAND /C INSTALL C:
 C:
 CD \
 MNGEPAM ADD SUPER\BUCK MTD C:\SUPBUCK SBUCK.EXE
 CD \HISTORY
 COPY X:\11-20-88.BSP
 CD \(your LAN directory)
 USE A: /D
*:APPLIC_2
*IF EXIST C:\HISTORY\11-21-88.BSP GOTO EXIT
*C:
*CD \(your LAN directory)
*USE A: \\(your server name)\GWHIZ
*CD \
*MD GWHIZ
*CD \GWHIZ
*COPY A:\GWHIZ\*.*
*MD GDATA
*CD GDATA
*COPY A:\GWHIZ\GDATA\*.*
*CD \
*MNGEPAM ADD GEE\WHIZ MTD C:\GWHIZ GWHIZ.EXE
*CD \HISTORY
*COPY X:\11-21-88.BSP
*CD \(your LAN directory)
*USE A: /D
 :EXIT
```

The changes made to the DISTRIB.BAT file will connect to the GWHIZ shared disc, create the proper directory structure on the user's PC and then copy the file into the proper directories. The major difference between the first and second application in the X:\DISTRIB.BAT file are that the first application uses the manufacturer's INSTALL process to get the files from shared disc to the user's PC. The second application just makes directories and copies files using MS-DOS commands.

Adding a Type 3 Application

To add a type 3 application, follow the same steps as for type 2 but install to your PC hard disc, look at the directory structure that was created, build the identical structure on a shared disc and copy the files there. Follow the rest of the type 2 process.

Batch files:
X:\DISTRIB.BAT | copies the application files to hard disc.
C:\STARTDIS.BAT | starts the distribution process.
C:\AUTOEXEC.BAT | automatically starts the at boot time.

History files
X:\HISTORY.BSP | used to check on existence of history subdir.
1 file for each application to be distributed.
11-20-88.BSP | contains application name and version number.

IMPORTANT !!!!! Once you have created your distribution process you should TEST IT on a PC similar to those of your PC user's To insure that the process works well. If you are loading several different configurations you should make sure that each works when downloaded.

Installation Verification Utility.

When you create a distribution process you need to know all of the files that the installation process added or modified. For example: Was CONFIG.SYS updated? Was a line added to AUTOEXEC.BAT? Were any hidden or system files added or modified? Where were all of the application files placed? To help you with these questions a verification utility has been created. This utility can be run just before a hard disc installation and run again just after the installation is complete. The utility will tell you just what files were added to the PC hardisc and if any files were updated. The utility consists of several FREEWARE utilities and some MSDOS batch files. Hewlett-Packard does not support this utility and is not responsible for its use or the results of its use in any way. This utility cannot be sold but can be distributed free of charge.

To receive a copy of the utility and the supporting documentation, contact your Application Engineer. The A.E. can receive a copy by sending an HPDESK message to Software Manager-OSD/hpd500/51.

A well thought out software distribution process can provide substantial benifits to your business by freeing both PC users and administrators from the tedious task of manually managing PC software. When combined with the other benifits of a Business System Plus solution such as printer sharing, file sharing, electronic mail and database access, the result is an organization that can concentrate on it's business objectives.

The distribution of applications as described in this document uses standard features of MS-DOS(R) and is not Warranted by Hewlett-Packard in any manner.

Integrating PCs, MACs and the HP 3000 with New Wave

Authors: Brent M. Ehrman, Senior Consultant
Crowe, Chizek and Company
2100 Market Tower
Indianapolis, IN 46204-2976

Paul S. Thomas, Associate
Crowe, Chizek and Company
2100 Market Tower
Indianapolis, IN 46204-2976

What is New Wave? Why has there been so much discussion about this product? What does it do? How is it going to affect my business? Will it have a major affect on the development strategies of our data processing department in the future?

These are often the questions we hear when the topic of New Wave is being discussed. There seems to be confusion as to what the product is and how to use it. The general consensus is that New Wave is a unique product which has a promising future, and that it is a product which will have a significant affect on the future trends within Information Systems.

In order to develop an understanding and an appreciation for Hewlett Packard's New Wave product, we believe it is important to take a step back and take a look at the trends within Information Systems in the 60's, 70's, 80's and into the future. Over time, there have been significant changes in a number of areas within the data processing environment. The most obvious changes have been the reduction in physical size of computers, the increase in processing speed of CPUs, as well as an ever increasing ability to communicate with each other. In addition to these, however, there have also been significant changes in end-user expectations due to a greatly increased understanding of the computer's power. The following brief summaries of the past couple of decades will help us see how these changes have evolved and where they are leading.

The 60's

Data processing in the 1960's consisted of large centralized mainframe computers. Data was processed in batch routines in which response time was measured in hours and days. These machines were large processing islands which had virtually no system connectivity or communication capabilities. They were typically locked away behind closed doors with strict environmental controls. Therefore, users had no interaction with the system. Users received weekly or monthly reports produced by these batch systems, but they had no hands on interaction or real understanding of how the computer worked. Without this hands on experience, it made it difficult for the users to understand how to utilize the power of the computer to help them do their jobs.

The 70's

On-line systems were introduced in the 1970's. Computers began to get more powerful; but, the typical on-line response time was still in the 5 to 10 second range. Communication between machines, especially machines made by different vendors, was practically non-existent. Computers were still kept in separate environmentally controlled rooms in which users were not allowed. However, the introduction of minicomputers brought the computer out into the user environment, which allowed users to become more familiar with the operations of the computer. Although users were still unsophisticated in their computer knowledge at this point, they were increasing their comfort level and beginning to recognize the potential of the computer.

The 80's

On-line and real time systems with 1 to 5 second response times are very common place. Computers have continued to get more and more powerful while shrinking in physical size. Most companies own hardware and software from a number of different vendors. Typically the different vendor's hardware systems operate separately from each other. There is some communication between separate vendor's hardware, but it is predominately batch in nature. Significant improvements have been made towards the development of easier to use software. On-line applications utilize menu-driven systems with on-line help capabilities. Numerous organizations and committees have been formed to develop standards for every aspect of data processing ranging from hardware communications to user interface consistency for software applications.

The introduction of the microcomputer has made a major impact on the business world. The enormous amounts of microcomputer software which has flooded the market has stimulated users' minds in regards to the types of tasks which can be performed on a computer. Users are becoming more and more sophisticated in their computer knowledge, often times developing their own software to perform required tasks. They are expecting more information on a more timely basis. They also have become a major influence on the direction in which automated systems are headed.

The Future

By following the trends of the past three decades, it appears that soon there will be on-line and real time systems with virtual real time response capabilities. There will still be uses for mainframes, minicomputers, and microcomputers; however, portable/laptops will become a major factor in the computer environment. They will become small enough and light enough that they will become an invaluable tool upon which we will perform innumerable tasks. We will continue to have multi-vendor hardware and software environments; but, emerging industry standards for operating systems, software, and communications will enable these systems to operate with full data integration. Companies will not only be concerned with sharing information between internal computer systems, but there will be increased communications beyond company boundaries. Connectivity and integration of systems will become a key consideration and a requirement when purchasing hardware and software; therefore, vendors will practically be forced to adhere to industry standards.

The biggest change however, may be in regards to the users' expectations. Users will be demanding higher levels of functionality, integration and consistency in their systems. They will not want to re-learn new software packages each time they want to perform a new task. More emphasis will be placed on the task to be performed and less on the specific details and intricacies associated with each piece of hardware or software required to perform the task. For example, it is not uncommon for a user today to use three or four different software packages (sometimes on different hardware platforms) in order to accomplish a specific task. One package may require the user to press Function Key 8 to exit from the screen, another may use the Escape Key to exit, while another may use the Tab Key. The user should not have to learn and remember these types of minor intricacies in order to perform a task, nor should they even have to concern themselves with which software package or which hardware platform they are using. In the future, the user's primary concern will be task oriented. The computer user of the future will not need to become a para-professional in data processing, as is practically required in today's environment, to be able to pass data between separate software packages and hardware platforms. The development of seamless and consistent interfaces will allow users to perform complex integration activities with minimal effort.

NEW WAVE'S STRENGTHS

Upon review of the trends over the past couple of decades, it is obvious that there will continue to be significant improvements in computer hardware which will allow computers to become smaller yet increasingly more powerful. There is also an indication that there will be even more drastic changes in software, and this is where New Wave will have a major effect on the market place. The trends of the past point towards systems which will place emphasis on the following characteristics:

- Ease of Use
- Consistent User Interface
- Information Sharing
- Highly Functional
- Seamless Integration

Let's take a look at each one of these characteristics and evaluate how New Wave applications will provide each feature and the affect it will have on the end-user.

Ease of Use

In today's computer environment, "user friendly" and "easy to use" are two phrases which are used so often to describe software systems that they practically have no meaning anymore. There has been a major thrust in the 80's to develop "user friendly" and "easy to use" systems. What do these vague phrases actually mean?

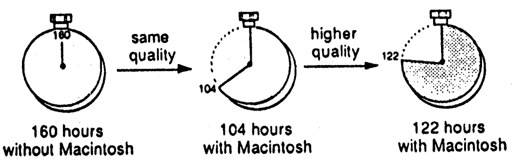
"Easy to use" systems in the 80's basically consists of the following features:

- Menu Driven,
- On-line Help,
- Function Key Utilization,
- Pop-up Search Windows, and
- Polite/Descriptive Error Messages.

Software developers have greatly improved the "ease of use" of applications over the past decade by utilizing these types of features. However, the users of the future are going to demand even easier to use systems which utilize graphics, colors and windows to provide a consistent working environment.

Although it is difficult to convince everyone of this, the graphical interface operating system, most commonly associated with the Apple Macintosh computers, provides the most "user friendly", easy to use software environment. Exhibit 1 presents figures from a Macintosh Benefits Study, performed by an independent third party (Peat Marwick Main & Co, 1987), which showed significant productivity improvements due to the ease of use of the Macintosh environment. Macintosh's use of icons, folders, pull down menus, and windows allows the user to become more task oriented and less concerned about the syntax required to run a specific software application. This design ties into the real world office environment; therefore, reducing the learning time required and allowing the user to feel comfortable with the system much more quickly than in DOS or any other command driven operating system environment. The easier it is for users to learn to use new software products, the more they will branch out and use new packages, thereby greatly increasing their productivity and effectiveness.

Exhibit 1: Macintosh Benefits Study



	Time saved with Macintosh
Managers	28%
Supervising Professionals	30%
Senior Professionals	29%
Professionals	50%
Overall	31%

New Wave allows users to use this type of icon driven system on DOS computers with DOS applications, as well as, MPE applications on Hewlett Packard computers. Applications may be encapsulated (i.e. built in) into the New Wave environment at varying levels. At the lowest level, a DOS or MPE application may be attached to an icon which simply allows the user, using a mouse input device, to point to the icon and to "click-on-it" (i.e. select it). For example, just like on a Macintosh, if a user wants to run the Microsoft Excel Spreadsheet Application, he simply points to the icon which represents Excel and "clicks-on-it". It is not necessary for the user to learn or memorize a string of commands to run the application.

At a slightly higher level of encapsulation, it is possible to link applications to specific data and have them represented by an individual icon. For example, a user could create a Monthly Sales Analysis Spreadsheet using Excel which he will use with different sales figures each month. Using New Wave, it would be possible for this user to create the Monthly Sales Analysis Spreadsheet and then link it to Excel and to an icon. At the end of each month, this user would "click-on" the Sales Analysis icon, and not only would it run the Excel software, but it would also pull in the Monthly Sales Analysis Spreadsheet. This type of application and data linking allows the user to concentrate more on the task of producing the Monthly Sales Analysis Report instead of which software package to use and which sub-directory the data is stored in.

The previous two examples of encapsulation referenced Microsoft's Excel, which is a DOS application. These types of capabilities are possible with any DOS application which is Microsoft Windows compatible. However, part of the power and attraction of New Wave is that HP3000 applications and data stored on an HP3000 can be accessed in the exact same way. An icon can be created which takes the user directly into an application on an HP3000 when "clicked-on".

With this level of integration it will be easier for the user to become task oriented and less concerned with the technical details, even to the point where the user will not care whether the software is running on a DOS machine or an HP3000.

Consistent User Interface

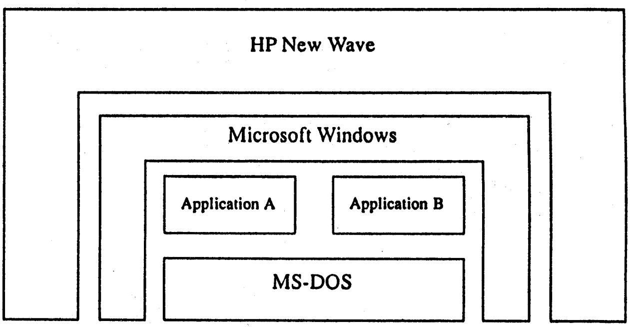
The concept of a consistent user interface is probably the most difficult concept to grasp simply because it does not exist in today's computer environment. It is hard to imagine all applications operating in a consistent manner even when they are all on the same hardware platform. It is even more difficult to imagine that there could be consistency when the software is operating on different hardware platforms. However, New Wave provides an environment in which this type of consistency can be developed.

Up to this point we have discussed a couple levels of encapsulation which makes the process of starting applications and linking those applications to specific data consistent. However, as we all know, each application has its own personality and intricacies. As we discussed earlier, one application may require the user to press the F8 function key to exit, while another may use the Escape or the Tab key.

By encapsulating programs at a deeper level, New Wave provides the capability to develop consistent up front user interfaces utilizing pull down menus. This up front interface can be used on all applications, the DOS as well as the MPE V or MPXL. For example, the user will not have to be concerned whether to press the F8, Escape, or Tab key. He will simply select "Exit" from the pull down menu, and New Wave will perform the exit.

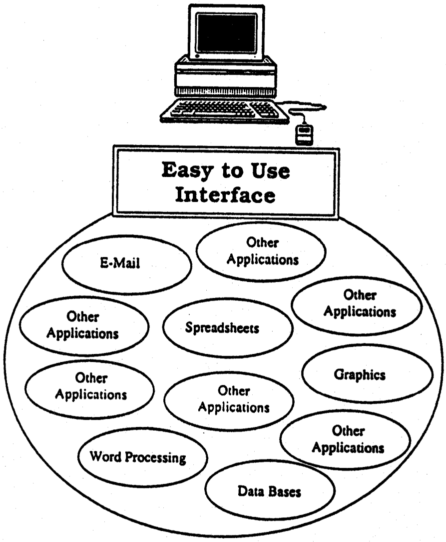
Exhibit 2 provides a simplified pictorial representation of the New Wave environment. In this diagram, we can see that Microsoft Windows envelops all of the applications as well as the MS-DOS operating system. New Wave then envelops Microsoft Windows as well as the applications and the MS-DOS operating system. Any tasks which are to be performed by the applications must, at some point, pass through the New Wave buffer. Therefore, New Wave has the ability to modify many aspects of the software at that point. New features, such as a cut-and-paste capability or user defined help screens may be added; data may be linked to another application; or any number of activities may be performed to alter the appearance of the original software application. By performing these types of activities on all of the software applications, New Wave is able to create a more consistent interface which allows the user to learn new software packages much more quickly and become more efficient at performing tasks.

Exhibit 2: The New Wave Umbrella



Another way to envision the New Wave environment is to recognize the fact that the user never has direct contact with the software applications. Every command that the user performs must pass through the New Wave buffer, and every screen or message sent to a user's workstation from a software application must also pass through the New Wave buffer. Exhibit 3 illustrates this concept. With this in mind, it becomes more clear how New Wave can have such a significant effect on developing consistent user interfaces.

Exhibit 3: The New Wave Interface Buffer



Information Sharing

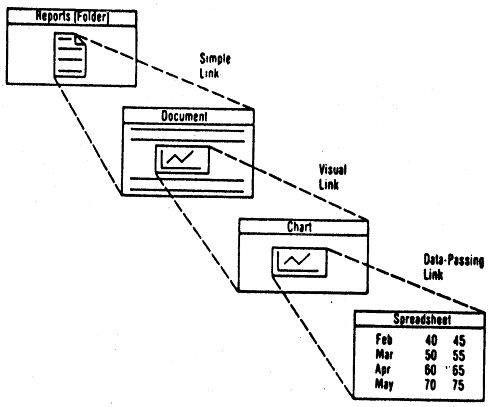
Many software applications today are able to utilize data which has been created by another application and occasionally on a different hardware platform. However, this is not always a simple process. It is often times difficult for a Data Processing professional to accomplish these tasks; therefore, it seems unfair to expect an end-user to be able to perform such a task. As a result, only the most advanced end-users even attempt to share information between applications. Most users would prefer to enter data two or three times before they would attempt to tackle the task of passing data between applications.

By utilizing "Agents" within the New Wave environment, a user can "click-on" an icon which would extract sales figures from an IMAGE database on a HP3000 and pass that data directly to an Excel Monthly Sales Analysis Spreadsheet. After the Excel spreadsheet has manipulated the data, it could then pass the data to a graphics package to create a Sales Analysis by Product Line bar chart. In this example, New Wave is not only passing data between DOS applications, but it is also using data created by an HP3000 application.

Agents, which make this possible, are system wide macros which allow you to record the steps or commands being performed. What is unique about Agents is that they also record the commands performed within the application so that macros do not need to be developed within each of the separate applications. As we saw in Exhibits 2 and 3, the fact that the applications are running under the umbrella of the New Wave environment, enables New Wave to record the commands performed in any of the applications. After the commands associated with a specific task are recorded, they may be attached to an icon. The commands associated with this Agent process are stored in a modifiable file; therefore, adjustments can be made to the Agent process without re-recording the entire process from the beginning.

In addition to Agents, "Hot Links" is another New Wave feature which will greatly increase the end-user's ability to share information between applications. When using Hot Links, the data is not duplicated as it is passed from application to application. It is stored in one central location and all of the applications which use that data access this one central location. Therefore, as the data is changed in a spreadsheet, all graphics, word processing, desktop publishing and other spreadsheet applications which are linked to that data will automatically reflect these changes. Exhibit 4 presents an illustration of how Hot Links may be defined. As the figures in the spreadsheet are changed, the chart associated with that data automatically is changed, and the copy of the chart within the desktop publishing document also reflects these changes.

Exhibit 4: Hot Links



By using a combination of Agents and Hot Links, users will be able to automate repetitive types of tasks, such as monthly sales reports, which could be taking days to produce currently. This will allow users to be much more efficient and effective at their jobs, which has been the primary goal of computer systems for years.

Highly Functional

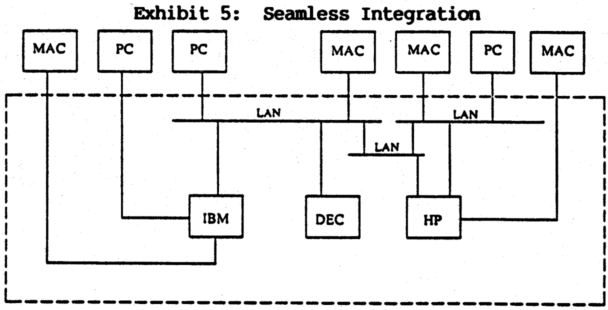
Users want to be able to use their computer to accomplish a wide variety of tasks. They want to be able to design spreadsheets, type a letter, generate charts, maintain their inventory, create overheads for a presentation, send a mail message to their co-workers, print payroll checks, etc... As far as they are concerned, they would like to be able to do all of these tasks using just one software package. Wouldn't it be great if there was just one package that was "all things to all people"? We all know that there is no such package; however, we are now able to provide the next best thing. Using New Wave, we can design and develop systems which have all of the required functionality by purchasing spreadsheet, graphics, word processing, electronic mail, database, and numerous other software applications and then integrating all of these different applications together under a common interface to make it appear as if it were one highly functional software package. Once again, Exhibit 2 illustrates this idea.

Seamless Integration

We believe that the challenge for Data Processing and Information Systems Departments in the future will be to become system integrators. They will need to use products like New Wave to utilize the strengths of existing applications both in the DOS environment, as well as MPE and others. The challenge is to create a system where the interface between applications and the process of sharing data is so seamless that the user does not know that it is happening. To the user, it should all appear as if it were one software application running on one computer, although in reality it may be numerous applications running on multiple hardware platforms.

The goal is to develop systems, like the one presented in Exhibit 5, in which a user could sit down at any workstation and access data off of an IBM, DEC, or HP system without any conscious realization that the data being used is not stored right there at the workstation.

Exhibit 5: Seamless Integration



PC, Macintosh and HP3000 Integration

In our work with New Wave, we have found the similarities between the New Wave environment and the Macintosh environment to be a definite strength. There are a number of companies who have both DOS computers and Macintosh computers. Within these office environments, it is important for users to be able to share information and to communicate with one another; however in most cases, this in not happening. The Macintosh users share some things with other Macintosh users, and the DOS users share some things with the other DOS users. The DOS users will not use the Macintosh because they are not familiar with the mouse and the concept of windows, and the Macintosh users will not use the DOS machines because they are spoiled by their mouse and windows (as you can tell, we are biased towards the mouse and windows).

To give an example of the type of things that can be accomplished using New Wave and other integration tools, we would like to describe a project in which we utilized New Wave. First, we connected an HP Vectra and an Apple Macintosh SE to an HP3000 Series 42. Using Tymlab's Business Sessions software, we were able to emulate an HP3000 terminal on both computers. For those of you who are not familiar with Business Sessions, it is an emulation package which is compatible with the Macintosh and with Microsoft's Windows on the PC. It provides windowing and mouse capabilities while emulating an HP3000 terminal (i.e. the mouse can be used to point-and-click on function keys, the window can be re-sized or moved, the user can jump in and out of terminal emulation mode into other applications, etc...).

We then used HP's Deskmanager on the HP3000 to serve as the network transportation device. Files were transported from workstation to workstation by attaching them to mail messages within Deskmanager. On the HP Vectra, we were using New Wave, Microsoft's Windows, Microsoft's Excel, Microsoft's Word, and HP's Advance Mail. On the Macintosh SE we were also using Microsoft Excel and Word as well as the recording capabilities available on the standard Macintosh operation system.

Using these standard off-the-shelf software applications, we were able to create a user interface which was consistent enough that the user could sit down at either the Macintosh or the Vectra and accomplish tasks without being concerned with which computer they were working on.

We created Excel spreadsheets and MS Word documents on the Macintosh, transferred them to the HP3000, attached them to a HP Deskmanager mail message, mailed them to the Vectra user, read that users mail through Advance Mail on the Vectra, modified the spreadsheet and document on the Vectra using Excel and MS Word, mailed them back to the Macintosh user on the HP3000, read the Macintosh user's mail and transferred the spreadsheet and document down to the Macintosh, and then printed them on a Macintosh LaserWriter Plus. All of this was done by encapsulating software applications, which already exist in the market place, into the New Wave environment.

By utilizing New Wave, we have been able to; create **easy to use** systems which have a **consistent user interface** and **share information** between applications, as well as provide a **seamless interface** between the multiple hardware platforms, thereby providing the user with a wide range of **functionality.** Some of the concepts presented within this document may have seemed rather utopian in nature, but these types of capabilities are here today. System integration between PCs, Macintoshes and HPs as well as other systems is not only what users will come to expect, it is what they will be demanding soon.

The HP NewWave Environment and MIS/DP

Bradley V. Husick
Hewlett Packard - Personal Software Division
3410 Central Expressway
Santa Clara, California
95051

The goal of this presentation is to familiarize you with HP NewWave, its unique contribution to your computing environment, and the relationship of the end user aspects of HP NewWave to the MIS/DP department and director.

First, what is HP NewWave, and why was it developed?
Today, HP NewWave is a PC software environment which works together with Microsoft Windows/286 and MS-DOS. Users add their choice of applications (MS-DOS, MS Windows, or NewWave) to make a solution. In the future, HP NewWave will be available for OS/2 and UNIX workstations. The development of HP NewWave resulted from research which identified four main business problems associated with computers:

Why Did HP Develop HP NewWave?

To Solve Four Major Business Problems:

- Computers are too hard to learn and use
- Bringing information together from different sources and formats is too difficult
- Repetitive tasks should be automated
- Investments in hardware, software, and training need to have a longer useful life

HEWLETT PACKARD

HP NewWave addresses each of these four business problems:

HP NewWave Solves Business Problems

Four Key Benefits:

- Easy to learn and easy to use
- Integration
- Task automation
- Protected investment

HEWLETT PACKARD

Let's examine each of these design goals and implementations in turn, and how MIS can benefit from these accomplishments.

1. Easy to Learn and Easy to Use

The MIS department often has time constraints in training end users. Each new application in DOS systems requires particular, customized training. The consistent and predictable graphical user interface makes HP NewWave and NewWave applications very intuitive. The fully interactive computer-based training in HP NewWave eliminates the need for classroom training. CBT lessons begin with the basics of using a windowing environment, and continue through more advanced uses of HP NewWave. The on-line help facility allows users to point to an item in HP NewWave and receive information about the item in a pop-up window. This help information is cross-referenced in a hypertext style, which allows users to see the information that they need quickly.

HP NewWave also allows users to context-switch between applications. For example, a user could have a word processor, spreadsheet, database, and terminal emulator open all at the same time, and switch from one to another by pressing ALT-TAB. This makes the one application active, putting all of the others on standby. Moving information from one place to another in this system is quick and easy.

2. Integration

Users often experience difficulty in accessing information which resides on a network. Upper management wants the easiest possible access to information without having to become computer experts. The combination of intuitive terminal emulation software and Agent technology can allow "pushbutton" access to information.

Once that information is downloaded to the PC, the task of presenting it in an understandable manner can be monumental. HP NewWave allows users to manipulate, link and combine information from multiple sources. This high level of integration allows users to create a "compound document" by simply selecting items with a mouse and dropping them into a document. For example, to include a graph or a spreadsheet in a word processing document, a user can simply pick up the graph data and drop it into the document window. HP NewWave also allows the user to edit items in the compound document by selecting them with the mouse. This single step process greatly reduces the effort required to assemble and manage different types of information. HP NewWave accommodates many types of information today. In the future, as independent developers make new types of data (like voice or video) available, these and others can all be added to HP NewWave without changing the environment itself or individual applications.

3. Task Automation

With HP NewWave Agents, a user can automate repetitive tasks such as the generation of standard monthly reports or the retrieval of information over a network without human intervention. When users perform such a task, they can set the HP NewWave Agent to record their actions. HP NewWave can be set to automatically perform any task that has been recorded at user request or scheduled by time and date.

The MIS/DP director is under constant pressure to justify the investment that the company has made in computer and information systems. Reporting information to management on a regular basis today is a time-consuming task. The complexity of these status or cost/usage reports often causes delays. The HP NewWave Agent technology can be used to overcome this problem. For example, MIS/DP personnel can compose a Agent tasks which assemble information on user, host, and job statistics, and generate graphs and reports for management. The Agent tasks are fully editable and the task language is completely documented.

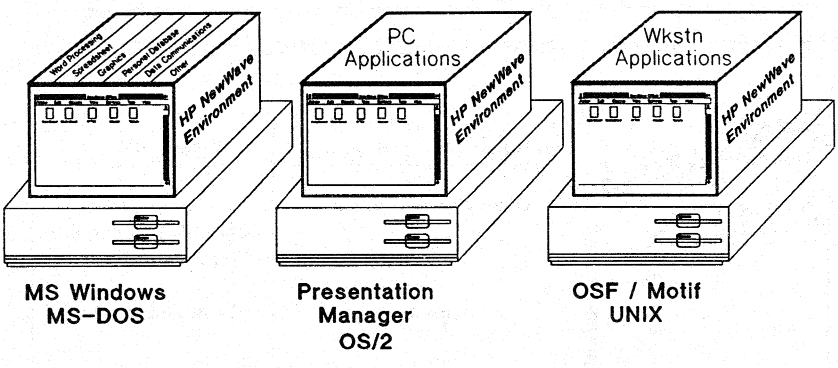
User productivity can be increased with the HP NewWave Agent. Simple repetitive tasks can be recorded and played-back by end users. More complex tasks will be the domain of the MIS/DP manager. One example of this type of task was shown in the early demonstrations of HP NewWave. The job at hand was to prepare a monthly sales report for management. The Agent task which was written to accomplish this started by launching Information Access/PC and downloading data from a sales database on an HP3000. These data were then assembled in a Lotus 1-2-3 spreadsheet on the PC. Graphs of the sales for the month were created with the data, and combined with text into a compound document report. This entire task was accomplished by the HP NewWave Agent, without the need for user intervention.

Another aspect of automation in HP NewWave is the linking of information throughout the PC. For example, a graph in one document can be included in other documents so that a change is reflected in all places automatically, saving valuable time for users. This is particularly useful for standard document or report formats.

4. Protected Investment

HP NewWave runs on industry standard hardware (80286 and 80386 PCs, expansion cards, displays, etc.) and industry standard software (MS-DOS, Microsoft Windows/286, and application software). Users can continue to use their favorite MS-DOS and Microsoft Windows software in the HP NewWave environment, saving on retraining costs and wasted time.

Under development are OS/2 and UNIX implementations of HP NewWave:



HEWLETT PACKARD

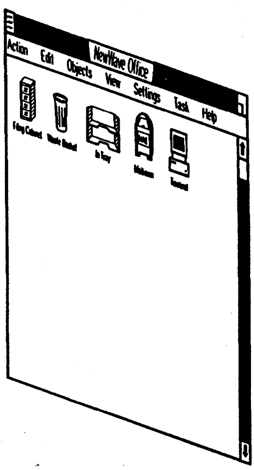
The MS-DOS / Microsoft Windows implementation of HP NewWave started with the Developer Kit which began shipping in March of 1988. The End User release of HP NewWave is scheduled to be released in the second half of 1989.

The OS/2 / Presentation Manager, and UNIX / OSF Motif implementations of HP NewWave are under development. As the products are more clearly defined, realistic schedules will be announced. Martin Johnson, of HP-OPD, is presenting a paper during this Interex conference which addresses the UNIX strategies relating to HP NewWave.

Summary

In conclusion, HP NewWave can provide real benefits to end users and MIS/DP personnel. The easy to learn and use design of HP NewWave can increase productivity and reduce training costs. Integrating information from different sources and formats is made easier, and repetitive task automation allows users to concentrate on their work, not on their computers. In addition, investments that companies have made in hardware and software are protected with HP NewWave.

HP NewWave...*more than a graphical user interface*



Behind the Interface...	Benefits
Object Management	Integration Compound Documents Information Links
Agents	Task Automation
On-Line Help	Ease of Use
Computer-Based Training	Ease of Learning

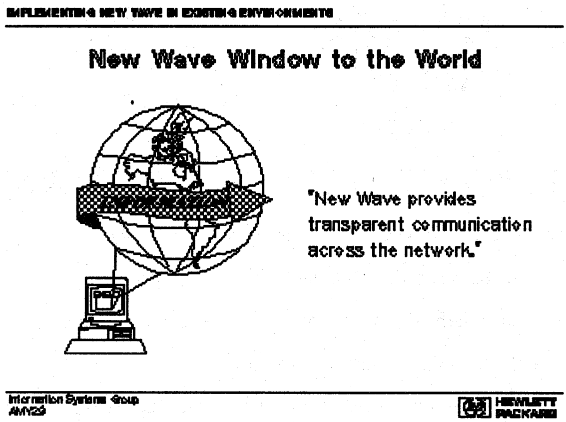
HEWLETT PACKARD

IMPLEMENTING NEWWAVE IN EXISTING HP 3000 NETWORKS

Alison McCallum-Varey

OPD Product Marketing

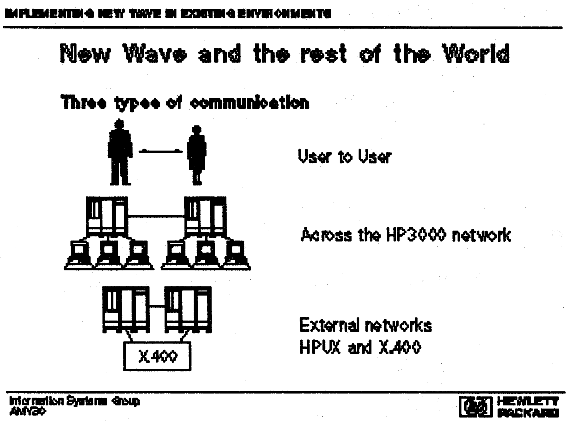
Introduction



One the goals of Hewlett Packard's NewWave environment is to give NewWave users a transparent means of communicating across the network. Making the communication transparent means that the NewWave user does not have to be concerned with the mechanics of distributing information across the network. All he or she has to do is send the information and the system takes care of its transmission across the network and its delivery to the recipient.

Ideally, all users on a system would get the benefit of transparent communications by implementing NewWave. In the real world, however, we acknowledge that existing customer investments in current wave software and hardware will mean that NewWave implementation may be more gradual. Not all users will be upgraded to NewWave at once.

Recognizing the need to accommodate mixed environments, NewWave has been developed not only to provide transparent access when communicating with other NewWave users but with the existing environment. This paper has been written to describe how transparent communications has been achieved between NewWave and existing systems. It will focus on the following three areas:-



1. Communication between NewWave and non-NewWave users.

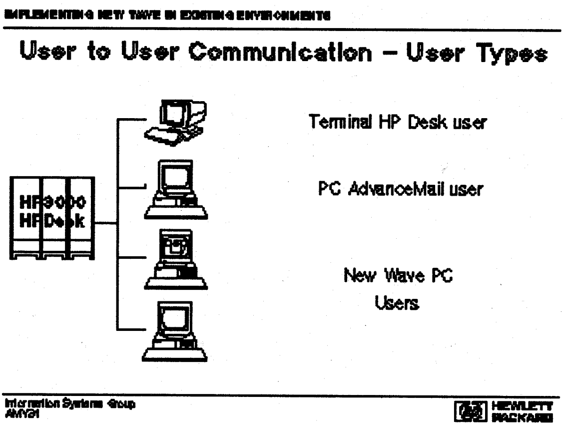
2. Communication across the HP 3000 network

3. Communication to other networks including HP-UX and X.400.

Communication Between NewWave and non-NewWave users

In looking at how transparent communication can be achieved, it is first necessary to identify who is going to be communicating with whom and what are they going to communicate. Once this is identified, it is then possible to discuss how this can be achieved transparently.

Who communicates?



In any existing HP 3000 environment there are likely to be different types of users. Our customers typically have a mixture of terminals and PCs. There are two particular types of current wave users with whom a NewWave user will communicate.

1. Terminal users who use HP DeskManager to communicate (this includes PC users who use terminal emulation to access HP DeskManager.

2. PC users who use AdvanceMail.

The current wave users will already be able to send, receive and manipulate documents created in a wide variety of formats. Key to successful implementation of NewWave into this environment is ensuring that if the NewWave user sends a document to a current wave user, he/she will be able to manipulate the received item and likewise the NewWave user can do the same to documents received from current wave users.

Looked at as a matrix here are the capabilities which every user will have.

IMPLEMENTING NEW WAVE IN EXISTING ENVIRONMENTS

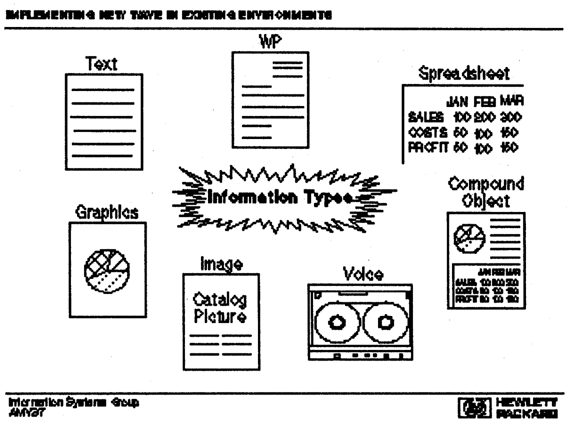
Communications between Users

Receive \ Send	New Wave	PC + AdvanceMail	Terminal + HPDesk
New Wave	✓	✓	✓
PC + AdvanceMail	✓	✓	✓
Terminal + HPDesk	✓	✓	✓

Information Systems Group
AMY32

HEWLETT PACKARD

Information types



Giving users communication capabilities means not just allowing them to send simple text messages but giving them ability to send any type of document and ensuring that the recipient can manipulate it. The types of documents which NewWave and current wave users can exchange are as follows:

Text

Text is the most common form in which information is distributed. Text is the simplest document type and contains no special formatting.

Word Processing Documents

There are very large number of word processors available, providing many formatting and DTP-like capabilities. As well as HP word processors such as Executive Memomaker and AdvanceWrite, HP will allow for the integration of third party word processors into NewWave - Word Perfect and MS-Word are two examples.

Spreadsheets

For users who need to manipulate financial or statistical data, a spreadsheet is a must. The two most popular PC based spreadsheets available are Lotus 1-2-3 and Excel.

Graphics

Graphs and illustrations are an increasingly important data form that users want to share with others, either as part of a WP document or as separate items. HP has its own graphics package Graphics Gallery which will be encapsulated in NewWave.

Image

Anyone who has a scanner can turn printed matter into electronic form and then send it around the network.

Voice

Voice annotation of messages is an exciting feature of NewWave.

Compound Objects

Many current wave applications can create the documents above without needing NewWave. However one of the key features of NewWave is that it can create compound objects from individual objects. It is possible to create "hot links" between individual objects so that if changes are made to one object, the changes are reflected in the other elements of the compound object - for example a spreadsheet could update a graphic and they could both be automatically amended inside a WP document.

This type of object is only found inside NewWave.

Packages

If an HP DeskManager user wants to group documents then the current wave option is to create a package which contains the collection of documents. No links are established between the items.

What happens when the recipient gets the information

The process of communicating information has two parts - the sending of the information and the receipt of that information. The information is useless if it cannot be manipulated by the recipient. There are three ways a recipient will want to be manipulate the message in this order of importance:

1. Read the message or parts of it.

2. Print the message or parts of it.

3. Edit the message or parts of it.

The recipient may want to do other things such as delete, reply, file or forward messages but these forms of manipulation would tend to follow on from the three major activities of reading, printing and editing.

How does this work in my existing environment?

To summarize, there are a number of conditions which have to be met to make possible the coexistence of NewWave and Current Wave users on the same system.

1. NewWave users have to be able to send any document type available in NewWave to other users on the system whether they are NewWave users, AdvanceMail users or HP DeskManager users. This includes compound documents.

2. The recipient needs to be able to manipulate the incoming information at the very least read and print it and, where the tools are available, to edit it as well.

It is important to understand that although ideally every user should be able to do whatever they choose with every document they receive, in practice there are some limitations to this which relate to the available software and hardware in any particular environment.

What HP have tried to implement is to give all non-NewWave users at least the capability of reading and/or printing documents they receive taking account of their environment. Where possible users will also be able to edit the documents.

Browsers and Converters

There are three methods of manipulating NewWave documents in HP DeskManager and AdvanceMail which enable users to gain the optimum access to them.

The first of these is what is called a Browser. It lets a user either read or print a document. It is simply a means of displaying the document contents but does not make any changes to it. A browsed document cannot not necessarily be edited. This method is highly useful in environments where the application which created the document is not available.

The second method of manipulating the document is called a Converter. This takes the document, opens it up and changes the contents to a new format whilst retaining as many of the original characteristics as possible. Given the large number of applications, all using different formats it is not always feasible to provide a converter for every combination. What Hewlett Packard have attempted to do is cover the primary converters and provided a mechanism for the introduction of new converters as they become available.

The third method is by using an available application. This is essential for editing documents. With the exception of text for which an editor exists in each of the products, many of the host and PC applications are optional extras and therefore would have to be purchased and installed separately.

The Overall Picture

The illustration belows shows the different combination of options available to users in the different environments who receive items from NewWave.

IMPLEMENTING NEW WAVE IN EXISTING ENVIRONMENTS

Sending from New Wave

Received by	Text	WP	Spread Sheet	Graphic	Image	Voice	Compound Document	Action
New Wave	✓	✓	✓	✓	✓	✓	✓	Read
	✓	✓	✓	✓	✓	✓	✓	Print
	✓	✓	✓	✓	✓	✓	✓	Edit
	Uses Application							Covert or Browse
PC + AdvanceMail	✓	✓	✓	✓	✓	X	Depends	Read
	✓	✓	✓	✓	✓	X	on	Print
	✓	✓	✓	✓	✓	X	Contents	Edit
	Browse	Browse + Convert	Uses Application	Uses Application	Uses Application	-	Done by HPDesk	Covert or Browse
Terminal + HPDesk	✓	✓	X	X	X	X	Depends	Read
	✓	✓	X	✓	✓	X	on	Print
	✓	✓	X	X	X	X	Contents	Edit
	Browse	Browse +/or Convert	-	Browse	Browse	-	Browse or Convert	Covert or Browse

Information Systems Group
AMY33

HEWLETT PACKARD

The NewWave environment has grown out of the existing environment which means it has capabilities not available in the current environment. Only when all users upgrade to NewWave will all the capabilities become available. In the meantime the aim has been to provide them with as many of their existing capabilities as possible and even added some. As NewWave develops, it may be that new capabilities will become available in the HP DeskManager and AdvanceMail environments. Both have the capability to add new browsers and new converters to make the receipt of information as transparent as possible.

IMPLEMENTING NEW WAVE IN EXISTING ENVIRONMENT

Receiving Messages in New Wave

- What is received: Any document
  Objects
  File Containers
- How they can be manipulated: Read, Print or Edit
- Converters involved on download
- Applications used in place of browsers

Information Systems Group
AMY29

HEWLETT PACKARD

Thus far, discussion has revolved around the NewWave user sending information to current wave users. However, the NewWave user will need to receive information as well as distribute it. Inside NewWave it is more straightforward. A NewWave user can receive any item from a non-NewWave user and it will be included in NewWave in one of two ways - either as an object or put into a file container for importing to an application.

Converters

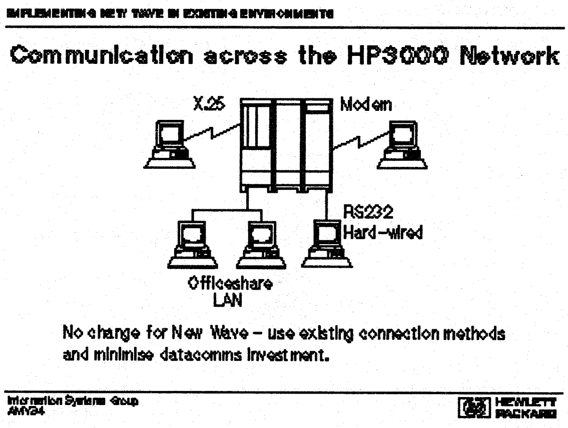
The NewWave user can specify any conversions he/she wishes to have made to the documents before they are downloaded to the PC from the host. This converter mechanism uses converters available on the HP 3000 and can be invoked for many existing applications.

Browsers

Browsers are used for reading and printing items where the application is not easily accessible. Because applications are more fully integrated into NewWave, they can be invoked more easily than in current environment and therefore NewWave objects are always manipulated using the application.

From a user's point of view this may appear to be quite a complex set of rules and cross references. However the end result is that far from it being complex to use, it is actually straightforward. That is because the user does not see any of these inner workings, all the converting and browsing is done behind the scenes - invisible or to use the words of the introduction - Transparent.

2. Communications across the HP 3000 network



In the first section, the emphasis was on how transparent user to user communication is achieved. A second aspect to this transparency is New Wave's desire to make access to the network itself transparent. That is to say, users do not have to be concerned with the whys and wherefores of data communication links. All the NewWave user needs to do is send messages and the underlying data communications methods, with help from the system manager, do the rest.

In terms of implementing NewWave in an existing HP 3000 environment, there are no special requirements for data communication links. NewWave has the same underlying data communication library as was used with AdvanceMail. This means that if you have been using AdvanceMail, then the same connection methods can be used. Those methods are as follows:

LAN Connections

Using OfficeShare plus LAN/Link 3000, users can connect via their standard LAN connection.

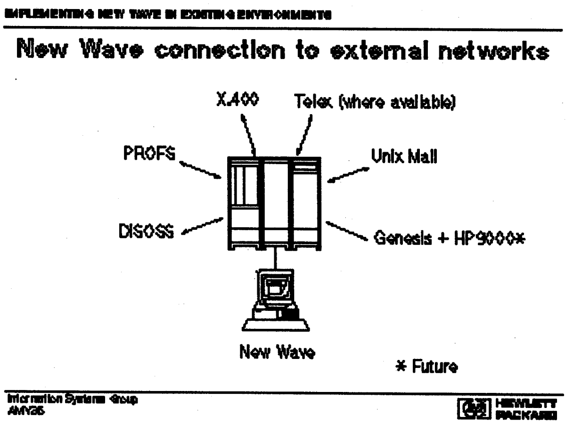
Serial Connections

For direct connections, the user simply configures their logon and passwords (both are encrypted for security).

For remote connections such as X.25 or modem, the system manager may have to write command files to negotiate the logon procedures required by these data communication methods. If these have already been written for AdvanceMail then they should work with NewWave Mail.

For those customers who have not implemented AdvanceMail but already have HP DeskManager, then they will have to set up the connections. However, the appropriate software will already be in place on the HP 3000 so there will be no extra expenditure on HP 3000 software to implement NewWave Mail.

3. Communication to external networks.



NewWave again makes use of the existing mechanisms to provide communication to external services outside of the HP 3000 network. At first release this will done via HP DeskManager's transport mechanism. The NewWave user simply sends his/her message to the recipients, HP DeskManager takes care of the routing of the messages. Depending on what is configured, HP Desk can send messages to a variety of destinations.

It also depends on what can be sent to the different systems, what kind of information can be distributed from NewWave - some can only receive text whereas others can receive binary files and therefore PC files.

IMPLEMENTING NEW WAVE IN EXISTING ENVIRONMENTS

Send from New Wave	Receiving System: Telex	PROFS	DISOSS	Unix Mail	X.400	Genesis
Text	✓	✓	✓	✓	✓	✓
WP	X	X	✓	X	*	✓
Graphics	X	X	✓	X	*	✓
Spreadsheet	X	X	✓	X	*	✓
Compound Document	<Converted and Text parts sent>					

Information Systems Group

HEWLETT PACKARD

Telex (Not available in the US)

NewWave users can send telexes out to the telex network via HP DeskManager and HP Telex. Similarly, a NewWave user can receive telexes via the same route.

Text is the only form of information that can be sent to the telex network.

PROFS and DISOSS

By means of HP OfficeConnect to PROFS and DISOSS products, NewWave users can send messages to the IBM environment. Links to PROFS are limited to text but PC binary files can be sent to DISOSS.

X.400

NewWave users can send to X.400 via HP DeskManager and OfficeConnect to X.400. In future, access to X.400 will be available in NewWave via HP's new mailing product for the HP-UX environment - project name Genesis.

X.400 is currently limited to sending text but as protocols are defined, other file types will certainly be permitted.

Unix Mail

Simple text messages can be sent to Unix Mail users from NewWave.

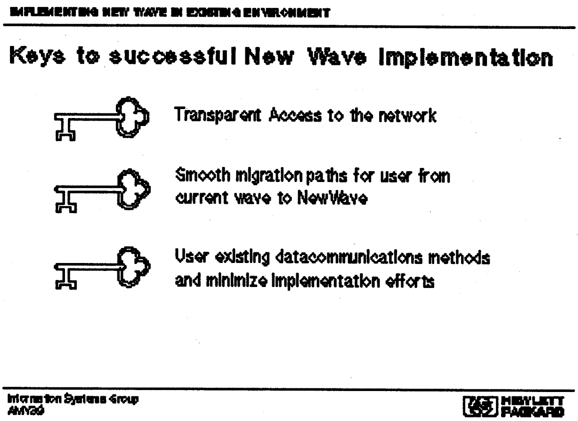
Genesis

As well as being able to connect directly to an HP-UX host running Genesis from NewWave, it will also be possible to connect via HP DeskManager to send text and PC files between the two systems.

Conversions

Any conversions of items going to external systems will be done at the gateway so the NewWave user does not have to be concerned about having documents in the right format - it will be converted or if not suitable to be sent returned to the sender.

Summary



In designing NewWave it was vital not just to provide transparent communications capabilities within the NewWave environment, but to ensure that those capabilities would extend into the wider network so that NewWave could be implemented in existing systems without customers having to upgrade all of their users to NewWave immediately.

In order to make the progression from current wave to NewWave as smooth as possible, the converter and browser mechanisms to allow exchange of information between New and current wave are available both to use. As customers move users from one environment to the other, they are not being cut off one another but can maintain same the communication capabilities as they had before plus the enhanced NewWave environment.

In already established systems, a major concern of any additions to the system is how much it is going to cost and how long it will take to implement. From the point of view of data communication links, NewWave can take advantage of what already exists and therefore where the work has already been done to connect PCs into the host, no additional effort is needed. Users can start getting the benefits of NewWave quickly and painlessly.

The combination of these three elements mean that whether you choose to upgrade all your users to NewWave at once or make it a gradual process, your existing system can adapt with the minimum time and effort.

Computer Viruses: Can They Be Prevented?

Bernard P. Zajac, Jr.
Data Security/Technical Support Manager
ABC Rail Corporation
200 South Michigan Avenue
Chicago, Illinois 60604-2402
Phone: (312) 322-4573 Telex: 756108

September 11, 1989
Brooks Hall/Civic Center
San Francisco, California

Acknowledgements

The author would like to thank: Congressman Wally Herger and Mr. Doug Riggs of his Washington staff; Mr. Ned M. Cole, Jr., Mr. Thomas J. Ryan, Mr. David L. Lathrop, Mr. Jeffrey A. Leith, Mr. Stephen J. Loyd, Mr. F. Stewart Mitchell, Ms. Mary J. Ryan, ABC Rail Corporation - Chicago, Illinois; Mr. Kirk W. Tabbey, Washtenaw County Prosecutor's Office, Ann Arbor, Michigan; Mr. Harry Hopping - Klienwort, Benson, GSI, Chicago, Illinois; Mr. Robert I. Brown, Schlussel, Lifton, Simon, Randa, Galvin & Jackier - Southfield, Michigan; Mr. James J. Ayres, Illinois Supreme Court - Chicago, Illinois; Mr. Bernard V. Boyle - Wyoming Department of Criminal Investigation - Wyoming. Especially, Mr. Stephen Garry, A. C. Nielsen - Northbrook, Illinois; Mr. Kevin P. Morison, Illinois Criminal Justice Information Authority - Chicago, Illinois and Ms. Dawn M. Holt for their critical critiques of the early drafts.

INTRODUCTION

Since the first documented computer crime case in 1958,[1] computer crime has come a long way. Today, we are dealing with what The New York Times aptly calls "the letter bomb of the computer age: computer viruses."[2] With electronic mail, viruses have become more than the letter bomb of the 80's: they have become a global threat[3]. Dr. Harold J. Highland, noted author and editor of Computers & Security, summed it up succinctly when he said: "We ain't seen nothing yet!"[4]

Viruses have infected networks and PC's. The latest entry into the virus fray is the Cornell Virus, which in a less than a day infected thousands of mainframe computers throughout the world.[5] Viruses restricted to PC's have also done so on a global scale. The Pakistani "Brain" virus, for example, infected an untold number of PC's as it traveled throughout the world wreaking havoc.[6]

---

[1]Donn B. Parker, Crime By Computer, (New York, NY: Charles Scribners & Sons), 1976, p. 34.

[2]"Letter Bomb of the Computer Age," New York Times, 5 November 1988, p. 16.

[3]Bernard P. Zajac, Jr. "Computer Viruses: The New Global Threat," The Computer Law and Security Report, May-June 1988, p. 3.

[4]Philip Elmer-DeWitt, "Invasion of the Data Snatchers!," Time, 26 September 1988, p. 64.

[5]Jim Ritter and Michael Gillis, "Computer 'bug' Bytes Colleges," Chicago Sun-Times, p. 1.

[6]Katherine M. Hafner, "Is Your Computer Secure?," Business Week, 1 August 1988, p. 67.

The History of Computer Viruses

Computer viruses are not a new phenomenon. They are just an old trick with a new twist: a computer trojan horse program that has the ability to reproduce.

Viruses are said to be the offspring of Frederick B. Cohen. Who, for his doctoral dissertation for the University of Southern California, created a virus in an effort to find a way to defend against self replicating programs. Cohen, now a professor at the University of Cincinnati, found they are next to impossible to defend against. Using two DEC VAX computers and a Univac 1108, Cohen found that a virus could spread throughout a computer in a matter of minutes![7] This was recently reaffirmed by the Cornell Virus, which infected and halted computers in a matter of minutes worldwide.[8]

Cohen's work with viruses was first made public at the 1984 National Computer Security Conference. A few years later, Rudiger Dierstein of the Deutsche Forschungs und Versuchsanstalt für Luft und Raumfahrt e. V. (DFVLR), introduced the concept of viruses to

---

[7]Bernard P. Zajac, Jr., "Computer Viruses: The New Global Threat," The Computer Law and Security Report, May-June 1988, p. 3.

[8]Michael Alexander, "Virus Ravages Thousands of System," Computer World, 7 November 1988, p. 1.

the European press in "Computer Viruses: A Secret Threat," a paper presented at the 1986 meeting of Securicom in Paris, France.

Since 1984, viruses have evolved into really two distinct types: Active and Passive. The active virus is a virus that damages or destroys systems. The Lehigh or Pakistani viruses are good examples of "active" viruses;" both destroyed systems. The passive virus, on the other hand, sits in the background and adds trapdoors or gathers information --- by far the most dangerous. The Cornell virus is a good example of a passive virus, its purpose was to gather passwords on attacked systems. However, a "bug" in the virus turned it into an active virus by mass producing itself and halting the infected systems.

How Computer Viruses Work

To study how viruses work, let's look at a simple virus: the Lehigh Virus, named after Lehigh University in Bethlehem, Pennsylvania, where it was first discovered.

This virus hid in a perfect place, the COMMAND.COM file. Perfect, because this file is executed every time something is entered at the keyboard. It is here that the DOS commands such as RENAME, DIR, and DEL are located.

According to the Lehigh University Computing Center, the virus actually hid in the stack space area of the COMMAND.COM file, which was why the COMMAND.COM's file size didn't change after it was infected. With some viruses file size is one way to detect a virus on your system -- not so with this one.

Once the virus was in place, whenever a user typed a DOS command, the virus would check to see if there was a non-infected COMMAND.COM file on the system. If so, it infected it and incremented a counter that kept track of how many other disks it had infected. The virus would then execute the user's DOS command. All this, unbeknownst to the user. From the users perspective, the computer just executed the DOS command and nothing more.

The virus continued this process of infecting computers and/or disks until the infection counter hit four; then it would totally erase the hard disk, including the boot sector and FAT (File Allocation Table) space -- an effective way of covering its tracks and rendering the computer totally useless.

Fortunately, this virus did leave a couple of telltale signs: the date on the COMMAND.COM file changed when it was infected -- you knew at least when you got hit; and the drive that contained the disk being infected would activate its write light when it was replicating itself --- you could see the virus infecting the disk.

To illustrate how simple viruses are, here is a simple PASCAL program for a virus. It does not include details for the subroutines or variable declarations, but it does demonstrate how simple virus structures are:

```
BEGIN   {Virus Logic}
    PERFORM check-if-infected;
    IF not-infected
      THEN PERFORM infect-computer;
    IF trip-condition = true
      THEN PERFORM destroy-disk;
END     {Virus Logic}

***** Normal Program Would Follow Here *****
```

Note, the first thing the virus does is that it checks to see if it needs to infect the computer. If so, it infects the computer and moves on to the next command. If not, it skips the infection process and executes the next command, which is to determine if the trip condition is true. The trip condition is when the virus should reveal itself and do what it was designed to do. This could be a number of things from transferring money to erasing hard disks. This condition could be just about anything: a date or a counter. As in the Lehigh University virus, for example, it was a specific condition --- it had to have infected four other computers. Other viruses, the Hebrew University and MAC viruses looked for a specific date.

Viruses can also be very sophisticated. The Cornell virus used a multi approach to attaching other computers: it used a known bug in the Unix Sendmail utility, it searched for passwords and had on onboard decryption routine, and had a method of finding what other computers were on the network.[9] This program was some 50,000 lines long.[10]

Some other techniques viruses use include "husking," in which the virus moves the contents of the target program to another location on disk and then inserts itself into the original file location. When the user tries to read the original file, he sees the original program because the virus points to the displaced program and displays it; not the real contents of the "husked" file. The Brain virus is a good example of this.[11]

However, the ease with which viruses can be created cannot be understated. Dr. Harold J. Highland, developed a demo virus just using the DOS batch file (BAT) commands in an AUTOEXEC.BAT file!

---

[9] Michael Alexander, "Anatomy of a Virus," Computer World, 7 November 1988, p. 157.

[10] Ibid., p. 157.

[11] Dr. Harold J. Highland, "The BRAIN Virus: Fact and Fantasy," Computers & Security, August 1988, p. 368.

Prevention

Viruses are extraordinarily simple to make. But how do we prevent them? Unfortunately, once a computer has been infected, it can be too late. The best way to prevent viruses: practice safe computing! Some examples include:

- o DON'T USE UNKNOWN SOFTWARE -- Including "shareware" software, software from bulletin board services (BBS), or bootlegged software. Use only software from a KNOWN vendor in a sealed container!

- o CENTRALIZE SOFTWARE PURCHASING OR HAVE AN "APPROVED" VENDOR LIST -- This prevents and deters the purchasing of unknown software.

- o USE A WRITE TAB ON "SUSPECT" DISKS -- This is one way to detect the Lehigh Virus and it prevents further contamination. When the Lehigh Virus tried to infect a non-infected disk with a write tab, the computer responded with a "WRITE PROTECT ERROR."

- o INSTRUCT EMPLOYEES ON THE DANGERS OF VIRUSES -- instruct them not to unload ANY unknown software to any company computer -- Viruses cannot spread if they are not given an opportunity!

- o IDEALLY, DECOMPILE NEW PROGRAMS, CHECKING FOR SUSPECT DOS CALLS BEFORE RUNNING -- If a program says it reads switches or calculates something, it better not contain DOS write calls!

- o MAKE ALL .EXE AND .COM FILES ON PC's READ ONLY -- There are several utilities on the market that can set the attribute bit to READ ONLY. This will prevent any unauthorized access.

- o WHEN USING "SWAP TAPES," RECOMPILE THE SOURCE PROGRAMS YOURSELF AFTER CHECKING THE CODE -- The source may say one thing and the object another.

- o CHANGE COMMON SYSTEM PASSWORDS! -- The recent Cornell Virus got in via known holes in Unix and it tried common passwords to get into other systems.

Currently, there are several commercial vaccination packages now available for the PC. However, they do not protect against ALL viruses: they mearly protect against some viral techniques we know of. Additionally, many of these viral filters work fine on an IBM clone, but they DO NOT work on a true blue IBM PC! It seems many of the filters were developed on clones and not the real thing.

Recently, ABC Rail Corporation participated in a study of computer virus filters for Computers & Security. As part of the testing process the filters were given directly to the end user to see: how easy they were to install; how "user friendly" they were; and how effective they were.

None of the virus filters that were reviewed were really acceptable. If they were effective, they were too hard for the user to use. Or, if they were very user friendly, they were not very effective. Which is really the old security problem: the more secure something becomes, the harder it is to use it. And the harder something is to use, the less productive it is.

When the famed Apple Virus hit, Apple released a free program called Virus Rx available from dealers that checked Macintosh computers for viruses by looking for "suspect" code in the system. There are now many anti-virus programs on Bulletin Board Services across the United States; however, extreme caution should be exercised when downloading anything from a BBS!

The only way to really protect against viruses is to not expose your system to any suspect software. This approach is fine for PC's, but what about large networks? Many companies have large Management Information Systems (MIS) departments to gather, condense and report information to top management so they can make informed business decisions. For many companies, this "information gathering" network can be global: Barclays and the Continental Bank of Illinois, for example, have international networks. Can these networks be protected against virus? Yes, just as easily as PC's!

There are many ways to secure large networks today with hardware and software, but, as in any security program, the weakest link is really one thing: people.

People are your greatest security asset and they are your greatest security liability. Educating people to the dangers of viruses will help curtail the uploading of suspect software on systems. So will enacting policies such as specifically prohibiting any foreign software from being loaded; centralizing software purchases to one department or developing an "approved vendors" list.

Detection

We have seen just how easy it is to create a virus and how to prevent viruses but how do you detect for one? Unfortunately, with the really good viruses, once you've detected it --- it's too late! However, there are a few methods available; some scientific and a some not so scientific. Some of the "not so scientific" methods include:

- o SETTING YOUR SYSTEM CLOCK TO THE FUTURE and then seeing if files disappear --- not the most elegant method, but effective.
- o LOOKING FOR "STRANGE FILES" ON YOUR SYSTEM - The "Cornell Virus," for example, had files named xNNNNNNN, where the N's appear to be random numbers.
- o NOTING STRANGE OR DIFFERENT LOGONS - When on a network, note if you see logons for communication programs at odd times. For example, a electronic mail truck delivering mail at an odd time.

Some of the more scientific methods for virus detection include:

- o HAVING MEMORY RESIDENT PROGRAM CHECK DISK CALLS - Many so called "virus filters" consist of a module that resides in memory that checks for certain types of disk calls or intrinsics.
- o CHECK FILES SIZES AGAINST A PREVIOUSLY ESTABLISHED TABLE - This will let you know if a virus has infected a file and increase its size. NOTE: This is NOT a fool proof method. Some viruses infect files and do not increase or decrease the host file's size.

- USING UTILITIES OR PROGRAMS SEARCH ALL FILES PROGRAM FILES FOR "STRANGE TEXT" -- Many viruses print messages such as "Ha Gotcha!" and the like. Using many utilities available today, you can scan entire systems looking for key phrases identifying "suspect" programs..

These are just some of the techniques that can be used to locate some viruses. However, they will not flush out all viruses -- the real creative viruses may never be found until its too late.

What If You Are A Victim?

What if it is too late and you are a victim of a computer virus attack, what do you do? This depends on how you were infected and what was infected. If your system is on a network you need to pull your system immediately off the network to prevent further damage or recontamination. Then you need to isolate the virus, see how it works, and take corrective action.

If you catch the virus early enough, you probably can use your backup tapes for a reload/restore, provided THEY HAVE NOT BEEN INFECTED. However, before the reload, the system (CPU) should be powered down to make sure nothing is staying in high (RAM) memory. A reload/restore is recommended because during a reload, most systems fully reinitialize all the disk packs.

The same technique would apply to an infected PC. However,

the "reload/restore option" becomes a little more problematic, as most PC users do not make regular backups as do most MIS departments. As a result, chances are that data will be lost. Before the reload/reload, it is suggested that a lowlevel format be performed prior to the standard format on the hard drive. This will more effectively "sanitize" the machine. Additionally, you should execute the DOS SYS command to overwrite and restore the boot sector, since some viruses reside there.
Many PC's today have on-board batteries to keep the CMOS memory alive. When powering the PC down, you should remove the batteries to also clear the CMOS memory, as there have been reports of viruses hiding in CMOS.

After the PC has been sanitized, you can reload from non-infected backups, should you have such a thing. But most often you will probably have to go back to the original software disks, including the operating system.

Legal Recourse To Viruses

Once you have your system back on its feet and operating, most probably you have spent a lot of time and money to do it. Your system has been unavailable and your users inconvenienced. What recourse do you have, legally? One software manufacture recently said, "You, as a user have a recourse, you just sue them [the software manufacture]!"[12] Interesting, but can you? The answer is "technically yes," but you will be blazing new legal ground. There is not much case law in this area.

Kirk W. Tabbey, head of the Washtenaw County (Mich.) Computer Crime Task Force, and an assistant prosecuting attorney in Ann Arbor, Michigan said, "You'll always have a criminal case if you can find the person who did it [created the virus] because a virus is a malicious act. Surreptitiously inserting a virus in a program is, in itself a malicious act, therefore a crime."[13] Currently, there is federal law which specifically deals with computer crime[14]. Representative Wally Herger (R-Calf) recently introduced the "Computer Virus Eradication Act of 1989 (H.R. 55) that calls for sanctions, both civil and criminal against computer viruses. But, this action is against an individual or individuals who created and/or inserted the virus. What about a the person who sold you the software or the software manufacturer? Can the they held liable? If so, what damages can be recovered?

It seems you can recover damages, but it is not a simple matter. James J. Ayres, an attorney with the Chicago firm of Magee, Collins and Lodge, and a part-time faculty member of DePaul

---

[12]Bernard P. Zajac, Jr., "Legal Options to Computer Viruses," Computers & Security, February 1989, p. 25.
[13]Ibid., p. 25.
[14]18 U.S.C. § 1030.

University's College of Law, points out that recovery can be approached in several different ways: it could be a pure contract law case between two or more parties; a Uniform Commercial Code case between buyer and seller; or a tort liability case. And within tort liability, it could either be a straight tort or a negligent tort, depending on the facts of the case. Or a cause of action under the Electronic Communications Privacy Act of 1986[15] could be initiated[16].

Ayres noted, "I think you would be hard pressed to argue that any commercially available software that comes in a box is a service." He said, "Customized software is more up the spectrum of service." Ayres said courts have held that information can be a product. If software is a good or product then the Uniform Commercial Code[17] has provisions for certain warranties.

The manufacture or publisher of the software has a certain responsibility to make sure the product is "virus free." Ayres points out, "Did the publisher know or should have [he] known" the software contained a virus? If so, then they are probably negligent. Explained Tabby, "If they can come into court and they can prove that they are 'state-of-the-art' for checking for

---

[15] 18 U.S.C. § 2510.

[16] Bernard P. Zajac, Jr., "Legal Options To Computer Viruses," Computers & Security, (To Be Published).

[17] U.C.C. §§ 2-312 -318.

viruses, and they missed this one. It'd be pretty tough, not only to hold them strictly liable, bit it'd would be pretty tough to hold them liable at all!"

As you can see, if you were the victim of a virus, you could have several options: go after the person who sold you the software, go after the publisher/manufacturer of the software, and if you know who inserted or created the virus the virus, criminally go after that person or persons.

Criminally charging someone for a virus or a computer crime is not new; as many convicted hackers know, it has been done and there is a body of supporting case law. However, civilly charging someone is new. The courts have yet to really address this.

As viruses become more virulent and prevalent, apprehension of the perpetrator becomes harder, victims, both corporate and individuals, will start looking to software manufacturers and vendors for: one, a higher level of assurance that the software is "virus free" and two, recovery for damages should they become a victim.

Summary

This paper shows that given enough hardware and software, we can protect most systems. However, we must remember that with every level of security we add, we suffer a decrease in the productivity of our users. It can't take them 10 minutes to log on. That is of course, unless our organizations are handling classified information.

With proper safeguards we can protect against MOST viruses. But as people create anti-virus programs and procedures others will come up with methods to beat them!

People are your great asset and they can be your greatest threat in any security plan, be it private industry or military. You can put all the hardware and procedures in place, but if your people don't endorse the security plan you can be compromised.

These are only a few of the things you can do to protect yourself from viruses. Viruses do not magically appear on systems, they are let in!

Viruses infect systems only when people let them.

Field Service Automation: Benefits and Pitfalls

By Asher F. Tunik - Computer Task Group Inc. (CTG)
One Commerce Center, Suite 1000
12th and Orange Streets
Wilmington, DE 19801

I. Introduction.

Automated systems have become common in most of the industries, providing variety of solutions in many different application areas. Accounts Receivable, General Ledger, Order Processing, Payroll, Inventory Control and Production Control automated systems were used since the dawn of the computer era. Then came an idea of automating the sales force. Quite a few sales force automation systems are currently available on the market.

For many years sales people had to rely heavily upon detailed record keeping either in their note books or in paper files to ensure that sales did not fall between the cracks. With introduction of an automated system it became unnecessary to keep paper files any more. Additionally management can measure effectiveness of marketing campaigns, telemarketing efforts and their sales force as a whole.

Just like a sales force, a service force has to rely heavily upon detailed record keeping. Service Representatives must know their customer base, what equipment each customer has, which machines are under a maintenance agreement and which are billable. If the same customer has equipment that is under maintenance agreement and some other pieces that are billable, the whole issue becomes even more difficult to resolve. Which Preventative Maintenance should be billed for, which one was already billed for (as part of a Maintenance Agreement), what parts are billable, in which case labor is covered? Add some machines under warranty, and the problems become even more complicated.

This presentation will discuss methodology and basic features that should be part of a Field Service Automation System.

II. You want to automate?!... What will come out of it?

Before you start with any project, it is usually a good idea to figure out where do you want to end up with it. Identify what your deliverables would be and when they will be delivered.

› Work with your service managers. Explain to them what your goals are, and how you can help their business grow. Encourage service managers to work with you at least during defining the initial system proposal.

› Define and qualify what service means to your customers. Automated Service Administration Package is a tool, which will enable the service organization to provide the exceptional service to your customers and therefore must be able to satisfy not only requirements of your business, but also the requirements and expectations of your customers.

› Set (or identify) specific guidelines for the service organization. These goals should be specific and measurable, and they should be based on customer research. Every point of customer contact should be included - telephone calls, letters and complaint handling as well as the delivery of service. Consider guidelines for speedy responces, for renewal of contracts, for contract quotations...

III. Reasons to automate.

Some of the major benefits of automating the service force are as follows:

o Turn the service organization from cost center into profit center.

Service organization does not have to be an overhead cost. It can be profitable! But, in order to make it profitable, the service organization will need proper tools in place.

o Provide a tool for the service representatives, so they can determine which machines are under a maintenance agreement, which machines are billable and which are under a warranty.

Such a tool will increase the accuracy of billings, customers will receive invoices faster, and invoices will be more accurate. For that reason service organizations will be able to save some expenses on labor (research each individual case, example: is labor billable or not? and if it is at what rate?) and customers will not dispute invoices, so there will be a decrease in volume of credit problems (example: you have billed me for the parts and I am covered under a maintenance agreement).

o Provide a tool for the management to gain better control, not only over the service organization, but over the entire business.

Management will be able to calculate Maintenance Agreement profitability by product line, by each model within the product line and have better idea about reliability of certain instruments. Also such a system will provide a manager with better understanding how the resources are distributed within the organization, who are the star performers and provide him or her with a tool for strategic planning.

o Provide an interface to the other automated applications in place.

Service representatives get to travel to the customer's site. If a certain machine moves (usually because its owner was transferred) the service rep can obtain up-to-date information about machine and its owner's location. Such a change can then trigger an update in the customer file, which in turn could lead to the new equipment sales and eventually more maintenance agreement sales.

o Allow more comprehensive and detailed analysis of the available market.

Since new system interfaces with all other automated systems, information will be available, about who is buying new machines. Upon expiration of the warranty period, these people will become potential customers for sale of a maintenance agreement. Also if life expectancy for a certain machine is 7-8 years, the sales force can approach the customer and potentially get a new sale. System will provide information about each product line, which product lines are more profitable than others and also provide information, why some product lines are more profitable than the others.

o Provide a method to standardize the approach to service administration, so that personnel turnover can be easily managed.

If all the information that a rep might need to know is on scraps of paper, it would be very hard to provide someone new with the information he or she needs to get the job done. However, if all the information is maintained in a standardized fashion inside of a computer, it is becoming just a question about how to train the "new kid on the block" to use the system.

o Provide means to renew contracts easily and accurately.

Renewal letters can be generated automatically, just by "pressing a button". They will be more unified, and have more accurate information about machines that need to renew (or buy) a maintenance agreement. Along with renewal letters, proforma maintenance agreements can be generated, which at the later date will either became active or will be deleted from the system (in which case machines will become billable).

o Relieve the service organization from cumbersome paperwork allowing more time to do what they do best - service equipment.

o Provide means of generating quotations for customers.

IV. Methodology of Automation.

A centralized IMAGE database will be needed to store the information. Having a centralized database will allow easier interface with the other major systems. Service Automation System will be able to interface with the

Inventory Control and Accounts Receivable (every time a certain part is used, system will "automatically" lower available inventory, not only for a tech but also for the whole available inventory). Every time that a customer gets invoiced, Accounts Receivable gets notified, that it should be expecting some income coming in.

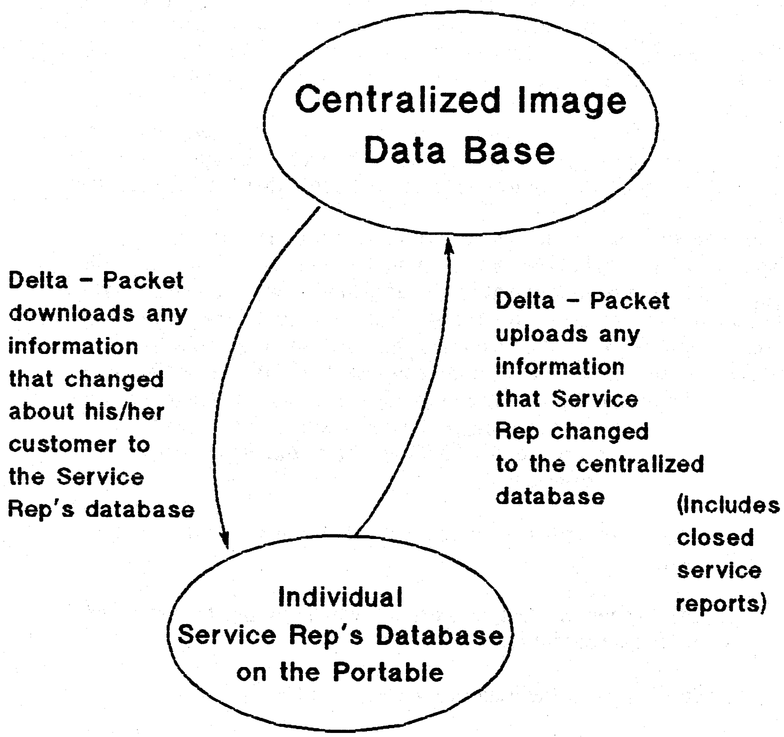
Every Service Rep in the field will get a portable PC, with an internal modem and software that allows to display, change and enter information about their client into the portable PC. Every night the Rep will have to dial into the host computer (using a little imagination that process can even be automated, so it does not require any human intervention). During that logon session information that was changed by the Service Rep about his/her customers will be uploaded into the mainframe, and any information that was changed by the central office will be downloaded into the portable PC. Information will be traveling back and forth in so called delta-packets.

Therefore, in order to have a successful logon session, each Service Rep will have to run a batch process on his/her portable PC in order to generate a delta-packet for upload. That batch process should run after all the daily transactions are entered into the PC. Similarly, at the end of the working day a batch process is needed to generate download delta-packets for each tech on a host computer.

Using commercially available communication packages and MS-DOS on the portable, it is possible to put the process of generating an upload delta-packet into the batch file. Once delta-packet is generated the same batch process can evoke the communications software, logon into the host computer and perform the swap of information. The same batch process then can apply the download delta-packet to the information in the PC. Similar batch process will have to run on the host computer side, in order to apply uploaded delta-packets to the centralized database.

So, in essence, at night, while the service rep is sound asleep, his portable will make make sure to upload his daily process along with all the generated billings and download all the information that has been changed since the last time of information exchange. Hardest part on the service reps part will be remembering to connect his portable into the phone line when he is done with the days work. Trick is to move only the data that have changed during the period from last information exchange.

Such a system will cut down on paper work. Field service report has to be filled up once (on a PC, by a service rep), there is almost no lead time for generating an invoice. Service Rep will have all the needed info on his portable to generate an invoice, and to figure out correct billings. The centralized system will have all the needed info for forecasting and other management functions, plus the ability to interface with every major system within the organization.


Centralized Image Data Base
Delta - Packet downloads any information that changed about his/her customer to the Service Rep's database
Delta - Packet uploads any information that Service Rep changed to the centralized database
(Includes closed service reports)
Individual Service Rep's Database on the Portable

Slide 5
AT6

V. Approach to automation.

As with any other system, the software functionality should be the primary reason as to how it is put together. The functionality of the service tracking data is quite different from most other systems that contrive corporate MIS. Service data contains information about productivity, about different markets, about profitability and reliability of certain product lines. It also contains sales leads and might even have a clues for new development.

Before implementing such a system, make sure that all the bases have been covered and you have a good understanding as to what the system should do in order to satisfy user's requirements and to fit into the "BIG" MIS picture.

The functionality that needs to be present in service force automation software is going to vary from organization to organization. There are some basic pieces that must be present in all implementations independent of organization. Most important the system must be easy for the service rep to use and fit into the "way" the business is handled. So before you start implementing such a system, it could be advisable to take a few field trips. If the system is easy for the service rep to use and data flows from portable PCs to the host computer and back, better service will be provided to the customer and more accurate overall data will be available for the management to make decisions.

Some additional features/functions that should be present in the Field Service Automation System are:

- Ability to generate customer lists for export to the other systems within the corporate MIS

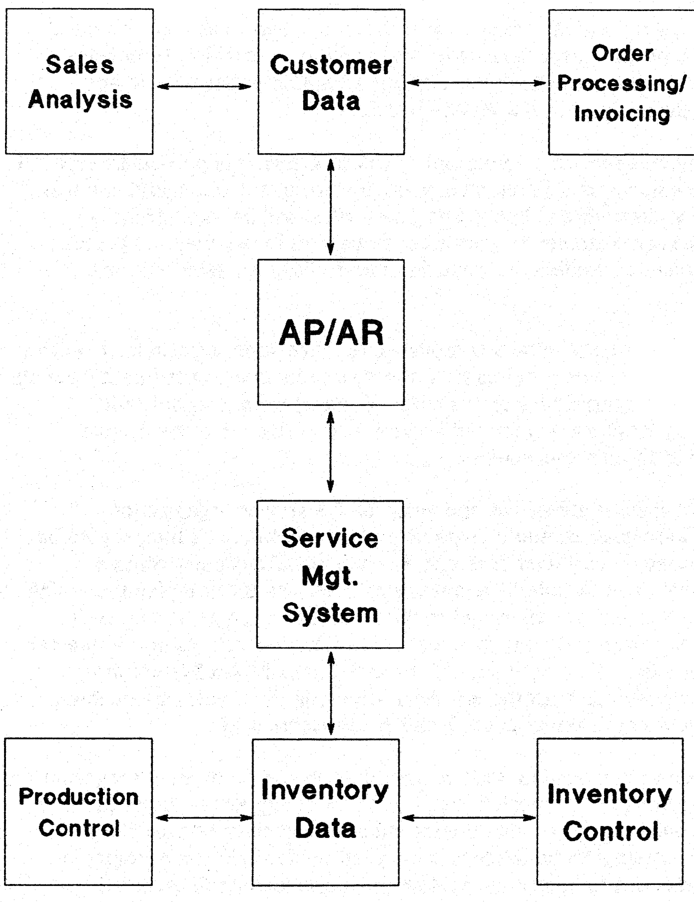
- Ability to look at the data in a variety of different ways

- Ability to generate good useful reports

- Ability to share information, so in case one of the reps visits an account for which another rep is responsible for, he/she can get information about that particular account easily

- Ability to interface with other existing data processing applications (Accounts Receivable, Inventory Control, Production Control)

> METHODOLOGY of AUTOMATION



These are just a few of the major requirements that the Service Force Automation System should satisfy. There could be many more things which could be of importance to your own shop (examples would be: audit trails, data dictionary maintenance, documentation, security of the proprietary data, Bill of Material or Indented Part List, Technical Data, Quality Control Data.). Therefore flexibility and ease of use in the service automation project are of the primary importance.

VI. You automated your Service organization... Then what?!

Customer service entails doing what it takes to ensure that your customers will have a positive experience with your product or service. It includes effectively dealing with the problems and complaints that arise in any business, particularly in the service business.

While customer service is important in any business, it is particularly critical to a service business. In such a business, the product is intangible and it is difficult for customers to know what the service will be like until it is delivered. Poor customer service hurts a business in two ways. It causes a loss of current customers and reduces opportunities for sales to new customers.

There are many tasks to be accomplished to make superior customer service a reality. Such service requires hard work and constant commitment from all levels of the organization starting with top managers and ending with contract coordinators. Automated Service Administration System must satisfy all of those requirements.

> Set specific guidelines and goals for the service organization. These goals should be specific and measurable, and they should be based on customer research. Every point of customer contact should be included - telephone calls, letters and complaint handling as well as the delivery of service. Consider such goals as speedy responses, frequent customer contact and reductions in the number of complaints. Automated Service Administration System must provide you with the means of achieving these goals, by enabling your organization to store and easily access data.

> Develop an ongoing staff communication and motivation program to ensure their support. Recognize that staff support is one of the most critical aspects of maintaining superior customer service. Data processing personnel that is assigned to maintain the Automated Service Administration System must also have their goals set

in order to provide the Service Organization with the ongoing support with the problems when they arise.

Providing exceptional customer service will not only enable your organization to sell more maintenance agreements and more service, it will also enable you to sell more machines (or what ever your final product is) thus increasing your market share.

VI. Conclusion

Those organizations that provide service should seriously consider automating their service force now. While most of the other major functional areas in an organization have been automated for a long time, service stayed mainly a manual or semi-manual task. The ability to get a handle on the service information will provide management and service reps with a powerful tool, which in turn will assure competitiveness of the organization in the ever changing market-place.

Bibliography:

1. Louden, Teri L. "The sale isn't over till the customer is happy". Modern Health. August, 1988.

2. Shirman, Mark P. and Lucia, Edward F. "Sales Force Automation - The Last Frontier". Presented at 1988 INTEREX Hewlett-Packard Business Users Conference, August, 1988.

MIS -- Preparing for the 1990's

Mitchell Kleiman
Ernst & Young
One Sansome Street
San Francisco, CA 94104
(415)951-3205

ABSTRACT

Technology in the business environment used to mean automation of existing business methods and processes. This provided those processes with faster throughput and sometimes extended capabilities. But focusing on minimizing the disruption caused by automation often limited the benefits provided. For example, the promised "people savings" and "paper-less office" were often never realized.

Today's more competitive business environment requires that the MIS function deliver business solutions, advantages, and real cost savings; in a timely fashion.

This paper addresses this new role for information technologies. A role in which the way MIS services have traditionally been provided is altered, and is now aimed at redesigning the business processes of the organization. Through an examination of the key areas of MIS (languages, tools, processing platforms and power, people, and methods), this paper will identify changes that are crucial to successfully deliver information resources in the future, and the impact of these changes.

THERE IS ONE THING THAT IS CONSTANT IN THIS FIELD:

CHANGE

****

The pace of technological change is accelerating, not slowing down or remaining constant. Information technology professionals will likely be the people expected to be in the know about changes and how new technologies can be applied.

The implication is that, in the future, executives who know less detailed information about technology and its applications will be called upon to make crucial information decisions that could impact the competitive position of the corporation. More and more, executives will need to depend on technical experts to make recommendations and critical purchase decisions.

It is not enough to be a good manager, good planner, and service-oriented. A chief information officer or MIS director needs to decide the when, as well as the what in strategic purchasing and positioning of hardware and software.

WHERE HAVE WE COME FROM?

- **AUTOMATING EXISTING BUSINESS PROCESSES**

- **DOING THE SAME THINGS FASTER**

****

Typically, the first applications of technology within a company is aimed at replacing the current manual system with an automated system; to do the work faster, more accurately, and to make information more available.

TYPICALLY:

- **COMPUTERIZE ACCOUNTING AND ADMINISTRATIVE FUNCTIONS**
- **THEN MOVE TO OPERATIONS**

****

The first functions to be automated in a company are typically accounting and administration (human resources, payroll, etc.) These are cost areas and if they can be done faster and with less people, they result in lower costs. These functional areas are fairly stable, transactions have a limited number of alternatives, and there are often many software packages to choose from on any hardware platform.

Once these easy areas are handled, the next area that often gets addressed is the operations side of a business. The operations side is often more complex and needs more customization; however, since it is usually organized to handle a large volume of transactions, information systems benefits can result in adequate paybacks. Again, there are typically several software packages to choose from on different computer lines, but often less choices available than for accounting packages, and often a particular platform will be stronger in a particular operations area.

So now that you have handled the easier systems: there appears (hopefully) to have been positive benefits from the implementation of computer systems resulting in cost savings or at least cost increase reductions, the staff are a little less overworked, and some reports get out faster than before.

It may appear there are potential benefits to computerizing other areas of the company, but those areas require more applications flexibility, the users are less technically sophisticated, and may be less patient with the process necessary for implementing technology. Against these pitfalls, the potential benefits may appear to be greater than for past functional areas -- particularly, if the people impacted by the change are managers and executives (higher cost of personnel, so a small benefit has a larger $ impact), or customer service or new products and services become available as a result.

NOW:

- **MOVING TO THE SALES AND MARKETING FUNCTIONS**

- **WORKING ON IMPROVING CUSTOMER SERVICE**

- **ASKED TO SHORTEN DESIGN LEAD TIMES AND PRODUCTION CYCLES**

****

Currently, many companies are implementing sales and marketing systems. These systems typically require more computing power, more communications abilities, and lots of flexibility (e.g., reporting formats, graphics). This is often a functional area in a company that requires total customization since each company does it differently, or so they say. Several manufacturers approach this business area with basic modules that serve as the backbone to a sales and marketing system. In addition, they require that the custom work development be done by either in-house programmers or software consultants.

Another current focus in the information technology field is directed at improving customer service. Executives have become concerned with the costs of the MIS function and expect a payback either in dollars or in improved (faster, better) service levels to the customer. Huge investments have been made in airline reservation systems, express package status tracking, and ATM networks to improve customer service and, if possible, to differentiate a company.

Information executives are being asked to provide real benefits of shortened product design lead times and shortened production cycles, especially in manufacturing industries, to improve the competitive nature of the firm. This is particularly critical for U.S. companies, given the lower labor costs and the competitive nature in other countries (particularly Asia).

MIS was often concerned with making new systems fit-in with the way the company had conducted business in the past, therefore resulting in as little disruption as possible. Operating guidelines such as these limit the potential benefits of implementing new technologies and often speed up the company processes, but do not result in substantial operating efficiencies.

Business conditions have changed radically over the past thirty years. International competition and improved global communications and transportation have led to a more global economy and a more competitive business environment.

In addition, the promised savings in people and paper have often never been realized. Executives are now scrutinizing computer projects more carefully and are more skeptical of the promised benefits.

This new environment requires that the MIS function deliver business solutions, advantages, and real cost savings; in a timely fashion.

THE NEW ROLE FOR INFORMATION TECHNOLOGY, IS TO FUNDAMENTALLY REDESIGN OR ALTER THE BUSINESS PROCESSES.

HOW?

- **REDESIGN BUSINESS PROCESSES IN ALIGNMENT WITH THE BUSINESS STRATEGY.**
- **DEVELOP FLEXIBLE, MODULAR SYSTEMS RAPIDLY.**
- **INCORPORATE APPROPRIATE NEW TECHNOLOGIES WITH OLDER ONES.**

****

It has become critical for information technology professionals to develop an understanding of the business and the critical success factors of that business in order to effectively apply technology solutions. In order to provide a competitive advantage or to stay competitive with industry leaders, IT professionals must stay abreast of corporate strategies and align the technology function objectives with the overall business objectives.

Executives will be less likely in the future to accept one- or two-year application backlogs, or long term implementation schedules. Systems must be flexible enough to keep up with the fast changing needs of the business.

As new technologies are developed, the timing element for implementation is crucial. Information professionals are expected to know when the time is right to take advantage of new technology, and how to merge its use with existing systems. This will require IT professionals to strategically plan and anticipate new developments. Current systems development either in modular fashion or through the insertion of "hooks" can be made ready for future use without major redevelopment.

NEW TECHNOLOGIES

- RELATIONAL DATABASES
- CD-ROM, WORM, AND DAT DEVICES
- SCANNERS
- OPTICAL CHARACTER RECOGNITION(OCR) DEVICES
- ELECTRONIC DATA INTERFACE(EDI)
- BAR-CODING
- DESKTOP PUBLISHING
- NETWORKING
- DISTRIBUTED DATABASES
- IMAGE PROCESSING
- ARTIFICIAL INTELLIGENCE

LEVERAGE THE BENEFITS:

- **ELECTRONIC MAIL**
- **AUTOMATIC MEETING SCHEDULING**
- **BUSINESS CONTACT SYSTEMS**

****

There is an enormous capacity of computer processing power on the desktops of American businesses. For the most part, that power is being under utilized. As desktop workstations are networked, additional benefits become available besides the separate applications advantages for which the PC's were originally purchased. Through the use of electronic mail, companies are finding they can reduce phone, travel, and express courier costs and enhance internal business communications. Other applications become easy to implement and can provide additional benefits once a network and desktop machine have been installed. Some of these applications include automatic meeting scheduling for employees, client contact tracking systems, automatic dialing systems, internal forms replacement, etc.

Once you have invested in a technology, look to see if you are getting the full value. Are there opportunities to leverage that investment?

KEY ARENAS OF INFORMATION TECHNOLOGY

- LANGUAGES
- TOOLS
- PROCESSING POWER/PLATFORM
- PEOPLE
- METHODS

LANGUAGES

The flexibility and quick response required for redesigning the business systems will require fourth and fifth generation languages.

Languages and related language development environments with modular coding (for reusability of code), easier maintenance, and power will take the lead.

Languages that work in the DOS and UNIX environments will begin to dominate the microcomputer and minicomputer markets. Take a look at the leading edge software applications and the various operating systems they are being developed for.

TOOLS

- Tools to organize data into information will become more important.
- Tools that allow data to be shared across systems will gain market share.
- Tools that make information available in any format from any program will be needed to integrate data from different systems and sources.
- Tools to insure the accuracy of information will become more critical as more and more information is stored in systems and becomes available.

[Code Generators, CASE tools, Database Add-on tools, Application Prototypers, Auto-Documentors]

PROCESSING POWER/PLATFORM

- We will need more processing power in the future and will need to obtain it (but it will be cheaper and provide more value).

- The future platform is going to be a desktop microcomputer (or commercial workstation).

  - the most cost-effective platform for delivering highly variable processing power needs locally without impacting other information users.

  - can adapt faster and put to use new technologies, and can talk with all other systems.

PEOPLE

- Stronger people skills - for delivering business solutions will be needed.

- Expertise in new technologies will become more critical.

- Expertise in industries will be required to grasp the critical success factors and align information function strategies with business strategies.

- Less "lone-wolf" and more "team-worker" type attitudes will be required as system complexity and size increase to meet business needs. Separate systems will be a thing of the past. Integration needs will require systems and people to work closer together.

- As complexity and scope increase, we will see a shift to contracting for the expertise instead of hiring industry and tool expertise.

METHODS

- Information Engineering
- Apply new technology
- Education
- Better Cost Justification
- Strategic planning
- Integration of equipment and software

HOW TO GET THERE

1. Move to personal computers and educate yourself and your staff about PC's.

2. Find out what new technologies are of interest to people and require them to keep up-to-date on those areas and to educate others - through topic papers, seminars, industry journals, etc.

3. Attend seminars and conferences, find out:
   - what is new,
   - what works,
   - what it can do,
   - what it costs,
   - where are things headed, and
   - how does it pertain to your company's business.

4. Do a study (either formal or informal) within your company as to what the current and anticipated business problems are and what executives are willing to pay to solve a particular business problem (or what it costs the company to not have solved it).

5. Talk with other MIS executives, how are they solving their companies business problems?

6. Talk with experts and find out what they recommend.

7. Test out new uses of technology within the MIS area.
   - make the projects useful, cost-effective, a learning experience for people.

TITLE: Opportunities for Increasing Productivity Using Electronic Printing Technology

AUTHOR: Bruce Toback

Handouts available at presentation.

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 3930

TITLE: How To Build An Executive Information System Using Today's Technology

AUTHORS: Terrence D. O'Brien and Janet Eden-Harris
Dynamic Information Systems Corporation
910 15th Street, Suite 640
Denver, Colorado 80202
(303) 893-0335

As the 1990's approach, subtle shifts are taking place that will have profound effects on the way we all see our roles as managers of information processing. In stark contrast to earlier years where CPU speed, application sophistication and benchmarks of performance were the standards with which we measured our contributions, we've begun to see an increasing emphasis on the true purpose of computing: now that we have stored and analyzed the data in heretofore unimaginable proportions, how do we get the data of interest to the people that need it WHEN they need it?

The concept of an Executive Information System (EIS) has actually been in existence for almost a decade in academic circles. With the increasing interest in providing decision-support systems to management, EI systems are now being analyzed for their practicality in today's computing environment.

An EI system can be the informational nerve center for an organization. By providing an effective mechanism for information development and dissemination, an organization can achieve the increased operational efficiencies that has been the promise of decision support systems. So just what is an EIS? David Friend, in an article entitled "The Three Pillars of EIS", has defined an EIS as a system intended to provide "easy access to individually specified mainframe information that can be interrogated and manipulated by nontechnical end users." To elaborate, a successful EIS will accomplish the following objectives for your organization:

1. Provide the right information to the right people at the right time and in the right format to facilitate timely decisions.
2. Provide an effective mechanism for information exchange between managers and subordinates and between departments.
3. Maximize the most important organization resource: time. Time of executives, information providers and data collectors.
4. Provide this information at an investment that is significantly below the value of the information.

An EIS should provide instant, ad hoc retrieval, with the ability to add new reports and screens quickly and easily. It should provide summarized information, tailored to each executive user or information provider. It should contain seamless interfaces to standard tools such as Lotus 1-2-3 for informational analysis. And it should utilize an inviting personal computer based interface using color screens, on-line help, menus and pop up windows.

So how can we create an effective EIS, using software and hardware tools existing today? It can be done, but first, let's put aside some assumptions:

1. EIS is actually a misnomer. It implies that "executives" will use the system we devise to access information. They won't. But their assistants and analysts--the "information providers"--will. The EIS must be designed with the informational needs of these information providers and top executives in mind.
2. An EIS does not have to incorporate artificial intelligence techniques or a natural language. At some point in the future, we may devise a system capable of utilizing AI and natural language in such a way that our requests cannot be misinterpreted by an unfailingly logical computer. But in truth, what users want today is a straightforward way to instruct the system to extract the data of interest. PC windows and menu systems have gained enormous popularity for precisely this reason. This is the current user interface for today's EI system.
3. An EIS does not require a relational database management system, nor is the relational database a particularly good platform for the EIS functionality. The current relational DBMS's do not contain a sufficiently flexibile retrieval mechanism to handle the data retrieval speed necessary to

support an EIS.

4. An EIS does not require extravagant new software systems. What it does require is a PC, user-friendly PC reporting and analysis tools (such as Lotus 1-2-3, DBase or Excel) and an instant and seamless retrieval mechanism on the corporate database.

Dynamic Information Systems Corporation: A Case Study

Dynamic Information Systems Corporation (DISC) is a Denver-based software firm providing database tools and user interfaces for the Hewlett Packard 3000 line of minicomputers. The corporation has grown rapidly, doubling in size in each of the past five years. With over 60 employees in five offices, the company has reached a "critical mass" of informational needs where word-of-mouth, memos and meetings no longer suffice to provide the information to the senior executives for decision-making. The company recognized a need to provide effective and timely information to the "information users" and primary decision makers. Using existing Hewlett Packard hardware and software, their own OMNIDEX DBMS and other available third party PC hardware and software, the company is developing a complete Executive Information System.

When analyzing the informational needs of the executives, DISC evaluated the needs of those that would be using the system extensively. We concluded that the analysts that need to access data to provide information for decision making usually have a few characteristics in common. They want data in a form they can use. They want it in a way they can understand and interpret. And they want it IMMEDIATELY. A PC equipped with Lotus 1-2-3 can provide the first. An effective interface product such as Walker Richer & Quinn's Reflection and DISC's OMNIVIEW that utilize the power of the PC and its extensive help system, color graphics and menu/command combination can provide the second. And a DBMS product such as OMNIDEX can provide the instant and flexible data retrieval needed to make an EI system practical. Other products that can give your EIS more functionality include a good production report writer product such as QUIZ or Reactor, a 9600 baud modem to give your EIS (and your executives/analysts) portability, and a PC graphics package to give the numbers "life."

Specifically, DISC has incorporated the following tools in building its EIS:

HARDWARE
HP3000 (Spectrum 925) as a database and communication server
Telemon Network engine for accessing outside information bureaus
    and dialing into customer systems
Fax system
Compaq and Vectra PC's with color screens
    (80286 and 80386 processors)
HP color printers and plotters for presentation graphics
HP Laserjet printers for word processing and desktop publishing
HP Scanner (scanjet)for scanning external documents for indexing
    into the corporate correspondence tracking system
Compaq portable PC's
2400 and 9600 baud modems for remote dial access by field sales
    reps

SOFTWARE

Accounting system

(Multiview was chosen because of the Cognos dictionary which can be used with OMNIQUIZ and OMNIVIEW

OMNIDEX-based systems:

- Sales lead tracking system
- Technical support tracking system
- Production tape cutting system
- Corporate communications tracking system

Electronic mail (Xpress by Robelle) used by every employee in the organization

PC-to-mainframe communication software (Reflection by WRQ)

Company-wide Roll-o-dex using an OMNIDEX indexing eliminated manual roll-o-dexes

OMNIVIEW (Lotus 1-2-3/HP link) can be used as a casual end-user interface or a programming language to develop a graphics based financial reporting system

Spreadsheet analysis (Lotus, Quattro and Excel)

Desktop publishing (Ventura)

PC word processing (Word Perfect and Word)

Equipped with these existing tools, DISC has built the key components of a functioning EIS system capable of providing data instantly to executives. Automatic exception reporting. Sales summaries. Budget preparation. Product cost analysis. On-line customer support. Automatic mailing lists. Users at all levels of the organization can now get the reports they need without usurping costly DP resources for every analysis.

indigo
Software Ltd.

560 Rochester Street
Suite 400
Ottawa, Ontario
Canada K1S 5K2
(613) 594-3026

April 4, 1989

Interex
680 Almanor Ave.
P.O. Box 3439
Sunnyvale, CA
94088-3439

Dear Sir or Madam,

Enclosed please find a copy of my paper #3932 for the Interex North American User's conference in San Francisco.

Sincerely,

Barry H. Gillespie
Director of Marketing

Advanced Electronic Forms Processing Techniques on the Hewlett-Packard LaserJet

Barry H. Gillespie and Donald F. Henderson, Indigo Software Ltd.
560 Rochester St., #400, Ottawa, Ont., K1S-5K2
(613) 594-3026

Introduction

The use of PCs and laser printers to replace pre-printed business forms is now an accepted practice, just as it has been on HP3000s and high end laser printers for a number of years. The simple replacement of existing pre-printed forms by electronic forms only begins to take advantage of this technology however. This paper will discuss several ways to expand electronic forms technology. MIS managers should begin to think of new and creative ways to use laser printers to perform data processing tasks that previously couldn't be done using continuous feed pre-printed forms and impact printers.

Applications that are written to work in conjunction with pre-printed forms are a major source of program maintenance expenditure in any MIS shop. If a form changes, even slightly, a progammer must go in and modify the code to adjust the overlay mask to the new format. This is a result of the fact that the order and the position of each data element is stored in the coded logic or the data maps of the program. This problem can be partially or completely avoided using electronic forms.

The data generated for pre-printed forms and impact printers is order and position dependent because impact printers print only one character at a time. Laser printers like the HP LaserJet are full page buffer devices. The image of the entire page is stored in printer memory, form and data, before anything is physically printed. This allows for other, more "intelligent" forms of data. Data files may be position independent, order independent, or both. This assumes, of course, that appropriate data merge software exists that takes advantage of these data types.

If the position of the data fields are somehow stored with the electronic form image itself, then the data file would simply have to be in the correct order. The data merge software would process the data one "piece" at a time and fill in the blanks sequentially. Each piece of data would simply be separated from the next by a known delimiter. In the PC world, comma delimited files, with ASCII strings enclosed in double quotes (") are common. Any database application has some sort of data export facility that can create such a file. On an HP3000, QUERY can be used to create a file where carriage return and line feed become the delimiter. The problem with this method is that if the form changes such that the order of the fields changes, the code to generate the data file must still be modified. The data file is order dependent.

Since the entire page is stored in the printer at one time, there is no need to send the data to the printer in the correct order if each piece of data could be tagged with its coordinate address. A file of the form:

X coordinate, Y coordinate, data

would achieve this purpose. The data merge software would simply have to position each data element at the given XY address. This method has a serious flaw also. If the position of a field on the form changes, the code to generate the data file must be changed. The data file is position dependent.

Is there a method that is both order and position independent? A modification to the position independent example above will do. In addition to storing a field's position with the electronic form, assign each field a unique name. A data file of the form:

field name, data

could be used. This of course implies that the data merge software has the intelligence to correctly interpret a data file of this nature and the forms design software allows you to specify where fields should be on the form and assign them names. The only time that the program that generates this data file will have to be modified is if a form modification adds or deletes a field. The actual code changes are much simpler. To delete a field, the code to generate that particular piece of data is simply deleted. Code for a new field can be added anywhere, since the order of the data is immaterial.

Once you have achieved this new indepentent mode of data generation, several other even more interesting ideas can be considered. Five areas will be discussed: complex multi-part forms, variable two page forms, selective multi-page forms, duplex forms and dynamic data-driven forms. All of these new techniques are possible because, when using electronic forms, the shape of the form template and the order in which successive templates are printed can be determined at processing time depending on the nature of the data.

Complex Multi-part Forms

Conventional pre-printed multi-part forms have two limitations.

With pre-printed forms, only a limited number of parts are possible because of the nature of impact printer technology. Once a multiple part form exceeds five or six parts, the bottom layers of the form become more and more unreadable, and the total thickness of all the parts makes printer jams more and more likely to occur. Electronic forms overcome both of these problems. Since each part is printed separately an unlimited number of parts can be printed. The forms fill-in software can take one set of data and ensure that it is re-printed correctly on all parts, whether the number of parts is two or twenty-two. Every part will be equally readable because each one is an original. Since each page is fed separately, no jamming problems can occur.

Pre-printed multi-part forms also demand that the data fields on all parts must be in exactly the same spot and be printed in exactly the same font on every part. Electronic forms eliminate these restrictions and allow the designer to make each part suit its function. The following purchase order example will illustrate these points.

The purchase order has four parts, the file copy, the vendor copy, the accounting copy and the receiver's copy. The file copy is shown in figure 1, with all data filled in.

The vendor copy is identical to the file copy except that the PO number is bar-coded, along with the purchaser's customer code for this particular vendor (see figure 2). More and more vendors are requiring that this sort of data be bar-coded to ease their data processing procedures. An electronic forms fill-in program should be able to write data in all the common bar code formats as well as in text fonts. The boxes for vendor and customer codes are larger to accomodate the larger area required for bar-coded data.

indigo
Software Ltd.

PURCHASE ORDER

PO Number	Vendor Name	Customer Code
ABC123	Master Widget Corp.	998-034

Date	Vendor Code	Terms
March 11,1989	MAR023	Net 30, FOB Destination

SHIP TO

MegaCorp Inc.
123 Main Street
Anytown, CA
90123

BILL TO

Same

Item No.	Description	Order Unit	Quantity	Unit Price	Total Price
WID001	Extra Small Widget	Case	3	21.00	63.00
WID002	Small Widget	Case	3	31.00	93.00
WID003	Medium Widget	Case	3	41.00	123.00
				TOTAL	279.00

FILE COPY

Figure 1
Part 1 of Purchase Order: File Copy

PO Number	Vendor Name	Customer Code
ABC123	Master Widget Corp.	998-034

Figure 2
Purchase Order: Vendor Copy - Vendor Code and Customer Code in 3-of-9 Barcode

The accounting copy requires an additional column on the right hand side of the page for the accountants comments and/or notes. Information such as check number, back-orders, short shipment amounts, and other useful information will be recorded here after the shipment is received. To accomodate this extra column all of the other columns must become narrower. The data on parts 1 and 2 is in 10 pitch courier. On the accounting copy (see figure 3) the data will be printed in 12 pitch courier, making each column 83% of its former width and allowing room for the blank "Accountant's Comments" column.

Item No.	Description	Order Unit	Quantity	Unit Price	Total Price	Accountant's Comments
WID001	Extra Small Widget	Case	3	21.00	63.00	
WID002	Small Widget	Case	3	31.00	93.00	
WID003	Medium Widget	Case	3	41.00	123.00	

Figure 3
Purchase Order: Accounting Copy - Extra column for accountant's comments

The fourth part, the receiver's copy has even more changes. The receiver shouldn't know what quantities of each item were ordered. He/she must do a physical count of the actual shipment. The "Quantity", "Unit Price" and "Item Total" data is removed and is replaced by "Quantity Received" and "Receiver's Comments" data (see figure 4). With pre-printed forms the dropping of data is accomplished by using specially cut (and expensive) carbons, or by having large black (and therefore unusable) areas where the data would normally appear.

Item No.	Description	Order Unit	Quantity Received	Receiver's Comments
WID001	Extra Small Widget	Case		
WID002	Small Widget	Case		
WID003	Medium Widget	Case		

Figure 4
Purchase Order: Receiver's Copy - Data is "Blacked" Out

It should be noted here that the data merge capabilities of the electronic form processor must be able to handle the output of data in multiple fonts, of suppressing data that mustn't appear on a particular part and of taking one set of data and having it appear correctly on all parts.

Variable Two Page Forms

Variable two page forms are a special way of handling situations where there is a fixed amount of header data for a form and where the the amount of detail data may vary from 1 to many pages. A standard company stock taking report is a good example. Small branches of a company may carry only a few items in stock, relying on quick delivery of other items from a nearby larger branch to meet customer demand. Larger branches and the main stores will carry progressively larger numbers of items. The first page of the stock report is shown in figure 5.

WEEKLY STOCK REPORT

Location Number	NN-002-123	Address	123 Main Street
			Anytown, TN
For Week Ending	March 12/89		34567-1234, USA

Stock Clerk Arnold P. Schwartz	**Signature**
Store Manager Janice R. Wainwright	**Signature**

Item Number	Item Description	Order nit	Re-order Level	Last Week	This Week	Order
PP-0001	Size 1 Widget	Box	50	68	24	100
PP-0002	Size 2 Widget	Box	50	14	98	0
PP-0003	Size 3 Widget	Box	50	53	13	100
PP-0004	Size 4 widget	Box	50	78	52	0
PP-0005	Size 5 Widget	Box	50	102	78	0
PP-0006	Size 6 Widget	Box	50	57	34	50
PP-0007	Size 7 Widget	Box	40	99	62	0
PP-0008	Size 8 Widget	Box	30	109	88	0

Figure 5
Weekly Stock Report - Header and Detail Lines

The top half of the page is taken up with fixed header information and sign-off areas for the store's clerk and the store's manager. The bottom half consists of detail lines. For a small store this one page may be adequate. But what about larger branches with many more items than will fit on one page.

With pre-printed forms the top half of the second page would be left blank, with more detail lines filled in on the bottom half. This would be repeated for the third page, fourth page, etc. until all data was processed. The top half of the page is wasted. With electronic forms the second page can contain only detail lines (see figure 6).

Location Number NN-002-123 For Week Ending March 12/89

Item Number	Item Description	Order nit	Re-order Level	Last Week	This Week	Order
PP-0009	Size 9 Widget	Box	30	54	34	0
PP-0010	Size 10 Widget	Box	30	29	77	0
PP-0011	Size 11 Widget	Box	30	18	54	0
PP-0012	Size 12 Widget	Box	30	46	26	60
PP-0013	Size 13 Widget	Box	30	78	51	0
PP-0014	Size 14 Widget	Box	20	29	14	40
PP-0015	Size 15 Widget	Box	20	17	37	0

Figure 6
Stock Report - Details Only Second Page

This "details only" page can be repeated the number of times necessary to print all detail information. This process uses half the paper of the pre-printed method and allows all branches, large or small to use an identical format.

This method could not be used with continuous feed pre-printed forms because there is no way of knowing how many "details only" pages will be needed until print time. It could be anywhere from 0 to 100 or more depending on location. Once again the electronic forms data merge software must be intelligent enough to handle the complexities of this situation.

Selective Multi-page Forms

Selective multi-page forms are very common in the insurance and banking industries. An insurance policy or a loan agreement has several sections, with one or more forms involved in each section. Depending on the options a client chooses, a subset of all the forms will be printed. This sort of application is not possible with pre-printed forms because the order in which the forms print and the actual selection of forms is driven by the data.

For instance client A may buy options one, three and six of a standard insurance policy, while client B will buy options two, three and five (see figure 7). The electronic forms processing software must be able to store all the forms together as one document and then select the correct forms templates based on the data.

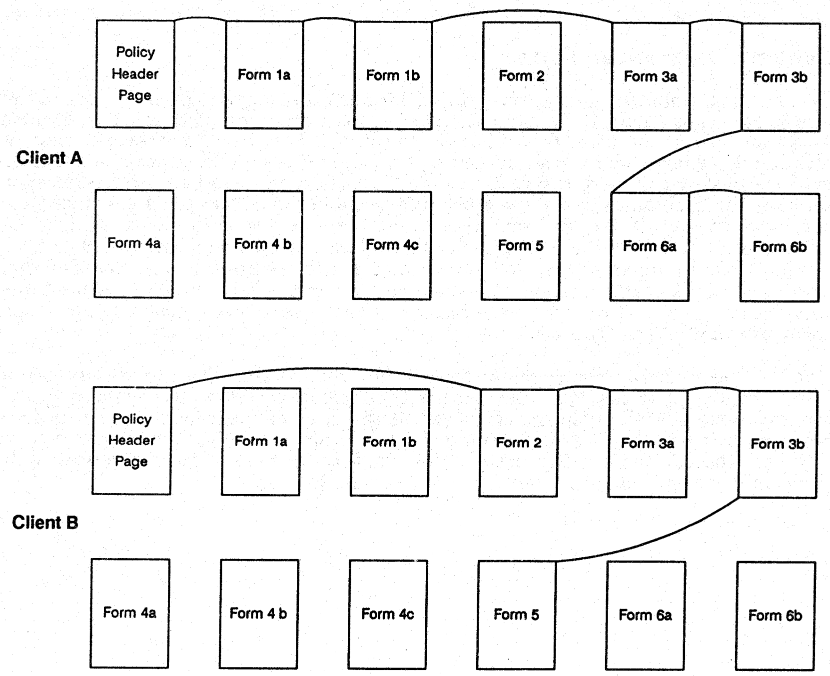


Figure 7
Two different subsets of the policy forms for different clients.

In each of the three preceeding examples all of the forms should be stored as macros in the memory of the LaserJet printer. This will greatly increase the throughput rate when multiple sets of data are being processed. Of course, the electronic forms processing software must have the sophistication required to load and later access these form macros correctly.

Duplex Forms

A duplex form is one that has information on both sides of the page. With pre-printed forms, a form can have template (ie. fixed) information on both sides of the page, but data may only be filled in on one side. Impact printer technology does not allow data to be filled in simultaneously on both sides at once.

With electronic forms and duplex LaserJet printers like the HP LaserJet IID and HP LaserJet 2000 this restriction is removed. Forms can be designed to have data filled in on both sides of the page reducing the amount of paper used. This reduces paper costs, as well as postage and filing costs. The electronics forms processing software must be able to take advantage of the duplexing feature of these printers.

This feature could be successfully combined with variable two page forms or selective multi-page forms, making either of these advanced techniques even more attractive.

Dynamic Data-Driven Forms

The last extention to normal pre-printed forms technology is dynamic data-driven forms. Here not only the data, but the shape of the form template itself is variable. An auto insurance accident report is a good example. The report has several sections, the claim's adjuster identification section, the accident scene description section, the vehicles involved section, the injured persons section and the police officers report section. The problem is that though the first two and the last sections are always of a standard size, the third and fourth sections may vary from a single vehicle and person to many vehicles and people. On a standard pre-printed form enough room is left for a fixed number of vehicles and people. If this fixed number is exceeded then additional pages of information must be appended to the form listing the additional vehicles and people involved. These pages are out of sequences, probably not in the correct format and easily lost.

Dynamic forms solve this problem by actually constructing the forms template at print time based on the data. Each part of the form is stored in printer memory as a separate macro. Unlike the macros used in the first four applications, these form piece macros use the LaserJet's relative addressing feature rather than the more common absolute addressing used in full page templates. Figure 8 contains the forms pieces required for our auto insurance example.

indigo
Auto
Insurance

Accident Report

A MaxiSure Company

Claim Adjustor	ID Number	Signature
Name		

Accident Scene Description	# of Vehicles	# of People	Road Conditions
What Happened			
			Date of Accident

Vehicle	Plate Number	State	Driver	Driver's licence Number	State	Vehicle #
Description of Damage						

Injured Person	Vehicle #	Name	Address
Description of Injuries			

Police Officer's Report	State	Badge Number	Unit Number	Report Date	File Number

Figure 8
The six template pieces for an Insurance Accident Report.
The last piece is simply a line that will clean up the bottom of pages.

The form processing software must be capable of determining which form piece is required for the next piece of data, and whether or not there is room on the page for that piece. If there is room, the template macro is executed, then the data for that template piece is filled in. After processing each piece and its corresponding data the printers cursor position is set to the bottom left corner of the completed part of the form, ready to process the next piece.

If there is no room for the next piece of the form the page is ejected and a new page is started. There may be standard bottom of page and top of page pieces that are always processed just before the current page is ejected and just after the new page is started. These pieces would give the form "clean" edges.

Figures 9 and 10 show two different results based on different data sets. Figure 9 is a form for a single vehicle accident with 3 people injured. Figure 10 is for a four vehicle accident with no injuries.

Conclusions

The five examples given here are all extensions over what can normally be done using pre-printed forms. What does this mean to today's MIS manager? Many of the processes that are currently taken for granted must be re-examined in the light of this new technology. The "We've always done it this way!" philosophy will no longer do. When thinking of converting from pre-printed forms methods to electronic forms methods the MIS manager can go beyond simple pre-printed forms replacement and take advantage of these more advanced techniques. Applications that before could simply not be done using pre-printed forms are now possible using electronic forms.

indigo
Auto
Insurance

Accident Report

A MaxiSure Company

Claim Adjustor	ID Number	Signature
Name		

Accident Scene Description	# of Vehicles	# of People	Road Conditions
What Happened			
			Date of Accident

Vehicle	Plate Number	State	Driver	Driver's licence Number	State	Vehicle #
Description of Damage						

Injured Person	Vehicle #	Name	Address
Description of Injuries			

Injured Person	Vehicle #	Name	Address
Description of Injuries			

Injured Person	Vehicle #	Name	Address
Description of Injuries			

Police Officer's Report	State	Badge Number	Unit Number	Report Date	File Number

Figure 9
An Insurance Accident Report.
A one vehicle accident with 3 people injured.

indigo
Auto
Insurance

Accident Report

A MaxiSure Company

Claim Adjustor	ID Number	Signature
Name		

Accident Scene Description	# of Vehicles	# of People	Road Conditions
What Happened			
			Date of Accident

Vehicle	Plate Number	State	Driver	Driver's licence Number	State	Vehicle #
Description of Damage						
Vehicle	Plate Number	State	Driver	Driver's licence Number	State	Vehicle #
Description of Damage						
Vehicle	Plate Number	State	Driver	Driver's licence Number	State	Vehicle #
Description of Damage						
Vehicle	Plate Number	State	Driver	Driver's licence Number	State	Vehicle #
Description of Damage						

Police Officer's Report	State	Badge Number	Unit Number	Report Date	File Number

Figure 10
An Insurance Accident Report.
A four vehicle accident with no injuries.

Sales Force Automation: Today

Michael Barry
Gateway Systems Corporation
2400 Science Parkway
Okemos, MI 48864
(517) 349-7740

Technological advances in the personal computer field are making sales force automation (SFA) an accepted tool in field sales force management. Many companies have provided laptop and desktop computers, as well as specialized software, to their field representatives. Others are debating where and how to start the search for a sales force automation system.

This paper will discuss some of the main reasons to automate, and look at some of the major U.S. corporations that have done so. I'll also be taking you through a needs analysis to help you decide what your company should be looking for when choosing sales force automation software and hardware.

Advantages

The first question to ask when deliberating the move to Sales Force Automation is "Why automate?" In a 1988 survey by *Sales and Marketing Management Magazine*[1], twenty-four top sales force executives responded to questions about their automation situation. Eighteen companies (75%) said they had already equipped their sales reps with computers, two others were in pilot programs, and two more were conducting feasibility studies.

The twenty-four companies surveyed had been chosen by their peers as having the top sales force in their respective industries[2]. Many of them attributed their success to the return on investment and intangible benefits their sales force automation systems provided.

There are five important advantages to sales force automation.

***Shorter sales cycle**
***Higher employee morale**
***Improved internal communications**
***Increased customer service**
***Increased sales productivity.**

---

[1] Thayer Taylor, "How the Best Sales Forces Use PCs and Laptops", *Sales and Marketing Management Magazine*, April, 1988, p.64-74.

[2] *Sales and Marketing Management Magazine*, Annual Survey, June, 1987.

When a company performs a cost justification for their SFA program, it usually includes at least one of these categories.

Shorter Sales Cycle

A McGraw-Hill study has estimated that by 1994 the average cost of each sales call will be over $500. Shortening the sales cycle means decreased business costs and less time for external influences to impact the deal. After automating their sales force, Stephen Korbecki at Inland Steel found, "They [field representatives] are able to make better decisions because the PC delivers information more quickly, and this is especially important when it concerns the bottom line on costs and prices."[3] The more information the sales rep has available during the initial call, the fewer times they will have to return to the client. A shorter sales cycle lowers company expenses, and in many cases will increase revenue since deals will close faster.

Higher Employee Morale

Hewlett-Packard pioneered a study of sales force automation productivity in 1986, using a test and control group. An intangible but obvious result of their automation effort was the increase in employee morale. Their test group sales reps felt their confidence and sense of professionalism grew due to an increase in their motivation.

Many companies express concern about how their salespeople will deal with automation, but Pepsi-Cola found their sales representatives reacted positively. "The salespeople are so excited over having data in a manageable form that they've taken to the program quite rapidly," said Kim Kelly of Pepsi Cola[4].

A key result of higher employee morale is lower sales rep turnover. The cost of losing a salesperson includes search and training expenses, and loss of business in the territory. This alone may justify the cost of a sales force automation system for some companies.

Improved Internal Communications

Communications were also a key in the HP study. Their daily internal meeting time dropped from 13% of the rep's day to only 7%. Rick Tancreto of Black and Decker found that their automation program brought the same results. "When a district manager wants to send details of a new promotion to the sales force, he puts a message in our mainframe. The next time the rep logs on,

---

[3] Taylor, p.66

[4] Taylor, p. 74

he's told there is a message waiting."[5] Rather than wasting time with a sales meeting, the information goes directly to the rep who reads it between sales calls.

Customer Contact Time

One of the biggest changes noted during the HP's pilot program was the increase in customer contact time. With a sales force automation system, their reps spent less time on internal meetings and travel (improved internal communications and shorter sales cycle), and were able to increase time spent with customers and prospects. Before automation, the reps spent 26% of their day calling on clients. After automating, they were able to increase the contact time to 33% of their day. As of May, 1988, that figure had reached 36%.[6]

Frederick Stephens of Gillette, also feels that increased contact time is important. "Making salespeople more knowledgeable about their accounts and the people they deal with is a critical issue if you want to become a preferred vendor."[7]

Increased time with the clients gives the salesperson a better understanding of the needs and wants of the people they service. It also lets them address any objections that may arise before they become critical.

Increased Sales Productivity

Finally, Hewlett-Packard and other companies have found that sales force automation increases sales productivity. HP figured a twenty-five percent increase in selling time produced an 8 to 13% gain in orders. Their automation program actually yielded a 27% increase in selling time, and sales rose accordingly. A good sales force automation system will also increase sales productivity by aiding the sales rep to determine the best prospects with the greatest sales potential.

Now that you've seen the benefits of having a good sales force automation program, let's look at some of the steps you should follow to make sure your system yields similar positive results.

---

[5] Taylor, p.67

[6] Thayer Taylor, "Improving Sales Force Automation", *Sales and Marketing Management Magazine,* May 1988, page 72.

[7] Taylor, p. 66

Preparing to Automate

Needs Analysis

Initially you may think only your sales reps will be using your sales force automation system. Companies with mature SFA programs, however, find that nearly everyone can and will benefit from it. Marketing, customer service, telemarketing, R&D, and training can all use the information gathered by the field representative. That is why it's important to assemble a planning committee made up of all potential users from throughout your company.

The committee must include sales reps, staff personnel, corporate field managers, and any others who need the information a good sales force automation package will provide. Everyone on the project team will have a fresh perspective of what they want the system to do, and you'll find the system becomes much more than the electronic mail and calendar tool you imagined.

The first task of the committee is to conduct a needs analysis. It should cover your company needs, your hardware needs, and your software needs.

The company needs analysis starts by identifying all the options the corporate system should provide. Then making the decision of which to implement first. This initial list should be short, three to five applications, and reflect both the corporate and individual's needs.

Once the initial applications have been chosen, how will you use them? Will your sales managers want to access mainframe data to aid them in setting individual and corporate sales goals? Will your telemarketers benefit from networking their computers and using a corporate-wide account management program? Should your sales reps send and receive sales and account information using corporate and personal databases? Can your shipping department decrease turnaround time if the orders come directly to the corporate warehouse from the field sales reps?

To view all your options, construct a matrix of users and databases for each application. List your initial applications, who will be using them, and on which machine. Decide where to store the data (PC or host database) depending on the user. The results will graphically demonstrate some of the functionality required of your system, like data transfer and synchronization, and the need for both PC and host databases. This information is crucial to correctly planning how your users will get the most out of your SFA system.

With the first applications committed to paper, your project planners must now decide which additional applications to add after the initial ones are working to everyone's satisfaction. This is important because you will be looking for software and hardware that are compatible with initial and future applications. Bad planning could lead to an expensive hardware or software upgrade as the system evolves. This step also helps identify the need for custom or pre-packaged software by indicating which future modules are required.

Many companies do an inadequate job of performing the corporate needs analysis. It is crucial, however, to the success of your sales force automation search. It pinpoints your applications, users, and databases, and initiates your hardware and software searches.

Software Needs

After defining the company needs, it's time to find the software. The software decision comes first so parameters are in place for the hardware decision. You will find that your software options come down to pre-packaged or custom systems.

Pre-packaged systems generally offer a cost savings in purchase price and are easy to install if they are compatible with your hardware and require little or no modification. Pre-packaged programs may severely limit your flexibility, however, when it comes to integrating the company's data presentation design or unique processing rules. New applications may also be difficult to integrate with the pre-packaged software as the system expands.

Many companies prefer to go with the flexibility and control of custom software. Custom software allows your programmers to develop applications specific to your company. These applications are easier to introduce to your users because the forms exactly fit their needs.

Good custom systems also provide communication between the host and PC databases for information exchange. This assures each user that the data they are using to answer customer questions is the most current information available.

Custom software eliminates many of the problems sometimes caused by adding new applications, too. The right software design should allow you to expand your sales force automation system in the future without any hardware or software compatibility problems.

Making the correct software decision is vital. Many software vendors qualify or eliminate themselves with a telephone call and brief questionnaire. This reduces your organization's sales force automation system evaluation cost and decreases your implementation time lag.

As with any system, a poor implementation of SFA software is expensive and of little benefit to the organization. A successfully implemented sales force automation system, however, benefits the entire company and pays large dividends. With the right software you should expect PC and host synchronization, company specific applications, user-friendly forms, and compatible software and hardware for all your applications. Don't settle for less!

Hardware Needs

After making the software decision, you can evaluate the hardware. Some companies already enjoying the benefits of sales force automation say their hardware decisions were the easiest part of the entire process[8]. By now you know how you will use your PCs and host(s) for each application. From there you may choose your hardware accordingly.

The host decision may be a little more involved than the PC one. You can measure the advantages and disadvantages of using an existing host or purchasing a new one by considering hardware and software compatibility, the increased data load, and your need for data centralization. Decide when the system will be accessed, for how long, and for what purposes. Also, how many concurrent users will it support? Much of the host hardware decision is mathematical and based on company expectations for performance.

The same companies which felt hardware decisions were the easiest to make also see laptops as the PC of the future[9]. The memory and storage capabilities are as extensive as a desktop PC, and the laptops allow the sales reps to work anywhere they have the time, need, or inclination to do so. Other considerations are memory and speed requirements, power sources, and screen types. Prices for the classes of computers are very similar and make your decision a relatively easy one.

Conclusion

It pays to perform a careful company needs analysis before you begin investigating sales force automation packages. Your planning group will be aware of exactly what to look for in a system and what questions to ask. It also aids the vendors you deal with by allowing them to demonstrate how their product meets your needs.

Sales force automation holds a great potential for many organizations, but only when correctly implemented and managed. Many companies have already installed sales force software systems and provided their sales representatives with computers. The returns have been remarkable, with full returns on investment often occurring during the first year of operation.

---

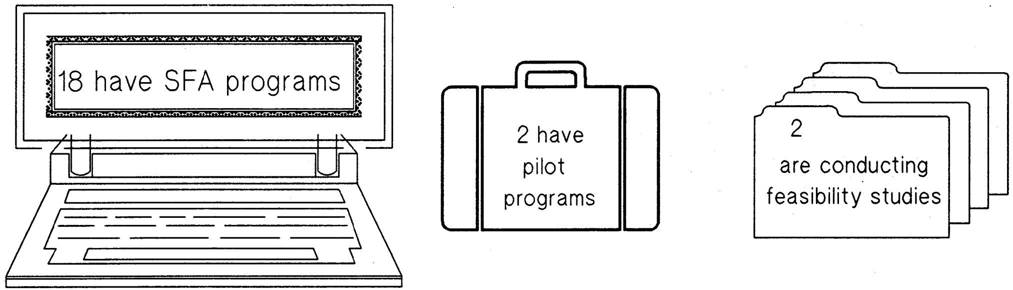
[8] Taylor, p. 74

[9] Taylor, p. 65

The ultimate decision of how and when to automate is up to each company. The ones who are the most well-informed and able to make the most knowledgeable decisions, however, will be the ones to use their sales force automation systems to their fullest potential and be the most pleased with the profitable results.

Why Automate?

22 Top U.S. Sales Forces



"Sales force automation is growing in a big way–at least among marketers rated by their peers to have the top sales forces in their respective industries. Eighteen of the top 24 companies (75%) participating in a Sales and Marketing Management survey have already put computers in the hands of their sales representatives. Two others have pilot programs and two others are conducting feasibility studies."

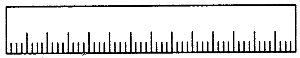
Reprinted by permission of Sales and Marketing Management Magazine, Copyright: April 1988.

Top Twenty Four

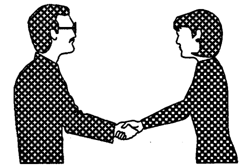
Armstrong World Industries
AT&T Microelectronics
Bergen-Brunswig
Black and Decker
Coca-Cola
Cooper Tire and Rubber
Du Pont Chemicals & Pigments
Eastman Kodak
Fleetwood Enterprises
Georgia-Pacific
Gillette
IBM
Inland Steel
Johnson & Johnson
Kimberly-Clark
Liz Claiborne
Merck & Co.
Motorola Communications
Northwestern Mutual
Owens-Corning Fiberglas
Pepsi-Cola
Textron
United Air Lines
Xerox

- Sales Force Automated
- Pilot and Feasibility Studies
- Not Sales Force Automated

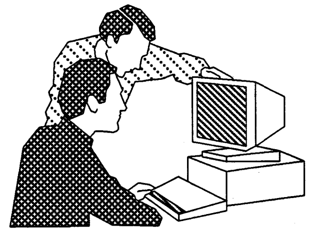
Five Major Advantages of SFA



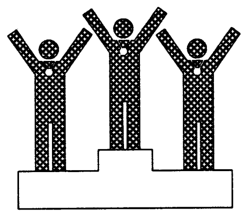
Shorter Sales Cycle



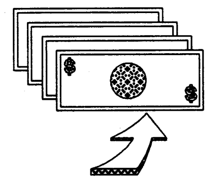
Increased Customer Service



Improved Internal Communications



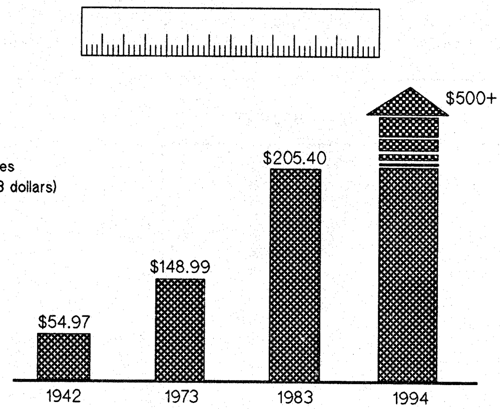
Higher Employee Morale



Increased Sales Productivity

Shorter Sales Cycle

Goods and Services
(All figures in 1983 dollars)



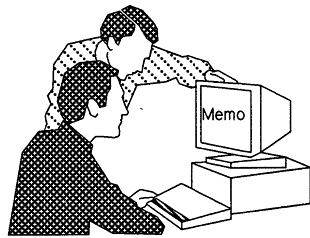
Average Cost Per Sales Call

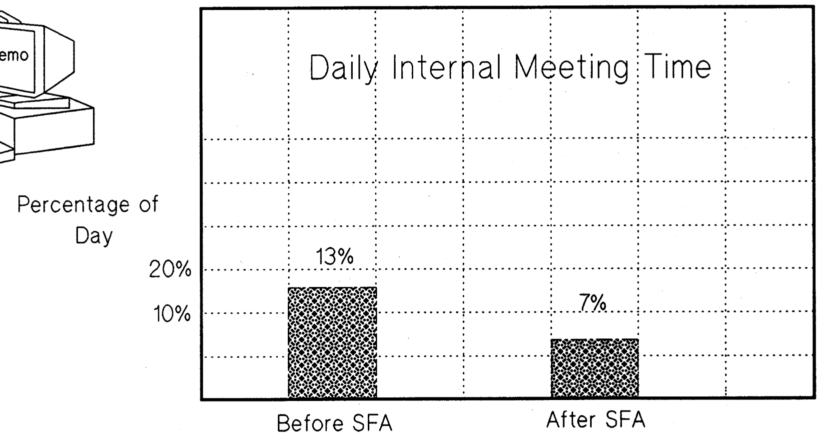
SOURCE: McGraw – Hill Research

"They [field sales representatives] are able to make better decisions because the PC delivers information more quickly, and this is especially important when it concerns the bottom line on costs and prices."

Stephen Korbecki, General Sales Manager, Inland Steel

Improved Internal Communications



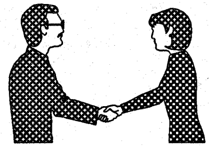


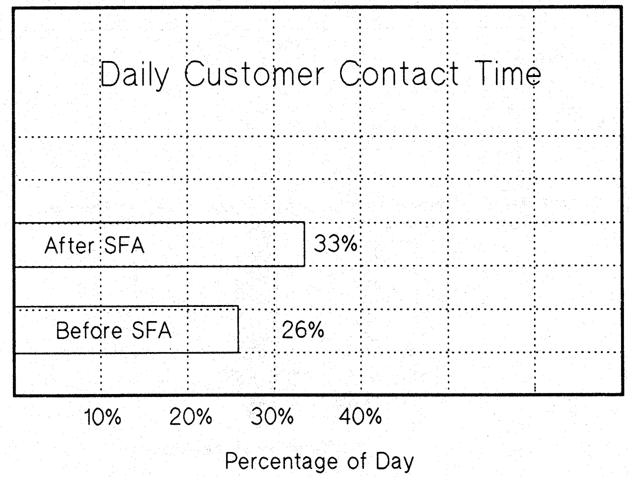
HP Sales Productivity Study

"When a district manager wants to send the details of a new promotion to the sales force, he puts the message in our mainframe. The next time a rep logs on, he's told there's a message waiting."

Rick Tancreto, Director of Sales Administration, Black and Decker

Increased Customer Service



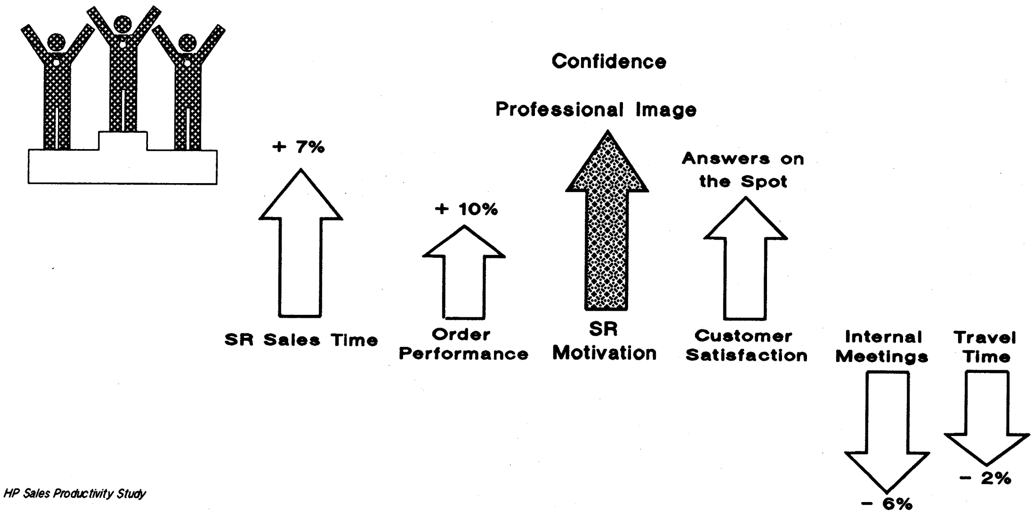


HP Sales Productivity Study

"Making salespeople more knowledgable about their accounts and the people they deal with is a critical issue if you want to become a preferred vendor."

Frederick H. Stephens Jr., Vice President of Business Relations, Gillette

Higher Employee Morale

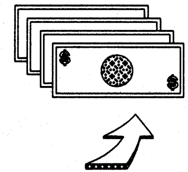


HP Sales Productivity Study

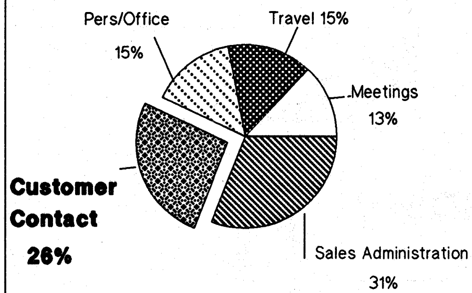
"The salespeople are so excited over having data in manageable form that they've taken to the program quite rapidly.

Kim Kelley, Pepsi-Cola

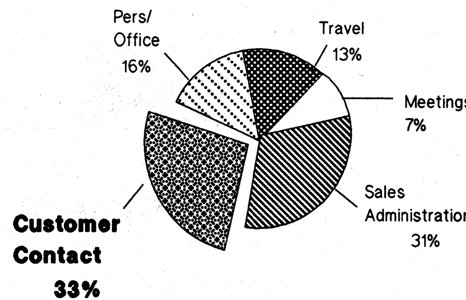
Increased Sales Productivity



Before SFA



After SFA



HP Sales Productivity Study

"A twenty-five percent increase in selling time produces, on average, an eight percent gain in orders, and, in some cases, as much as thirteen percent."

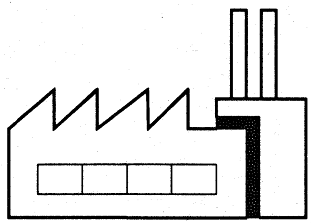
Benjamin J. Menold, Field Productivity Manager, Hewlett-Packard

Needs Analysis

Analyzing Sales Force Automation Needs

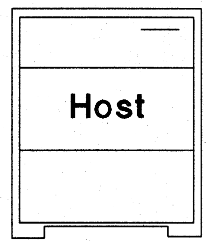
1

Company Needs



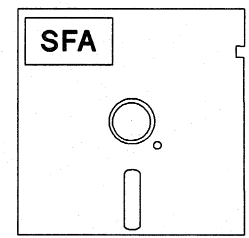
2

Hardware Needs



3

Software Needs



Company Needs

Company Structure and Resources

Headquarters

Personnel	Equipment
Management	PCs
Support	Host

Field Offices

Personnel

Sales Reps

Equipment

Some Desktop PCs

Telemarketing Office

Personnel

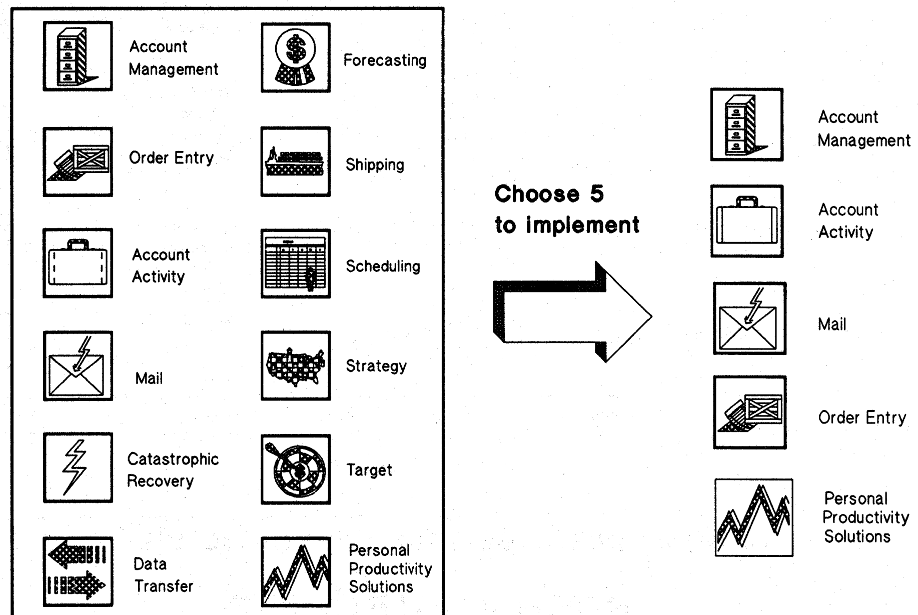
Telemarketers

Equipment

PCs

Company Needs

Sales Force Applications



Gateway Systems Corporation

Company Needs

Analysis of Application Users and Databases

Five Applications to be used	Who will input the data?	To which database?	Who will use the data?	From which database?	Who will analyze the data?	From which database?
Account Management	Telemarketing Sales Rep	Host	Management Sales Rep	PC, Host	Management	Host
Account Activity	Telemarketing Sales Rep	PC	Telemarketing Sales Rep	PC	Management	Host
Mail	Sales Rep Management Support Telemarketing	Host, PC	Sales Rep Management Support Telemarketing	PC, Host	NA	NA
Order Entry	Telemarketing Sales Rep	PC	Support	Host	Management	Host
Personal Productivity Solutions	Sales Rep	PC	Sales Rep	PC	Sales Rep	PC

Gateway Systems Corporation

Company Needs

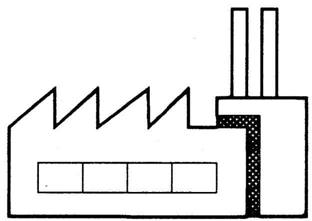
Conclusions

- Must be flexible enough to handle current and future applications
- Data will reside on PC and host
- Data will be transferred between PC and host
- Security system to screen out invalid users

Analyzing Sales Force Automation Needs

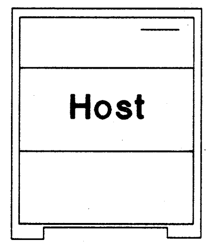
1

Company Needs



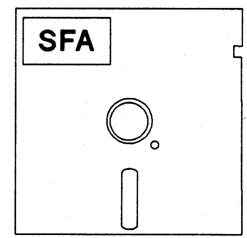
2

Hardware Needs



3

Software Needs



Hardware Needs

Personal Computers

- 286 or 386 microprocessor
- 20 or 40 megabyte disk
- 1200 or 2400 baud modem
- Battery operated or not
- Laptop or desktop

"The desktop ties the salesperson to a facility. You want him to be able to do what has to be done wherever he may be."

Ralph Kreidel, Owens-Corning Fiberglas

Hardware Needs

Host Computer

Considerations:

- Compatibility
- Increased data load
- Data centralization
- Existing or new host

"Buying hardware is probably one of the easiest choices you have to make."

Ron Greenwell, Motorola

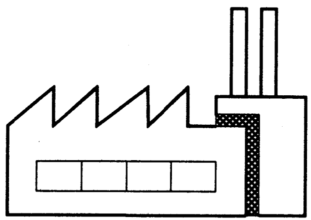
Hardware Needs

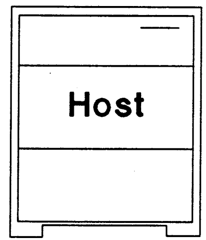
Conclusions

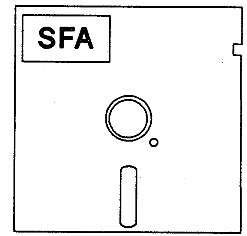
- Host and PC must be able to communicate
- Must be able to transfer information between them
- System must be flexible for future additions of new applications
- Need a security system
- Must support multiple users and multiple databases

Analyzing Sales Force Automation Needs

1	2	3
Company Needs	Hardware Needs	Software Needs







Gateway Systems Corporation

Software Options

	Stand-Alone PC vs.	**Fully Integrated**
Performance	Rep's Laptop or Desktop only	Will operate independently or communicate with other company computers
Investment	\$200-\$600 per PC	50%-150% of hardware cost

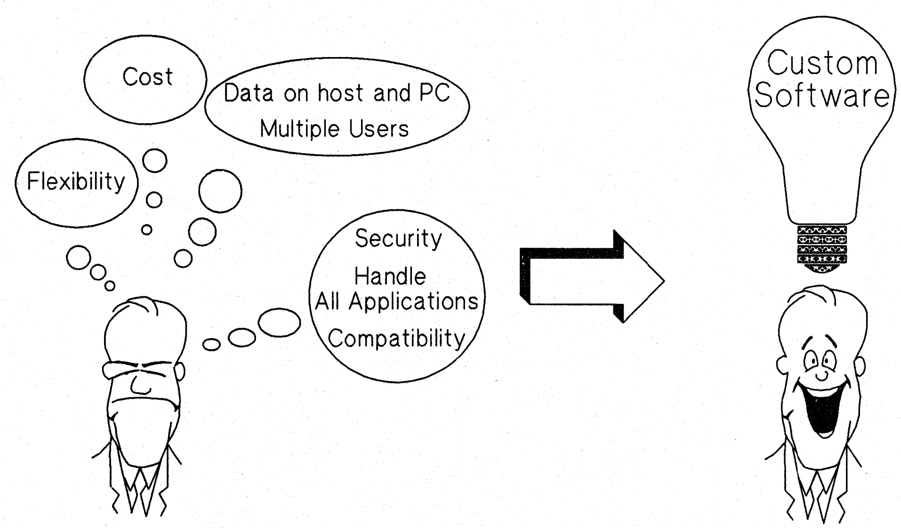
Software Needs

Turning Data into Functionality

Examples	Pre-packaged	Custom
Territory Management	✓	✓
Custom Profile	✓	✓
Call Planning	✓	✓
Call Reporting	✓	✓
Prospect Information	✓	✓
Contact Information	✓	✓
Account Activity	✓	✓
Product Information		✓
Competitive Information		✓
Order Information		✓
Sales Projections		✓
Calendar	✓	✓
Expense Reports		✓
Electronic Mail	✓	✓
Data Transfer (Error Detection and Correction)		✓
Sales Aids (Spreadsheet, Word Processing, etc)		✓
Sales Management Information		✓
Future Development (CD videotext, CBT, etc)		✓

Gateway Systems Corporation

Sales Force Automation Decision



Problems

Solution

The Desktop Computing Network

Edward S. Milbury

Gandalf Technologies Inc.
350 East Dundee Road, Suite 201
Wheeling, Illinois 60090
(312) 459-6630

INTEREX HP Users Conference
September 11-14, 1989

The Desktop Computing Network:

Bringing The Power of the PC and Networking Flexibility

to

Today's Knowledge Workforce

Today's organizations increasingly are made up of larger percentages of "knowledge workers," individuals who need access to personal productivity applications, the ability to communicate efficiently with other staff, the ability to access information bases - all at a cost which equates to the payback in increased performance.

There are a variety of traditional methods to deliver this support, including host-based networks, PCs, and PC/LANs. Unfortunately, none of these are ideal for all types of users for a number of reasons, which results in some potential users lacking access to various on-line system support.

The following discussion will address a new approach to satisfying these user requirements, which I will call the "Desktop Computing Network," an architectural concept with significant new benefits to both users and providers of departmental and enterprise-wide networking and processing services.

Historical Development:

The decade of the 1980's brought about a fundamental change in the computing architectures of large organizations, as chip and memory technology made it economically feasible to offload processing tasks from the traditional central mainframe and minicomputers to intelligent desktop workstations. Hardware cost/performance begot DOS applications software; availability of DOS software drove increased hardware sales! While this has been a happy series of events for both users and suppliers, a perplexing question remains: "Why hasn't this revolution been even more successful?"

Studies by FOCUS and others show that while 20 percent of our "knowledge workers" (people who would benefit from automated database access or information processing) now have some form of intelligent workstation on their desktop (about the same as those having host-based terminals), the remaining 60 percent have neither. This potential market continues to elude the PC market suppliers, leaving 6 out of 10 of our workforce still resorting exclusively to the telephone and the pencil!

Continuing Problems:

The factors slowing implementation of desktop processing are many, but certainly include:

- **Complexity - PCs still require enough technical knowhow (configuration, operating system commands, file backups, security) to discourage some users and increase training costs.**

- **Evolution - The speed of technology change and unclear upgrade paths increase risks and delays implementation decisions.**

- **Security - Release of mission-critical databases from centralized management is often inappropriate.**

- **Cost - While the decline in "relative" cost vs. performance over the past few years was dramatic, many desktops can still not justify the total cost of dedicated PC availability (equipment, peripherals, software, service, training).**

The Operating Environment:

Let's review the typical "operating environment" in most organizations today. While no two are alike, a walk-thru would likely find desktops with:

- **Non-intelligent terminals attached to one or more host-based systems using direct-connect or contention-based networks (switched or LAN based).**
- **PCs, both standalone and LAN networked. Possible gateway access to other resources.**
- **Single-purpose terminals (such as dedicated Word Processors).**
- **Desktops with no keyboard display, or desktops with devices for multiple applications.**

Staff with sufficient rank or job functions to clearly justify the hardware and support costs already have PCs, most of which are periodically upgraded to track the cost curve downward. Some other users obtain PC access as the result of "power-user" upgrades, and still others time-share machines using variations of "sneaker-net" schemes. The remainer have not yet justified access due to any number of the above delay factors, or because their need for usage is only occasional.

So with the dichotomy of acknowledged user need but yet insufficient purchase justification, is there anyone moving to solve the problem?

The answer appears to be yes, given recent availability of the "clustered processors" and the "computing network" and their increasing market acceptance. What are these products and how do they solve the user "access-gap"?

Clustered Processor:

The basic premise is that most computing resources, even microprocessor-based hardware, should be shared on a demand basis. Reduce the fixed cost per desktop to the minimum (essentially a keyboard and screen) and place all hardware, software and peripherals in a common pool! While not always the appropriate solution for a full time "power" user, the same total dollar expenditure for an organization may allow each individual user access to a much better processing resource for the partial daily period PC applications are typically used. Further, the administration can be offloaded from non-technical knowledge users to centralized support staff, giving greater overall efficiency and reliability. As only low-speed keystrokes and screen painting information flows between the centralized processing and the desktop, the need for high-speed LAN transport to every desk (with its associated cost) can be eliminated.

Implementation of this concept has appeared this year in a variety of solutions:

- Add-on processors and monitors for PC bus expansion slots now allow several users to share hard-disk files, software, and common PC hardware.

- Higher performance microprocessors and multi-user operating systems allow additional user sharing of available processor cycles.

- Add-on processors to central hosts allow occasional access to DOS applications from terminal devices.

The Computing Network:

The above solutions are now addressing the need to expand DOS application availability at the small workgroup, departmental level. Where users are in a confined work area and technical support is provided to each departmental grouping, the above solutions are growing in popularity.

The design architecture of Computing Networks, however, follows the basic principles of traditional LAN technology, with the exception that the physical collocation of processing modules, servers, gateways and bridges permits exceptional information transport rates among all subsystem elements. Rather than requiring serial bus transport over cable with its significant packetizing and control overhead, high-speed parallel busses virtually eliminate the negative effect of transport bus loading seen on large LAN networks.

Taking full advantage of sophisticated Network Operating Systems such as Novell's Advanced NetWare, multi-drive servers can provide gigabyte storage with full fault-protection redundancy, as well as bridge sub-LANs or individual intelligent workstations. New storage technology, such as CD-ROM readers, are also directly supported.

Computing Network Architecture:

The following elements would make up the typical Computing Network Architecture:

1. Desktop: Most popular ASCII-type display devices can be supported, including HP block-mode devices and Hercules graphics terminals (even IBM Display tubes and Macintosh machines using their terminal emulation modes). Local printing is available from the terminal if a printer interface is available. PCTerm style terminals can be used to get full 25-line display and PC-style keyboard. Desktops with PCs are also supported on the same system using terminal emulations (e.g. when users wish to access remote network files or applications).

2. Connectivity: Rather than wiring every desktop to support high-bandwidth transfers, a 19.2 Kbps path over existing twisted pair is normally used. Networking products such as short-haul modems, efficient multiplexers, and even voice/data line sharing equipment support inexpensive connectivity.

3. Communications Bus: This bus and its intelligent controller provide end-to-end connectivity paths upon user request. The same desktop device may select a path to one or more local host computers, to a wide-area network, or to an application processor within the Computing Network (the distance between the desktop and the processor is not restricted). Should all application processors be in use, the network provides prioritized queueing for the next available resource.

4. Application Processor: Once assigned, each user has dedicated use of the full power and memory of each processor. Menus allow application selection, and loading is automatic. All network servers and resources are available. Upon session completion, logout clears the memory and returns the processor to the pool for reassignment.

5. **Processing Bus:** All data transport between the processors, peripherals and servers is via high-speed parallel bussing, which increases performance over distributed cable-based LAN transport schemes. Systems include a master File Processor (which runs the operating system and File Server functions), Server disk(s), I/O Servers (for shared printers or modem pools), and Gateway servers. The Processing Bus allows bridging to other nodes or to network PCs. Peripherals such as Floppy Disk Drives, Tape Streamers (for automatic Hard Disk backup) and management consoles are included in the system design.

User Advantages:

This architecture can have benefits to both end users and system management:

Users with terminals can be given access to DOS applications without changing the desktop device (or adding a second device), and without rewiring for LAN transport speeds. With minimal training, users can menu-select the desired application for automatic loading, and terminate with a simple logout at completion.

Users are freed from the logistics of configuring desktop PCs, getting applications running, backing up data files and protecting data and software on floppies. They can select any available application package and share files with all other users.

It is likely users of Computing Networks were not able to previously justify a personal PC and the associated cost because of only occasional need or other reasons. However, the small incremental cost of each additional new user will now make access justification for all staff quite easy.

Management Advantages:

The management of PCs, PC-based databases, networking and overall costs is more and more becoming the concern of finance staff, IS management and operations staff (as opposed to only the local department people). Taking a top-down view, delivery of DOS applications via a Computing Network should be considered when one or more of the following circumstances exists:

- o Need for centralized control of PC applications software (e.g. revision level control, file exchange, document standardization, sharing of packages, virus prevention, etc.).
- o Access and data base security is essential (e.g. centralized management of who gets to what, and who can extract protected files).
- o Users have existing terminals (may be optimum for host-based applications).
- o Large population of occasional users (may be senior staff who don't want training, or those who want to do an occasional spreadsheet).
- o Remote access to LAN functionality needed (DOS application plus LAN networking with geographic limits).
- o Users have low PC literacy (unless usage will be frequent or continuous, reduction of users' need for literacy has high payback).
- o Shared access to database files (including CD-ROM)
- o Potential for mainframe offload (if the life of a host processor can be extended by offloading highly interactive users- e.g. E-MAIL - cost savings are likely).
- o Need for low per-user cost (it may be advantageous to spread available budget to more users with this low per-user cost architecture).

Application Examples:

This architecture will support the bulk of the current DOS applications available, limited only by the characteristics of the display device (e.g. character-mode tubes cannot display bit-mapped graphics). Popular Word Processing packages, spreadsheets (including Lotus 1-2-3), and database packages and other office-support software are highly compatible.

In addition, most well-behaved custom or vertical applications would be supported under the DOS environment.

Applications which are uniquely attractive include:

- LAN-based E-MAIL packages. Inexpensive application packages can now support both PCs and terminals in sophisticated E-MAIL networking (both local and multi-site, enterprise-wide). This provides the potential to offload this port-intensive application from mainframe systems.

- Distributed LAN Networking: PC networking applications which previously were restricted to departmental size environments can now support users at unrestricted distances between desktops. For example, inexpensive database sharing is now possible at low per-user cost regardless of physical location of user and processing node.

- CD-ROM Utility. By combining the centralized application processors with locally attached CD-ROM readers, easy remote access to the new CD-ROM information libraries is possible for large numbers of network users.

- Virus Isolation: As these networks can be configured with diskless workstations and terminals, it is possible to isolate all software loading to the control of the system administrator only. This can provide minimal exposure to external influences in critical systems.

The Innovation Evolution:

The Computing Network is a product that is available in the marketplace today, and vendors are continually enhancing the variety of environments which can be supported. They are expanding support to wider variety of existing terminals, providing configurations for small and large user populations, integrating popular operating systems, and offering flexible options for tailoring processing power and networking.

While the Computing Network doesn't solve every user or every network need, it appears to be a logical complement to extend the benefits of personal productivity software further down in the organization. It answers many of the problems which have prevented PCs from reaching more user desktops.

So it appears the trend toward universal access to personal productivity tools continues:

- The "Innovators" showed us the feasibility of desktop processing via microcomputers.
- "Early adopters" of microcomputer hardware created the market for DOS applications software.
- The proliferation of DOS applications attracted the technology "Followers" which drove PCs into mainstream business.
- Networking made the PC truly viable as a workgroup system.
- Clustered Processors are now reducing cost and complexity of departmental systems.
- Computing Networks are making universal access to DOS applications an economic reality in the large organization.

The 1990's will likely see Computing Networks architecture becoming a mainstream solution to large user groups needing both universal access to all resources from one user interface, including those applications previously requiring intelligent desktop workstations. While not the answer to every application, it will attract the attention of the large but yet still unserved market of potential DOS users.

The Desktop Computing Network:

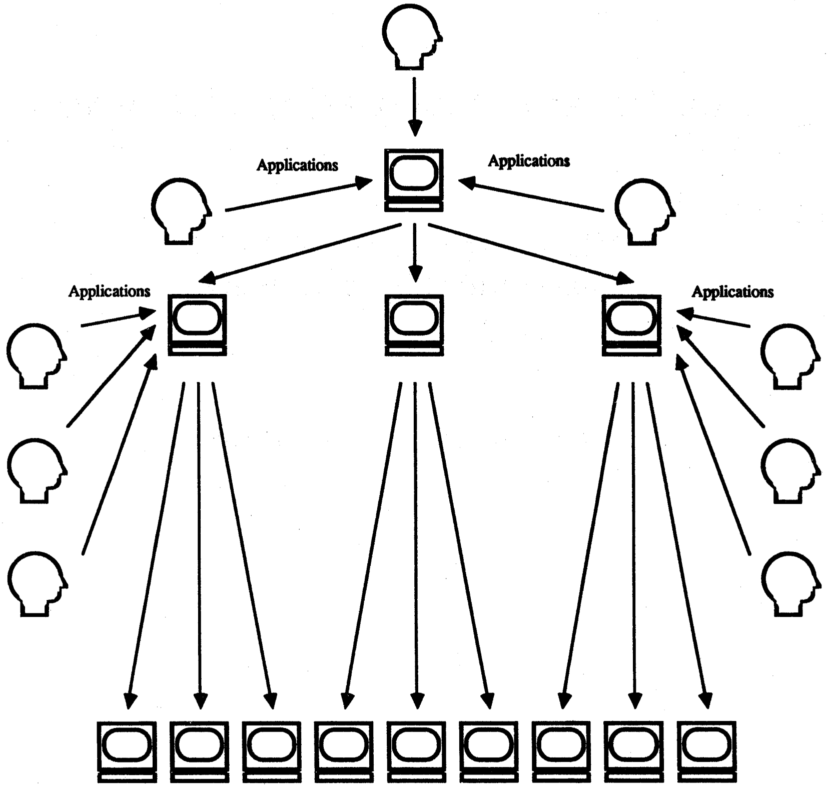
Bringing the Power of the PC and Networking Flexibility

to

Today's Knowledge Workforce

Edward S. Milbury

Gandalf Technologies Inc.



Historical Development

Factors Delaying PC Implementation

- o **Complexity**
  - Training
  - Non-Productive Time
  - Maintenance
  - Skill Retention
  - User Reluctance

- o **Evolution**
  - Technology Changes
  - System Maintenance
  - Upgrade Paths

- o **Security**
  - Database Management
  - Physical Security
  - Access Control

- o **Cost**
  - Hardware Cost Down
  - SkinWare Costs Up

The Operating Environment

- o Host-based Terminals
- o Stand-Alone PCs
- o Networked PCs/Departmental LANs
- o Dedicated Processors
- o Empty Desktops

Clustered Processors

Premise: Computing Resources Should Be Shared

Only User Keyboard/Display Is Dedicated

Processor/Software/Peripheral/
Networking Shared

Payback: Less Idle Resources

Centralized Support

Low-bandwidth Pipes to Desktop

Examples: PC Add-On Processors/Monitors

PC Time-Slice Operating Systems

Host Add-On Systems for DOS Applications

The Computing Network

Basic Architecture Principle:

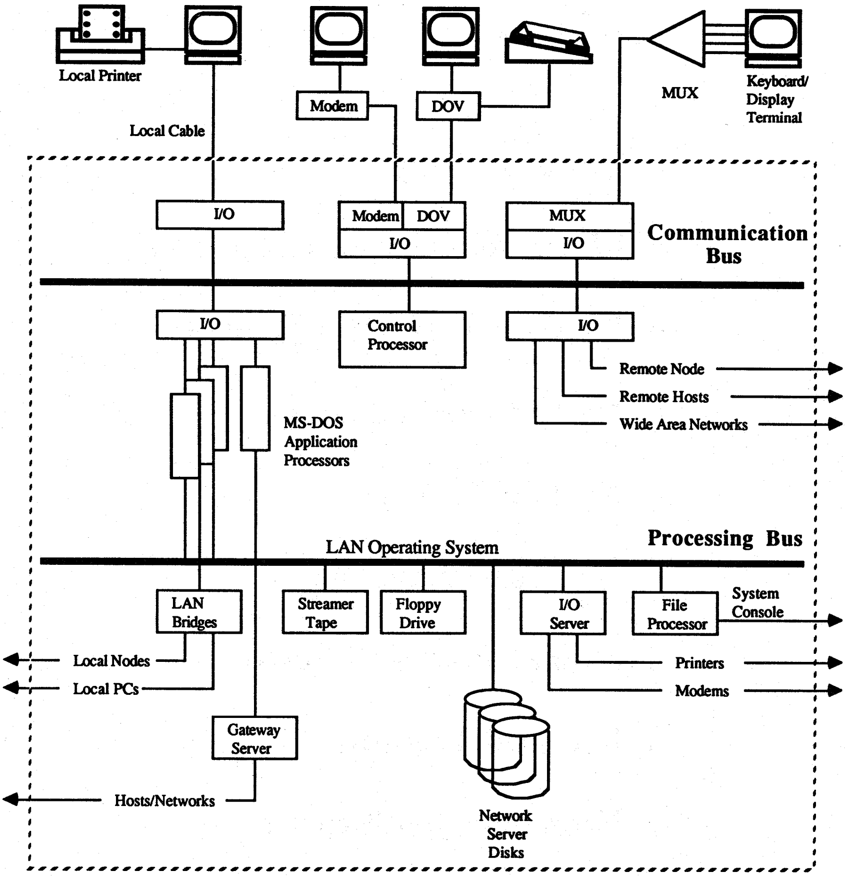
Collocation of Processors, Servers, Gateways, High Speed Buses, Memory, Management Tools

Node Access from Networked Terminals (Local, Short-Haul, Modem, MUX, Data-Over-Voice)

Shared Access/Dedicated Sessions

Full LAN Operating System Support (Management, Fault Tolerance, Diagnosis)

Central Management, Software Selection, Resource Allocation, and Security



Computing Network Node Architecture

User Advantages

- o Single Display
- o Menu Selection of Application
- o No Hardware Management
- o No Software Loading
- o No File Backups
- o No Disk Protection
- o Shared Files and Software
- o Reduced Training
- o Economic Justification

Management Advantages

- Centralized Application
- System Security
- Use of Existing Installed Equipment
- Broader Access to DOS Applications
- Simplified Training
- Sharing of Software, Files and Peripherals
- Low Per-User Cost

Application Examples

- o Defacto Standard Office Support (WP, Lotus 1-2-3, DBase IV)
- o Vertical Applications
- o E-MAIL
- o Distributed Access
- o CD-ROM Sharing
- o Virus Isolation

Innovation Evolution

- o Desktop Processing Feasibility
- o Large DOS Application market
- o Application availability drove business usage
- o Networking enabled group productivity
- o Clustered processors reduced desktop cost
- o COMPUTING NETWORKS are making true universal access to end user applications practical

Competitive Leverage: Using Information Services

Pete Koester
Hewlett-Packard Company
100 Mayfield Avenue
Mountain View, CA 94043

Introduction

Through the course of this talk, I will layout a series of concepts related to developing information services from a strategic perspective. We will begin with a discussion of why investing in information services is a low risk way of expanding your business. Then we will move into the best ways to start developing information services in your company, structurally. A quick tour of three illustrative business scenarios will follow in a discussion of product life cycle considerations. We will wrap up with a few salient points about the changing relationship between customers and vendors.

Some of these concepts carry over from traditional product marketing strategy; many of them do not. You, as the current and future champions of information services, speak with great accuracy about the needs of customers and their applications. However, from the perspective of upper management within your corporation, a broad strategic framework for selling the concept of information services is needed. This presentation is intended to introduce you to some of language and thinking that you can successfully use to develop information services in your company.

Information Services: A Definition

For the purposes of this presentation, an information service is defined as,"a value-added means of delivering information to end-users". The information may currently be unavailable to the end-user, but generally it is delivered in some form within your organization. The end-user may be the companies' ultimate customer or internal users that support the customers' business.

Information services which add value usually do so through a combination of a user interface software and electronic media.

Examples

	Software		Media
*	Full-Text Search	*	CD ROM
*	Field Search	*	Online via Modem
*	Hypertext	*	Floppy Diskette

Broadening the Base Business

We hear more and more about acquisitions and mergers in the press every day. Often these investment activities involve companies buying into areas unrelated to their current lines of business. Numerous studies show that such use of company resources, more often than not, are unsuccessfull and result in early divestment. Companies are generally more successful by growing through product line extension or growth into closely related businesses.

Examples

* Marriott offers lower cost, lower service hotels; a product line extension.
* American Express offers a charge card in addition to its credit card; entering a closely related business.

Information Services: Sticking to Your Knitting

The great lesson to learn from the unsuccessfull use of capital outside your base of business, is to find creative ways to grow within your area of expertise. Investment here is far more likely to yield long-term success and stave off competitors. By examining the critical information linkages in your company and developing better distribution methods and tools, you can make a big difference in the productivity of your workers and your customer's workers.

Why its Important

Studies of work patterns all show that skilled labor will become increasingly scarce. In addition, labor is the most difficult component of total costs to reduce for most industries. An important goal for any organization in the 1990's is efficient use of skilled labor.

The workforce increasingly will consist not of assemblers and office clerks, but of knowledge workers. Like traditional workers, these knowledge workers need tools appropriate to the task at hand. Office knowledge workers require access to company financial reports, inventory levels, sales, etc. Technical knowledge workers need data on operating requirements, workarounds and methodologies.

Not only internal users need this information. Increasingly, the difference between vendors of products will not be measured in millions of instructions per second or the lowest amount of breakage. There will be little or no differentiation in measures of this type as technology outruns traditional measures of performance. A possible means to distinguish your company's products, is to differentiate the "augmented product"; the entire collection of goods and services the customer perceives as the total product. A critical component of the augmented product is the quality of support materials you provide and the value they deliver to your customer. By marrying an appropriate interface to needed information for each set of knowledge workers, you can create a distinction in the market for your product.

Implementation at the Corporate Level

The cost and process of developing information services is a non-trivial task, typically requiring cooperation from several operating groups. Information streams within your company must be merged and standard formats for the information's presentation chosen. Synergies and the strategic nature of an investment in information services are most easily recognized at the corporate level. Additionally, a champion in the organization should be tapped for the purpose of developing the necessary linkages. The champion must have the support of upper management and have the ability to evaluate the importance of various information sources.

Differentiation at the Business Unit Level

Each strategic business unit(SBU) in your organization has the potential to use information services in a different way. One division's customers might be very familiar with online information systems. Another might have no experience at all with such systems and rely on a direct link to customer service representatives. For a financial application, a field oriented interface with numeric screening might be important. For another, keyword search with boolean operators might be appropriate. For packaging the product, the SBU which actually runs the business should make the critical decisions.

Prioritizing Your Efforts

One of the critical tasks you will face in getting an information product off the ground is determining what information should be included and in what order. Generally, you will have window in which information can be added after product release, but acceptance of your product, like any first impression, will ultimately determine its success. I recommend that you create a weighting scheme to help you determine the criticality and priority of information sources using internal and customer feedback. Rate the importance of the information against its relative accessibility for your product.

Information Priority Table (To be included in the presentation)

Capitalizing on your success

Once sources have been identified, and a plan for implementation of standards and systems for merging information streams developed, you must make critical product line decisions. These decisions should be made using good product line logic. Are you best to sell and package the information or should your distributor? Do you want to stabilize a sensitive part of your product line by making the information service a no charge feature of an existing product? What is the message you want to deliver to the market with the introduction of your information service?

The following three case studies will give us a chance at looking at the packaging of an information services product under different stages of the product life cycle.

CASE STUDIES (To be included in the presentation)

Case #1

Case #2

Case #3

Suprise! Its the Information Age

Companies that buy airliners now require that their specifications, parts lists, etc. be delivered on CD ROM optical discs. Why? Certainly not to make the manufacturer more efficient. They require this service because they have recognized the value of their own knowledge workers. How far away are your customers from requiring all the information about your product reside in an easily accessible electronic format? Would it make a difference right now if they could? Would it make a difference in your relationships with your customers if they could?

The Technology Trap

A common query I often hear about developing information services regards the next generation of technology. What about the next generation of optical disc technology? Expertise in developing information services is not dependent on the delivery platform, or media as it is sometimes called. A few years ago, a company that developed information services on floppy diskettes or Bernoulli cartridges was far ahead of the competition in developing the next generation of product. Today a company which invests in merging its information streams to deliver a CD ROM product will be in a much stronger relative position than its inactive competitors to produce a next generation of media product. Media is relatively cheap; developing the infrastructure is expensive. Do not get trapped into thinking that you will be investing in a process with a limited life-span based on the technology.

Throwing the Keys Away on Your Customers

The point of developing information services and positioning your company to deliver a higher value augmented product is simple. In the competitive marketplace, the nature of the relationship between the vendor and customer is changing. You, as the vendor will be counted on to help your customer be more competitive, cut costs and develop better methods. Companies which accomplish these goals will effectively, "lock in", their customers to an ongoing relationship. That is competitive leverage.

Optical Storage -- Finding a Home

Connie Doster
Hewlett-Packard
700 71st Ave.
Greeley, CO 80634

- Computer Industry Trends
- MO Technology - A Short Tutorial
- Positioning for Success
- The Main Applications

Good day. My name is Connie Doster. I represent Hewlett-Packard, and I am an Optical Product Marketing Manager at Greeley Storage Division in Greeley, Colorado.

The title of my talk today is "Optical Storage -- Finding a Home", to which many people may be tempted to sigh, "At last!"

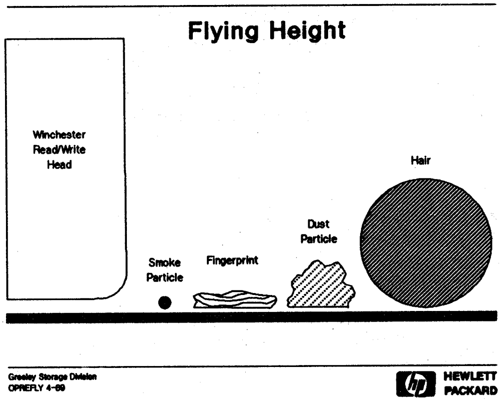
I'll begin with a short tutorial on Magneto-Optical Technology, since I believe that it will help you understand our conclusions about positioning. As part of the tutorial, I'll show some cost and performance comparisons with existing technologies. Next, I'll discuss some of the major trends in the computer industry which are relevant to the topic of optical storage. Then, I'll step through the positioning that HP is using with the introduction of optical products, and finally I'll highlight the main applications that Optical will be used for.

Unique Features of Optical recording

- Low cost/Mbyte
- High track density
- Greater flying height
- Removable media
- Durable Media

So, to get started, let's get into some of the details of optical recording. The most promising benefit of optical recording is a cost per megabyte much lower than today's magnetic disk drives. Fixed hard disks today cost from $30 to $15 per mbyte depending on the performance and capacity of the drive, while rewritable optical will cost between $10 and $1 per mbyte depending on whether you need 600 Mbytes or 60 Gbytes of storage capacity.

Because data is recorded via a laser, as opposed to a magnetic head, optical disks can store 10 to 20 times as much data as a magnetic disk on a given unit of area because of the higher track densities. In fact, optical track densities are roughly equivalent to their bit densities. This is due to the ability of the laser to focus down on a very small area on the media.



I'm going to switch slides here briefly to discuss the point about greater flying heights, and then I'll get back.

Unlike magnetic disc drives where the read/write head is flying several microns above the surface of the media, an optical drive's head resides up to four thousand times higher thus eliminating the possibility of a head crash. On the scale of this particular slide, an MO head flies about 7 or 8 feet off the surface of the disk.

Unique Features of Optical recording

- Low cost/Mbyte
- High track density
- Greater Flying height
- Removable media
- Durable Media

In turn, this greater flying height also allows the media to be removable, thus providing the opportunity for unlimited storage, security of data, and transportability of information.

The media itself is very durable, withstanding fingerprints and even minor scratches on its surface. They are not susceptible to damage from magnetic interference, radiation, heat, or common office mishandling. Writable media is further protected with a hard plastic cartridge. The projected life expectancy of the media is easily in excess of 10 years. If information needs to be stored that long (or longer) on 1/2" tapes, they must be retensioned periodically. A major advantage of optical media is that no user intervention is required to maintain the integrity of the data.

The combination of removability and durability in turn yields perhaps the most profound benefit of all, namely that it's possible to design a robotic mechanism which can automatically load and unload disk cartridges from a storage compartment to one or more optical disk drives. These devices, known as autochangers, or jukeboxes, will provide immense storage capacities at very affordable prices.

Jukeboxes are in use already in tape cartridge based configurations, as well as in large format WORM devices.

Rewriteable Optical Technologies
- MO
- Dye Polymer
- Phase Change

There are several types of rewriteable technologies under investigation; including, magneto-optical (MO), dye polymer, and phase change.

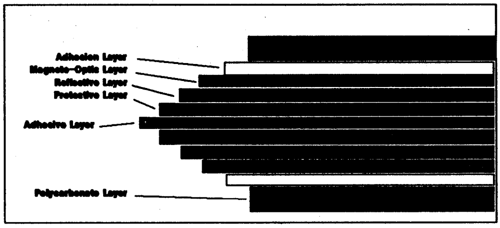
MO is considered to be the most advanced of the erasable techniques and the most reliable. MO media can be erased and written-over repeatedly, similar to hard discs. Retention in excess of 10 years is a fairly common specification. I'm going to come back to MO in more detail, but let me first cover the other two briefly.

Dye-Polymer technology uses a translucent plastic disk with a colored layer which absorbs heat from the drive's laser beam. A bump is created on the area heated by the laser. Reading a Dye-Polymer disk is similar to reading a CD-ROM -- the bumps reflect light differently than the flat areas in between. One significant drawback is that the media wears out after 1,000 to 10,000 write cycles.

Phase-Change technology uses a plastic disk with a special metal layer. Heat generated by the drive's laser changes the molecular structure of spots on the metal layer from an amorphous state to a crystalline state, and back again. To read, differences in the brightness of the reflected light from the amorphous spots and crystalline spots are detected. As with Dye-Polymer technology, the Phase-Change disk will not endure many rewrite cycles.

These technical issues may be restated in due time, but MO is clearly in a more advanced state at present.

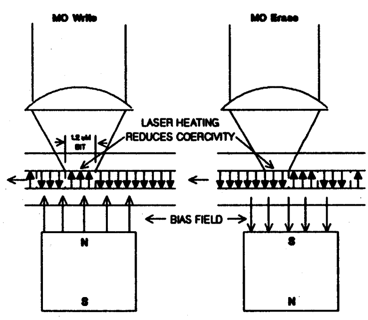
Magneto-Optical Media



Greeley Storage Division
OPRELAYR 4-89

HEWLETT PACKARD

I'm going to go into a little more detail on Magneto Optical Technology, variously known as Thermal Magneto Optical, or TMO, or simply MO. MO is basically a magnetic recording technique which uses optics for an assist. The actual recording layer which holds the recorded information has a very high magnetic resistance. As you can tell from the picture, the media is double sided, but must be physically turned over to write on the back side.



Greeley Storage Division
OPREWORK 4-89

HEWLETT PACKARD

The magnetic field required to alter the bit direction varies greatly with temperature. At room temperature, the magnetic field required presence of a normal magnetic field is to heat up a small spot on the surface disk to its "Curie point". At this high temperature, the film's magnetic properties change, allowing the drive magnet to alter the magnetic polarity of the bit. The direction of the magnet's polarity determines whether the bit is a 0 or a 1.

In today's optical disk products, the recording process requires two steps: an erase pass, and a write pass. The "erase pass" sets all the bits on the area to be written to a known state. To do this, the magnetic field is set to record all zeros, and the laser individually heats up each spot as the disk rotates. For the subsequent write pass, the magnetic field is set to record ones, and the laser is turned on as each appropriate area rotates under the beam; when those bits which are meant to be left as zeros pass under the beam, the laser is left off, and the bits are therefore left unchanged. The reason that two passes are required is that the writing magnet is large enough that its magnetic field can't be reversed in time for each bit that passes by -- its polarity is changed once per disk rotation. Therefore, random writes require 2 disk rotations to complete. Writes on new or erased disks, which are known by the host computer to contain all 0s, could be performed in one disk rotation. This two step process is automatically handled by the disk drive controller.

To read data, MO drives take advantage of a physical law known as the Kerr Effect, which basically says that the magnetic field affects the polarity of reflected light. You detect the light's polarity, and you know which way the magnetic field was set.

5.25" Rewriteable Optical Standards

Format Standardization Efforts:
- Physical Cartridge Dimensions
    - Cartridge format defined = ISO Draft Standard
    - Same as 5.25" WORM Cartridge
- Physical Media Format
    - Agreement reached on all technical issues for Continuous Composite format in ANSI (X3B11) and ISO committees (January '89)
    - Manufacturers supporting Continuous Composite:
    - Hewlett-Packard, Hitachi, Maxtor, Olympus/Ricoh, Phillps/DuPont Optical, Sony, 3M
- Manufacturers supporting Sampled Servo:
    - LMS, Pioneer
- Manufacturers supporting proprietary formats:
    - Canon, Maxtor

While I'm on the subject of MO technology, I'm going to discuss standardization, because I believe that the wide acceptance of standards will be a primary factor in determining the acceptance of optical technology in the industry.

With a solid standard, data interchange is made possible, and, because of high production quantities and competition, the media will cost less than non-standard media. A standard will also assure that drives will be available to read your data 10 years from now.

The "Continuous Composite" format which defines the tracking, sectoring and data format for MO is supported by most companies that are developing Rewritable-Optical products. Supporting companies include Hewlett-Packard, Maxtor, Olympus/Ricoh, Sony, and Hitachi. In addition, there are a number of media companies providing CC media, including Sony, 3M, Philips Dupont Optical, Daihatsu and others. In fact, all currently announced rewritable optical mechanisms either support Continuous Composite or are a proprietary format (such as Canon).

Some manufacturers also support "Sampled Servo," which is another format being considered for MO. We support the Continuous Composite format because it's easier to adapt for future improvement, it has the widest support, and will be in large scale production first.

Hewlett-Packard has actively participated in the ANSI 5 1/4" MO committee, and actively participates in the ISO committee, developing the standards along with the other recognized industry leaders.

So up to this point we've talked about the advantages of optical technology, why we've chosen Magneto-Optical over other technologies, a little about how MO works, and about format standards. Next I'd like to focus on performance and cost trade-offs in comparison with hard disks and tapes.

Performance Considerations
- Access Time - Lots of short transfers
- Track to track time - Data locality
- Transfer rate - large sequential file transfers

Disk performance is a complex issue, and many parameters affect it. Access time is often looked at, alone, as the most important performance point -- "How long does it take, on average, to access the file?" If you're shopping for an "access-intensive" system disk, this is an important specification. One consideration of MO technology todeay is that the average access time is 2-6 times longer than high performance hard disks. While hard disks are pushing under 20 milliseconds access, announced MO mechanisms have access times in the 40-100 ms range. This is because of the complexity and resulting mass of the optical heads.

Consider, also, the degree of "locality" that your files will have -- Your frequently-accessed files may be close to each other on the disk so that track-to-track seek performance is the key issue.

When retrieving large files, access time becomes less important -- once the drive has located the beginning of the file, transfer rate becomes the important performance point- "How fast can it transfer the data into computer memory?" When storing many files sequentially, access time is not an issue at all. Sustained throughput and frequency of media changes are the concerns. Transfer rate of MO optical disks is reasonably close to that of magnetic winchesters.

Comparative Performance Evaluation - Random Access

When comparing optical to a hard disk, performance is the major disadvantage. Seek time is fairly long for average transactions over the entire surface. However, short stroke seeks over a fairly small area are reasonably close to hard disk performance specifications. Looking at the throughput side of the performance equation, reading is accomplished about as fast as a hard disk, but writing is half as fast since two passes are required.

The slide here compares I/Os per second for various disk drives - a measure which takes into account both access time and throughput. The optical disk is shown as a range between the "best case" of reading in a highly localized data segment (short stroke read), and the "worst case" of writing in random locations (average write). You can see that with applications requiring high performance, the 795x disks are the clear winners. For access intensive applications (1 Kbyte records) the optical drive's worst case performance is about 5 times slower than the hard drive. But for average UNIX transactions (8 Kbyte records):

- Rewritable optical is 3 times slower than a hard disk, and 3 times faster than a floppy with worst case performance.

- Rewritable optical is only 20% slower than a hard disk with best case performance.

- Average write is 17% slower than average read. This is because most of the time is being occupied by seek time and overhead.

For very large files (100 Kbytes), such as images, the fast throughput helps reduce the peformance penalty. Here optical performance is approaching hard disk capabilities.

Comparative Performance Evaluation - Backup

When comparing optical to a tape drive, however, performance looks great. For fast backup with convenient file access, rewritable optical can't hardly be beaten. This slide shows both the quoted transfer rate for each product and also the effective "backup rate." This backup rate is calculated by adding the time required for rewinding, changing media, loading and unloading media each time a piece of media is filled up. 1/2" tapes boast fast transfer rates, but their limited capacity per reel, rewind and loading times cut the effective backup rate significantly. Here Rewritable-Optical is:

- 10 times faster than the 9144A 1/4" cartridge tape

- 3 times faster than 1600 bpi 1/2" tape

- About the same as 6250 bpi 1/2" tape.

Comparative Cost Evaluation - One Drive + Required Media

In an attempt to compare the differing technologies fairly, we have compared cost by adding up the whole solution -- the cost of the drive, along with as many pieces of media (or as many fixed drives) required to satisfy a particular storage need.

Here we find that rewritable optical is 1/3 to 1/30 the cost of hard disks, depending on capacity. At 650 Mbytes it's about 2.5 times more than 1/4" tape, and 10% less than hard disks, but at 3000 Mbytes, optical is 2 times more than tape and 10 times less expensive than hard disks. It's worth pointing out that on any of these technologies, it becomes rapidly impractical to backup more than 5000 megabytes unless you're using an autochanger.

Storage Choices

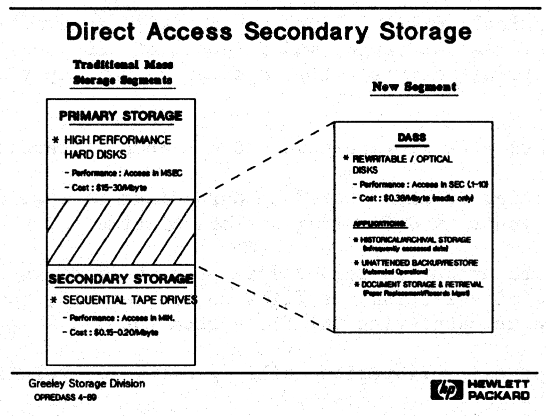
Rewritable-Optical storage is the choice for storing less frequently used data requiring low cost and high convenience.

Rewritable autochangers allow capabilities not available with any other technology. They offer vastly expanded capacity which can be used to keep infrequently-used information online. Convenient access is not sacrificed as it is with tape.

Hard disks win where high performance is the top priority. The need for short access times justifies the higher cost per megabyte.

WORM is the solution for legal archive security because you can't alter the data (without detection) once it's been written.

Tape drives win as the most cost-effective way to store sequential information- clearly the winners for simple hard disk backups. Because the per-unit media cost is low, tape will win when small quantity data-interchange is needed.



Positioning for Successs
- Primary Storage
- Secondary Storage
- Direct Access Secondary Storage

Given the performance and cost characteristics of this technology, as well as the previously unmet user needs discussed earlier, Hewlett-Packard believes that optical storage is a perfect fit for a new layer in the storage hierarchy.

Traditionally, mass storage solutions have fallen into one of two categories -- Primary and Secondary storage.

Primary storage is typically one or more fixed magnetic hard disks. It's fast, random-access storage with moderately high capacity -- used as the online system disk.

Primary storage is used to store applications, heavily accessed databases or files the user is currently working on and using extensively. It's also used in applications needing virtual memory, fast data processing, report generation, and complex calculations.

Secondary storage has consisted of one or more offline storage devices -- usually a 1/4" or 1/2" tape drive, or flexible disk on smaller systems. It's used primarily to backup the system disks. These devices are also used for logging transactions, distributing software, archiving historical data, and exchanging data between systems.

The main gaps between Primary and Secondary storage are access time and cost per megabyte. Average access time for hard disks is measured in tens of milliseconds. Access time for information on tapes is measured in tens of seconds for a mounted tape, minutes to hours for a tape in the library. The cost of magnetic disc storage is about $15 to $30, while tape storage costs 30 to 40 cents per megabyte.

Rewritable-Optical will fill this gap. With an average access time in the .1 to 10 second range, and a cost of 20 to 40 cents per megabyte, optical drives will create a new layer in the storage hierarchy.

Studies of hard disk storage have shown a couple trends:

- A large portion of data stored on hard disks is static -- a great deal of data isn't used for long periods of time

- Users are reluctant to remove data from their disks because of the inconvenience of selecting the files to be removed, and the added inconvenience of retrieving the files from tape.

Rewritable-Optical libraries will provide a solution to these inconveniences:

- Because of the tremendous cost advantage of Rewritable Optical disks over hard disks (pennies versus dollars per megabyte), data can be kept in a library where it will be inexpensive, yet still easy to retrieve.

- The job of managing hard disk space can be automated -- older, static files can be migrated to inexpensive storage while the system is running.

- Using file server technology, these benefits can be brought to PC and workstation networks.

- The autochanger will allow these applications to be "transparent" to the user- it will eliminate the need for user/operator intervention and will "manage" the disks.

Direct Access Secondary Storage, or DASS, will offer exciting new possibilities in three major areas -- archival storage, unattended backup, and document (image) storage. Whether the system uses an autochanger multi-disk system, or the stand-alone optical drive, the applications are basically the same. A small system may initially work effectively with a stand-alone drive and a few disk cartridges. A large system will need an autochanger, capable of keeping track of files on many disks and swapping disks automatically.

The following applications summarize the particular solutions offered by DASS.

Archival/Historical Data Storage

DASS could replace tape as a more convenient and reliable archival data storage medium. Whether data is archived for a week, a year, or ten years, it can be accessed and updated easily.

From the user's point of view, archived or historical data is as accessible as if it were still on the primary storage disk -- with somewhat slower response time.

DASS provides a way to compare and process information collected over a long period of time. Simulations could be performed and saved for further analysis. All revisions could be kept, allowing change control. Because the media is removable, applications requiring data security can become more convenient. Since the media is compact, sending information off-site for disaster recovery is not as cumbersome as it is for 1/2" tapes.

The 10-year minimum life of the disk means no worry over data safety. Users can retrieve the files any time, without operator intervention. Operators will not need to re-tension tapes to keep the archived files readable.

I'm reminded of a conversation I had recently with the head of a large data center. Like many MIS managers, he constantly has to delete old data sets to make room for newer ones. Inevitably, whenever he gets done restructuring his data bases by deleting year-old data, the president of the company comes by and wants to run a report comparing current information to older data to examine trends, whether it's inventory management, or payroll expenditures, or whatever. So the DP department has to roll off the current data, reload their archive tapes, run the reports, delete the old information, and reload their current information, and start all over again. With a DASS device, a computer system could potentially access the old information directly, because it would be cost effective to keep it online.

Unattended Backup

System backup and recovery can be fully automated with an optical library, eliminating the need and cost of an operator. Networks of PCs and workstations could be backed up at a central system overnight, between shifts, or as a continuous background activity.

The ability to perform backups without operator intervention will greatly increase the reliability of a system -- a large percentage of unplanned downtime is used to recover from human errors.

Document Storage

Optical libraries will be ideal for managing the "information explosion" of image/text documents, making access to information in the office, manufacturing, and lab environments convenient and efficient.

As document storage becomes more demanding, combining text with scanned images, storage requirements increase dramatically. Printing and scanning devices with higher resolution and color will add to the storage demands. A printed page of text takes about 2K bytes. A scanned image might run a megabyte or so. A full-color image with reasonable resolution can take upwards of 20 megabytes. As voice data becomes more common in the office environment, storage requirements will take yet another quantum leap.

Insurance records can be stored locally, available for convenient retrieval. Technical writers could have a huge historical library of text and graphics, ready to incorporate into new documents.

Optical storage, because of its unique attributes, will quickly find a home in the computer industry as a low cost way to keep old information online. Its superior convenience features compared to ape will justify its use in many applications, especially historical archives, unattended backup, and document management.

It is not a replacement technology. Rather, it is an enabling technology, which allows users to keep more of their information online, available, and therefore useful.

Optical Storage -- Finding a Home

- Computer Industry Trends
- MO Technology - A Short Tutorial
- Positioning for Success
- the Main Applications

In closing, I hope I've convinced you that optical storage has indeed found a home. It is no longer an answer looking for a question; rather it's an excellent match for many of the major trends taking place in the computer industry today.

CPU, Memory, or Disc?
The Basics of Performance Analysis

Jason M. Goertz
Mattedor Computer Services
Bellevue, Washington
206-746-0212

Introduction

Virtually every HP 3000 (and its associated System Manager) has experienced either long term or temporary problems with system performance. When this happens, what can be done? Why does it occur at all? How can we as system managers determine the cause and fix it?

This paper is intended for people who are novices at performance analysis. I will attempt to describe the basic theory of this sometimes arcane field in such a way that a beginner can understand the fundamental principals involved. This will lay the groundwork to understanding the HP 3000 performance characteristics and, more importantly, characterize *your* HP 3000 performance. Along the way, we will visit some of the tools that are available to make this job easier. I'll discuss how to *interpret* what these tools tell you.

The Three Basic Resources

The very first thing to realize when discussing performance analysis is that the computer system is a finite resource with limited physical properties. These properties are dictated by the laws of physics, which at this writing are yet to be changed or transmuted. In other words, it is important to realize that the computer system has a finite capacity for work, and the users' *expectations* must be set with this fact firmly in mind. I have gone into more than one performance analysis only to discover that the thing I had to adjust was not the computer or the application, but the user. Once the limitations of the computer are realized and accepted, the process of analyzing and adjusting the performance of the system is much easier to accomplish.

The computer system consists of three primary resources: **CPU**, **Memory**, and **DISC IO**. For the purposes of this discussion, the three will be defined as follows:

CPU - Central Processing Unit cycles. This is the part of the computer that actually performs arithmetic and logical operations. It is here that calculations are performed, data is edited, etc. Useful work is accomplished in this portion of the system.

Memory - Real semiconductor RAM memory. It is here that the data and code being used by the programs is stored. Due to the manner in which all computers in use today operate, data and code **must** be in real memory in order to be acted upon by the CPU. Note that we are **not** discussing the speed at which the memory is accessed, an important factor when discussing microcomputers. When talking about performance of an HP 3000 computer, the **size** of the memory and how it is managed is much more important.

DISC IO - This represents the speed at which data can be brought into (and written out of) real memory. This is very important, since most data processing commonly performed today (especially on HP 3000's) depends heavily upon disc storage technology. Several things contribute to this figure, the most common being the number of effective disc IO's per second possible by the system.

These three resources are the ones which every application in the world must have available in order to run. Each application uses them in varying amounts, adding to the complication of analyzing their interactions. For example, a CAD (Computer Aided Design) application will generally use a great deal of CPU, since many calculations must be performed to do the graphics manipulations common in these applications. Likewise, a great deal of memory is used by CAD. A "typical" HP 3000 business application tends to go heavier on the disc IO than on the memory, since most of what these applications do is store and report data to and from disc structures. It is important to not only understand what the computer is capable of, but exactly what the software is requiring of the hardware in order to accomplish the goals set forth by the users.

To make analyzing the performance of these three resources a bit more difficult is the fact that the three resources are interdependent. It takes CPU to perform a disc IO, and it takes disc IO to manage the real memory resource. It is precisely this interdependence that takes performance analysis from a science to somewhat of an art. In mathematical terms, there are more variables than equations, which makes the problem impossible to solve. In other words, it is often difficult or futile to exactly quantify the interdependencies of the three resources (equations), which makes solving for the unknown quantity (variable) sometimes a matter of wetting the finger and sticking it into the wind. Before delving into the technicalities of CPU, Memory and Disc, it is important to keep in mind that "bad performance" is a very subjective thing. Let's digress for a moment and discuss this fact.

Perceptions

Each of the three primary system resources is finite in quantity. Figure 1 shows this relationship. Very simply put, performance problems arise when the software load on the system requires more of one of the resources than is available. It is important to note that on modern multi-processing systems an additional dimension must be discussed, that being time. In other words, a given amount of work can always be performed by the system, but what differentiates a "fast" system and a "slow" system is the time frame in which that workload is performed. Dayend processing that takes 14 hours is unacceptable. The time element also varies by the business demands of the company. For example, monthend processing that takes 4 days may be perfectly acceptable to some businesses, but not others.

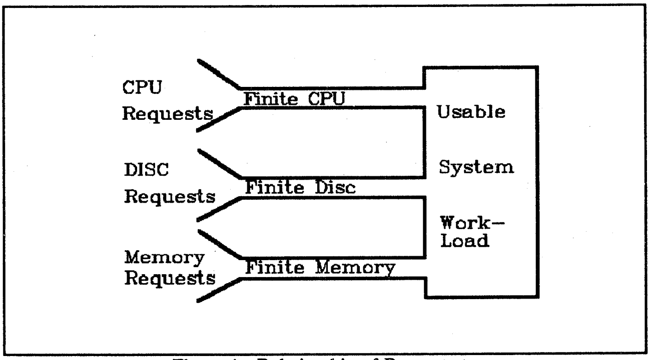


Figure 1. Relationship of Requests to Availability of System Resources.

All of this leads up to the fact that the definition of "good" or "bad" system performance ultimately relies upon factors totally outside the realm of the machine and the software. With this in mind, I would like to offer the following as the definition of "bad performance":

> *Poor system performance is when a computer system does not perform its workload in the time frame that its human operators and managers perceive it should.*

A key word in this definition is *perceive*. Many times bad performance is only perceived, based upon an individual's *expectations*. The following scenario will show how this can occur.

XYZ corporation decides that it needs an HP 3000 to do its business computing. It buys one, and in a very wise move, decides to bring the applications online one at a time, spreading out the work necessary by the DP staff and lessening the impact on the end user community. The machine is brought in and the first application, Accounts Receivable, is implemented. (XYZ corp has a lot of problems collecting bills, and this was decided to be the highest priority application). The two people who do data entry in AR are very happy with the system, and since they and the DP staff are the only ones on the machine, response time is instantaneous. The AR clerks go merrily on with their jobs, accepting one/tenth of a second response time as "normal".

As the months pass, the other two applications are brought up on the same day. The first is the warehousing application, which involves 70 terminals scattered among 3 sites. Additionally, Accounts Payable is brought up on the same day, again with two clerks, who sit right next to the two AR clerks. When all this is brought up, the response time goes to 2 seconds. For this many terminals, this is a very good figure. The DP staff was conscious of performance from the outset, and worked hard at making sure the application was efficient. The AP clerks see the two second response time and accept this as "normal". However, the AR clerks, who have lived with no response time for months, find this new creeping pace of the machine almost unbearable. The four clerks sit side by side, see the exact same response time, and two find it acceptable and two do not.

The key here is that the AR clerks' **expectations** were set early on in the process. They were still able to get their work done, but the "slow" response times sent them up a wall.

Going a bit further with this scenario, the system performed fine until the first crunch hit, and every one of those 70 terminals was banging away trying to get product out the door. Response times went up to 15 seconds. The system got so bogged down, in fact, that shipments from the warehouses couldn't get out in time, missing shipping windows necessary for the marketability of the product. Because of this, hundreds of thousands of dollars of product were of less use, or no use, to XYZ's customers. In this case, the machine was not performing well enough to meet the business needs of XYZ. Notice that this occurs only rarely, just during a crunch.

In the second case, the machine truly does not perform as well as it **must** to meet the needs of the company. It is possible to have *perceived* performance problems as well as *actual* performance problems.

The bottom line of this whole discussion is that acceptable system performance must be defined before successful performance analysis/tuning can be done. This definition of acceptable system performance must consist of both a quantification of the workload desired and the time frame in which it must be performed by the system. The time frame used for this quantification can vary depending upon the type of workload. Batch work typically is measured in a certain number of transactions or reports running in a given amount of wall time. For example, "all 7 dayend jobs must run in 10 hours or less". Online performance is usually measured in transactions per hour, or specified with a certain response time.[1] A good definition of online performance might be "the

---

[1]. **Response time** - The time from the completion of one terminal read until the machine is ready to accept more input. For example, the time from when the ENTER key is pressed on a VPLUS screen until the screen is cleared and another screen of data can be keyed.

application must be able to process 2000 transactions an hour" or "the application must process a transaction with half second response time". Using this terminology, it is possible to establish a performance goal for an application quite nicely.

Steps in Performance Analysis

Now that we have discussed the objective versus subjective aspects of performance analysis, we will now get to the meat of the subject and discuss how performance analysis is done.

The analysis of an HP 3000's performance typically follows some set steps. While each analysis is a bit different, the following description usually applies to them all to some degree or other:

1. The performance goal is set or specific questions regarding system performance are asked.

2. The current performance level of the system is measured.

3. If the current level does not match the desired level (a forgone conclusion, or the analysis wouldn't have been needed to begin with), the performance problem is found and adjusted.

4. Steps 2 and 3 are repeated until the performance is at an acceptable level. If the level cannot be modified enough, then 1 (the expectation of the users) is adjusted.

Virtually all the papers and discussions of performance done up to now have always centered on 2 and 3. This paper will focus on them as well, but the importance of doing steps 1 and 4 cannot be overlooked. As with any endeavor, a goal must be defined, or there is nothing to shoot for. In addition, it is *critical* to realize that the analysis may come to an impasse, and the expectation must be reset. Sometimes this is simply caused by the fact that the system cannot physically do what is needed or expected. Sometimes it is caused by the user not having deep enough pockets to invest in the hardware and/or software to solve the problem. In either case, the performance analysis is a failure if 4 is not executed when necessary.

The remainder of the paper will center on the steps 2 and 3 above, with emphasis being given on the basics of these points. The reader is urged to find other sources, including qualified performance consultants, if more detail is desired. It is NOT the intent of this paper to train a performance consultant. What is the intent is to make the reader familiar enough with the subject to at least ask the proper questions.

Measurement

The first thing you must do to determine if the system is operating with performance limits is to measure its current performance. The way the system is measured is dictated by the goal of the performance. Simply put, the best way to determine if the system is performing well is to set up a standard set of tests which can be repeated, and measure the time they take to complete. If the performance goal is to have the dayend batch jobs run in 4 hours, then data must be gathered on that mix of jobs and see if they run in 4 hours. This may seem trivial, but this escapes many people. Such a test is generally called a **benchmark**[2].

The steps in performing a benchmark are as follows:

---

[2]This term probably derives from the days with technicians (working at a "bench") would write down figures on the bench ("mark" the "bench") derived from measurements.

1. Determine what job mix will measure the desired performance characteristics. If online response time is the hallmark of the desired performance goal, setting up a batch job is not the correct thing to do. Jobs are usually easier to measure than online performance, since they can be run more easily.

2. Make sure the benchmark is **repeatable**. While virtually no benchmark is absolutely repeatable, some margin for error is generally acceptable. Repeatability implies that all the variables are held constant. For example, no other work can be on the system, as this will take system resources from the benchmark. The data should be the same for each run, and so on.

3. Perform the benchmark some number of times. Good scientific method dictates that at least three runs be done and averaged. This is not always possible, so do as many as is feasible. For something like the dayend processing example above, each night's run could be considered a benchmark.

4. Keep careful track of the results. This may be done by simply saving the $STDLIST's from the runs. In any event, keep a record of the pertinent data of each run.

This benchmark setup should be saved so that when step 3 of the performance analysis is done, the benchmark can be repeated to determine if any tweaking helped.

As you may be guessing by now, doing performance analysis in this rigorous manner is time consuming. In most cases, the current work load that your system is already performing serves as the benchmark. Formal benchmarks are usually only done under formal circumstances, such as when a vendor is trying to sell more hardware, or a comparison is being made for others to study. The type of benchmarks we as system managers are interested in do not need to be designed and built to the smallest degree.

As mentioned, benchmarking batch jobs is relatively easy. Just stream the job, and check the elapsed time. If other factors of the application's performance must be measured (ie, number of disc io's, etc) then this gets a bit more complex, and will be covered in a minute. To benchmark online performance, however, is a bit trickier. To repeatedly benchmark a set of online transactions is difficult, since entering them in a repeatable fashion requires a human being (or worse, human beings) to behave in a perfectly repeatable fashion. Some tools do exist to *simulate* human beings entering data into terminals. HP for years has used one called TEPE (Terminal Emulator and Performance Evaluator). TEPE was very user hostile and setting up the scripts was very difficult. A newer tool which has come out of HP's Roseville division is called Wrangler, and is apparently easier to use and setup. I have not used it, so I can't say for sure. In any event, neither of these tools is available to the general public (read: you or I) and won't be considered here. Telemon, the company who builds the Type Ahead Engine has a derivative of the TAE called AUTOMAN. It works with software on the HP 3000 and can record a set of keystrokes and play them back with the same pauses (and mistakes) as when the transactions were entered. I have used this on one occasion, and it works quite well. It is not trivial to set up (as with most sophisticated devices) and uses two ports on the machine. One port for the normal connection to the system, the other for the software running to gather the logfile of keystrokes. Most users of small shops cannot afford this. However, it is really the only option available to the normal user, and definitely recommended to those who are serious about benchmarking online performance.

In summary, there really isn't a good way to measure online response. The best way is to put code into the application to take timings of how long different events take. Most application programmers do not think about performance, and so this is almost never done. A product by Mattedor Computer Services of Bellevue, Washington called IOSPY will measure response times by looking at the IO entries in the system. This generally will suffice, but instrumenting the application is a superior method if it can be accomplished.

Performance Tools and Peeking at the System

If the application does not perform as desired and tuning is necessary, the obvious question is "What do I adjust?". This question is not easily answered. Since software is, by its nature, ephemeral, it cannot be examined by our five senses. We must extend those senses inside the machine to get some idea of what is going on inside. We do this by using any one of a number of tools that exist for this purpose.

Most of the tools that exists on the HP 3000 for general performance analysis rely heavily upon a portion of MPE called the Measurement Interface, or MI for short. This consists of a set of procedures and inline code buried at various portions of the system software. The procedures are called to arm the interface, and various data segments are built which the inline code uses as counters and timers. As an event occurs, the code in MPE dealing with that events goes out to these various data segments and increments the proper counter to reflect that event completing. If a timer is required for that event, then the time is extracted from the system and updated. The program using this data is responsible for extracting this collected data from the MI tables, processing it and formatting it as it wishes, then presenting it to the user. The MI collects a great deal of data on various aspects of the system, the primary resources being (you guessed it) CPU, Memory and Disc.

There are several tools which allow the monitoring of these resources. The original one is called OPT (for Online Performance Tool) from HP. It has been around for about 10 years. Other third party vendors have also built their own OPT-like tools. Carolian has SYSVIEW, which offers the same basic functionality as OPT but without the nice graphics interface. Another is PROBE/3000[3] from Strategic Systems Incorporated of Seattle. It offers all of the important functionality of OPT plus a few ideas in the data presentation that OPT does not have. The newest addition to this line of products is called CIA (for CPU/IO Analyzer) from Facer Information Design in Australia, marketed in the US by Tres Associates of Austin Texas. The screens in CIA are not as dense and complicated as those in the other two products, and has a good presentation of the fundamental performance data needed for system analysis. The screen layouts used in the remainder of the paper are recreations of those from CIA.

While I will at times in this paper be comparing and contrasting the products, this should not be construed as a product review or endorsement for any of the four. All have their advantages and disadvantages, just like any class of products. If your are interested in more information on the three, contact the individual vendors for demos. I chose CIA for this paper simply because I felt the screens lend themselves better to this particular discussion than the others.

All of these tools offer display **contexts** which present data relating to one of the three resources. OPT, PROBE and SYSVIEW separate the CPU, Memory and IO contexts to different screens. Thus, the CPU context would offer data regarding the CPU usage, Memory context for memory, etc. All the tools also have a global screen, which combines the some data from the other screens. Refer to Figure 2 for a sample CIA Initial screen. This is always the first one presented to the user, to allow him or her an idea of what is going on with the entire system. CIA has one unique feature which can be seen on this screen. The global CPU and Disc activity is always presented at the top of the screen, allowing a better comparison between the individual process data and the overall performance of the system.

It is important to realize that using these tools involves two things. First, you must understand what the data displayed means. Second, you must know how to *interpret* the data. This is the more difficult part. As mentioned before, since the interactions of the various parts of the system are difficult or impossible to precisely quantify, it takes some experience and knowledge to know whether the data displayed is significant or not. I will attempt to give some guidelines. In general,

---

[3]. OPT, copyright Hewlett-Packard Co. SYSVIEW, copyright Carolian. PROBE, copyright Strategic Systems Inc. CIA, copyright Facer Information Design.

the documentation accompanying the tools gives guidelines for their interpretation.

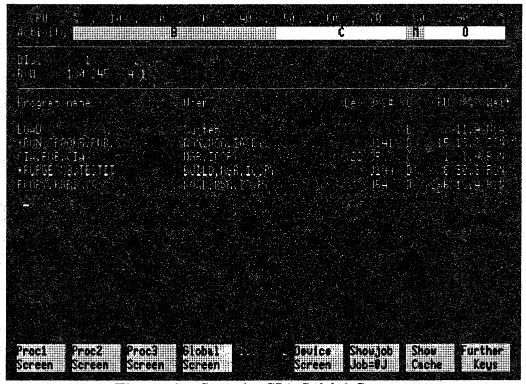


Figure 2. Sample CIA Initial Screen.

CPU

The CPU bar represents a breakdown of how the CPU is spending its time. Essentially, the CPU spends some of its time doing overhead functions, such as dispatching processes, managing memory, completing IO's, etc. The goal is to maximize the time the CPU spends on user processes (useful work as far as we're concerned) and minimize the time it spends doing overhead. The unit of measurements is always expressed as percentages, with each type of function being assigned its own abbreviation. Refer again to the top of Figure 2 for a sample CPU bar. The abbreviations and the definitions for the various CPU states are as follows:

B - CPU Busy. Percentage of time spent doing useful work for processes.

I - CPU Idle. Percentage of time the CPU has nothing to do.

P - CPU Paused for Disc IO. Percentage of time the CPU was paused waiting for disc IO to complete, and nothing else was ready to run.

M - CPU spent on Memory Management activity. Percentage of time the CPU was swapping in processes, performing global and local garbage collection, etc.

O - CPU spent on overhead functions. Due to the way the system software is structured, this number is not directly measurable. In reality, this number is derived by subtracting the sum of the others from 100.

V - CPU time spent waiting for Memory Management (Virtual) IO.

The MI actually measures many more separate types of CPU activity. The above list represents a composite, and are the activities normally displayed on the global screen. The ones listed above are usually all that is really necessary to track, unless there is a very odd or specific problem.

The goal is to have CPU busy be as high as possible, with the others being as low as possible. Any amount of CPU on Memory Manager usually indicates some sort of memory problems, although this is easier to determine from the other contexts. A high overhead figure can indicate a great deal of disc caching, disc IO or terminal IO activity. Paused for disc will be high on a disc bound system. With disc caching, it is rare to see paused for disc. Idle time, of course, represents excess CPU capacity.

In summary, the CPU figures break down how the CPU spends its time. High percentages of either Memory Management or Paused for disc can indicate problems in these areas. If the Busy figure is high and the workload is still not being accomplished in the time frame necessary, then faster or multiple CPU's are warranted.

Memory

There are three different aspects of memory that can be observed with a OPT and PROBE. The first is how much memory is present in the system, and how much of that memory is occupied by what types of objects. The second is a breakdown of the types and number of Memory Management events are executed by the system. The third is to compare the ratio of launches to swaps. All of this data is available on the Memory contexts in PROBE, OPT and SYSVIEW.

Examination of the types of objects and how much of memory they take up is useful in getting an idea of what type of object is being accessed the most. If most of memory is filled with code segments, then it is these code segments that are being accessed the most. Generally, the objects are broken down as follows:

Resident MPE - Memory taken up by MPE data structures (tables) and code segments which must be present in memory at all times. This can be reduced by reducing table sizes. At one time this took up a significant portion of memory. With the larger memory sizes we have today (4 Megabytes and above), this is of less concern.

Code Segments - Objects which contain executable machine code. Since code is sharable on the HP 3000, this usually is not a great concern.

Stacks - Data segments used as stacks by the system. These are specially formatted areas, one assigned to each process.

Data Segments - Data segments which are not specially formatted. Many things can be contained in these XDS's (eXtra Data Segments), such as non-resident system tables, file system control blocks, Image control blocks, etc.

Disc Cache Domains - Areas of memory used by disc caching as its buffers.

The useful thing to be gleaned from this display is what type of object dominates memory, and thus what part of the system is being used the most. Usually this will be the cache domains, as they tend to fill all available memory.

To understand the second aspect of MM activity, the Memory Manager events, it is necessary to understand how the Memory Manager performs its job, and to understand its goals.

First and foremost, the Memory Manager exists for one thing: To manage the real memory resource. This involves allocating space for objects (data and code segments, cache domains) and making those objects present in memory. It will try do to this in the most efficient way possible. The Memory Manager will go through a set procedure, and at each step check to see if the object it is trying to bring into memory is in. If not, it tries the next step in its procedure. The sequence of steps is designed to impact the system as little as possible. As each of the different things is tried, the overhead on the system increases. It is the relative frequency of these different Memory Manager actions which is measured by the MI, and reported by PROBE. The different events are listed below, in order of increasing impact on the system.

IMI - This indicates the segment was already being brought in on behalf of someone else. This is done for segments that more than one process would access, such as a file control block. In essence, the Memory Manager doesn't have to do anything at all.

Overlay Candidate- Recovering an Overlay Candidate involves, literally, flipping of a single bit. ROC (Recoverable Overlay Candidates) are created by the Memory Manager because these objects have not been accessed in a while, and the MM figures that this area of memory might be used soon. If however, the segment is referenced, the ROC bit is flipped and the segment is made "present" in memory.

Found
Free - This is done when the segment is indeed not in memory, so has to be swapped in. However, the Memory Manager finds a free area of memory to bring it into, and Space merely has to post a disc IO to swap it in. The disc IO imposes some load on the system, but the fact that an area of memory already existed makes the impact lower.

Deferral/
GiveUps- The difference between these two would require a much more in-depth understanding of the Memory Management algorithms. Suffice it to say that these indicate that the memory load was such that the Memory Manager gave up rather than try to bring the requested segment into memory, usually because it determined that it would have to kick something else out that was being referenced often. If this figure is high, then memory problems are definitely indicated.

In general, the higher the percentage of the lower impact actions, the better the memory situation. High numbers of Give Ups indicate a memory problem.

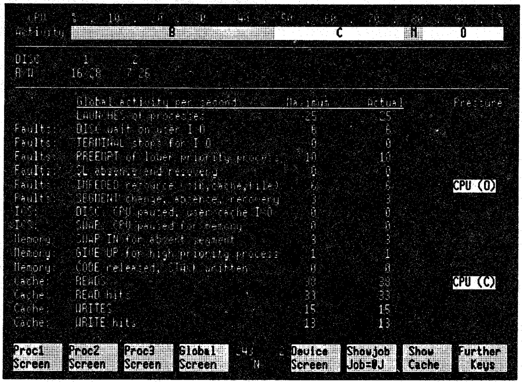


Figure 3. Sample CIA Globals Screen

The last facet of memory is a derived number. The ratio of launches to swaps is a good, simple barometer of memory action. A launch is merely the dispatch of a process, or allowing a process to have the CPU for its time slice. Normally, this is never more than 300 milliseconds, and is usually less, due to the nature of processes on HP 3000's. They tend to give up the CPU voluntarily while waiting for some event to occur, usually an IO. A swap is a memory management term used to indicate one or more objects (ie, segments) had to be brought into memory for a process to run. If the ratio of swaps to launches is high, then memory problems exist. Take the worst case, the number of swaps and launches is the same. This means that 100% of the launches requires a swap. Put another way, every time a process is given the CPU, one or more objects it needs is not in memory. This definitely indicates a memory problem. A good rule of thumb is that this ratio should be about 10-20% at the outside, after the system has reached a steady state. It is normal to see a burst of swaps when programs are being run, for example. It is important to make sure all the programs are running, data bases open, etc, before taking any serious measurements.

All of this discussion is well and good, but frankly, most of the items discussed above are not very useful to the average system manager. They all are really just used to determine if a memory problem exists. The CIA Global screen (Figure 3, page 9) shows the key items and automatically flags those it feels are a problem area. In the screen shown, CIA has flagged IMPEDED processes and CACHE reads as problems areas. Indeed, the workload that is represented here was such that these types of problems would be produced. A word of prudence should be noted here. Currently, the criteria for flagging these lines as problems is hardcoded into the CIA program. As the CIA manual warns, these should be factored in with the workload, CPU speed and other aspects of the machine, and possibly ignored for certain system loads. However, it does give a good starting point for the performance analysis.

DISC

The final primary system resource is Disc IO. The relative IO rates are important when considering whether IO is the bottleneck. Generally, disc IO rates of 35 to 45 IO's/second per disc can be expected. This will vary with the CPU model and IMB/GIC configuration. If the disc IO rates shown by this screen are consistently high, then more discs/IMB's/GIC's might be warranted. Figure 4 shows CIA's display of this data.

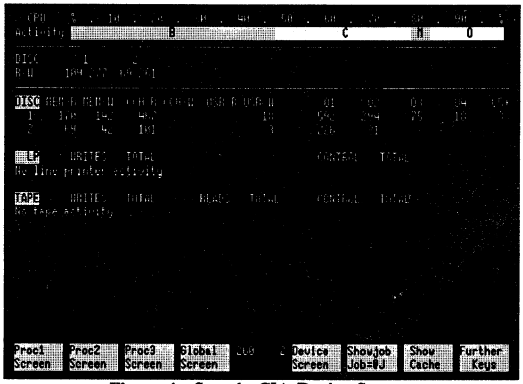


Figure 4. Sample CIA Device Screen.

Since the advent of disc caching, physical disc IO is never a bottleneck on a cached system. This is because the system generates several logical IO's (performed as a memory to memory move by the CPU) for every physical disc IO. Simply put, the current CPU's are not fast enough to generate enough logical IO's to saturate the physical IO channel. Logical rates of several hundred IO's/second would have to be performed to reach this point, and even the Series 70 doesn't have that much power.

Because of the importance of caching on a modern HP 3000, PROBE has a special screen devoted to caching data. The MPE command :SHOWCACHE provides some data, and Figure 5 (page 11) shows an example of the output of this command. In addition to the data shown by :SHOWCACHE, PROBE does some other calculations which make the data easier to digest. Specifically, the read hit/write miss figures show how well caching is doing.

The fact that caching is very prevalent today brings us to an interesting note. Since the advent of disc caching, the systems have essentially become purely CPU bound. In fact, if disc caching is working perfectly (ie, we never have to wait for a disc IO), then the system is CPU bound. Assuming adequate memory (another fairly recent phenomenon, the current price of memory chips not withstanding) the system bottleneck is purely CPU. While this might sound bad, it is not so in the long run. Building faster CPU's is much easier than building faster discs. Disc drives are bound by some very nasty laws of physics having to do with physical movement of the heads, rotation of the disc, etc. CPU's, on the other hand, can be built to incredible speeds, limited primarily by the cost of the machine and, ultimately, by the speed of light. HP's whole philosophy of the Spectrum line is based upon the principles of building simpler, potentially blazing CPU's. Unfortunately, the system software is chewing up more of that resource, but eventually this should even out.

Tweaking

Assuming we have read the various displays in CIA (or OPT, PROBE, etc), what do we do then? This, of course, is the million dollar question. Ideally, we look at the displays, determine the bottleneck, then either modify the application to use less of that resource, or add more of that resource to the system. If memory is a problem, then we might be able to modify the application to use a smaller stack, or we can add more memory. The same with CPU. In general, it is easier (and usually less expensive) to throw hardware at the problem than to modify the application. This

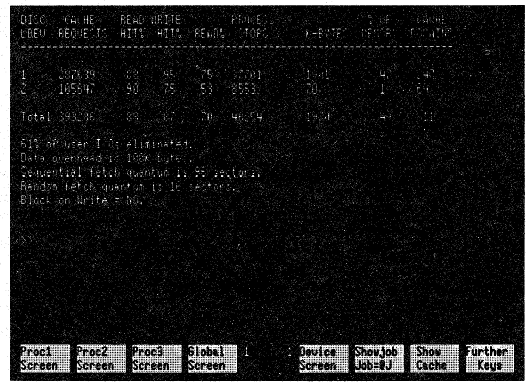


Figure 5. Sample :SHOWCACHE Display

is simply because most performance problems necessitate a re-design of the application, not modification after the fact. Very simply, this is because few people consider performance goals when designing an application, one of my favorite soapbox subjects. Anyway, it sometimes is possible to get noticeable gains in performance by tweaking a few things in the application. The 80/20 rule applies here: 80% of the performance gain is achieved by 20% of your effort. In other words, some good gains can be made by a little effort, but any more gains require finer tuning and more effort. Usually the gains are smaller in the later stages of tuning.

It is beyond the scope of this paper to deal with all of the "tricks" regarding the tuning of a system and application. A very good book exists that is chock full of ideas. *Taming the HP3000*[4] by Robert Lund does an excellent job of giving very practical hints on performance. Not only practical, but **numerous.** This book is full of good ideas in many areas. Implementing any one or two of them will more than justify the cost of the book. He also presents a primer on analysis, covering much the same ground as this paper.

Some Other Notes

Most of the tools mentioned offer some other capabilities besides online reporting. Most can print reports to the line printer at periodic intervals. Using this method, a batch job can be run which logs performance data all the time, giving the system manager some long term data by which to monitor the system.

Along those same lines is batch logging capability. Some of the tools log the raw performance data captured from the MI to a disc file, and a set of tools is shipped with it to report the data. The raw logfile is extracted in some manner (to flat files or an IMAGE data base) and some form of graphic interface is provided. PROBE allows character graphs to be drawn, or offers an interface to different graphics packages. CIA offers the ability to download data to a PC and then use the user's favorite graphics package for the reporting. The graphs produce are often comparable to those provided with HP's HPTREND facility.

Where Else Can We Go

As with many specific, specialized areas of computers, performance analysis is an area where experience counts. If a system manager feels he or she cannot adequately analyze the performance of their system, many avenues exist to help with this.

---

[4]. Robert A. Lund, Performance Press, P.O. Box 151 Albany, Oregon 97321. Copyright 1987, ISBN 0-945325-01-0.

The largest avenue, of course, is Hewlett-Packard. Every HP office (or at least area) has trained performance specialists who can look at your system and determine the bottlenecks. HP has several "pre-packaged" performance analysis products, such as HPTREND and HPSNAPSHOT.

Many sites, however, feel that there can be a conflict of interest by having HP do the analysis. The feeling is that they will just find that more hardware is needed, then use the analysis as an opportunity to sell more iron. Sometimes this is warranted. Many times as an HP Performance Specialist I was called in when it was a forgone conclusion that hardware was going to be purchased. I was to just answer the questions "What and how much?". Some sites will want an independent opinion, and that is where an independent performance consultant can be brought in. Generally, these people are ex-HP SE's who were trained in performance analysis, and can give a good appraisal of the system. Usually they are cheaper than HP, and/or can give a much more customized report, answering specific questions. (The amount of flexibility exhibited by HP in their analyses is usually dependent upon the individual performance analyst and his or her management.) Appendix A lists some of the independents. This is by no means a complete list, and only reflects those with whom I have personal knowledge.

HP and these independents will also teach performance analysis. HP's teaching is in the form of a class that is sold with OPT/3000. Strategic Systems has a course which has been taught several times. The other consultants can generally be hired at consulting rates to teach members of the DP staff about the use of tools, interpretation, etc.

Conclusion

Performance analysis and tuning is an iterative process where a system bottleneck is identified, corrected, and the system re-measured. Much experience is needed, generally, to find the correct bottleneck and fix it. However, tools, books, and training exist to allow the system manager some amount of independence in analyzing and correcting his or her own system, and thus gain much better use of their HP 3000 computer system.

Appendix A

West Coast Companies Offering
Performance Analysis Services

Allegro Consultants, Inc.
2055 Woodside Road, Suite 170
Redwood City, CA 94061
415-369-2303

Hewlett-Packard
Local sales offices
Contact your Sales Rep for Information

Mattedor Computer Services
5105 Highland Drive
Bellevue, WA. 98006
206-746-8123

Robert Lund & Associates
34130 Parkwoods Dr. NE
Albany, OR 97321
503-327-3800

Strategic Systems, Inc.
10502 11th Ave. NE
Seattle, WA 98125-7506
206-525-3309

Also: Contact SIGCONSULT via INTEREX.

MPE Performance Tools - A Chronology

By Mark Michael, MCS 3, CSS/HP Technical Support
Hughes Aircraft Company
Communications and Data
Processing Division
P.O. Box 9399, C07/2063
Long Beach, CA 90810-0399
(213) 816-6017

The following is meant to be a meaningful blend of my experiences and knowledge on the subject of performance tools, acquired over the more than ten years that I have been working with HP3000 computer systems. The opinions offered within are my own, and should be considered as such.

One area of systems software for which the HP3000 community has been given the "Rodney Dangerfield treatment"[1] for many years has been in the area of performance measurement. Recently, there has been a major expansion of products and services made available that purport to satisfy the thirst of the HP3000 community for performance tools. In this paper, I intend to provide a summary (from a personal perspective) of the development of the better-known products and services in this area.

First, let me provide a definition or two[2]. What do I mean by "computer system performance"?

In a computer system, PERFORMANCE is judged by the computer's ability to complete requested work without unreasonable delay.

In other words, your HP3000 provides what you consider good performance if, as the system manager, you receive an acceptably low number of complaints about system response. This definition is entirely subjective in its content, but it reflects reality and lends itself well to threshold-type measurements which can be numerically analyzed. For example, you may conclude, via a performance tool, that there is a correlation between a substantially higher number of user complaints and when measured response time on your system exceeds a three-second average.

[1]"I tell ya, I don't get no respect!" (with apologies to Mr. Dangerfield).

[2]Although, this paper is not meant to be a performance primer.

What do I mean by "performance tool"?

A PERFORMANCE TOOL is a set of programs used to record, analyze or forecast system performance.

I perceive two types of performance tools as being useful for computers doing actual work[1]:

BOTTLENECK-FINDING (what's wrong right now) is different from CAPACITY PLANNING (what's going to go wrong *n* months from now).

Obviously, it is better to prevent bottlenecks before they occur, rather than wait for them to catch you unawares. It's better for *n* to be greater than 6 months, rather than 1 day; this allows you time to complete the acquisition of an upgrade, or take some other corrective action, in a more orderly (and less panic-stricken) fashion.

What attributes do I look for in a resource measurement?

- It should provide a threshold-based measurement where appropriate (i.e., 90% CPU utilization, or 2.8-second response time versus 3-second maximum acceptable response time), or a population count where appropriate (such as terminal, disc, tape or printer I/O operations per second or per shift).

- It should be measurable via a well-known (ideally, supported) programmatic interface. The MPE V/E Measurement Interface qualifies, but just barely (more on that later).

- The data for a given week/month/quarter should be comparable to previous weeks/months/quarters, thus requiring a well-understood method for indexing results prior to and after a system upgrade. I prefer to keep the current month on the "hundredth percentile", and index any historical data downward (using a coefficient of relative throughput). This requires less explanation to the uninitiated ("We're at 70 percent and will run out in about 5 months", versus "The current value of 235 was derived by ....").

- A sampling of data points over time should be open to usage of statistical methods for extrapolation without serious distortion for the time period required for taking corrective action.

- The resource should be one that identifies a risk to system performance if it is exhausted, and for which corrective action can be taken.

[1] As opposed to benchmark systems, an entirely different area for discussion.

What principal resources do I think a performance tool should report on, assuming the limitations of the MPE V/E Measurement Interface, log files, etc.?

CPU UTILIZATION

My experience with this has been that either of two measurements have been useful:

- o Percent utilized (time busy [not paused] versus elapsed time) over a period of time (such as a work shift), or
- o CPU busy hours during the total period of hours elapsed (such as hours CPU busy during day shift, Monday through Friday, plus Saturday from 8 AM to 12 noon).

In both instances, the measurement is usually expressed as a threshold-style measurement (how soon until I'm out of gas), and both require indexing of each month to correct for past and planned upgrades. I prefer indexed percentages, with the current month's available capacity equal to 100. I usually recommend that an upgrade be ordered for delivery prior to a given CPU exceeding 80 percent utilization for a projected month (assuming that no further modeling of growth in CPU resource requirements is available).

DISC SPACE UTILIZATION

I have found that a measurement of percent utilized for all disc drives on a system is a good "trip-wire" measurement to warn of impending problems. Further tuning, file placement, etc., requires more detail than most people can digest quickly when attempting (in 5 seconds or less) to determine if any problem exists yet.

Device-level breakouts, as well as trends by application are also useful in determining the course of corrective action to take, where application disc usage can be defined without severe additional overhead. For example, if disc space usage by applications on a system can be broken out at the group level, then the existing MPE :REPORT command provides a supported source of raw data for reporting.

This measurement is usually expressed as a threshold-style measurement (how soon until I'm out of gas), and requires indexing of each month to correct for past and planned upgrades. Again, I usually recommend that corrective action begin when capacity is expected to exceed 80 percent within 6 months (depending on the volatility of usage on the system).

RESPONSE TIME

A generally accepted definition might be: elapsed time minus think time divided by the total number of terminal reads for the sample to be analyzed. This information is available for each process from the MPE Measurement Interface.

Many service-level agreements require that certain response-time thresholds never be exceeded. Thus, a threshold-style measurement is in order, without indexing of data. Most products available for MPE systems today do not include, as part of their analysis or reporting facility, the ability to provide configurable "confidence intervals". Such a facility would provide graphical support for a service-level agreement which specifies that 95 percent of all transactions would occur in *n* seconds or less[1]

TERMINAL ACTIVITY

The MPE Measurement Interface provides a count of terminal I/O's for each process for its lifetime. It turns out that the count provided is fairly representative of the actual number; for example, a VPLUS program has a single terminal I/O counted for each combination of VSHOWFORM and VREADFIELDS, with the Measurement Interface handling the variations in the number of handshakes that occur between the terminal and the HP3000.

Are there any resources for which there is no straightforward method of reporting?

REAL MEMORY DEMAND VERSUS SUPPLY

Although a considerable number of counters are available from the MPE Measurement Interface, it is not clear how to best utilize them to answer the following fundamental questions:

- Is my HP3000 computer running short on real memory?
- How much more do I need, if any?

DISC I/O THROUGHPUT

The counters available from the MPE Measurement Interface for disc I/O are in I/O's per second; a lot of averaging must be swallowed for this measurement to be useful. Another set of counters are for the number of entries found on the Disc Request Queue for a given disc drive at a given time; this allows for a measurement that is less dependent during interpretation on knowing the type of disc drive(s) being measured and the distribution of disc I/O's being performed by address or length.

The measurement I would like to have would be device utilization as a percentage of the theoretical maximum throughput, based on the throughput of the slowest device in the path between the CPU's memory and the data on disc; given the high volatility of activity from millisecond to millisecond for a given path as described, it is unlikely that the achievable maximum will exceed 35 percent of the theoretical maximum.

WHAT'S WRONG WITH THIS PICTURE?

Many people will stop at this point, thinking that simple linear projections of future activity is enough to catch any potential performance problem. This would be true, if you never intended to add a single enhancement, new application, new

[1]The choice of 95 percent is not a coincidence, since it can be easily computed using the second standard deviation. But, is it attainable? Therein, I may be letting the amateur statistician in me get carried away....

user or any other externally-induced variation. System performance trends would be based solely on doing the same thing from month to month. Linear projections ("last month was 30 percent, this month was 40 percent, next month will be 50 percent") have their place, where no other source of information is available. However, there is a requirement to know something of the business activities that your computer system supports. With such knowledge, you can make an attempt at forecasting your requirements in a way that simple linear projections can never satisfy.

A number of useful papers on this subject have been presented over the last several years, many by HP performance specialists from both the factory and the field. As an example, the idea of paying attention to the underlying business trends for capacity forecasting came from Tony Engberg of HP's Performance Technology laboratory. His papers on modeling have been extremely informative.

One other warning. The MPE V/E Measurement Interface is officially undocumented, unsupported and requires Privileged Mode to access it. Although it hasn't moved much, this is a point of concern and sets an uncomfortable precedent for MPE/XL Measurement Interface access.

With the above definitions, expositions and cautionary notes in mind, let's begin by turning the clock back to 1979. The dates given in the headings below are largely when I first saw the product, and may not accurately reflect when they were formally announced or shipped.

THE NEOLITHIC ERA (PRE-MPE IV)

Fresh in my new position as system manager of Beckman Instruments' development HP3000 Series III (remember those?), I went back-to-back to the "A Programmer's Intro" and "System Managers'" (SM) courses at Hewlett-Packard's Fullerton office. In both courses, some time was devoted to contributed library tools, including OVERLORD, SOO (Son of OVERLORD), SILO (Son-in-law of OVERLORD), TUNER and others. I did some further exploration of system performance after the classes. Usage of these tools led to my first system failure.

With this rude introduction to Privileged-Mode (PM) programming side-effects, I determined to find out what we could safely measure, how we could measure it, what we could do about it, and how much it would cost (both to measure and to fix). With MPE III, there wasn't much that we could reliably obtain. HP SE's had two tools that were used sparingly to measure system performance:

- RTM, a menu-driven tool capable of measuring global system resource usage. It was notorious for causing system failures from MIT to MIT, and was the predecessor to OPT/3000,

- SAMPLER, a product that originally involved both hardware and software. A board containing a second system clock was installed into a Series III. This second clock allowed the interruption of the system in a fashion analogous to a device; the "driver" for the second clock scanned through system tables when the second clock interrupted the system. Although it was too expensive to use continuously, SAMPLER did provide information about program segment usage and instruction counts far beyond any other tool available at the time. SAMPLER was the predecessor to APS/3000.

MPE IV

Our department at Beckman Instruments had been privy to information about a new family of HP3000 computer systems under development at HP, known as the "Integrated Computer Family" (ICF). The first member of this family to be introduced would be code-named "Grizzly". The operating system kernel for the new systems was to be rewritten to include better dispatching and memory management. A key feature of the new operating system was the inclusion of a vastly improved memory management scheme, as well as a better dispatching mechanism. The code for both of these functions operated as part of the device driver for the system clock (thus ensuring a regular, consistent and sharable access to the system at less-than-a-second intervals).

SAMPLER was recoded to eliminate the requirement for hardware; it now simply added itself to the "shared clock interface" queue, along with the dispatcher/memory manager code. And, RTM changed too. ...

Upon receipt of our first HP3000/44 ("Grizzly") system, our HP support engineers used SOFTWARE.CREATOR to install the operating system. Under this mechanism, your system had to have enough disc space to allow *all* HP-supplied software to fit on your system; the SE then deleted the products you didn't want. When prompted for a product called "OPT/3000", we all scratched our heads and figured, "what the heck - let's see what it is". When we ran OPT.PUB.SYS, we all became so excited, we showed it to my management. I was enrolled in HP's new Performance Optimization course that day. This appeared to be what we had been waiting a long time for.

It turned out that the MPE IV lab had anticipated the need to do a lot of tuning in order for MPE IV to properly execute. The previous mechanism in use for measuring how completely a given operating system was traveled, and how well it performed was known as MMSTAT. This usually required additional source code statements calling MMSTAT routines, resulting in a completely different set of object code than what was released to the public. To short-circuit this process, the MPE IV kernel included a "Measurement Interface" (MI), which consolidated much of the information that RTM had to work very hard at obtaining.

All was not completely well, however. Although MPE IV allowed a significant performance improvement over MPE III, the improvement was primarily due to the elimination of some rather inefficient memory-management and file-system buffering code. Systems that apparently were CPU-bound were now spending most of their time waiting for disc I/O. And, ...

... almost ALL of the PM utilities such as SOO and TUNER now both lied and frequently caused system failures! Without OPT/3000, most HP3000 users were now "blind" as to what may have been causing poor system performance.

OPT AND MPEDCP (1982)

OPT/3000 provided a supported mechanism for studying the following global resources:

- o Global CPU Utilization, and CPU states
- o Global Memory allocation by type of segment
- o Global disc I/O by disc drive logical device number (LDEV)
- o Global tape and printer activity by tape/printer LDEV

Originally, OPT did not provide any visibility to process-level data. However, we were able to confirm by benchmark that the global CPU numbers were within acceptable limits. With the Series III, we were able to hook up a Tesdata hardware performance monitor to a pin on one of the CPU boards, which allowed us to validate the overall CPU busy number.

There was a problem with using OPT in that environment. For example, a user would call up and say that he/she had a problem with performance on a Series III in, say, Paris. We would connect via dial DS and spend most of a day transferring OPT to that system. Then, usually the following day, we would ask the site to reproduce the problem. More often than not, the CPU was saturated. We had no supported way to determine what program may have been saturating the CPU, since OPT didn't tell us that at that time. Given the expense involved in upgrading the CPU, we had a tough time selling global-only data to management.

Additionally, much of OPT's functionality was carried over from the RTM tool; a large amount of its functionality was spent on diagnosing MPE III performance problems. For example, there was a section of memory utilization display code in OPT for displaying information about memory segments too small to represent graphically; it turned out that the MPE IV memory manager had been optimized to never create any segments below that limit, anyway.

At this point, two things happened that helped a great deal. First, enough information began to leak out of HP's labs to allow the improvement of contributed utilities (at least to the point where they no longer caused system failures and *could* be made to be reasonably truthful), as well as the initial development of third-party products for performance measurement. Second, HP's field personnel developed MPEDCP as a "snapshot" tool, measuring and reporting on just about everything that happened on the system - for one hour.

MPEDCP utilized global, device and process-level MI data and produced an exhaustive report. For a time during the early 1980's, Beckman had HP run several MPEDCP runs per year, spread around a network of several dozen HP3000 systems. MPEDCP had some limitations, in that it didn't "trend" any of its data based on performance history, and it wasn't available as a purchased product. It did mark the first time, however, that we had a tool available to ensure a fairly accurate resolution of performance bottlenecks at the process level as they happened.

Hewlett-Packard began using MPEDCP internally in developing benchmarks. This increased HP's dependence on useful, accurate snapshot-style performance tools.

SYSVIEW (1983)

Although a number of contributed programs were now available for usage in bottleneck-finding, only OPT/3000's logging facility was available for ongoing, continuous performance measurement. Also, no supported product was available for inspection of process-level activity.

In 1983, I became aware of a product from Carolian Systems International called SYSVIEW. This product's claim to fame was as a "supported OVERLORD", or "OPT plus a lot more". It used the MI to display (and optionally, log to disc) data about global CPU, memory, disc I/O, free space and response time (!). It also included process-level information for the first time, allowing us to see which processes were hogging a given system.

SYSVIEW included more of the information we wanted, but still required a lot of jumping from screen to screen in order to study a process' activity within a global context. Its logging facility would allow for ongoing monitoring of system activity, once disc space became cheap enough to allow for the large volumes of data anticipated.

After moving to Hughes Aircraft Company in mid-1984, I was asked to put together an internal paper describing the set of performance tools we would need to service our internal customers. During this time, Hewlett-Packard released MPE V/E, and our department had its hands full with system updates. In addition, the MI moved again, causing many contributed programs and third-party products to behave erratically. By mid-1985, when we returned to the subject of system performance, we were pretty sure what we wanted.

HPTREND, HPSNAPSHOT, HPCAPPLAN (1985/1986)

Hewlett-Packard laid a bombshell on us at our Hughes-internal HP3000 users' group meeting in mid-1985. One of their speakers discussed three new products, intended for introduction within a few months, that taken together would provide us with substantially what we wanted (or, so it first appeared). The three products were:

- o HPTREND, which had a collector job running continuously on a measured system, and periodically shipped its data back to HP for further reduction and analysis,
- o HPSNAPSHOT, a descendant of MPEDCP (via the System Performance Evaluation Project, or SPEP, tools set), costing $5000 for the first analysis,
- o HPCAPPLAN, a product which would take HPSnapshot data and use it as input to a workload modelling facility. In theory, this product would allow us to play "what if" games once we had a clear enough idea of what would be included in the projected workload.

We began a formal project for selection of performance tools and the development of a capacity planning service. Our first activity was to set up a meeting with an HP team working on HPTrend at the factory. We needed to begin an iterative process of design refinement which could lead to a product providing what our management was looking for.

Unfortunately, the people responsible for HPTrend at HP decided that they were unable to participate in developing any enhancements along the lines we had suggested. They cancelled the initial meeting and declined to further participate, even though we had included a right of first refusal for Hewlett-Packard in our project plan (after all, they wrote the operating system, so they would be best positioned to supply a supported product). We then turned to the third-party market to solicit further participation.

SYSPLAN - HUGHES C&DP MAJOR PROJECT (1986)

We prototyped a performance measurement system, using Carolian Systems' SYSVIEW for data collection and Cognos' POWERHOUSE for reporting. Our requirements analysis included sending the prototype reports and charts around to various managers involved in approval of system upgrades. After considering their feedback, we determined that the new product would have to:

1. Automatically collect data continuously on global and process-level system activity. The ten-minute sweep approach used by Sysview would allow too much process-level data to go unaccounted for (we failed to record data for processes whose entire lifetime fell between sweeps); we needed data collected at process termination, too.

2. Make available statistical data capable of interpretation according to the guidelines discussed earlier in this paper.

3. Allow for automatic retrieval of performance data from remote systems to a central administrative site.

4. Satisfy internal and external (i.e., US Air Force Program Office, etc.) audit requirements.

5. Provide charts and reports that had our management's approval.

6. Provided genuinely useful information.

After distributing a Request for Proposal and entertaining bids, Hughes Aircraft Company's Communications and Data Processing organization selected Carolian Systems International to write the product, initially to our specifications. During 1986, the first release of SYSPLAN was developed. We implemented the first production version on more than two dozen systems during late 1986 and into 1987.

SYSPLAN is driven by the MI, plus some other miscellaneous data (such as disc free space maps, for calculation of disc space usage). It contains three major modules:

- o COLLECTOR, which runs as a batch job on the system and logs performance data continuously to disc.

- o REDUCER, which runs periodically, statistically reducing the raw data down to cumulative, hourly, normalized global and process-level data.

- o REPORTER, which provides written reports and DSG charts in conformance with the format approved by Hughes management.

Once the first release of the product was in place and Hughes management had their first real visibility into HP3000 system performance, their reporting requirements changed so much that our department ended up writing a backend interface for Sysplan into a non-DSG graphics environment. This backend is written in Powerhouse, and provides the charts seen in Figures x through y.

Sysplan gives us everything we had seen in Sysview, plus the process-termination data we found we could not do without, plus built-in statistical reduction of the data. This statistical reduction includes both global data, plus application-level data, where applications are defined as collections of identified processes.

Nothing is perfect. We found that, until the Spring 1989 release of Sysplan, its reducer took far more resources than we had anticipated (the current version of the product, available as of this writing, takes a fraction of the time to do the same work as earlier versions). Also, statistical reduction data was accumulated on a monthly calendar that did not conform to the Hughes fiscal calendar, complicating our interpretation of the data for those systems having regular, monthly activity variations. We found that DSG graphics were too slow and CPU-intensive to be practical on a large scale. Also, once we began to use Sysplan, we realized that we hadn't insisted on the ability to include disc-paused states in our CPU utilization graphs. In spite of these drawbacks, we have found it quite useful, and charts using data derived from Sysplan have been used successfully (and accurately) to justify system upgrades in a number of instances.

Since that time, Sysplan's further development has been influenced by several other HP Major Accounts who also have invested a significant amount of time, effort and money in acquiring and using capacity planning tools.

Of course, measurement and trending still will not predict changes in system performance without additional analytic modelling, simulation or other forms of analysis....

PROBE/3000 (1988)

As time passed, other competitors became welcome additions to the choice of tools available for bottleneck-finding. Strategic Systems, Inc. introduced PROBE/3000. It included much of the same information as Sysview, but presented in a form that was superior in many respects to either OPT or Sysview.

The Probe/3000 user interface is function-key driven. Its initial display takes many of the concepts that went into OPT (in 1981 and earlier) and Sysview (in 1982) and builds on them. I found that, in the version of Probe that I reviewed in mid-1988, I rarely needed to go beyond the main display in order to locate and diagnose a system bottleneck.

Probe/3000 also includes additional functions not found in other products. For example, a global "DBUTIL SHOW"-like capability for examining database usage is included.

Again, nothing is perfect. I found that SSI's first attempt at a statistical reduction and reporting facility was too rough to be useful. In the mid-1988 version I reviewed, the manual did not include any examples of its operation, and the reporting programs blew up on Pascal I/O library errors when I attempted to use them.

Even at that, Probe was and is today an excellent bottleneck-finding tool. I look forward to improvements in their product as competition heats up. As of the date of submittal of this paper, I have not yet been able to test the MPE/XL version of Probe, but hope to have completed testing prior to presentation of this paper. Also, in all fairness, I intend to review the MPE V/E version of Probe again, testing the revisions made to its trending facility.

CIA (1988)

Facer Information Systems has made available a product called CIA (which stands for CPU and I/O Analyser). It includes an interactive bottleneck-finding program, plus a background facility for continuous performance measurement. The data collected from this facility can be used to generate one or more of a series of analytical reports or histograms. I intend to perform a more exhaustive analysis of this product during the interval from submission of this paper until its presentation.

HPGLANCE (1989)

Hewlett-Packard dropped another bombshell with the introduction of HPGlance. This product provides many of the first-level bottleneck-finding-at-a-glance displays found in preceding products. It appears to be enough for interactive troubleshootings. It is available now for both MPE V/E and MPE/XL. It is supplied by the operating systems vendor (which, for better or for worse, may give their support staff prior knowledge of MI changes). It also is priced very aggressively. I find a lot in HPGlance to recommend it.

The one drawback with HPGlance for some people is what others may consider a principal strength: it is provided by HP. On the one hand, the HPGlance development and support staff may have better access to information about the MI, and may be in a position to influence changes in MI design. On the other hand, since the hardware, software and measurement tools are all supplied by HP, some people may find it more difficult to get justification for upgrades past their management when using this tool. After all, who is validating the numbers but the company from which we would be buying the system upgrade?

LASERRX (1989)

Without any prior warning to us, after we had attempted to work with HP during 1985 and 1986, HP in late 1988 introduced a product called LASERRX. This product purports to fulfill the intended function of what we had tried to get in 1985. The product includes a number of innovations relative to competitive offerings, but at a price (literally).

LaserRX performs continuous collection of data in a fashion that is analogous to (but a bit more primitive than) Sysplan. A periodic sweep of the MI tables is performed; but, instead of writing that data to disc for later analysis, the data is reduced continuously, too. The best analogy I have heard is that Sysplan's Reducer is like a square chunk of fudge candy, while LaserRX's patented reduction facility is like a thin layer of fudge cake frosting. Both end up using about the same amount of ingredients/resources; one appears to be "bigger" in usage than the other, depending on your viewpoint. The application definition facility in Sysplan is more comprehensive, and reduction can be re-run in Sysplan, while (as of this writing), it can't be re-run in LaserRX (they're supposed to be working on this).

Presentation of performance data occurs on a 286/386 PC equipped with a CD-ROM drive, Microsoft Windows and a hi-res color monitor (VGA preferable, EGA is ok). Considerable flexibility is available in modes of presentation; data can be graphed in multiple, overlapping windows and compared visually. A limited amount of calendar configuration is available (weekends can be excluded, as they can in other competitive products). Additional resources can be reported, including (finally!) disc I/O device utilization as a *percentage* (rather than the somewhat artificial "I/O's per second").

The existing version of LaserRX does have some "release 1.0"-type deficiencies:

- o No trending facility is included natively in LaserRX. This can probably be done by exporting to either Lotus 1-2-3 or Microsoft Excel. As of this writing, I am attempting this, and should have results by the time of presentation.

- o If I am incorrect in my assumptions about the structure of an application when I define it in SCOPE (the LaserRX collector), I can't go back and re-reduce the data. This greatly impedes any iterative refining process that would otherwise be used to better understand current usage of the system being monitored.

- o There is no fully configurable calendar (company fiscal calendar, company holidays, split shifts, etc.). The only way around this is to take the low-level detail data and produce your own charts (as we did with Sysplan).

- o If I don't have a properly configured PC on hand, the initial expense of setting up LaserRX could be substantial. It turned out that I had such a PC available (a Vectra ES/12 with HP LaserROM CD-ROM drive, plus Windows/286 2.1). Some people may consider this to be an advantage, where they have multiple HP3000 systems and would prefer not to incur any additional system load on any of those HP3000 systems in order to generate their reports.

- o The SCOPE collector doesn't have any mechanism to switch log files. This is an elementary oversight, which precludes backing up the log file data properly without bringing down SCOPE (an unacceptable situation). We found during testing that we lost the first few days of a month-long collection, due to SCOPE "wrapping around" inside its log file without telling us. This one has to be fixed before SCOPE can be trusted as a collection mechanism.

All things considered, LaserRX is a formidable entry into the capacity planning segment of the performance tools market. LaserRX and Glance are the first of a series of products that HP intends to make available during 1989 and 1990 for performance management.

MPE/XL - THE ANSWER IS 42 (1989-?)

Many of the readers of this paper will be acquainted with Douglas Adams' THE HITCHIKER'S GUIDE TO THE GALAXY series. In it, a race of people on a distant planet in the dim past had asked a super-supercomputer for "the answer to life, the universe and everything". After millions of years of processing, it responded "Forty-two". When they asked, "what was the question", the computer said simply, "I forgot - you didn't ask me to remember the question".[1]

The moral of that story as I see it has to be, "ask your questions carefully, remember to take good notes, and don't trust alien computer systems".[2]

The HP Precision Architecture MPE/XL operating system has put performance monitoring of HP3000 computers back to the equivalent of "Neolithic" times. Not only are we no longer able to get the answers we can obtain from the nearly-supported MPE V/E MI, we can't be sure our questions make any sense anymore. The tools should evolve much faster this time than last. Of course, not only has the data available to collect changed, but many of the assumptions about system behavior also must change. The type of system activity considered necessary to provide optimum response time and batch throughput is still not well-understood (as of MPE/XL 1.1) in all cases. A graphic illustration of this was the large number of papers on performance and internals at SCRUG '89 (all presented by non-HP employees with one or two stellar exceptions, all well-attended), but few takers for panel members at the MPE/XL-specific roundtables (there were a few brave souls). How will we find our way out of this wilderness?

The last time, HP made available the requisite source code for the operating system and anonymously contributed utilities which illustrated mostly correct ways of accessing the performance data found within the bowels of MPE. This time, the pickings are slimmer, with the predictable result that it is taking far longer for performance tools to appear than most MPE/XL customers think that it should.

HP, to its credit, is working on a number of tools. Also, to HP's credit, when faced with a performance/uptime crisis with MPE/XL 1.0 beta sites, HP's management put everything else on hold and assigned the best people they could get out of every lab to fixing MPE/XL 1.x. This process put the aforementioned tools (and even the underlying MPE/XL MI code development) on hold, wreaking some minor havoc with introduction schedules.

Fixing the near-term problem will probably involve fuller disclosure by HP of how HPGlance/XL and HPLaserRX/XL intend to obtain their data. Also, consideration of inclusion of the MPE/XL MI in the HP Architected Interface Facility (AIF) project is now underway. The good part of that is that the AIF would provide a supported MI (for the first time!). The bad part is that this may mean that access to the underlying raw data will be permanently restricted (if this

[1]Does anyone remember what the question was?
[2]With apologies to Mr. Adams.

had happened in the late Seventies and early Eighties, Adager, DBGENERAL, Sysview, Probe, CIA, Silhouette [now sold by HP], SpeedEdit, Network Engines, NetBase and a host of other products [using PM or system tables, or both] would not exist). IBM has tried for many years to enforce such a policy with their operating system products, sucessfully with System/3X and AS/400 systems, less successfully with System/370 operating systems. We should all consider what advantages and disadvantages there are for Hewlett-Packard to imitate IBM in this respect.

My modest suggestion is to do both - AIF *and* tables, plus one more thing. Please, HP and the third parties, develop and utilize an AIF; I'll sleep better nights, knowing that the production computers don't have any non-supported PM code running anywhere on them. Please, HP, allow *me* as system manager to set the level of privilege available to non-kernel programs *at the time I boot the system* ("production" boot versus "beta" boot). Please, HP, allow third parties access to the low-level data; the AIF won't include everything we may find we later need (the AIF project, by definition, must have as its result a constantly evolving product for it to be successful). And, finally, Interex, please serve as some kind of an honest broker in this regard, perhaps even registering Interex as an AIF "vendor" in order to facilitate access to AIF documentation on behalf of the rest of us.

System Performance

Developing A Strategy

By Laurie Facer
FACER INFORMATION DESIGN
PO Box 270 Epping, Australia 2121
Phone: 011 61 2 484 3979 Fax: 011 61 2 484 5709

There comes a point in every System Manager's lifetime when he feels that he should "really do something about system performance". However, the reality of the situation is that controlling system performance is more easily said than done.

Good system performance on a given CPU is not achieved by the implementation of one simple idea. In reality, the system performance experienced on a machine at any one time is the accumulation of many decisions made over time and often made in isolation to each other.

The multiplicity of factors involved in controlling system performance, does not diminish the fact that system performance is crucial to the success of any DP installation and ultimately the organization itself.

Why A Strategy?

The starting point in tackling system performance is the recognition that system performance is not the result of any one decision. Rather it evolves over time and is the result of many decisions.

The operating environment of the HP3000 is extremely dynamic. There are many variables that make up the total environment and, indeed, that environment changes continuously. The environment is dynamic in two ways. Firstly, given a particular environment structure, the interaction of processes within that structure changes constantly and secondly, the environment structure itself changes over time.

Let me give an example. A machine that has exactly the same file structure and data on two days may behave differently on each day. The reasons for this are manifold - different pattern in logon times, two programs run together on one day but not the next, somebody decides to do a KSAM generic search, etc.

The structural environment has remained unchanged, but performance has varied markedly.

Of course the structure of the environment itself is constantly changing with the addition of new hardware, the introduction of new systems, converting an IMAGE database system to an OMNIDEX database system, etc.

Another complexity in system performance is the inability to find many absolute rules that will always ensure good performance. One rule may work beautifully for one installation, but have horrific results for another. These rules even change on the same machine, depending on what is happening at the time.

For example, very large blocking factors may be fantastic for overnight batch processing, but kill the system during the day for data entry and online reporting.

To further add to our problems, system performance is not obtained by just concentrating on one variable. System performance is controlling many variables.

Obtaining the most from your HP3000 is a balancing act - everything must be kept in balance so that bottlenecks are not created. If bottlenecks do occur, then waiting occurs, and poor responsiveness results.

The number of variables to determine that balance are enormous - files balanced across discs, on-line processing versus batch processing, memory size, code segmentation, blocking factors - and the list never seems to end. Just to think about where to start gives one a headache. And just after you have it all sorted out, you go to a user group meeting and somebody tells you something that totally contradicts the thing you just spent three weeks in implementing.

So, when tackling system tuning we are faced with three basic problems:

a) Dynamic Environment - moving target syndrome.

b) Absence of many absolutes - the theory of system performance relativity "Everything is relative to everything else".

c) Unlimited factors - how long is a piece of string syndrome.

Given these problems, obtaining system performance is not just a matter of twigging a few bits - it is a continuous operation that needs constant management. If the system manager is to obtain control over system performance, he needs to first develop and then implement a clearly defined strategy.

The Strategy - A Plan Of Action

In developing a System Performance Strategy, three tasks need to be performed:

a) System Performance Goals set.

b) Action Plans detailed.

c) Resource requirements defined.

I need to emphasise at this point that system performance is a project just like any other software project. The size of that project depends on how far you want to go in controlling system performance.

There are no "go fast" buttons on the HP3000. As stated earlier, system peformance is the final result of many decisions. The better control you have over the decision making process, the better your system performance will be.

Unfortunately, many system managers are not in the position to exercise total control over the system. This fact needs to be recognised in the strategy and goals set accordingly.

System Performance Goals

Before goals can be set a definition of system performance must be formulated. The definition that I propose comes from Neville Silverman, the author of the system performance product, CIA.

Neville proposes the following "System Tuning Definition":

At the process level:

Minimization of CPU usage.

Minimization of Disc I/O usage.

At the Global Level:

Minimization of on-line Process response time

Maximization of Global Process CPU usage.

Minimization of Queue lengths on each Disc drive.

Minimization of Memory Manager activity.

Minimization of Dispatcher activity.

At the Process level we are looking to minimize the resources required by each process to perform their allocated function. This implies control over decisions made at the programming and system design level.

At the Global level we are looking to maximize the "effective" use of resources and minimise the overhead in using those resources. We are also looking to minimize bottlenecks such as queueing on disc drives.

You may wish to construct your own definition of system performance, but I believe the above definition to be a good starting point.

Having obtained a definition of system performance, the next step is to construct a series of goals. These goals should be set in such a way that their achievement can be measured. To be able to this you may require one or more system performance measurement tools. Preferably, the purchase decision for that tool will be made after you have defined your system performance strategy.

The goals that you set should be tailored to your organizations particular requirements. I will propose four simple goals for a fictitious installation.

Goal I

Session Response Time is to average less than three seconds between 9 am and 5 pm Monday through Friday.

Goal II

Overnight production will complete by 6 am Monday through Friday and 5 pm Sunday on weekends.

Goal III

"A" priority reports will be completed one hour after requested.

"B" priority reports will be completed four hours after requested.

"C" priority reports will be ready for distribution 6 am next day.

Goal IV

Client Account Enquiry - Will take less than 4 seconds to enquire on a client's account at any time.

Bank Reconcilliation - Will be completed at 4 pm Monday through Friday.

Unmatched Deposits Report - Will always complete within 5 minutes of start.

My first three goals are general goals representing the processing standards that are expected from my installation. The fourth goal is specific to the requirements expected from specific applications. These are the goals by which I will rate my system performance.

Developing A Strategy

Once the goals have been defined, a strategy for attaining those goals needs to be developed. The strategy that I would recommend has five components:

1) A System Performance Monitoring System that reports progress in goal attainment - Goal Reporting.

2) A System Performance Monitoring System that provides enough data on system activity to allow the capture of system performance culprits - Activity Reporting.

3) The definition of areas of system performance that most urgently need attention.

4) The creation of projects to tackle the renegade system performance areas.

5) Performance review procedures for new programs and systems.

The Goal Reporting system highlights the system's performance against the stated goals. Only if there is a deviation from these goals, need any action be taken.

If action does need to be taken, activity reports must be available to allow detailed examination of system activity. These reports should be able to be turned on or off depending on the need for troubleshooting.

Once you have gained control over system performance, the need for projects to examine system performance problems will be greatly diminished. If, however, you are faced with a badly performing system, you will need to tackle one thing at a time and progressively gain control over performance.

To maintain control over system performance, a set of performance review procedures is essential. You have just spent a lot of time and, perhaps, money to get better system performance. The procedures you implemented to gain that performance need to be propagated in new programs and system designs.

Goal Reporting

For Goal Reporting, I will concentrate on just one of our objectives - Session Response times. This goal is generic to a well performing system and is the most applicable to the vast majority of HP3000 installations.

Goal I stated that our system is required to have a Session Response time that averages less than three seconds between 9 am and 5 pm Monday through Friday. To determine if this goal is being fulfilled, we need to be able to measure Session Response times. The problem with this is defining what a response time is.

Superficially, response time is the time it takes to receive a response after hitting a key to send data. But in reality, life is a lot more complicated than that.

For example, if an HP3000 was only running one process, the response time using 9600 Baud would be faster than a response time using 1200 Baud. Sometimes it is hard to know what is a request for something to be done and when the task is complete. V/3000 sends characters down during machine "think" time - these should not be mistaken for user data transmission.

A further example of difficulty in measuring response times is highlighted by HPWORD and online QUERY reports. HPWORD is continously transmitting and receiving characters. We need less than three seconds response time for that. However, if someone requests an online serial read from QUERY, it is a bit unrealistic to expect a three second response time for that process.

Response time will vary widely by the minute, as the exact functions that the machine will be performing at any one time is totally unpredictable.

There are products available that will measure device response times, eg, TMS from Orbit, and these are very handy for measuring responses of particular devices.

Another approach that can be taken is the one similar to that taken by CIA's Session Reponse Time Report.

This report shows, for sessions divided into "C" and "D" queues, the average response times for specified time intervals for all active session processes. Just as importantly, it shows the wait times that processes experienced for system resources. The response time provided by this report is the total wait time divided by the number of Terminal Faults.

Recognising that not all processes can be treated equally, a third column is provided showing response times for individual processes nominated by the system manager.

This report has three major advantages:

a) Individual processes can be excluded from overall figures and examined separately.

b) The response time is a good approximation of wait times for processes and ignores transmission rates.

c) The wait times highlight the areas causing most delays.

This, or a similar report, can provide our benchmark for determining how Goal I is being achieved.

As stated before, system response times are the result of the sum of the individual parts. Having established the benchmark report for overall performance, the next step is to monitor the individual parts. This is done through the activity reports.

Activity Reports

It is convenient to group system activities under three general headings:

Global Activity

Disc Activity

Process Activity

Global Activity

Global activity can be divided into three major parts:

CPU Activity

Memory Manager Activity

Dispatcher Activity

According to our System Tuning definition, we want to maximize efficient CPU utilisation, minimize Memory Manager activity and minimize Dispatcher activity.

The CPU is where the work gets done. The objective should be to maximize the time CPU spends on working for processes and minimize the time spent on performing other tasks.

For convenience, most performance analysis products divide the work performed by CPU under the following or similar headings:

B - Busy

P - Pending I/O

C - Caching

M - Memory Manager

I - Idle

V - Virtual Memory

O - Dispatcher and overhead

A machine under stress will see low Busy percentages and high percentages in other areas. To gauge bottlenecks in a system, the CPU busy state is a good place to start. High percentage levels for each state defined above points to the following problems:

B - Processes are obtaining a high work rate from CPU. The higher percentage time spent in this state the better.

P - CPU is pausing a lot and wating for I/O. If you have a lot of processes running and you are obtaining high "P" percentages, then your machine is incurring I/O bottlenecks.

C - Caching percentages will be high if the system is spending a lot of time managing disc caching. If you have constantly high "C" percentage and low "P" percentages with many processes running, you may need to rethink your disc caching strategy.

M - When memory is scarce, Memory Manager works overtime. High memory activity can also point to bad program segmentation.

V - Similary, when "V" is high, this indicates a lack of memory, as transfers to and from virtual memory are made necessary.

O - Overhead represents the CPU utilisation of Dispatcher. Dispatcher controls entry into the CPU. If it is constantly high, your machine is under stress.

If we look at CPU busy states, along with Dispatcher activity, we have a very powerful indicator as to how things are performing.

The CIA product represents Dispatcher activity through "launch" rates. A launch rate is the number of times that Dispatcher launches a process into CPU. If Dispatcher has high launch rates, it means that the available resources are having a hard time keeping up with processing demands.

Earlier, I stated the theory of system performance relativity, that is, everything is dependent on everything else. The above indicators are a classic example of the application of that theory. You cannot run a system performance analysis tool, see a high memory percentage and then assume you have memory problems. You must look at the CPU busy states and the launch rates relative to each other and your benchmark report - the Session Response Time report. To do this, you must be able to cross reference by time of day and be able to look at the day in its entirety.

If response times are high, launch rates high and the busy state of CPU is low, obviously you have a machine that cannot cope with processing demands.

If, on the other hand, you have low session response times and your launch rates are low with CPU spending most of its time in the "busy" state, your machine is coping very well.

Disc Activity

In monitoring disc activity, there are two areas that need attention:

Disc queue lengths

Relative work rates of each disc

Disc queue lengths point to the inability of a disc to cope with transaction volumes. If queues are continuously sitting at a length of six or greater, then that disc is under stress. If the system disc is continuously sitting at queue lengths of six or greater, then the whole system is under stress.

It is absolutely critical that the queue lengths on system disc be kept low. MPE keeps its tables and directory on system disc. If you open a file on disc 2, MPE has to go to the system disc to look up the directory before going to disc 2. If there is excessive queueing on the system disc, then access cannot take place on disc 2.

It is also important to maintain balance across discs. Those discs with extensive queues could have those queues reduced by shifting active files to the discs with lower or no queues.

It is imperative then that you have an activity report showing the queue lengths on each disc over periods of time. When using this report trends can be highlighted and corrective action taken.

Process Activity

To obtain total control over system performance, the system manager needs as much information on process activity as possible.

The process information required includes when the processes ran and the resources they required during their execution time. This detailed process information can then be crossed checked against the global and disc reports to give a full picture of machine activity and how well the machine coped with that activity.

Defining Problem Areas and Projects

Once the system performance monitoring system is in place, system performance problem areas can be defined and projects to tackle those areas established.

I stated earlier that system performance rules are rarely absolute. Having said that, I will outline several reasons for poor system performance and give the symptoms to look for. If you have a poorly performing machine, you need to start somewhere, and what follows is a list of some of the areas to start looking. I will give these examples in the form of cases.

Case I - Too many processes running simultaneously.

This is a relative easy one to pick. When there are too many processes screaming for attention, Dispatcher works overtime. Firstly, you will notice very high launch rates and associated with this will be very high pre-emption rates. This is the result of high levels of faulting (loss of access) in the CPU as MPE tries to resolve the imbalance between available resources and process requirements.

You will also notice that the CPU busy state will be low and the other CPU states (particularly overhead) will be high.

Obviously session response times will be poor. You will also notice that CPU faulting rates will be high and in particular preempt rates will be very high.

To resolve this situation, besides making all processes more efficient, you have two major options.

The first option is to upgrade the CPU. Leave things as they are and upgrade. This is an expensive option and is great provided that the system load does not keep growing. Sooner or later you are going to either run out of CPU upgrades or money.

The second option is to spread the work load. If you have heavy workloads during the day, but light workloads during the evening, you have the ideal opportunity to ease the daytime workloads by running at night. I strongly urge all installations to look at this option. The night hours are often full of unused processing power.

Workloads can also be spread by ensuring that the job limit is not high. If there are more than say three jobs running concurrently with sessions, then this will cause stress. Remember that jobs do not wait for terminal faults. They will take all the resources they can until they are impeded or pre-empted by a higher priority process. Pre-empts cause dispatcher a lot of work and will impact session response times.

Case II - Lack of Memory

Under these circumstances, Memory Manager works overtime. You will see the CPU spending a lot of time on memory, virtual memory will be high, and, if you have caching, CPU time spent on caching will also be high. Again the busy state will be low.

Also, pay attention to memory allocation rates and the memory cycle time.

Memory allocation is the rate per second that attempts are made to find space in Memory and to make an absent segment (i.e. in Virtual Memory on disc) present. The Memory Cycle rate is the rate per second that Memory Manager went completely through Memory. Each cycle will endeavor to force out Memory segments that should really remain in Memory.

The simplest and most effective solution to this problem is to increase your memory. However, the problem can be eased by ensuring a good balance of workloads and if you have disc caching on, then turn it off (either for all discs or for selective discs). The dynamics of caching are such that the only effective way to control disc caching is to have it managed it automatically. The product CIA provides this facility.

Case III - I/O Bound System.

I/O is potentially your worst enemy for system performance. It is the slowest part of the machine and is all important for any processing.

An I/O bound system will show the CPU with a high percentage of time waiting for "Pending" I/O.

Most noticeably, you will see a high I/O rate for disc drives and extensive queueing on each disc.

The remedies for I/O bottlenecks seem to be endless. Make sure that the workloads on each disc are well balanced by moving active files across discs. Ensure that there are no active files on the system disc. Review blocking factors on files. Review file structures for efficient access to records and ensure that programs are efficiently using disc accesses.

There is enough to be done in the I/O area to ensure several projects. It is also important to constantly review I/O variables in new systems and programs.

Case IV - Some Processes Are Very Inefficient

Weeding out the inefficient processes is a very important part of system performance. You need to be able to identify the culprits so that they can be corrected or at least recognised and rescheduled for less busy times of the day.

If we go back to the System Tuning Definition, there are two things that we require processes to do:

Minimize CPU usage

Minimize Disc I/O

The method I suggest for tracking down the most harmful processes is as follows:

Look at the Session Response Time Report for the worst periods of response time. If you are using a report such as CIA's Process Level Statistics Report, list all the programs that were running during the bad response time period.

Highlight those procedures that have a high CPU second time in relation to run minutes and that have a large number of sectors moved. These are the processes that are chewing up CPU and I/O.

Each program then needs to be individually evaluated to determine if:

1) They should be running at these times of day

2) If they can be made more efficient

An alternative approach is to look at the Process Level Statistics Report for the entire day and highlight those processes that use the most CPU and shift the most sectors of disc. Making these processes more efficient could help system performance overall, not just at particular times of day.

Defining Resource Requirements

If you have a badly performing system, it is going to cost money to correct it. Either a hardware upgrade will be required, or resources will need to be allocated to review existing programs and systems.

Hardware upgrades are solutions that yield quick results. Unfortunately, they may not be the best solution. Software evaluation is the best long term solution, but it takes time.

If you undertake software review and evaluation you will ensure maximum utilisation of hardware resources and have greater control over your system. The potential benefits of this approach are substantial, but you must be prepared to spend time and effort.

There are four prerequisites for undertaking a successful system performance review of software.

Firstly, like all successful software projects, you must have management on side. You will be making recommendations that may cause disruption to current procedures and require investment in new systems.

For example, the installation that is suffering from poor process scheduling will need to reschedule resource hungry processes to "out-of-hours" processing. This means shifting those reports that do endless serial reads during the day to processing overnight. It may be hard to convince some users that they cannot expect to get their reports until the next day.

That same installation may need to invest in a good job scheduling and dispatching system and a product like Omnidex to eradicate serial reads. To make such an investment, management needs to be onside and to understand the goals being sought.

The second prerequisite is technical knowledge. It is no good trying to make a system perform better if nobody has the technical competence to do so. But if you are a system manager of a small shop who has little technical expertise, do not dispair. Technical expertise can be obtained from several sources.

If you have a good system performance monitoring system in place and a strategy for tackling system performance, you can harness the knowledge from others and obtain knowledge from experimentation. The whole point of a system performance monitoring system is to reveal what the system is doing. This information obtained on your machine can be utilised in discussions with third party vendors, HP System Engineers, and consultants. You are no longer at the mercy of a lot of "maybe it is because ofs" and "it probably would be better ifs". You can provide figures that can be evaluated and you can experiment with suggestions and measure the results. You have an ideal learning environment.

The third prerequisite is to have the ability to change your operational environment. Many system managers feel that because they are dependent on a third party supplier and cannot write programs themselves, that there is little they can do about the performance of the software. This is not the case. There are many changes that can be made to improve system performance that do not require one line of code to be changed. This is particularly so in the area of I/O. Master sets in databases can be given more efficient capacities, detail datasets can be better reorganized, blocking factors on files can be made more efficient, files can be better spread over discs, and the list goes on.

There is a lot of literature on ways to improve system performance. Set yourself a project of improving I/O and armed with this system performance material and your system performance monitoring system, clean up existing I/O bottlenecks.

Also, tell your third party supplier that you have performance problems and then show them the reports from your system performance monitoring system - it might help them to "concentrate their minds" if they know that someone is watching.

The last prerequisite required to successfully review software for system performance is the ability to be able to establish software review and implementation procedures. Obtaining good system performance is a constant battle. There are operational procedures that must be performed on a regular basis, programmers and systems analysts must be aware of the repercussions of their programs and system designs, and users must be able to communicate expected increases in transactions.

The system manager's work is never done.

System Performance Monitoring System

So far I have outlined the elements in developing a system performance strategy. An essential part of that strategy is putting a system performance monitoring system into place.

As well as providing the information necessary to develop and implement a system performance strategy, the system performance monitoring system has two additional major functions:

1) Communicate system performance progress to management

2) Facilitate capacity planning

Communicating System Performance

System managers invariably find that the two hardest things to sell to management are system upgrades and utility software.

If there is a system performance strategy in place and the system performance monitoring system is reporting goal achievement, then management will be better able to relate to the system managers requests.

The requests are no longer - "we must buy this upgrade product because we have performance problems". Rather the request can be presented as a well-defined need and solution.

Management can now be approached with the proposition - "As can be seen by the CPU graphs and the Disc Analysis graphs, we need to purchase this software to overcome an I/O bottleneck. This should reduce our current response times from 5 seconds during the day to less than 3 seconds".

Management will be much more receptive to a request that shows potential savings.

Capacity Planning

Capacity planning is simply an extension of system performance monitoring.

When undertaking capacity planning, there are two things that you need to know:

1) What will be the growth rates of my current systems

2) What new systems will I need to carry

In looking at system growth rates it is essential to weed out the important from the unimportant. Using the process analysis reports, it is easy to construct a report that shows the most resource hungry systems on your machine.

Once you have isolated those processes, the next step is to see how well you are performing against your system performance goals. This gives you a tolerance factor. You can then make assumptions about growth rates for each system and see the possible extra workloads that will be placed on the system.

For example, if the accounts system takes up 50% of processing requirements and this system were to increase by 20%, then if we are sitting on the response time goal, we will be pushed over our goal targets.

On the other hand, if we are well below our response time goal, and the system accounted for 20% of the system resources, then a 20% growth rate can be easily accomodated.

For new systems, it is best if you can approximate that system to existing systems, and then extrapolate on the existing system.

Capacity planning is a very imprecise science, but at least we now have the performance information that will allow us to make educated guesses.

In Summary

Developing a system performance strategy is essential in gaining control over an installation. If the performance of a machine cannot be defined then basic decisions on hardware upgrades, software purchases, etc, just become a shot in the dark.

There are three elements to developing a system performance strategy:

1) Putting in place a System Performance Monitoring System

2) Formulation of system performance goals

3) Implementation of system performance projects

The first is required to measure progress and obtain information for sound decision making.

The second is required to obtain a clear communication of system performance objectives.

The third is required to enable system performance to be tackled in a rational and rewarding manner over time.

AUTOMATING

THE

PERFORMANCE MANAGEMENT

FUNCTION

by

Brian Duncombe
c/o
Strategic Systems Inc.
11050 5th Avenue N.E., Suite 101
Seattle, WA 98125
(206) 362-2231

INTRODUCTION

A monumental volume of material has been presented and published over the years in an attempt to help the users of HP3000 computer systems maximize the performance of their systems. This material has generally focused on specific aspects of performance within the HP3000 environment and has provided tips, guidelines and rules of thumb for tuning up the system for maximum performance. In spite the magnitude of this material and the degree to which it is repeated from one source to the next, human nature has prevailed and many if not most of the suggestions have gone unheeded. To a large degree, this apparent apathy has been due to the conflicting demands on the time of those within the organization best equipped to practice the art of performance management. Even when the suggestions are heeded, the tuning of the system is then often neglected until the performance of the system becomes a problem again or the practitioners of the tuning reads an article that motivates them to further review and action.

This paper presents a number of ideas that can help automate the tuning of the system or at the very least minimize the effort required to keep the system in tune.

CONSOLE OPERATIONS

Whenever we have procedures that require human intervention, we have the potential for delays due to the human factor. In many of these instances, it is not only the process requesting or requiring the manual intervention, but the complete system. While the effect of delays in human intervention is often overlooked when studying the performance of the system, they do indeed cause degradations in performance and are therefore a target for optimization.

SYSTEM UPTIME

The worst possible performance problem occurs when the system is not available. In these cases, the question is not when will the system get faster but rather when will the system even be available for use. The causes of system unavailability are probably infinite but a very few account for the majority of the actual occurrences. The following constitute some of the ones that can be addressed in a practical way:

System Failures. When these occur, the system becomes unavailable until someone intervenes and re-starts the system. Correcting the actual cause of the system failure is beyond the scope of this paper but recovering as quickly as possible after a system failure would be a noticeable performance increase. While some shops staff the computer room with an operator, many HP3000 installations manage very well without a dedicated operator. It is not unusual to find HP3000 sites where the nightly batch streams are submitted on the way out of the door and the system is left to run unattended throughout the night. In these cases, the occurrence of a system failure may go undetected for long periods of time. It continually amazes me how long an interactive system can be down before the event is reported. By automating the notification of system failures as well as extended power failures, we can reduce the occurrences of coming in the next morning only to find that the nightly batch processing stopped just after we walked out. Several vendors provide hardware solutions to this type of problem including Telemon (Console Engine) and Design/3000 (CallBack/3000).

System Backup. Even though the system is still functioning, there are procedures required as part of the management of the system that preclude access to certain functions within the system while they are being performed. The most obvious of these is system backup. While backup is a necessity of our environment, it usually requires exclusive access to the files on the system while it is processing. A number of tools exist that attempt to minimize the period when the files within the system are unavailable. In most cases, these tools employ a smarter algorithm and data compression in combination with free disc space to store the backup prior to writing it out to the archive medium. Tools such as BackPack (Tymlabs), HiBack (HiComp), Backup/3000 (Orbit) and OnLine Backup (Orbit) attempt to minimize the impact of backup on the availability of the system.

OUTSTANDING REQUESTS FOR AN OPERATOR REPLY

In many cases, a request for an operator reply can go unacknowledged for some period of time. If the program making the request is part of the critical path for processing, the delay can directly affect the performance of the system. In the case of tape, simply configuring the device as an "autoreply" device can sometimes eliminate this delay. Of course, this is limited to a single tape mounted on each tape device and is prone to errors should another tape be mounted when the autoreply is satisfied.

DISC CACHING

Although disc caching will normally improve disc subsystem performance on the "classic" HP3000 architecture, there is ample opportunity to improve the benefits of disc caching by adjusting the available parameters. Depending upon the current load on the system, the availability of excess main memory and the types of accessing being performed, it is advantageous to adjust the fetch quantums and in some circumstances actually disable caching altogether on one or more of the disc drives. While rule of thumb guidelines exist for this tuning, in order to apply them effectively you must know what is happening within the system at the current moment. Most of us do not have the time to constantly monitor the system and we therefore either ignore the tuning of disc caching or we tune for the general case.

Tuning for the general case. If we don't have any better input to the tuning process, we can usually improve the performance of the disc caching software by using the available MPE command to set the random fetch quantum to some higher than default number and then monitor it with the showcache command for a while to see if we can improve it a bit more. Generally speaking, setting the random fetch quantum to approximately 60 sectors ("cachecontrol random=60") is a good compromise setting.

Customizing generalized tuning. If we can somehow determine that the caching parameters should be set depending upon the time of day, then we can set up a series of batch jobs that can be streamed at fixed times of the day (using the ";AT=" parameter of the stream command) and have the jobs re-stream themselves. Of course, if we have access to one of the job scheduling packages, we can be more elaborate in our specification of when to adjust (eg. after a certain job has completed).

Continuous monitoring and adjustment. Since conditions within the system can change almost unpredictably, the best method for managing disc caching is to use software that continuously monitors the system and caching effectiveness and applies the generally accepted rules for tuning the caching throughout the processing day. A contributed program (DCO, disc cache optimizer) is sometimes available through the SE field service organization and does a fairly good job of dynamically tuning the disc caching subsystem. Several third-party products exist that as part of their function will tune disc caching to some extent. These products include Gofaster (Strategic Systems Inc.) and KLA/Express (KLA Associates).

DISC FREESPACE

The availability of disc freespace can affect performance in two general ways. In the first place, a shortage of freespace or very fragmented freespace on a disc drive can cause the system to perform sluggishly when performing any disc space management (especially on ldev#1). In the second instance, having insufficient disc freespace or freespace that is in small fragments can cause a program to abort simply because it cannot get the space that it needs to continue processing.

<u>Deleting unused space.</u> Most computer systems tend to collect disc files that are really not required to be available on-line. For the most part, this occurs simply because someone forgets to purge the file when they are finished with it. In other cases, the files may be historical and could easily be stored to tape and then deleted. Still other files appear on the system as a result of program aborts leaving intermediate workfiles.

The reports produced by regularly streaming a job that runs free5 will act as an stimulus to at least think about the subject. It will also provide a historical trail of how disc freespace varies over time.

You can automate the archiving and purging of system logfiles by instituting a procedure that is carried out at regular intervals and involves storing the log###@.pub.sys files and then purging them.

You can make a habit of purging all k###@ files (editor work files) at regular intervals. The contributed library contains at least one program aimed at automating this procedure.

Using the "store" command with the appropriate date specifications, you can store all files not accessed since a certain date and then purge them from the system. If someone subsequently needs the file, it is available from tape and the user is gently reminded that they should be managing the file themselves and keeping it off of the system when not required. The MPEX tools from VESOFT are particularly helpful in this type of disc space management. A fairly recent addition to the available software that is aimed directly at automating these types of procedures is DiscMaster (Unison) which allows you to specify your criteria for when to compress, when to archive and when to trim disc files to reduce wasted space at the end of the last extent.

<u>Balancing freespace across the available disc devices.</u> In most cases, it is more convenient and easier to manage a system when disc freespace is relatively evenly distributed amongst the available disc devices. For those systems that contain disc devices of varying storage capacities, the default device configuration will cause the smaller discs to fill up faster than the larger discs. To make the balancing more automatic, you should use the sysdump facilities to perform "device class" changes and for the device class "disc" specify ldevs in proportion to their storage capacity. For example, a system with two 7925 disc drives (120MB each, ldevs 1&2) and two 7933 disc drives (404MB each, ldevs 3&4) should specify the device class disc to be "1,2,3,3,3,4,4,4" in order to distribute new file space proportional to the capacities of the devices. The same technique can be used for other device classes that you frequently use (eg. spool).

Defragmenting disc space. Most HP3000 installations would benefit by regularly compacting the freespace on the disc drives. As long as the situation has not been allowed to deteriorate drastically and there is sufficient volume of disc freespace on the drive, the VINIT subsystem supplied with MPE can be used to condense a disc drive utilizing the "cond" command. This can be setup into a batch jobstream (":run pvinit.pub.sys" gets you the vinit subsystem) and on a regular basis, you can defragment your disc drives. You might schedule one or two disc drives per night or do them all every weekend depending upon how quickly the discs become fragmented again as well as how available processing time is. When freespace on a disc drops below a certain value (about 10%), the disc freespace usually becomes very fragmented and the effectiveness of the "vinit cond" function is severely reduced. In this case, the only alternative is to move some files from this device to another and then try again. If this still doesn't work, the final alternative is to "reload" your files.

IMAGE SET CAPACITY

When it comes to performance tuning, the subject of the Image subsystem has received perhaps more attention than any other single area. This is only logical since Image represents the primary data management tool for most HP3000 users. In addition, because data is concentrated into databases that are then shared by many users, the database becomes a natural focus for efforts that will be leveraged amongst many of the applications and users thus maximizing the tuning effort. While there are many particular aspects that can be addressed, the management of dataset capacities is one that can be automated to some extent with little or no difficulty.

Exceeding the capacity of the dataset. When a dataset is full, attempts to add entries to it cause errors which usually force the programs to terminate abnormally. This can be looked at as severe performance degradation. The solution, of course, is to increase the capacity of the dataset which is fairly time consuming and if a set fills up in the middle of the day, can be a major disruption of services. What we would really like to do is detect and correct this trend before it becomes a problem.

Monitoring capacities with query. The simplest technique is to schedule a job to be run regularly that simply uses query.pub.sys to perform a "form sets" listing for the database. This provides a concise and easily collected printout that can quickly be reviewed for pending problems.

Using Howmessy (Robelle) to monitor database capacities. This program, which is available to Robelle customers or the contributed library version (dbloadng) which performs the same functions but at a considerably reduced speed, provides a great deal more insight into the status of the database and it's internal organization. Scheduling a batch job to run this program at regular intervals will provide you with an excellent tool for quickly reviewing the state of your database.

Automating capacity management using supported tools. Many of the popular database management tools provide a function to review dataset capacities and automatically adjust them according to simple rules specified by you. Generally, these rules include the sets you wish to manage combined with the upper and lower limits for dataset percent fullness before a capacity adjustment is invoked. These tools include DBGenrl (Bradmark), Flexibase (Proactive) and DBTune (HiComp).

Master dataset synonym chains. As records are added into master datasets, the algorithms used to locate the record based upon the value of the key can begin to deteriorate and cause collisions to occur. When these collisions happen, the subsequent records that would normally be placed at the same disc location are linked together into synonym chains and located as close to the primary location as possible. This means that when we later access these records using the key value, we may be unknowingly traversing a linked list of records with it's reduced efficiency. In addition, as the dataset fills up, it becomes less and less likely that synonyms will be located near the primary location. The report produced by Howmessy (or dbloadng) will highlight the percentage of entries that are synonyms, the length of synonym chains and the maximum distance that may be required to be searched in order to find a free space to place a new synonym. The techniques discussed in the preceding capacity management section are equally relevant to correcting synonym problems. In general, a master dataset will not begin to exhibit synonym problems until it is about 75% full although there are of course many exceptions to this general rule of thumb.

Changing dataset capacities. A variety of methods are available for changing dataset capacities. In most cases, they are very time consuming and require exclusive access to the database while they are being performed. For this reason, the goal of dataset capacity management should be to detect the need early enough to schedule the correction into a time when it is the least disruptive. While you can simply use dbunload, a change of the capacity in the schema, a rebuild using dbutil and dbload to perform capacity changes, this method requires that you unload and reload all datasets when it is usually one or two datasets that need adjustment. This reason alone should justify acquiring one of the supported utilities that allows a single dataset to be managed. The supported tools include Adager (Adager), DBGenrl (Bradmark), DBMgr (DISC), DBTune (HiComp) and Flexibase (Proactive).

IMAGE DETAIL SET ORGANIZATION

Just as capacity management of Image databases is an important tool for globally optimizing the performance of most HP3000 computer systems, so too is the practice of periodically re-organizing the data within detail datasets. Over time, older entries are usually deleted from datasets and new entries are added. Since Image tries to re-use deleted space within a dataset before it assigns new entries to previously unused areas of the dataset, records chained together on a common key value tend to be widely distributed within the dataset. When we subsequently read up or down a chain, the probability of related entries being physically adjacent on the disc medium decreases with time. By periodically reviewing the internal state of our datasets and where necessary re-organizing the entries to improve physical placement we can sometimes gain substantial improvements in performance. Like most other good ideas, this one tends to slip to the bottom of our to do list until we are forced to address severe performance problems. By implementing a procedure of first determining the timeframes in which individual datasets become disorganized and then setting up scheduled procedures to repack the datasets during more convenient time periods, we can keep our databases performing at higher efficiency levels.

<u>Detecting the problem.</u> Programs such as Howmessy (Robelle) and DBLOADNG (contributed library) will provide statistics that reflect the state of individual datasets. The key indicators are "Ineff Ptrs" and "Elongation" which highlight chains in which the related entries are not physically adjacent on the disc. Since many detail datasets have multiple chains with which they can be accessed, we can really only pack the entries for one chain per dataset. This means that we must look for chains with longer average lengths combined with our knowledge that the chain is frequently used.

<u>Correcting the problem.</u> In order to re-pack the related entries for a detail dataset, we must logically unload the dataset in sequence by one of it's chains and then reload it. This can be done using dbunload, dbutil (erase) and dbload although just as capacity management using this technique requires all sets to be processed, so does chain reorganization. Again, the preferable method is to use one of the supported database maintenance utilities such as Adager (Adager), DBGenrl (Bradmark), DBMgr (DISC), DBTune (HiComp) and Flexibase (Proactive). Once you have determined the datasets that will benefit from this activity and can estimate the frequency with which it should be carried out, you can schedule individual jobs to perform the re-packing on a regular basis.

JOB SCHEDULING

Most HP3000 shops run some batch work. In fact, batch processing is a significant part of the total loading on many systems. Since batch processing is by it's very nature not time critical in the same way as interactive response is, the existence of batch oriented workload provides us with a portion of our processing workload over which we can exert some control. There are several objectives that batch processing allows us to address including:

Workload balancing. One of the generalized techniques for performance improvement is to spread the load out so that some of the workload that is normally done during a peak processing period can be deferred until some time when the system is less utilized. Batch workload lends itself very nicely to these attempts at levelling out the workload. By adjusting the limit on the number of concurrent batch jobs, we can keep them from impacting the system during the severe peak processing periods of the day and then allow them to run during the periods of the processing day when the system is normally less utilized. The simple technique of scheduling jobs using the "AT=" parameter of the stream command allows us to determine when a job will be submitted to the system. By using this technique to submit a job (with ;hipri) that uses the limits command to adjusts the number of concurrent batch jobs, we can implement a plan to minimize batch processing in the 9:00Am through 11:30AM peak interactive processing period and defer the batch processing to lunch time.

Keeping the system busy. In many cases, when an operator is forced to get involved in making decisions about such things as what to run next, the system is often left relatively idle for the duration of the operator think time. In addition, the operator is sometimes not readily available when the decision must be made and the delays are consequently compounded. By minimizing the requirements for human intervention and the use of job scheduling software, these types of delays can be avoided. When the jobs have little or no inter-dependencies, the use of the limit command in conjunction with the ";AT=" parameter of the stream command can prove sufficient for keeping the system working. As the number of job inter-dependencies and complexity increases, supported software tools that provide more extensive control over the scheduling of batch jobs such as MAESTRO (Unison), JMS (Design/3000) and OCS (Operation Control Systems) may make good business sense.

Automating recoveries from program aborts. One of the primary reasons for human intervention in many batch processing environments is simply to handle the case of program aborts and the attendant recovery procedures. In many cases, these procedures have been comprehensively defined and could be included either as conditional logic within the jobstreams themselves or through the more comprehensive facilities of one of the supported job scheduler packages.

PRIORITY

A multi-user operating system such as MPE can really only service more than one requirement for processing resources at a single time. Because of this, an algorithm has been devised that attempts to distribute the available resources according to the relative "priority" that is assigned to each of the competing tasks. To achieve this prioritization, the various users of the system have been grouped into what amounts to four distinct entities identified as the linear, CS, DS and ES scheduling queues. These scheduling queues are then assigned ranges of priorities so that conflicts can be resolved based upon both the generic queue that a process is assigned to as well as the specific priority within the range that is currently assigned. By default, we reward "good" interactive users (CS subqueue) at the expense of "bad" batch users (DS and ES subqueues). Since it is normally advantageous to provide optimum service to interactive users, this scheme is generally a good one. The MPE operating system also gives us the ability to reconfigure the priorities of these queues to suit our specific circumstances. In order to customize the system to our current processing objectives, we can do the following :

Schedule jobs to adjust the TUNE command. Depending upon our objectives, we can configure the TUNE command to allow cpu intensive interactive processes to sink in priority below those of our batch jobs or we can give higher priority to our batch processing. In addition, we can exercise some control over how quickly a particular job sinks within the priority range for the assigned queue. This gives us some degree of control over the workload. By identifying at which times of the day we wish to set the tuning parameters to certain settings, we can automate these actions using batch jobstreams (containing our tune command) that are submitted at the appropriate time using either the ;AT= parameter of the stream command or one of the available job scheduling systems.

Use a "Performance Manager" software facility. If we are interested in a finer control over the way particular processes are prioritized, we can use one of the third party software packages that have been designed with this in mind. These packages allow you to define such things as additional subqueues as well as identify which processes will be assigned to which subqueues based upon the program file that they are running and / or the logon identity of the user. With this degree of control, you are able to specify "good" and "bad" processing loads at a much finer detail. Of course, the parameters within these systems can be adjusted depending upon the time of day as an aid in adjusting your specific processing objectives throughout the processing day. The tools available for this type of control include Gofaster (Strategic) and KLA/Express (KLA Assoc.).

TOOLS MENTIONED

Adager	PO Box 2358 Sun Valley, ID 83353 (800) LDD-REGO
BackPack	Tymlabs Corporation 811 Barton Springs Road Austin, TX 78704 (512) 478-0611
Backup/3000	Orbit Software 319 Diablo Rd., Suite 218 Danville, CA 94526 (415) 837-4143
CallBack	Design/3000 Inc. 860 Lancaster Dr. SE PO Box 13086 Salem, OR 97309-1086
Console Engine	Telemon 492 Ninth Street, Suite 310 Oakland, CA 94607-4098 (800) 622-0630 (916) 622-0630 (503) 585-0512
DBGenrl	Bradmark Computer Systems Inc. 4265 San Felipe Avenue Houston, TX 77027 (713) 621-2808
DBMgr	Dynamic Information Systems Corp. 910 Fifteenth Street, Suite 640 Denver, CO 80202 (303) 893-0335
DBTune	Hi-Comp Hinrichs GmbH Eichenlohweg 24 2000 Hamburg 60 West Germany 49-40-630-40-11
DCO	sometimes distributed as part of TELESUP account
DiscMaster	Unison Software Inc. 415 Clyde Ave. Mountain View, CA 94043 (415) 968-7511
Flexibase	Proactive Systems Ltd. Central Court, Knoll Rise Orpington, Kent BR6 0JA England 0689-77933
GoFaster	Strategic Systems Inc. 11050 5th Avenue NE, Suite 101 Seattle, WA 98125 (206) 362-2231

TOOLS MENTIONED continued

HiBack	Hi-Comp Hinrichs GmbH Eichenlohweg 24 2000 Hamburg 60 West Germany 49-40-630-40-11
Howmessy	Robelle Consulting Ltd. 8648 Armstrong Road RR#6 Langley, British Columbia Canada V3A 4P9 (604) 888-3666
JMS	Design/3000 Inc. 860 Lancaster Dr. SE PO Box 13086 Salem, OR 97309-1086 (503) 585-0512
KLA/Express	KLA Express Clearwater, FL (813) 784-5976
Maestro	Unison Software Inc. 415 Clyde Ave. Mountain View, CA 94043 (415) 968-7511
MPEX	VESOFT 1135 S. Beverly Drive Los Angeles, CA 90035 (213) 282-0420
OCS	Operations Control Systems 560 San Antonio Road Palo Alto, CA 94306 (415) 493-4122
OnLine Backup	Orbit Software 319 Diablo Rd., Suite 218 Danville, CA 94526 (415) 837-4143

HOW TO ANALYZE DATA BASE PERFORMANCE

Randy Safier
Proactive Systems
1411 S. Woodward, Suite 201
Bloomfield Hills, MI 48013
(313) 333-7200

Abstract

This paper discusses the parameters that affect the user response time and throughput of IMAGE data bases. Procedures and tools available for obtaining the necessary diagnostic information are reviewed. Guidelines for optimizing performance are proposed and ways to implement the necessary changes are discussed.

INTRODUCTION

This paper is about getting the best performance out of your data bases. Performance problems often arise in IMAGE data bases. Often a resolution of the problem is critical for on-going operation of your application systems. But there can also be inefficiencies hidden inside your data bases which are taking a toll on performance without you being aware of them. They simply sit there eating up some of your expensive resources. What I want to discuss is attacking these two types of problems with the minimum expenditure of effort, staff time and money. The proposal is that we can use the principle of an Expert System to achieve this.

Performance monitoring and improvement is an ideal candidate for an expert system. There is an abundance of knowledge about IMAGE data bases and their performance characteristics in the literature. If you look through any of the Interex Conference Proceedings you are likely to find discussions on various aspects of IMAGE Databases and I have brought together some of these in a bibliography at the end of this paper. It is not that knowledge is unavailable, there is quite a lot of it. It is not that it is difficult to understand, but that it is difficult to retain all the knowledge. An expert on IMAGE data bases may have no problem with the quantity of knowledge but for someone who is designing and writing systems, or managing a Data Processing Department, it is almost impossible to remember every little quirk, every little rule that you must or musn't follow when you design a data base. It is also difficult to find enough time to regularly monitor all the parameters that affect operational performance.

Firstly we need a system for measuring the appropriate parameters - looking at all the sets and the structure of a data base. Secondly we could use those parameters with a rule-based system to come up with recommendations about what should be changed. And then we can actually implement those recommendations (although at that point some manual intervention may be necessary).

In the rest of this paper there are four areas I will cover:

1) I will analyse what we mean by performance and what it is that affects performance.

2) I will look at the measurement of the various parameters relevant to IMAGE data bases, and which ones are applicable and useful.

3) I will look at what rules we can actually adopt by comparing parameters with norms, or by looking at parameters in conjunction with each other, to try and decide what changes could usefully be made to a data base.

4) I will look at the consequences of actually implementing some of those recommended changes.

FIGURE 1

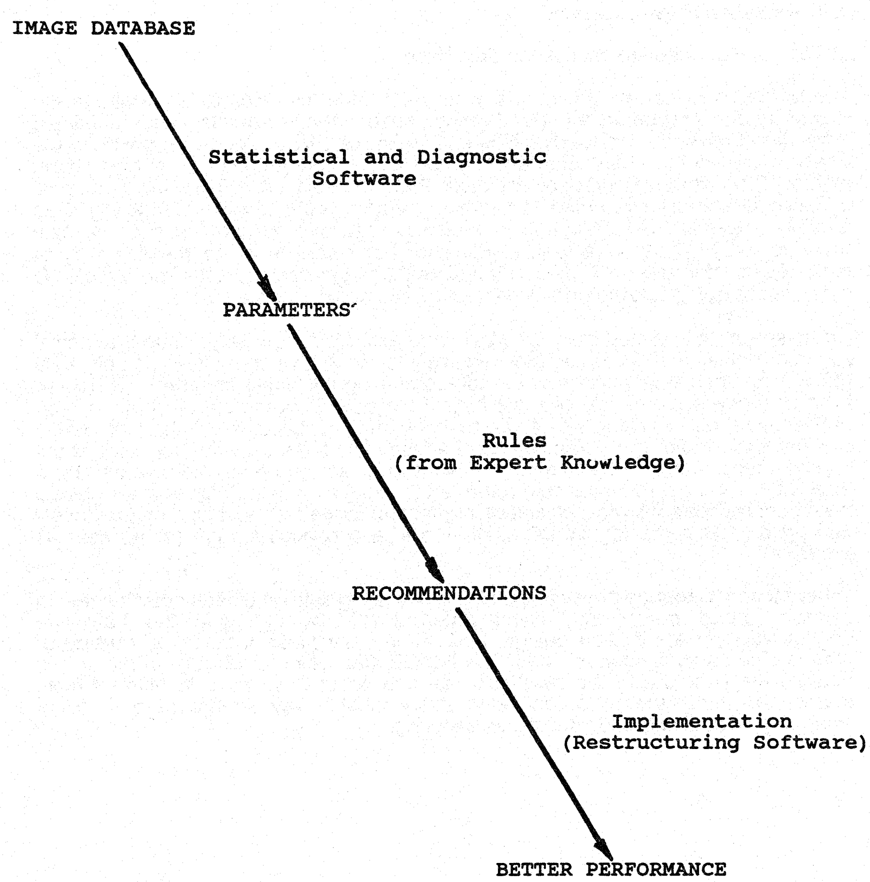


IMAGE DATABASE
Statistical and Diagnostic Software
PARAMETERS
Rules (from Expert Knowledge)
RECOMMENDATIONS
Implementation (Restructuring Software)
BETTER PERFORMANCE

PERFORMANCE

What are the primary factors that affect performance? In relation to IMAGE data bases, performance depends on four main factors:

a) The structure of the data base (i.e. the design of it and the surrounding application).

b) The content of the data base (i.e. the data it contains).

c) The location of the data base.

d) The software used to access the data base.

The prime constraint on the above is disc I/O. Most commercial applications are limited by disc performance rather than cpu speed. We immediately should make a distinction between physical disc I/Os and logical disc I/Os. We all are aware of the principle of caching - that when a disc I/O is made more data is brought in to main memory than was requested, on the basis that if the next call asks for the next serial piece of data then it will already be in main memory and that will avoid having to do another physical I/O. Requesting a disc access and finding the data in main memory would then be a logical I/O, and that leads us on to the first way of improving performance - if we can translate these physical I/O calls into logical I/O calls we are going to save ourselves a lot of time waiting for the disc.

But even on non-cached machines the same thing is happening in that the logical records (entries in IMAGE) are blocked together so that when we ask for a physical I/O we get a series of consecutive entries coming into main memory. Of course both of these systems are of most help if the actual entries we want are indeed consecutive, and anything we can do to make them consecutive will therefore help. Having made those two points we have already laid the foundation for most of the improvements that can be made. We want to make sure that when we ask for a physical I/O we get the maximum return on it - both by making sure that we get the maximum number of logical pieces of information as we can, and also by previously making sure that the logical information is in a consecutive form (or as near as possible).

Other means to reduce physical disc I/O are locating sets on different disc drives (to minimise head movement), using deferred I/O, by changing the hardware configuration or application design, and so on. But most of these lay outside of what we can actually control within the IMAGE data base. And also on the end of that list we should add the overhead that can occur if we have insufficient main memory, or things are not organised in a very suitable way, so that there is a lot of swapping of data and code segments occuring.

FIGURE 2

DATABASE PERFORMANCE

THE STRUCTURE OF THE DATABASE:

Poor design or ineffective normalization.

Unneccesary, or not enough, Paths into Sets.

Detail Sets that should be Master Sets, etc...

THE CONTENT OF THE DATABASE:

High Percentage Full Master Sets with inefficient Secondaries.

Excessively long chains, sorted or otherwise.

Messy Detail Chains etc......

THE LOCATION OF THE DATABASE:

Remote Databases accessed over DS/NS lines.

Mutually-accessed Sets on same Disc Drive etc.....

THE SOFTWARE USED TO ACCESS THE DATABASE:

TurboIMAGE or IMAGE.

TurboIMAGE/V or TuroIMAGE/XL.

Poorly written user programs etc....

PERFORMANCE IN MASTER SETS

So let's return to our theme of getting the maximum possible return of logical I/Os for each physical I/O and see how that is effected on an IMAGE data base. We first need to review the structure of the IMAGE data base. Let's look at the structure of a Master Set (and that includes Automatic Master Sets) and the problems that can happen. The set can be thought of as a contiguous strip of disc divided up into Blocks (see Figure 3), and the Block would correspond in normal circumstances to what is read with one physical I/O. Within that Block there can be several potential entries, as defined by the Blocking Factor, and if you are working on a cached machine you will be reading in a multiple of Blocks so you may in that case be getting in perhaps 30 or more Blocks with one physical read. Now the whole point about Master Sets is that each entry is accessed by the use of a Search Item which makes it unique and the position in the Set where the entry is to be found is determined by applying an algorithm to that unique key.

If our Search Items are reasonably random and the algorithm is doing a good job then we would expect our entries as we put them into the Set to be fairly randomly scattered throughout the length of the Set. If the data base is getting fairly full it is possible that when we want to put another entry into the position which is calculated from it's Search Item that it will already be occupied by an entry which has previously been put there. In that case what IMAGE does for us is find another vacant slot as close as it can, hopefully within the same Block as the first (known as the Primary) entry. The new entry that we have just put in is then called a Secondary. Now IMAGE keeps pointers attached to all these entries so that if we want to find the one that went into a Secondary position, the algorithm tells us the position that it should occupy and we find that it's in fact occupied by another entry. That entry then will be able to point us along to the Secondary. It could be a long chain of Secondaries until we find the particular Secondary that we are looking for, if indeed it's in the set at all, and then if it isn't we will find the end of chain.

So, this structure gives us our first opportunities for increasing efficiency. If the set is becoming very full up then we are exponentially more likely to find that any new entries we add will try to go to positions already occupied by Primaries. So we will be getting longer and longer Secondary (also called Synonym) chains. Another possibility for long Synonym chains is that the algorithm may not be very suitable for the type of data of the Search Item that we are using and that it is not producing a very random pattern. In other words entries will be "clustered" together through that Set and that also can raise the possibilty of high numbers of Secondaries, and longer Synonym chains. Now we have already said that IMAGE will put those Secondaries as close as possible to the Primary entry and if they are in the same Block as the Primary then we only require extra logical I/Os to find it and not extra physical I/Os. So a high proportion of Secondaries doesn't necessarily matter, particularly if we have a large Blocking Factor. What we can say is that Secondaries are really only a problem where they occur frequently outside of the Block containing the Primary.

However, it's not only reading along the Synonym chains that we need to consider. We also need to consider what happens if a new entry is to be put into the Set and what would happen if it was directed towards not only a position that was already occupied by a Primary but also that all the other potential entries in that block and succeeding blocks had been filled up. This could occur either because the Capacity has nearly been used up or because the entries are clustering together in one part of the Set. Now in either of those two cases, the HP3OOO is going to have to do a lot of physical I/Os in order to find a vacant slot to put that new entry into. In some ways that might sound like a far-fetched problem but there is one specific quirk in IMAGE which can and does cause it to happen. If the Search Item of that entry is of a character data type such as type X,U or Z then randomised hashing will usually produce a good spread of the entries across the Set. If however, the data type is of a numeric type (that's type R, I or J) then a different algorithm is used. What happens is that the numeric value is simply taken, modulo the Capacity, to be the Set position for that entry. Now if the Search Item value never exceeds the Capacity that will be fine because every entry by definition would have a unique position and that would work even better than the randomised hashing technique. If the Search Items are frequently greater than the Capacity, then by taking a modulus, we can get a lot of repetition in terms of the entry position into which that entry is to be put.

One further problem that is frequently encountered with numeric data types for Search Items is the "Four-word Search Item". In the case of a Four-word Search Item only the upper thirty-two bits are used by the algorithm to locate the entries in the Set. Now this means that unless the values of the Search Items are more than a certain amount then all the entry positions are going to be returned as zero giving you just one long Synonym chain. That is something we've actually seen on a users data base resulting in a Synonym chain of two thousand entries. The strange thing about it is that IMAGE will run quite happily. When you ask it to do a DBGET it will find the entry for you. It will take a little while, but it will do it and when you ask it to do a DBPUT, it will do it but again it will take some time. And you may well find that end-users grow accustomed to it. They don't ask any questions, they just think that the computer takes a long time to find things. It's only when something else changes that the problem can come to light. So here is one of our golden rules:

Don't assume that everything's fine just because the problem hasn't surfaced!

There may well be gross inefficiencies hanging around your data bases consuming resources and waiting for the day when they can turn into a critical problem.

One possibility for improving the performance of a Master set would be to increase its Blocking Factor. That is to say, increase the size of the Blocks allowing more entries to be packed into them. In some cases that may well be of some help. However, there are two or three things we need to consider before doing that. First of all the size of the Buffers that IMAGE uses to hold these Blocks in main memory has to be the same for all the Sets in the data base. So we need to decide on the Buffer's size and then change the Blocking Factors of all the Sets, the Masters and Details, to bring the Blocksize up as close as possible to the chosen Buffer size. Now that's good because it makes sure that we make absolute maximum use of all the physical I/Os and of the memory space that we are using to store the results of those physical I/Os.

However, increasing the Blocking Factor may not show much improvement because cached reads are bringing into main memory a consecutive series of Blocks in any case for each I/O. Also, in a machine with a shortage of memory, larger block sizes mean more memory usage and potentially more swapping of segments. However on many non-cached machines, there certainly will be some improvement by increasng and optimizing the Buffer sizes of all the Sets. Later in this paper I will discuss how Blocking Factors and Buffers interact and thus affect performance.

To recap then for Master Sets we should look at:

a) The Number of Secondaries.

b) The Percentage Full.

c) The Clustering of Entries.

d) The Block size and Blocking Factor.

e) The Search Item type.

FIGURE 3

```
                         MASTER SET STRUCTURE

-------------------------------------------------------------------
!  :   :   :   :  !   :   :   :   :  !   :   :   :   :  !   :   :   :   :  !  :  :
!  :  :X  :  :X  !  :X  :X  :X  :  !  :X  :   :  :X  !  :X  :X  :   :X  !  :  :
!  :   :   :   :  !   :   :   :   :  !   :   :   :   :  !   :   :   :   :  !  :  :
-------------------------------------------------------------------

      BLOCK 1           BLOCK 2          BLOCK 3          BLOCK 4          . . . .


                         DETAIL SET STRUCTURE

+-------------+
| DELETE CHAIN |
+-------------+
             |
             |
             |------------+-----------------+--------+
             |            |                 |        |
-------------------------------------------------------------------
!  :  :  :  :  !  :  :  :  :  !  :  :  :  :  !  :  :  :  :  !  :  :
! X: X: X: X: O! X: X: O: X: O! X: X: X: X: X! X: O: X: X: O! X: X:
!  :  :  :  :  !  :  :  :  :  !  :  :  :  :  !  :  :  :  :  !  :  :
-------------------------------------------------------------------
                 |            |         | |             |  ^
    BLOCK 1      |  BLOCK 2   | BLOCK 3 | |  BLOCK 4    |  |
                 |            |         | |             |  |
                 |------------+---------+-+-------------+  |
                 |                                         |
                 |                                         |
                 |                                   HIGHWATER MARK
    +-------------+
    | SEARCH ITEM |
    |    CHAIN    |
    +-------------+
```

PERFORMANCE IN DETAIL SETS

Let's move on to look at the structure of a Detail Set. A Detail Set has a very different structure consisting as it does of chains of entries which are linked together by pointers (see Figure 3). All the entries in any one chain have a common Search Item value which is also shared by an entry in a Master Set. Indeed they can be linked in entirely different and seperate chains to other Master entries where that chain would contain a common value in a different data item. Entries are generally added to the end of a Set until such time as deletions take place and any holes left over by these deletions are linked together into what is known as the Delete Chain. The beginning and end of the Delete Chain are stored in the user label of the Set so that entries can be re-used and the available entries can be found without the need of a serial read.

Let's take first of all serial access to the Detail Set. That is going to take the minimum of physical I/Os when all the entries are packed together and none are deleted. The more deleted entries there are the longer will be the necessary physical reads to get to what is called the Highwater Mark (ie the logical end of the Set).

The other sort of read is the chained read where we are picking up each of the entries in a chain which is related to a particular Master entry. That's going to be at its most efficient if all the entries in that chain are placed contiguously in the Set, and conversely its going to be inefficient if consecutive entries on that logical chain are dotted all over the Set, particularly when the next entry is not in the same Block as the previous entry. Again the arguments that I used when talking about the Master Set and the Blocking Factor and the Block Length equally apply to the Detail Set. We can get more logical entries in larger Blocks subject to not restricting the number of Buffers available, and the amount of available main memory.

There is an approach that is fundamental to improving the performance of both the serial and chained read on the Detail Set and that is to take the entries out, sort them into order of their chains and then put them back in. That process is called Re-packing. It helps in two ways, first it means that in a chained read every time you read a Block of entries, you'll automatically get the next series of logical entries in the correct sequence; and secondly by removing the Delete Chain entries are completely packed together, reducing the number of physical I/Os necessary to get right through the Set from the beginning to the Highwater Mark in a serial read.

Another feature of IMAGE structure which is often used on Detail Sets is the sorted chain in which the particular entries which go to make up the chain are sorted on the value of a particular item. This places an overhead only on DBPUTs and DBDELETEs, but it can be a very substantial overhead if the chains are long because the DBPUT will need to read through the chain. In fact it reads backwards from the end of the chain to find the appropriate point in that chain to put the new entry into. People are fairly wary of using sorted chains, probably with good reason, but they do have good uses provided there are not too many entries on a chain or providing that the entry length is quite small and the Blocking Factor is reasonably large, or provided that the entries are added in more or less sorted order. In those cases the DBPUT overhead will be minimised.

For Detail Sets then, we should look at:

a) The Messiness of the Primary Path.

b) The Number of Deleted Entries.

c) The Percentage Full.

d) The Block size and Blocking Factor.

e) The Number of Sorted Chains.

BLOCKING, BUFFER SIZES AND BUFFSPECS

The thing we should be aware of with Buffer sizes is that there is a limited space available in main memory for storing these Buffers. That space is 32K words less about 6K depending on the actual nature of your Database (see the IMAGE manual or the IMAGE Handbook for the exact details). In order for IMAGE to work efficiently in terms of putting new entries or deleting entries, a certain number of Buffers are required to support that operation. It depends on the number of paths affecting the set into which an entry is being put or deleted from (again the IMAGE Manual gives the relevant calculations). For any particular data base, this figure of the minimum required number of Buffers can easily be calculated. It is considered as a minimum because if the data base has less than this number of Buffers then on certain DBPUTs some of the Buffers will have to be flushed and re-used in order to complete the DBPUT - and obviously your programs will have to wait while this is done.

The number of Buffers for a specific Database can be defined by running DBUTIL and setting the "Buffspecs". These specify the number of Buffers IMAGE should use for a range of users; eg 8 for up to 4 users, 10 up to 6 users and so on. Another performance factor comes into play here - when the number of users of the Database changes and IMAGE has to change the number of Buffers it has to lock all access to that Database until the operation is complete. For a Database with a dynamic number of users this could give lumpy response times. We would recommend setting the same number of Buffers for all numbers of users unless your environment requires different numbers of Buffers at different times. It is difficult to find any knowledgeable discussion on how IMAGE uses it's Buffers and so it is not easy to recommend what the maximum number of Buffers should be - perhaps as a rule of thumb one could specify twice the minimum number for all users. This is an area I would like to see explored further in later papers.

Having set to the Database at least the minimum number of Buffers required this can then be used to calculate the potential maximum size that each Buffer can be. The Buffers are contained in main memory in an Extra Data Segment which has a maximum size of 32K words. Some of this 32K is taken up by other IMAGE data but in TurboIMAGE there is a well defined maximum space available for Buffers. In this case it is a simple matter to divide that space by the maximum number of Buffers specified in the Database's Buffspecs to obtain the potential size of each Buffer. As we have said, all Buffers are of the same size and by comparing each Set's Blocking Factor to this Buffer size we could potentially increase throughput by increasing the number of entries in each physical I/O. Balanced against this we must recognize that we are increasing main memory usage by increasing TurboIMAGE's Buffers. However Turbo's memory requirements are significantly higher than IMAGE anyway because of it's increased number and size of Extra Data Segments. The Buffers are located in only one Extra Data Segment per data base and so this increase should be relatively insignificant on all but heavily memory-constricted machines.

LOCKING AND TRANSACTION LOGGING

That concludes the discussion on the relevant structure of IMAGE so that now we can build up parameters with which we can measure efficiencies or inefficiencies within that structure. No discussion of performance however is complete without mentioning Locking and Logging. There's one golden rule of what not to do with locking and that is to have a keyboard input after a Database Lock but before the Unlock. That is the quickest and most effective way to halt an IMAGE Database, particularly if the operator goes off to lunch before pressing carriage-return. Other than to say that locking should be coordinated, ie everyone should be doing it the same way in an installation, the rest of locking theory is outside the scope of this paper.

The idea that performance will be degraded by turning on transaction logging is really something of a myth. Figures that we have seen and our own experience will indicate that it's virtually unnoticable that transaction logging is operating. That doesn't however go for ILR. If you want the additional convenience and security that ILR will give you, you will reduce the speed of your transactions to disc by about half. Although of course there are no overheads there for read intensive software.

MONITORING PERFORMANCE

How do you determine whether any of the above problems are present in your data bases? There are a number of pieces of software around which will read through sets or entire data bases and give you a certain amount of statistical information, including the parameters we have touched on already. Probably the most well known are DBLOADNG or HOWMESSY with the former being in the Contributed Library. Our own experience comes from the FLEXIBASE product which we produce and which contains the DIAGNOSE module in it. This goes one stage further in that it not only gives you the parameters, it will also analyse the figures and make recommendations as to what should be done to improve performance.

In running through these parameters I'm going to pick out the salient ones; obviously basic information is given such as the name of the Search Item, its type, what is the Primary Search Item, the number of items in the set, the entry length and the number of paths and so on. These figures are easy to obtain from many sources such as QUERY and together with the Capacity and the Block Length, they are fundamental to making any decision about the set. Let's take the Master Sets first. Apart from the Percentage Full, which is a very important statistic, information is given on the average length of the Secondary chain. A Primary with no Secondaries is considered to have a chain length of one. Now the most important statistic that comes out within DIAGNOSE for a Master Set is called the Percentage Inefficient and we define it in this way:

Percentage Inefficient = The number of Secondaries in the set which are located in a different Block to that of the Primary, as a percentage of the total number of entries in the set.

This is the prime indicator of any inefficiency in the set because it tells us how many extra physical I/Os we are going to have to make to read the entries. We should bear in mind that on a cached machine, the situation won't be quite that bad because it's likely that the position of the Secondary is in a fairly close block and therefore likely to be in the same Cache Domain.

The other statistic that needs to be looked at in conjunction with the inefficiency is the "Clustering Factor". This is concerned with the distribution of entries throughout the set. If they are clustered together we can get problems because we are going to have an increased number of Secondaries, and because in order to find a position for a new Secondary we are going to increase the number of I/Os necessary. The Clustering Factor is a general measurement of the amount of clustering in the set overall.

Now for a Detail Set it is important to know the Number of Deleted Entries, that is the number of holes in the set. But even more important for Detail Sets is the Percentage Messy. If all the entries are contiguously loaded in the right order for each of the chains then we would say that it was Zero Percent Messy. So a perfectly Repacked Detail Set would have Zero Percent Messy. The higher the percentage figure, the longer and more inefficient would be the chained access along the Primary Path and it is important to note that in DIAGNOSE it is only a measure along the Primary Path. In this context the Primary Path should be the most commonly used access path with the longest chain lengths. Repacking along this Path would then return the most benefit. Parameters mentioned in other pieces of software or in the literature are the Expected Number of Blocks, the Average Number of Blocks and Elongation etc. We went specifically for a single figure which is concerned with the actual performance criteria, that's to say the number of Primary Path chain pointers which potentially make extra physical disc I/Os.

With a cached machine there is more that can be said on this subject. Thus far in the literature a chain is considered to be Messy if it contains non-contiguous entries, particularly if they point to other Blocks. In a non-cached environment, chained access down this Path would consume more disc I/Os than needed because the extra Blocks would have to be read in from disc causing extra physical I/Os. However on a cached machine there is a much greater chance that the other Blocks will have been read into already existing Cache Domains, resulting then in logical rather than physical I/Os. DIAGNOSE examines this possibility by quoting two further Messiness Percentages - the Percentage Messy for 16 Sector Cache Domains and the Percentage Messy for 96 Sector Cache Domains. These two figures extend the sizes of the "block" used to measure Messiness to incorporate the highest and lowest Cache Domain sizes. By comparing these three figures we can get a feel for how well Caching will eliminate the extra disc I/Os caused by Messy Detail chains. They also give a qualitative feel for the extent of a Set's Messiness - a high Percentage Messy which reduces rapidly when measured at 16K and then disappears at 96K gives a good picture of the way in which that Set is actually Messy.

FIGURE 4

PARAMETERS AFFECTING PERFORMANCE

MASTER SETS:

PERCENTAGE FULL

PERCENTAGE OF SECONDARIES

CLUSTERING

BLOCKING FACTOR

BLOCK SIZES

NUMBER OF BUFFERS/BUFFSPECS

SEARCH ITEM TYPES

DETAIL SETS:

PERCENTAGE FULL

PERCENTAGE OF DELETED ENTRIES (HOLES)

PERCENTAGE MESSY

BLOCKING FACTOR

BLOCK SIZES

NUMBER OF BUFFERS/BUFFSPECS

SORTED CHAINS

THE RULE-BASED SYSTEM

It is possible for the software to go on another step and apply rules to examine those parameters in order to make recommendations (see Figure 5). The DIAGNOSE module of FLEXIBASE does this and therefore saves you the effort of analysing the data yourself. Currently in the DIAGNOSE module we can produce a dozen or so different recommendations, all of which are based on their own set of rules. One possible recommendation is "No Action at this time", which is, of course, the most desired one as it tells us that everything is alright and that none of the parameters are pointing strongly to any particular problem.

Let's take a recommended Capacity Change as a simple example. Of course it isn't as easy as saying that if the Capacity is more than such a value then there should be an increase. The rule for a Master Set actually reads:-

If the Master set Capacity is greater than 20 and the Set is more than 80% full, unless it's less than 90% full and less than 5% inefficient entries, then recommend new Capacity as the nearest prime number to give 75% full, as long as the Capacity increase would be greater than 1.

Now that is a bit of a mouthful but it does build a number of important constraints into the basic "bump up the Capacity if it's more than 80% full". It is concerned both with the absolute number of entries and also with the inefficiency. In other words, if the inefficiency is still low then the set is allowed to be more full before a recommendation will occur. Other rules might recommend Capacity Changes for different reasons, for example to reduce the Secondary inefficiency.

Other recommendations are concerned with optimising the Buffer Usage by increasing Blocking Factors, Repacking along the Primary Path if a certain Percentage Messy is reached for Detail sets and changing the type of the Search Item if we're in one of those Numeric Key problems. See Figure 6 for an example of a DIAGNOSE report. Sometimes one notices a situation where there is a Primary Path with a maximum chain length of one (that's a one to one relationship between the Master and the Detail). It is quite pointless to make that the Primary Path, assuming there are other paths of course, as the only point of a Primary Path is in Repacking the set and no possible improvement can be made on a chain length of one. DIAGNOSE would recommend that you change that.

Structural damage can be detected also. Even though the software only does serial reads of sets, it can detect broken chains. That's because, in line with a paper that came out in 1987 by an Australian S.E. (see "Diogenes" in Bibliography), it is possible to detect broken chains by producing a hashed total of the forward and backward pointers. Also by totalling the number of entries and comparing that with the chain length totals in a Master set, missing Chain Heads can be detected in most cases. Other checks are also performed.

FIGURE 5

RECOMMENDATIONS

CHANGE CAPACITY (MORE THAN ONE REASON)

CHANGE BLOCKING FACTOR

CHANGE BLOCKMAX

CHANGE MINIMUM NUMBER OF BUFFERS

CHANGE SEARCH ITEM, TYPE

CHANGE PRIMARY PATH

REVIEW NEED FOR SORTED PATH(S)

REVIEW NEED FOR ORDINARY PATH

CHANGE NON-PRIME CAPACITIES (CONDITIONAL)

REPAIR STRUCTURAL DAMAGE

REPACK DETAIL SET ALONG PRIMARY PATH

NO ACTION

FIGURE 6

SAMPLE DIAGNOSE FUNCTION OUTPUT

```
<><> DIAGNOSE/3000 <><>      Version C.04.03     (THU, OCT 29, 1987, 10:51 AM)


DATABASE STATISTICS AND DIAGNOSIS FOR DMISC.DIRECT.HQ              Page:  1

Global Statistics:
  Image Level C            - (TurboIMAGE)
  Global Buffer Length (words)    549    No Logid
  Number of Sets                   11    Database Created    SUN, JUN 21, 1987
  Number of Items                  38    Database has never been DBSTORED
--------------------------------------------------------------------------------

  9) Set (Manual)    ADDRESSES:
  Search Item - ENTRY                     Type Z6
  Number of Items                   8     Entry Length (words)           175
  Number of Paths                   0     Blocking Factor                  2
  Capacity                        113     Block Length (words)           361
  Number of Entries                35     Total Number of Blocks          57
  Percentage Full               31.0%     Clustering Factor              .01
  Percentage Secondaries         8.6%     Average Secondary Chain       1.09
  Percentage Inefficient         2.9%     Logical Device Number            1
     RECOMMEND - Increase BLOCKING FACTOR to   3 to optimize buffer usage

 10) Set (Manual)    NOTES:
  Search Item - ENTRY                     Type Z6
  Number of Items                   2     Entry Length (words)           178
  Number of Paths                   0     Blocking Factor                  2
  Capacity                         59     Block Length (words)           367
  Number of Entries                 0     Total Number of Blocks          30
     RECOMMEND - No action at this time

 11) Set (Detail)    DETAILS:
  Primary Search Item - KEY2              Type X16
  Number of Items                  13     Entry Length (words)           108
  Number of Paths                   7     Blocking Factor                  3
  Capacity                        114     Block Length (words)           409
  Number of Entries                36     Total Number of Blocks          38
  Percentage Full               31.6%     Number of Deleted Entries        0
  Percentage Messy             100.0%     Percentage Deleted             .0%
  % Messy - 16 Sector Cache     33.3%     Number of Sorted Paths           0
  % Messy - 96 Sector Cache       .0%     Logical Device Number            1
                                                       Chain Lengths
  Path  Search-Item    Type     Master-Set          Average     Max      Min
   1     KEY1          X16      KEY1                  1.3        9        1
   2    !KEY2          X16      KEY2                  1.1        3        1
   3     KEY3          X4       KEY3                  1.3        9        1
   4     KEY4          X4       KEY4                  1.2        3        1
   5     KEY5          X4       KEY5                  1.4        9        1
   6     KEY6          X4       KEY6                  1.1        3        1
   7     ENTRY         Z6       ENTRY                 1.0        1        1
    RECOMMEND - REPACK set along Primary Path
    RECOMMEND - Change Primary Path from KEY2 to KEY5
    RECOMMEND - Increase BLOCKING FACTOR to   4 to optimize buffer usage

------------------------- DIAGNOSIS  COMPLETE ---------------------------
                                         (THU, OCT 29, 1987, 10:52 AM)
```

RECOMMENDED CHANGES

IMPLEMENTING THE RECOMMENDATIONS

Those are the rules for producing the recommendations. Now to move on to the next stage which is to actually implement those recommendations. Some people have asked for automatic implementation of those recommendations. We have always held out against that because we feel that at this point we require some manual intervention for several reasons.

Firstly there will be circumstances, either because of the way the data base is being used, or the particular environment that it's being used in, that mitigate against the recommendation. For example, maybe there is a shortage of memory or disc which would prevent increasing capacities, or maybe a particular Path or Set is accessed so rarely that there really is no point in reorganising it. In these cases there really does need to be someone who says "Well, we'll look at the recommendation and either accept it or reject it because of circumstances that the software doesn't know about".

Secondly, implementing some of these recommendations can require long run times. Most installations run under pressure and any additional work such as Capacity Changes, Blocking Factor Changes or Repacks need to be scheduled.

We feel the third reason against an automatic system is that if software like DIAGNOSE is run regularly (e.g. once a month) on a data base then there really won't be many recommendations and therefore the demand on the Database Administrator's time will not be that great. From a scheduling point of view, it is quite important to be aware of the relative run times of the various types of recommendations that need to be implemented. Detail Capacity Changes for example can be done extremely quickly because they can employ fast techniques. However, a master Capacity Change can only be done by taking all the entries out of the old set, purging the old set, building a new empty set and putting back the entries into it (although in many cases actual DBPUTs can be avoided). FLEXIBASE, and also a few other products are capable of implementing these changes. Blocking Factor Changes, Search Item Changes and Path Changes will all require some sort of an unload/reload of that set. Some changes can be very simple, such as the Primary Path Change which is simply a change to the Root File and can be implemented almost instantaneously.

Repacking a Detail Set is an interesting challenge because the problem which you are trying to solve (the very long chained read times or the long DBPUT times due to sorted chains) are going to be the things which also slow down the Repack if the Repack uses standard DBPUTs. We tackled this problem last year by bringing out a new Repacking module which doesn't use DBPUT at all but instead builds on the structure which is already being used by DBPUT. It goes in and basically reorders pointers and resequences entries and it has paid off handsomely. For example, there was one set of several tens of thousands of entries which had three sorted paths and a Repack on that set used to take by the DBPUT method around 8 hours and it now runs in about 25 minutes. More than that; because it is actually building on what's already been done by DBPUT, if the Percentage Messy is quite small then there is really very little physical work for the Repack module to do. A Repack based on DBPUTs might run 30 percent faster if the Percentage Messy was very small, but the new Repack would probably run 5 times as fast if the Percentage Messy is very small. This means that it is quite feasable and desirable to do very frequent Repacks because not only do they not take very long but they keep the good performance on chained reads ongoing.

CONCLUSION

The main thrust of this paper has been to show that Database Performance can be improved more effectively by the use of an Expert System approach. The goal of any Database Administrator or System Manager in this respect should be to obtain the best performance from his or her system so as to satisfy user's demands while cultivating an aura of effortless competance. A regularly run system which can spot hidden inefficiencies or potential catastrophies (such as DATASET FULL right in the middle of a four hour overnight batch run), coupled with adequate software tools to implement it's recommmendations, will prove invaluable to every computer site which uses IMAGE data bases. The Expert General Knowledge coded into the software tools by their creators together with the Expert Local Knowledge of the Database Administrator or System Manager, when combined, enable data bases to be more efficient and provide users with greater performance.

Bibliography:

Papers and Articles:

"Evolving Performance Guidelines - An '87 Update" - Mark Tolbert. Interex Las Vegas 1987 Proceedings Vol 3 paper 3183. 1987

"Trends In Image" - C. Bradley Tashenberg. Interex Las Vegas Proceedings Vol 3 paper 3180.

"Database Dynamics" - F. Alfredo Rego. Interex Madrid 1986 Proceedings pp203-218.

"Squeezing the Last Bit out of your HP3000" - Bob Green, David Greer & Mike Shumko. Interex Las Vegas 1987 Proceedings Vol 2 paper 3077.

"The Three Bears of Image" - Fred White. Interex Anaheim 1984 Proceedings paper 73.

"Image/3000 Performance Plannig and Testing" - Walter Gioscia & Dennis Heidner. Interex Montreal 1983 Proceedings paper 98.

"Overview of Optimizing" - Bob Green. Thoughts and Discourses (see below).
"Improving Your Performance" - R.E. Van Valkenburgh. Interact September 1987.

"Diogenes - Searcher for an Honest Data base" - Jim Kramer. Interex Las Vegas 1987 Proceedings Vol 2.

Books:

"The Image/3000 Handbook" (and the TurboIMAGE supplement) -Various Authors. Wordware, Seattle.

"Thoughts and Discourses on HP3000 Software" - Eugene Volokh. Vesoft, California.

Making It Fly - Optimizing Applications for MPE XL

Sid Smith
Hewlett-Packard Company
9255 SW Pioneer Ct.
Wilsonville, OR 97070

Introduction

The introduction of the Series 900 Precision Architecture systems by Hewlett-Packard has provided an opportunity for users of HP systems to dramatically increase the number of users and the application workloads on their systems. Many users have discovered significant boosts in application throughput simply by restoring their application onto a Series 900 system from an MPE V system. Others have improved throughput even more by recompiling their code to use the Native machine instructions on the HPPA system. And some have gone so far as to modify their code to take advantage of the new architecture and reap even greater throughput benefits.

This paper will first discuss the basic concepts of MPE XL and HP Precision Architecture as applied to system performance, and then provide tips on how to optimize the performance of your application using the basic concepts. The paper will concentrate on those components of HPPA that the application programmer has some control over: the hardware architecture will not be discussed in any detail. As appropriate, results of tests performed on an HPPA system will be provided to substantiate the optimization tips.

MPE XL and HP Precision Architecture Basic Concepts

"It looks the same to me..."

To the user of HP 3000 systems, nothing has really changed. Well, maybe response time is a bit better, and reports are available sooner. But it looks the same, doesn't it?

The guts of the Series 900 systems is totally different from the HP "Classic" systems: the hardware design allows for 64-bit addressing, versus 16 bit addressing on Classic systems; support for multiple processors is provided; a reduced number of hardware instructions is implemented (RISC); the operating system was totally rewritten; and a host of other design features were implemented. The overall design of HP Precision Architecture allows for few parts, greater reliability, and much improved performance over the Classic 3000 systems. The improved performance is derived from a few key features of the hardware and operating system: Reduced Instruction Set Computing (RISC), intelligent pre-fetching of data, gathered writes, and mapped I/O.

The RISC concept

Much has been said on the subject of RISC, lately most of it positive as the rest of the industry attempts to catch up to HP. Since the subject of this paper is performance, I will let others speak on the intricacies of RISC. However, a brief explanation of why a RISC system is generally faster than a complex instruction machine is in order.

RISC is based on an 80-20 rule: 80% of all instructions executed on a computer are done by 20% of the total number of instructions available. A RISC system limits the number of instructions to a subset of those on a complex instruction system. Each instruction (with a few exceptions) should be able to execute in one cycle, as opposed to several cycles on a complex system. Because there are fewer instructions, intelligent compilers were built to better utilize the general purpose registers and the reduced instruction set. The intelligent compilers are able to reduce the total number of instructions required for a task to about the same number of instructions required for a complex system. The net result is: roughly the same number of instructions, each instruction executing several times faster equates to faster execution of user programs. While this explanation of RISC has been greatly simplified, it should provide the reader with a basic understanding.

In order to take full advantage of the HPPA architecture, the program must be compiled with one of the Native Mode compilers. Each NM compiler offers various levels of code optimization. Optimization level 0 provides no additional optimization beyond the minimum done by the application. Level 1 optimization provides some additional code optimization, and level 2 provides the most optimization. Various levels of optimization are supplied to allow the programmer to effectively debug the code before fully optimizing it. Each level of optimization makes certain assumptions about data alignment and placement to optimize register usage and reduce code, but in so doing adds complexity to the debugging process. In some cases, level 2 optimization can reduce the amount of code executed by 5% or more.

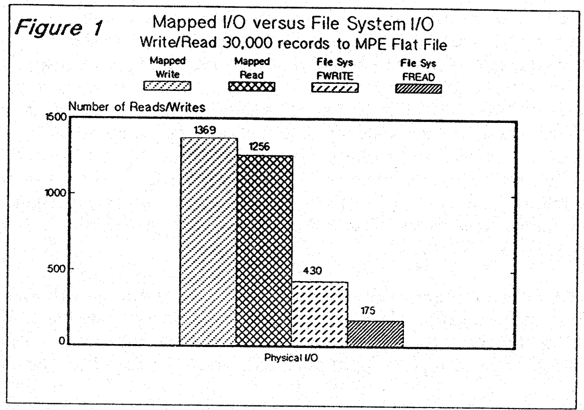
If the user is unable to compile the code into native mode, the program will still run using the MPE V instructions. The MPE V instructions are emulated by actually executing MPE XL instructions. To improve performance of compatibility mode code, an Object Code Translator (OCT) is used. OCT produces a much enlarged program file and performs optimization of the emulated code. When executed, a program that has "been OCTed" will still run emulated MPE V code, but will run 10-20% faster than the original CM code because of the optimization done by OCT.

Intelligent Pre-Fetching of Data

Data on MPE V systems is stored on disc in logical blocks. The user specifies the blocking factor when the file is built, then data is read from and written to disc in logical blocks. Depending on the blocking factor, anywhere from 1 to several records are retrieved with each physical read and placed into a intermediate buffer area in memory. The MPE file system then de-blocks the data and transfers it one record at a time to the user stack. In order to reduce the number of physical I/Os, disc caching software was introduced several years ago. The disc caching software will read up to 96 sectors (256 bytes each) of a file into an area of memory referred to as a cache domain. While disc caching can substantially reduce the I/O load on a system, it can also dramatically increase the CPU load by having to manage all the cache domains.

On MPE XL, data is stored in 4096-byte pages. The user can still specify a blocking factor, but the system ignores the blocking factor in favor of accessing the file in physical page increments. Because data is always accessed in pages, main memory is divided up into pages of equal size, thereby reducing the overhead required to manage memory. Load and Store instructions are used to read or write data to disc and buffers are no longer used. Data is simply moved one or more pages at a time from disc to memory, and then to the user's stack (one record at a time). The CPU overhead required to manage user data is much less than on an MPE V system.

When the user uses the MPE XL file system or TurboImage, the operating system evaluates the address of each record requested. If data is being read sequentially, regardless of the intrinsic used to perform the read, the operating system will request multiple pages of the file to be read into memory. Figure 1 shows the number of physical reads required to sequentially read 30,000 records from a file. The efficiency of the pre-fetch algorithm results in an average read hit-rate of 90% or higher on most MPE XL systems. The number of physical reads on an MPE XL system is less than on an MPE V system and the CPU overhead required to manage the I/O is considerably less.



Gathered Writes

MPE XL further reduces physical I/O by not physically posting data to disc until one of several situations occurs: the file is closed, the program calls FCONTROL(6), the system Transaction Manager is used and a Transaction Manager Checkpoint occurs, the memory occupied by the data is required by another process, or the FLOCK/FUNLOCK intrinsics are used. By waiting to post data, the system memory manager is able to use some intelligence by gathering all contiguous pages and performing one physical write. Figure 1 shows that it took only 430 physical writes to post 30,000 256-byte records. In other words, each physical write posted an average of 4 pages of the file.

Mapped I/O

The concept of mapped I/O is a bit more difficult than the other concepts previously discussed. Very basically, every byte of data in memory and on disc has an associated address, called a virtual address (virtual, because it is not a real, physical address). When the user accesses a file through the file system, the file system checks for EOF, increments the record pointer, does pre-fetching if appropriate, updates the file label, etc. The end result of a file system read or write is a request of the memory manager to retrieve or post one page of data at a specified virtual address.

MPE XL allows the user to bypass the overhead of the file system and communicate directly with the memory manager by supplying the virtual address within the program. Figure 4 shows the CPU time required to read and write 30,000 record to a file using the file system and mapped access. Note that while the CPU time for mapped I/O is less than the file system, the wall time was more due to the pre-fetching and gathered writes used by the file system.

Accessing a file mapped is best when access is random. A typically excellent use of mapped I/O is a large table that might be accessed through an extra data segment on MPE V.

Optimizing Your Application

Native Mode or Compatibility Mode?

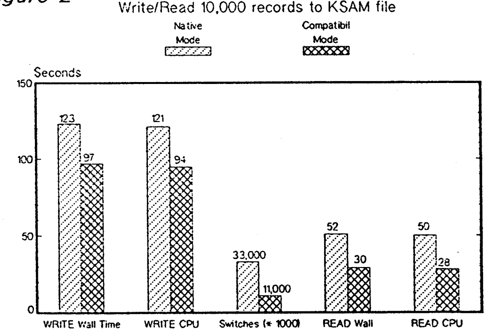
When possible, you should always be ready to compile your code into native mode. How's that for an answer? Actually, there are circumstances when code should be left in compatibility mode.

When subroutines (system or user) are still in compatibility mode, such as KSAM (at the time of this writing), the overhead of "switching" between NM an CM can be great enough to eliminate the advantages of the NM compiler. A switch occurs when code is executing in one mode and a subroutine in the other mode is called. Parameters must be passed from the main routine to the subroutine. The NM stack is able to access the CM stack, but the CM stack does not even know of the existence of the NM stack. Consequently, a CM program calling a NM routine can pass the address of the parameter, while a NM routine calling a CM routine must pass the entire parameter. The more parameters there are, the greater the overhead of switching from NM to CM there is. The following graph (figure 2) shows the wall time, CPU time and number of switches for a program reading and writing 10,000 records to a KSAM file.

In general, unless your program exclusively calls CM subroutines, it should be moved to native mode. For example, an application that uses TurboImage data bases, but calls a CM subroutine to parse the user requests should most likely be moved to NM. Ideally, the subroutine should be moved to NM, or (better still) moved inline into the main program.

When a program remains in compatibility mode, it should always be Object Code Translated. The OCT will provide 10-20% improved performance with no loss of functionality.

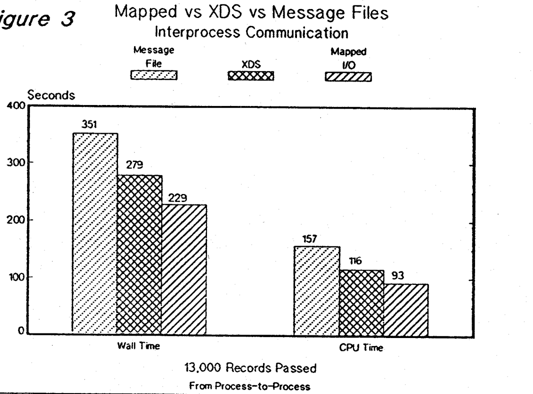
Figure 2



Interprocess Communication

An application that requires some form of interprocess communication provides a dilemma for the performance specialist. The code for message files is still in compatibility mode, but will be moved to native mode in the future. Figure 3 shows the wall and CPU time required to send an 80-byte record from one process to another using message files, an extra data segment, and a mapped file. An application currently designed to use message files should not have to change, unless an immediate gain in throughput is required. The user code must become more complex when using a mapped file to coordinate process synchronization. In my test, each process suspended when reading or writing to the mapped file and was re-activated by the other process when the record was read or written. In no case should extra data segments be used for interprocess communication. A set of contributed routines is available to map XDS calls into mapped file access.

Figure 3



Accessing Tables

In no case should an extra data segment be used for table access. The best performance will be realized when the table is small enough to be read directly from disc into the user stack (remember, no stack limitations with MPE XL) and accessed as an array. If the table is too large to spend time reading into memory, it should be accessed as a mapped file. Access to a file as mapped is achieved through the HPFOPEN intrinsic and a type of variable called a pointer. Since pointers are only available in the PASCAL and C programming languages, the COBOL programmer must use a subroutine written in PASCAL or C (see figure 6 for sample code).

Which optimization level to use?

The answer to this question is simple: use the highest level of optimization possible (COBOL only allows 0 and 1) in which your program still works. Remember, the optimization process must make certain assumptions about data alignment and procedure calling to be most efficient. If the assumptions are incorrect, the results can be unpredictable. Code should always be tested both before and after optimization for reliability.

Quantifying the performance gain from various optimization levels is more difficult. In general, the system will be executing user code between 10% and 20% of the time and system code the remaining time (TurboImage, file system, memory manager, etc.). Fourth generation languages typically spend more time in user code than third generation languages, and CPU-intensive applications such as spreadsheets or modeling packages may spend even more time in user code. The more time spent in user code, the greater the benefit to be gained from higher optimization levels.

To illustrate, imagine that you have the means to reduce the number of steps required to prepare a meal. The total time required to prepare the meal will be the sum of your time plus cooking time. If cooking time represents 80% of the total time, reducing your time will have little effect on the overall preparation time. However, if cooking time represents only 20% of the total time, optimizing your steps can greatly reduce the overall preparation time. So it is with code optimization on MPE XL: the more time spent in user code, the greater the performance gain through code optimization.

Accessing MPE Flat files

There are two rules when accessing flat files:

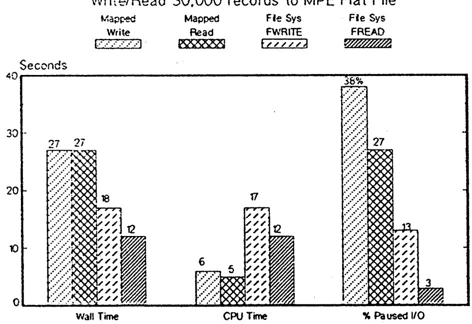
1. If accessing the file sequentially, use the MPE file system to access the file (FREAD/FWRITE).

2. If accessing the file randomly, access the file mapped by using HPFOPEN and pointers.

The ability of the file system to do pre-fetching and gathered writes can improve the throughput by decreasing the physical I/O. When pre-fetching and gathered writes is not an issue, as with random access, using mapped access can improve throughput by decreasing the CPU time required to access the data.

When building a flat file, don't worry about blocking factors, except as it applies to disc space utilization. MPE XL does not use the blocking factor when accessing the file. Flat files should be built with the default (* = unlimited) number of extents. If a number of extents must be specified due to disc space limitations, specifying 1 (one) extent will provide the best performance. Specifying 32 extents can decrease the throughput when writing to a file sequentially.

Figure 4 Mapped I/O versus File System I/O
Write/Read 30,000 records to MPE Flat File



In tests we have run, disc fragmentation has not proven to be a detriment to transaction throughput. Disc fragmentation is only an issue when contiguous space is required by a program, or by the system when updating to a new release of the operating system. Packing discs will not improve system performance.

TurboImage Buffers

By the time most people are reading this paper, Release 2.0 of MPE XL will either be available or soon available. Release 2.0 will contain enhancements to TurboImage that will eliminate the buffer area, except for the header area. Data will be accessed by mapped I/O. Prefetching will be done by the memory manager and gathered writes will be allowed through the system transaction manager. In other words, don't be concerned with buffer specifications.

Intrinsic Level Recovery and Autodefer

There is no situation in which ILR should be used. Figure 5 shows the relative performance of adding records to a stand-alone master data set when ILR or autodefer are enabled.

TurboImage uses the system transaction manager to ensure data integrity. The transaction manager maintains images of each transaction that modifies data in a disc log file. The disc log file is updated at least every second (usually more often on a busy system). In the event of a system interruption, the operating system will automatically recover any transactions that were not completely posted to the data base. Intrinsic level integrity is maintained, but logical consistency is not. Image transaction logging must be used if a logical consistency is required

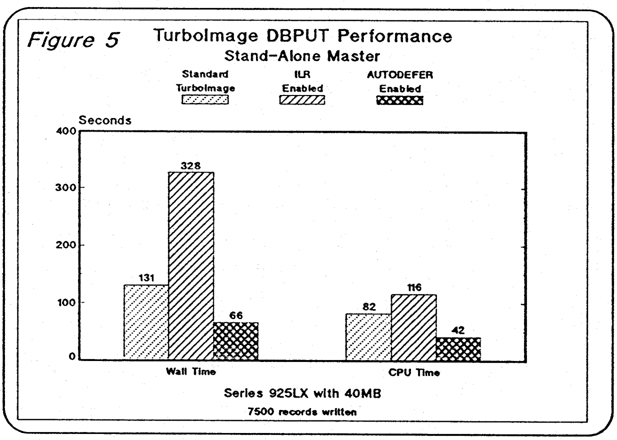


Figure 5 TurboImage DBPUT Performance

for your application.

The rules for autodefer remain as they were with MPE V. If the application can be restarted from the beginning (this assumes you have stored the data base), and the application is DBPUT and DBDELETE intensive, then enabling AUTODEFER with DBUTIL can improve the transaction throughput. The downside, of course, is the lack of data integrity in the event of a system interruption. It is unknown at this time what the performance of autodefer will be on Release 2.0 of MPE XL.

Summary

Because most applications use TurboImage exclusively, application optimization will be a simple matter of recompiling into native mode. For those users with a more adventurous spirit, time spent optimizing the application for MPE XL can provide many fun-filled hours of entertainment. I encourage any of these adventurous people to experiment, and to work with those features of MPE XL and HPPA that make the system unique in the industry.

Making your application fly on MPE XL can be easier than you previously believed.

Figure 6 - Accessing a file mapped from COBOL

HPFOPEN Variables:

See the MPE XL Intrinsics manual for additional information on the HPFOPEN call.

```
01   FILENUMBER              PIC 9(9) COMP.
01   STATUS-PARM.
     05 STATUS-INFO          PIC 9(4) COMP.
     05 SUBSYSTEM            PIC 9(4) COMP.
01   FILE-NAME-OPT           PIC 9(9) COMP VALUE 2.
01   FILE-NAME               PIC X(10) VALUE "&TESTFILE&".
01   ACCESS-TYPE-OPTION      PIC 9(9) COMP VALUE 11.
01   ACCESS-TYPE             PIC 9(9) COMP VALUE 1.  {Write access}
01   LONG-MAPPED-OPTION      PIC 9(9) COMP VALUE 21. {18 for SHORT}
01   LONG-POINTER            PIC 9(18) COMP.     { 9(9) For SHORT }
01   DOMAIN-OPTION           PIC 9(9) COMP VALUE 3.
01   DOMAIN                  PIC 9(9) COMP VALUE 0.  {New File }
```

Pointer variable used for adding or subtracting to the value of the pointer address of LONG-POINTER:

```
01   POINTER-VAR             PIC S9(18) COMP.
```

Open the NEW file for mapped write access:

```
     CALL INTRINSIC "HPFOPEN" USING FILENUMBER, STATUS-PARM,
          FILE-NAME-OPT, FILE-NAME, DOMAIN-OPTION, DOMAIN,
          ACCESS-TYPE-OPTION, ACCESS-TYPE, LONG-MAPPED-OPTION,
          LONG-POINTER.
     IF STATUS-INFO <> 0 THEN
        DISPLAY "HPFOPEN ERROR, INFO = ", INFO
        CALL INTRINSIC "QUIT" USING 1.
```

Copy the pointer returned from HPFOPEN to our variable:

```
     MOVE LONG-POINTER TO POINTER-VAR.
```

Write to file using mapped access:

```
     CALL "MAPPED-IO" USING POINTER-VAR, BUFFER-AREA.
```

Figure 6 - continued

Increment pointer to write next record:

```
COMPUTE POINTER-VAR = POINTER-VAR + RECORD-LENGTH.
```

At the end of program, mark the EOF of the file, and close the file:

```
CALL INTRINSIC "FPOINT" USING FILE-NUMBER, LAST-REC-NUM.
CALL INTRINSIC "FCONTROL" USING FILE-NUMBER, 6, CTL-CDE.
     { FCONTROL '6' forces all data to be flushed to disc
       and forces posting of the file label }
CALL INTRINSIC "FCLOSE" USING FILE-NUMBER, 1, 0.
```

PASCAL Subroutine "MAPPED-IO":

```
$STANDARD LEVEL 'EXT_MODCAL'$   { Use MPE XL extensions }
$SUBPROGRAM$

program dummy_name;

type

      Record_Type                = PACKED ARRAY [1..80] of CHAR;

PROCEDURE mapped_io (VAR Long_Pointer : GLOBALANYPTR;
                     VAR The_Record : Record_Type);

VAR

      Pointer_to_Record  : ^$EXTNADDR$ Record_Type;

{ Most of this will be totally confusing for those not conversant with PASCAL. The
'$EXTNADDR' above is only necessary if you are using the LONG mapped option
(extended addressing). The '^' symbol specifies a pointer type. }

BEGIN

{ Copy the GLOBALANYPTR (a special type in PASCAL) to a 'typed' pointer so that
it can later be dereferenced (don't worry about what this means) }

      Pointer_to_Record := Long_Pointer;

{ Write record to file with mapped access }

      Pointer_to_Record^ := The_Record;

END;
BEGIN {Main}
END.
```

HP3000 System Performance
Real Customers/Real Solutions
by James A Hepler
Hewlett-Packard
Novi, Michigan

System Performance and Resource Utilization is a growing concern in the industry and across the HP3000 user community. As a Performance Specialist for the past five years and an Account SE for seven years before that, I have helped many customers identify and resolve performance problems in the real world. It is the purpose of this paper to provide case studies of how customers resolved performance problems on their systems in many cases without major costs so that other HP3000 customers may be made aware of some of the possible problems contributing to degradations in system performance. All of these situations described herein actually happened to Real Customers and a Real Solution was found to resolve their problem.

Case 1: The system is slower than it used to be after a major conversion from another vendors system which no longer met the customer's growth needs.

This customer hired an outside consultant to assist them in a conversion from a small system to an HP3000 Series 37 with a single disc drive. The conversion was necessary, because the existing system was already overburdened and could not be upgraded. The HP3000 hardware configuration was recommended by another third party. The application used indexed file methods and a major new application was to be added. RPG and KSAM were selected by the customer since the applications already existing were in RPG with ISAM.

The conversion of existing applications went smoothly and performance was acceptable until the new application was written and put into production. Suddenly on-line response degraded to an average response time of 17 seconds where it had been 6 seconds before. The 6 second response had been acceptable since it was much faster than the previous system had given.

An examination of the utilization of various resources on the system showed two obvious bottlenecks - CPU and Disc I/O. The CPU was almost consistently 80% or more Busy and the single disc consistently did more than 70,000 Physical I/Os per hour. It was determined using modeling tools that if the I/O bottleneck was removed, the CPU would become busier. Approximately 63% of the I/Os were reads. 46,000 of the 70,000 I/Os were from the new application and about half of the CPU time being used was also from that application. There was no doubt where the source of the problem was.

The application was an unusual one which read information from a foreign device on an asynchronous port almost continuously and then posted data to a KSAM file which was essentially a logfile of data transmissions. There was an inherent problem in that many records needed to be logged, but the biggest problem was that many records were being logged even when no data was received from the device.

Since 24,000 I/Os per hour were from the converted applications and 46,000 I/Os per hour were from the new application, it was decided to acquire a second drive to spread the load. The 70,000 I/Os per hour is about 19.44 per second or more than the rated value of most HP disc drives. Clearly the I/O problem was the most serious since it would be impossible to increase the speed of the I/O being done without replacing the disc drives with faster drives. The disc hardware cannot access the data any faster than it already is accessing it. A second disc drive was added and a Reload performed.

At the same time, the application was rewritten to more intelligently post log records especially when no data was actually transmitted. There was also a restructuring of the data key (index) so that there were fewer duplicates allowing a more efficient internal structure for KSAM. (See B-tree discussion in HP documentation.)

The net impact of these changes was to reduce the amount of Disc I/O from the 46,000 per hour previously to about 18,000 per hour - a significant improvement. Also, most of the I/Os which were eliminated were Writes to disc which would have a substantial benefit in supporting a move to MPE Disc Caching.

At this point there was about a 50% improvement in response time for the on-line applications, but more improvement was desired since there was to be some growth within six months. The system utilization was reexamined and the average CPU Busy was now in the 80-85% range at peaks with higher values reached with batch reporting jobs executing. The Disc I/O was relatively balanced among the two drives, but user programs tended to be Paused for I/O still. There was still a very favorable read/write ratio with almost 75% reads on average.

With a favorable read/write ratio like this, it is normal to turn on MPE Disc Caching as a way of reducing the physical I/O and thus reduce the wait time for I/O and improve response time and thruput. However, the real world is not always that simple. This customer did not have disc caching software on their Series 37 and also did not have sufficient excess memory or CPU time available to allow disc caching to function properly anyway. A CPU Upgrade was needed to allow the advantages of disc caching just outlined.

A remarketed Series 52 was acquired with sufficient memory to support MPE Disc Caching. With MPE Disc Caching turned on, and the increased memory availability for disc caching, the physical disc I/O was decreased to where the I/O delays experienced on the Series 37 were almost gone. The excess memory had another unforeseen affect in that the KSAM Extra Data Segments tended to stay in Main Memory all the time speeding up the access time even more. This was exhibited as a reduction in Memory Manager overhead.

It was helpful to have a faster CPU as well. The net impact was that the average online response time was now about four seconds and that the online users added did not change this average response time significantly over the six months of growth which followed. About 40% more simultaneous users were added. Incidentally, the new application also functioned admirably well even after the growth.

Summary: This case exhibited three separate but interrelated problems. The first was that the "new" application was inefficiently written. The second was the excessive amount of disc I/O for one drive. The third was that a CPU upgrade was needed mainly to be able to implement MPE Disc Caching, but also to sustain expected growth. Upon correcting these problems, system throughput and response time improved dramatically.

MPE V to MPE XL: If this Series 52 customer ever migrates to the MPE XL environment, they should be able to expect the same kind of excellent response and throughput they are now experiencing even in compatibility mode with RPG assuming a suitable hardware configuration is chosen for the migration.

The "new" application (asynchronous device access) should be reviewed or tested with the distributed terminal controller under MPE XL to be sure it still functions properly.

Of course since MPE Disc Caching was part of the solution to this customer's problem the question should be asked, how is this need fulfilled in the MPE XL environment? The answer is simple - the MPE XL I/O system does not need disc caching, but does a much better job of utilizing memory and CPU speed to eliminate even more physical disc I/O than MPE Disc Caching did in the past.

Case 2: Customer did a complete Reload over the weekend to reduce disc fragmentation and now their users are complaining about poor response times and certain batch jobs seem to be hung.

This customer had a Series 58 with a TurboImage application using nine different data bases all open by the major online application. Since no coding changes had taken place, the problem must have to do with file placement as a result of the System Reload.

It was discovered that almost all I/O on the system was taking place on logical devices 3 and 5. Disc Caching was enabled but the physical I/O rates were 16 and 18 I/Os per second respectively. The queues were very long on 3 and 5 also with a length greater than four 26% of the time on ldev 3 and 18% of the time on ldev 5.

File accesses were examined showing that the Reload had redistributed the files in a perfectly terrible manner. Prior to the Reload, files and therefore the physical disc I/O had been distributed approximately evenly across all disc drives. Subsequent to the Reload, the 5 most commonly accessed files were now on drives 3 and 5.

To compound the problem further, the locality of the files was also poor. The affect of this was to cause longer and more frequent head movements increasing the length of time to do a physical access on the average. The disc drives were doing the legendary "HP Rhumba" caused by the excessive head bounce.

The analysis of the individual file accesses showed that by purely bad luck, the reload had caused certain Master data sets to be located near the internal areas of ldev 3 and the Detail data sets they were linked to (for most accesses) was physically located at the beginning of ldev 3. This caused the exaggerated head movement resulting in the longer queue lengths on ldev 3. A similar problem occurred with files on ldev 5.

Temporarily, the Master Data Sets were Stored to tape and Restored to ldevs 2 and 4. This gave acceptable performance for the remainder of the week. Following the system backup on Friday evening, a Reload accounts was done with selective Restores of the 6 most accessed files first followed by the rest of the system files spread at random. The selective restores were done in such a way to balance as equally as possible the physical disc I/O. This effort was successful and the system performed similar to the way it had prior to the disastrous reload of the previous weekend.

Summary: Physical disc I/O imbalance can cause serious degradation as this customer discovered by innocently doing a Reload. For this reason it is important to ascertain the most commonly accessed files and distribute them appropriately across the available drives. There may be other factors affecting this such as file size, chain lengths within a database, block sizes, Controller contention, type of disc, etc. These other factors are harder to quantify than the raw amount of physical I/O per file. In other words it is easier for the customer to determine this and control it himself than the other factors. Most of the time this is all that is needed to receive optimum performance from the disc drives on an HP3000.

MPE V to MPE XL: The disc access situation on MPE XL is much different because of the major improvements to the I/O system to take advantage of Precision Architecture. The probability of a physical disc I/O bottleneck being created under the MPE XL environment is much less than it is under MPE V. There have been very few cases to date where this has been a problem, but it is possible and should be considered if there is a Performance problem under MPE XL. It is probably not a migration issue to be concerned with and in fact chances are that whatever your disc I/O situation under MPE V it will be improved under MPE XL with sufficient Memory and CPU availability.

Case 3: An interactive application using a fourth generation language to access a TurboImage data base normally has a 2-4 second response time, but occasionally takes over a minute, even in a stand alone (single user) environment. This system is a Series 58. The application has logic which calculates certain chemical factors in a substance manufactured and then ascertains whether or not the compound is hazardous to human health or the environment.

After many unsuccessful attempts at finding flaws in the complex logic of the application, it was decided to approach the problem from the other direction. Perhaps by reviewing the accesses of the data, we could determine what the problem was. PROFILER was used to trace the TurboImage accesses. Eureka!! It turned out that normally there were only 4-5 DBGETs per transaction, but that the occasional problem showed as many as 85 DBGETs!! The question was why?

The program logic seemed to be fairly straight forward, and the 85 DBGETs were tracked back to one simple statement of fourth generation code which was essentially a read of the data base. However, the internal logic of the fourth generation language in interpreting that simple read sometimes caused 85 DBGETs and sometimes only 4-5.

Further research on the internals of the fourth generation language showed that if given criteria for finding a record which included more than one key, it would make a decision as to which chain to read along to find the correct record. Our program logic for establishing which key value to look for many times came up with only one possible key, but several other non-key criteria. The fourth Generation language would then only be able to read that one key or read serially.

An examination of the key structure of the data base showed that certain key values had very long (as long as 500 on one chain) chains because the key value was too common. The key in question was not the primary path and therefore tended to be physically fragmented as well which added to the access time when there was a problem. To further compound the problem, the program logic was rather convoluted in determining whether this was now the correct record needed after each DBGET, and sometimes required reading a chain in another data set to make that decision.

There was a very elegant solution to this problem. It turned out that we could shorten the length of the long chains by a factor of ten by adding a word of data which was already available to the key that was causing the problem. This was done, causing the long response time to be reduced to about 8 to 10 seconds which was considered acceptable.

Summary: While the real problem here was the long chain length, the difficulty in finding the problem was because the simple coding which is the advantage of using fourth generation languages can sometimes make a problem in logic difficult to find, because the internal processing of the fourth generation language may not always be obvious. Many fourth generation language manuals now have logical flow charts showing the internals of their I/O logic in particular. (See TRANSACT documentation for examples.)

MPE V to MPE XL: Other than the possibilities of larger data bases and more users on MPE XL systems, there is no migration issue here assuming the fourth generation language is available for both environments.

Case 4: An application which used a fourth generation language (TRANSACT) to access a remote TurboImage data base through a leased line at 4800 baud using DS/3000 takes three minutes to delete a single record. The CPU was a Series 70 at the local end and a Series 68 at the remote end.

The programmer had tracked the problem to the DELETE statement in the TRANSACT program, but couldn't figure out why it took three minutes to execute that one TRANSACT statement.

A DS trace showed that the single DELETE statement created a remote DBOPEN followed by a series of remote DBGETs. Following each DBGET, the record found was returned to the local system to allow the local TRANSACT program to determine if this was the correct record to delete. If it was not, another remote DBGET was issued. It turned out that the DBGET issued was a chained DBGET and that the average number of DBGETs issued was around 85 for each DELETE statement.

Two minutes and 20 seconds of the 3 minutes delay was caused by data communications delays and turnarounds due to BiSynch communications. The rest of the response was because of the chain length, etc.

This problem was fixed by creating a son process on the remote system to do the delete rather than doing it through the Remote Data Base Access facility of TurboImage. This eliminated the costly line delays and reduced the time of this transaction to about 30 seconds which the customer was willing to accept in their environment.

Summary: Both Case 3 and Case 4 are examples where the advantage of a fourth generation language made a problem analysis take a different path than most programmers are used to. There is an important message here to not overlook the possibility that the many things a fourth generation language does for the programmer may make problem analysis more interesting.

A second message here is that data communications delays are many times overlooked as part of a performance problem. This may be true no matter what the protocol or communications environment may be.

MPE V to MPE XL: There is not a direct migration issue here other than changes in the environment under MPE XL may alter existing data communications delays and perhaps shorten them. The fourth generation issues previously mentioned in Case 3 may be at issue here also.

Case 5: Customer has a Series 70 under MPE V with an online TurboImage application. There has been much growth over the last year and response time has gradually degraded to an unacceptable level. The application is written in COBOLII.

The system never shows more than about 65% CPU Busy and there are no disc I/O bottlenecks visible. The main cause of delay is Impedes. Further examination of the data base locking strategies and queues reveals no problems. The Impedes are on the Global Data Base Control Block.

Using modeling tools, we see that adding more users just makes the problem worse and that the CPU will never be busier because the Impedes keep users from getting to the

CPU. If the Impede is removed, the CPU will eventually become saturated.

Part of the reason for the Impede is that the application opens the data base twice for each user causing more access paths to the data base. This is necessary the way the data base is designed and was not a problem until the number of users reached around 45.

Since this was an application from a third party, the supplier began a major redesign of the data base structure in order to meet the needs of their growing customer base. This redesign allowed each user to only open the data base once and to actually access the data faster.

Summary: This is a perfect example of data base contention on the control block. There can also be similar problems such as contention on the Buffers of the data base or files which can severely limit access to data once a certain number of accessors is reached. More on that in a later study.

MPE V to MPE XL: This could be a complicated migration issue because of major changes made to TurboImage in the MPE XL environment. This situation will continue to evolve with future release of MPE XL. There is the potential issue of more users accessing the data base under MPE XL, but the customer in this case study moved to the MPE XL environment and continued to grow without ever running into another contention problem. The important thing for now is that the improvements in TurboImage XL have helped remove this limitation.

Case 6: Several months after migrating from a saturated Series 70 to a Series 950, Performance seems to be worse at times, especially when certain batch jobs are executing. The major applications are TurboImage written in COBOLII with some third party fourth generation tools used for report generation in batch jobs.

Examination of the Wait Reasons showed what turned out to be contention for data base buffers. The buffer settings were at the defaults of 8(1/2),9(3/4), etc which didn't seem to be a problem on the Series 70. The application however does cause many opens and closes of the data base at times rather than keeping it open at all times for a particular process.

However this is probably not contributing greatly to the problem since there are usually at least 25 users in the data base at any given time. The real problem is that there are just not enough data base buffers available since there are only 17 configured.

DBUTIL was used to increase the number of buffers to 200 for any number of users. (See DBUTIL in HP Documentation.) Immediately performance was noticeably improved. There was much less waiting for buffer availability and Image Overhead was decreased.

Summary: Under MPE V the Data Segment Size limitation also limited the number of buffers which could be configured for a TurboImage Data base since they all had to fit in one Segment. Under MPE XL this limitation has been removed. There is no reason to keep the data base buffers configured small under MPE XL because of the increased memory sizes under Precision Architecture.

MPE V to MPE XL: Always reconfigure the TurboImage Buffers as part of the migration. There is no reason not to. A value of 200 is a good place to start.

Case 7: The customer added Main Memory to their Series III and Performance actually got worse. Unbelievable!

The customer had needed additional memory for quite awhile and had drastically changed his TUNE parameters in an effort to give his on line users better response time and had been moderately successful. Because DQ (batch) jobs very seldom were able to get their segments into memory before being preempted, they never really got to execute and use CPU time. They just caused the Memory Manager to thrash and either Giveup or the DQ process would be preempted by a higher priority process.

After Main Memory was added, the DQ processes were able to be ready to execute when they arrived at the CPU and were able to process more and in some cases lock up the data bases and a KSAM file which the on line users sometimes accessed as well. This was caused partially because the DQ processes were CPU intensive and the DQ quantum had been increased to a large number (2000). This meant that now that the DQ process was in memory he was able to lock certain data structures until his quantum expired, and then sometimes got reprioritized at a CQ level because a CQ process was waiting for him to release his lock. The application should not have allowed the DQ process to lock structures needed by the CQ.

Other cases had the DQ processes just using the CPU for their whole time slice and the CQ users waited for the CPU as a result.

Resetting the TUNE parameters corrected a lot of this problem, but a review of the locking strategy was also done. A batch job should not lock out an on line user.

Summary: It is possible to cause extra system overhead and other scheduling problems by inappropriate settings with the TUNE command.

MPE V to MPE XL: This is a very complicated issue for migration because of the many changes in the MPE XL environment. The next case will discuss this further.

Case 8: A customer has a Series 950 under MPE XL using NS to a Series 70 under MPE V doing Remote Data Base Access, Virtual Terminal Access, and Remote File Transfer, etc. There are sporadic instances of poor on line response times and some cases where one batch job monopolizes the DQ and sometimes the whole CPU.

The TUNE parameters of this system are at the defaults.

The history of this case study is very long and involved. The problem eventually was narrowed down to a variety of priority problems under MPE XL. Many of them were related to NS and were difficult to detect. Many were as a result of other situations within MPE XL.

There have been and still are many discussions going on as to the whys and wherefores of priority problems under MPE XL and there have been many bugs fixed by the erstwhile HP lab engineers and many more will have been fixed before this is read by the public. One theory that has been espoused is that the improvements in the I/O system under MPE XL have allowed priority boosting that was needed with MPE V to become a problem because it is not needed under MPE XL.

In other words, the priority boosting under MPE V was not a problem under MPE V because during the delays to complete I/O other processes were able to execute. The I/O on MPE XL being so much faster does not have these delays and certain processes can now hog the CPU. Whatever the background of these types of problems is is not important. The solution to these problems is important.

There are several scenarios of priority problems which can impact a MPE XL system. One scenario has a system process executing on behalf of a user doing a remote file transfer monopolizing the CPU gradually eventually locking out other processes until it is done with the transfer. The problem is caused by the process not decaying properly within it's queue and in the case of a batch job in the DQ, it sometimes gets relaunched at the top of the CQ where it should not be.

This can result where a batch job in DQ does a REMOTE HELLO. The result of this is a remote session in CQ and if a remote file transfer takes place, a local process will be started in CQ on behalf of the user to do the transfer. This may create some problems since a batch job is now functioning in the CQ instead of the DQ where it was intended. This can be avoided by adding the PRI=DS parameter to the REMOTE HELLO. This will cause the local process to also be in the DQ. (Note: if you want a file transfer to dominate the CQ you are on your own!)

There are other similar problems that have occurred on a variety of customer systems. Some DQ processes will monopolize the DQ. This can be eliminated by flattening (setting base and limit to the same value) the DQ. This customer found that flattening the DQ to the value of the CQ limit caused the entire system to run more smoothly.

While many of these priority problems have already disappeared and more will, the flattening action has worked around many of them. (see TUNE Command in HP documentation)

The other factor which may be affected by the System Manager/Operator is the Quantum or minimum or maximum amount of CPU time available to a process each time it arrives at the CPU. There have been many cases where too large or too small a value have caused excessive dispatcher activity or other forms of sporadic response times. (see Case 7)

More often making this value too small is the problem which emerges. Typically as a system gets busier and on line response time begins to increase, the system manager attacks the DQ to "help" the online users and this will sometimes help, but sometimes extra overhead is introduced which can make the problem worse.

There is no doubt that use of the TUNE command is an optimization and that it is very easy to alter the performance of a system for better or for worse so caution is urged. Also, once a change is made, give it time to 'settle in' before making a second change. There may be certain processes that need to clear out before the new TUNE command becomes completely active.

This customer's Series 950 had 110 interactive users with an eight job limit set. There were certain batch jobs such as a third party spooler, a job scheduler, and an asynchronous device driver which were active at all times. The queues were set as follows:

```
CQ  200,200,152,200
DQ  200,200,160,200
EQ  200,200,201,220
```

The EQ was seldom used. The overlap in the DQ caused CQ processes that decayed enough to have to wait for batch jobs. On a normal system this may not be a problem and in fact may be desirable at times. However, in this case there was a problem.

The application used a fourth generation language to access a TurboImage data base and also used VPLUS. Some Network File Transfer, Remote Data Base Access, and Virtual Terminal activity was common.

The system was typically 99% CPU Busy with no individual process using more than 6% of the CPU in any given minute. One part of the interactive process created a son process in DQ to do a 'large' inquiry occasionally. These processes and other batch processes very seldom got CPU time. On line response time was very sporadic with intermittent delays of as long as a minute. Many CQ processes had decayed to below the DQ base.

The TUNE command was used to reset to:

```
CQ  200,500,152,200
DQ  1000,1000,200,200
EQ  1000,1000,201,220
```

The DQ was flattened to the limit of the CQ which forced a normal round robin action within the queue and with any CQ processes that had decayed to the limit. The increased CPU values allowed processes that arrived at the CPU to process to completion more often or until they did I/O or were preempted.

The impact to the system performance for this customer was dramatic. It was as if we had changed gears and replaced old worn out spark plugs. Our 'engine' was now firing on all cylinders and running smoothly.

CQ processes no longer decayed to where they waited for DQ processes. DQ processes no longer waited for other DQ CPU hogs. Dispatcher overhead was greatly reduced as a result. The average CPU busy dropped from 99% to a more reasonable 80-85% range. This system was still very busy, but the thrashing caused by improper TUNEing and the aforementioned priority problems had been removed.

Summary: Many customers have benefited from flattening the DQ to the limit of the CQ. While it may not benefit all customers, it is a simple thing to try to see if it will help. Also, do not set the CPU min-max too small or too large!

MPE V to MPE XL: As stated before, the CPU size and other operating parameters contribute to the determination of the proper optimum values for the TUNE parameters. The migration issue is that the parameters will probably be set differently under MPE XL then they were under MPE V.

Epilogue: All of these customers were successful in finding a solution to their performance problems. These brief case studies may give you some ideas of the types of factors affecting system performance. It is recommended that the concepts be your guide rather than these specific instances as every system has a different situation and different opportunities for tuning for performance.

It is my hope that you will carefully consider the cases here where customers found a solution without buying hardware. There have been cases where a system upgrade was purchased with no change in performance which is an event we all would like to avoid. Many times there is something just as effective with a lot smaller price tag. Good Hunting!!

DISTRIBUTED DATA BASES

Randy Safier
Proactive Systems
PO Box 7102
Bloomfield Hills
MI 48302
(Tel: 313-333-7200)

Introduction

"Distributed data bases are the most significant development in the commercial data base world since the first genuine relational systems" (Chris Date at Large Data Base Conference).

"Distributed DBMS: in search of wonder glue" (title of article in Datamation by E.D. Myers).

The above are a couple of recent quotations that show the current level of interest in distributed data bases, ie. data bases that can spread over more than one computer.

The Problem

If you manage the data processing facilities of any medium to large organisation (ie. with more than one office location) then you have the problem of designing the computer facilities to meet the different application needs of the users at each location at minimal cost.

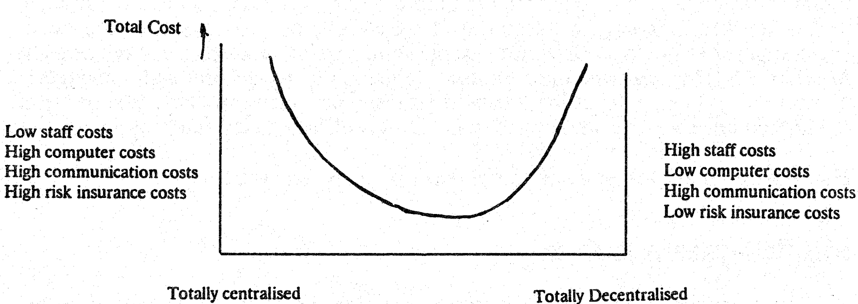
In the stone age of computing the answer was simple - buy a large IBM mainframe, put it in head office, place dumb terminals in all appropriate locations linked back to the central system and everyone will be happy - except maybe your finance vice-president but you can always think of some technical reasons he won't understand for not doing it any other way. Managerially this is great - all your data processing staff and facilities are in one location so they are easy to control and you can build a center of expertise. All the applications can easily interact as they are on the same computer. Someone will mention Groschs Law (the power of a computer goes up with square of its cost) so there must be big economies of scale etc, etc. Your friendly IBM or HP salesman (if you are considering a 950) will always back you up on this kind of proposal.

Unfortunately the economies of scale no longer apply - multiple small computers can give you more "effective" throughput at less cost than a single large system, e.g. how many Micro3000s can you buy for the cost of a Model 70. With a single large system you are placing all your eggs in one basket - you are very vulnerable to computer failure or "acts of god". In addition it totally ignores the communication costs.

The prices of computers are dropping much more rapidly than the cost of telecommunication links. In addition if you look at almost all real situations where locations are spread geographically the communication costs are a very high proportion of the project budget (particularly if you capitalise the on-going costs). For the reasons given above no PTT has ever proposed a centralised telephone switching system!

However it is very easy to go to the other extreme. This was proposed at a board meeting I once attended - "Why don't we ditch our expensive centralised minicomputer and buy 50 PCs to spread around". I don't think I need to explain the problems this would have created a few years ago in trying to share data between applications/programs. The cost of this solution in terms of the chaos, additional clerical procedures and management time must also be very high.

So there has to be some optimal mid point between total centralisation and total decentralisation for any organisation which minimises the costs and yet supports the application needs. There are always many possible solutions and the options are increasing all the time so it is usually necessary to cost out a few alternatives to see which is most economic.



Distributed Systems

OK so we have settled on a distributed system of some kind. These are characterised by:

- They comprise more than one computer (or processing unit).

- The computers are interconnected (even if only occasionally).

- Significant interaction takes place between computers.

- A single system image in the user interface is preferable.

- Many design alternatives are available.

Note that the logical mapping of application system needs onto physical systems is constrained by technological capability and cost. The latter are changing rapidly. However the mapping is likely to be approached by looking at the clusters of processing (both machine and human) and the dependency between processes. The resulting analysis may well come up with the same labels as we conventionally use, e.g. inventory control is an application characterised by the processing of stock records. The degree of interaction, volumes of data flows, required fault tolerance, organisational impact and other such factors can affect how these clusters are perceived.

Partitioning between processing in a network is usually based on a mixture of locational division and functional division. Geography has a large impact on the communication cost so applications are often grouped together and run at one site (this is the "branch computer" scenario where all functions for a location are run on one computer at the location with the same system being replicated ar each branch). Alternatively the network may be split functionally to accord with managerial structure, e.g. finance department have their own computer and associated network; production have a different network with possibly different computer equipment.

The concept of vertical and horizontal partitioning is relevant here.

Single System Image

As regards single system image it would certainly be ideal to provide data and service location transparency, i.e. the user could call upon services or data anywhere in the network using the same commands and without knowing where the data or processing was located. This can be made easier if you have:

Homogeneous Transparency - all nodes in the network run the same computers with the same operating system and data base management system.

However in real life most users have a mix of equipment so it would also be nice to have:

Heterogeneous Transparency - where different nodes run different DBMSs. Obviously this is much more difficult to achieve but progress is being made towards this possibility with the OSI standard, the increasing similarity of relational data bases, the possible standardisation on SQL as the data base access language, etc.

Replication

In an ideal world you wouldn't have replication of data in a network (ie. multiple copies of the same data). One of the early selling points for IMAGE and other DBMSs was the ability to avoid data redundancy. However if you look at any operational data base it usually still has it - not necessarily because of poor design but because you need to optimise data retrieval.

In networks you need it for the same reason. For example, to retrieve a record from a local computer is going to be much faster than from a remote system - if the ratio of enquiry to update is high then it can be more cost effective to hold multiple copies (the comms traffic can be minimised and hence the cost). You may also find it more economic to balance the workload over multiple locations by having replication. In addition you need multiple copies for reasons of back-up and redundancy. Unfortunately data networks (particularly international ones) are not very reliable and even the best computers (eg. HP3000) are going to be down occasionally.

Rule of thumb: do not design any large network on the principal that all of it will be there. Because failure probabilities are multiplicative, even a 99% up time for any one node means a high probability of partial system failure on the network as a whole if you have more than a few nodes or links.

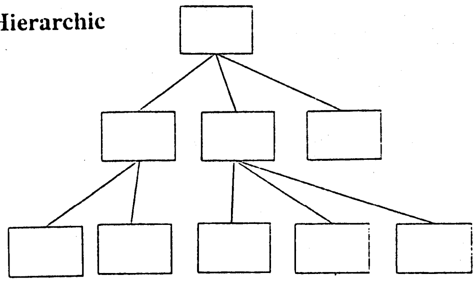
Replication is therefore useful but synchronisation and recovery (eg. rollback) on failure must then be considered.

In addition if you have replication you need data replication transparency plus preferably query optimisation. There is not much point in having a local copy of the data if your end-user report writer chooses by default to take data from the remote system.

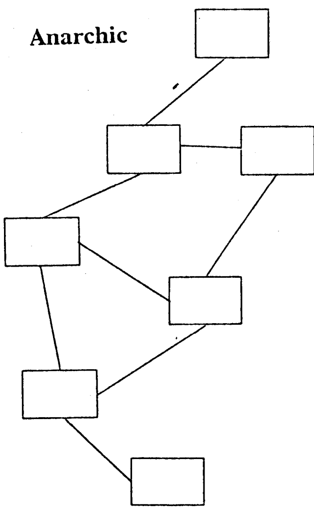
Network Structure

Distributed processing networks fall into basically three structures: hierarchic, anarchic and single connection.

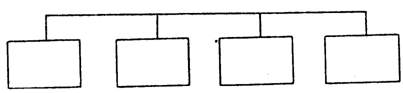
Hierarchic



Anarchic



Single Connection



The State of the Art

Well lets see how HP products such as IMAGE and DS/NS match up to the above requirements and also look at other products.

Now IMAGE was designed on the principle that a data base was a single logical unit. There still isn't a multiple data base recovery option on the same computer let alone on a distributed system. The ability to access and update a data base on another node in the network is provided but you need to know where it is (you can provide limited location transparency by simply using log-on file equates). If the remote data base or link is out of action then your local application aborts. In addition if you have more than a few remote "sessions" then your computer processor load is excessive.

You have data replication with SILHOUETTE, typically a whole data base, and with BACKCHAT where you can also select specific data sets or even records.

You certainly don't have hetergeneous transparency (you can't link a TurboIMAGE data base on HP to an IBM or DEC data base for example).

However things are progressing. There is HP Net Delivery which at least gives you callable intrinsics to help you pass data around a network in "batch" mode even if you need to do a lot of ad-hoc programming. There is Speednet from Infocentre which attempts to map an IMAGE data base (with local replication) onto PCs plus Datasofts MIRAGE.

There is HP SQL which is a more industry standard form of data base even if it may not be ideal in the short term for high volume, transaction processing, commercial systems.

Lets have a look at other suppliers. The two major relational data base suppliers are Relational Technology (with Ingres) and Oracle Corporation (with Oracle). Both run on a variety of DEC, IBM and HP-UX hardware - Oracle is also being ported to the HP3000.

Ingres Star has the following implementation schedule for distributed DBMS functions:

Phase 1 (available now) - Location transparency, multi-site read (single site update).

Phase 2 - Multi site update, data replication.

Phase 3 - Gateways to non SQL systems (eg. IMS)

Oracle are developing SQL*STAR, Tandem have ENCOMPASS and HP are doing some r & d on relational distributed data bases (but don't hold your breath waiting for the resulting product).

As you can see the main thrust in distributed data base development is on relational data bases. This is for one main reason - they are simpler and more industry standard than most network data bases (eg. limited data types) and they do not have the problems of embedded pointers (simple tables are easier to replicate).

The Proposed Solutions

To cope with the problems of data location transparency, query optimisation, etc, most academics in this field propose a global data dictionary which holds information on the location and content of each data base (plus its type if one is aiming for heterogeneity). This has to be replicated on each node (or a sub-dictionary maintained of those parts relevant to each node).

To handle the problem of transaction atomicity (ie. how to ensure consistency and recovery on failure when a transaction can initiate multiple updates on multiple nodes) there are three possible methods:

1. Two stage commit. Effectively the local system asks for a lock on each remote system - if it is given then the transaction is applied and then each commit checked - if any one node hasn't processed the transaction then it is rolled out on all the others (obviously automatic deadlock avoidance is also required in this case). This is currently the favoured technique but bearing in mind the amount of communication that takes place for a single update I wonder if any commercial system with other than very low data volumes could function using this method. It also assumes that all the network is operational which as already pointed out is a big "if".

2. Time stamping with inconsistency checking. In this case records are allocated a time stamp so that they can be serialised when applied to a remote data base (or if a later transaction is discovered as being applied to the remote data base before the local one then it is rolled out). A variant of this technique is used in our BACKCHAT product where it is possible to turn on a check that the remote record being updated is still consistent with how it appeared to the local updating process - if it is the update is applied - otherwise exception processing routines are entered. This method can cope with time delays between local and remote systems (due to communication time or temporary node/comms failures).

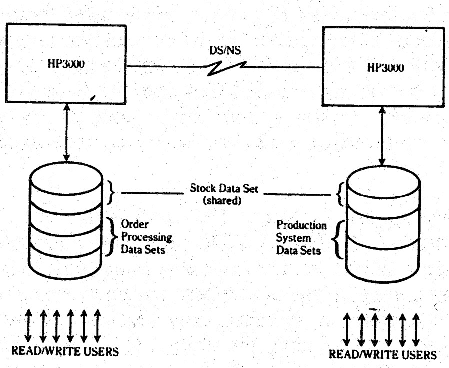
3. A pragmatic approach where the data base software imposes no limitations but the application design has to be done to avoid conflicts - this is not as difficult as it may seem in practice (this is what has to be done in most clerical systems for example).

The Route We Took

We had the problem a couple of years ago of meeting the data replication requirement for a particular application to be based on IMAGE data bases running on multiple HP3000s. The approach we took was to use the IMAGE logging mechanism to collect information on local updates and by passing them around a network perform the remote processing. The resulting product is called BACKCHAT and it has been used to meet many different distributed systems requirements.

We have also recently added a module called BACKBONE to provide data location transparency. In this case IMAGE calls are intercepted and routed automatically to a remote system using NetIPC as the communication method. A "global data dictionary" is used to hold configuration information and enable easy switching of routings. In neither case do application programs need to be changed. In practice these two technologies are to a certain extent interchangeable but they have different operating characteristics and degrees of fault tolerance so that one or the other, or a combination, is selected to best meet the users needs.

It is very easy to build the following kind of application using these techniques:



In the above diagram a part of the data base is replicated over both systems. Users on both computers can update the same data. Recovery is automatic whenever or wherever a failure occurs.

On each node in the network (which is independently configured and therefore can be totally anarchic) is a configuration file which contains information on the "shared" (or "replicated") data bases/sets, the incoming/outgoing communication paths and processes, etc. Each node uses IMAGE in the normal way - no special programming is required.

Examples

The following are examples of the kind of application solved by this approach:

1. FM Insurance

REQUIREMENT: to share data and spread processing over 4 HP3000s based in different countries around the world.

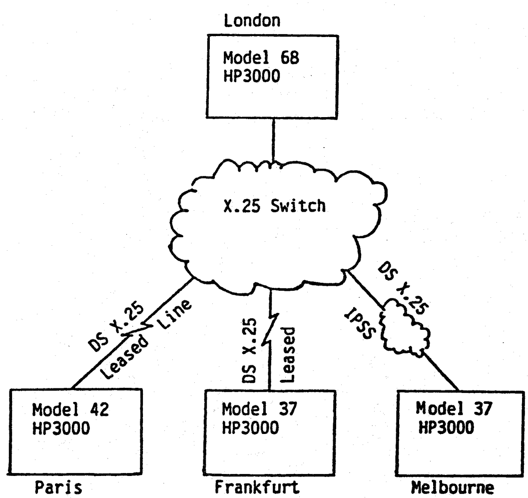
Because their clients are multi-national companies it is necessary for different offices of FMI to share information - for example the same client account number must be used irrespective of where a new policy is created (only by such means can total risk and loss rates for particular clients be easily summarised). Therefore a lot of common reference information needs to be maintained even though changes such as new accounts or uninsured risks can arise in any of the locations. However both for legal reasons and because much of the processing and file storage requirements were specific to local offices, a centralised approach was not considered feasible. The DS/3000 communication software could be used to transfer information... but it hardly solved the above application requirement. Moreover with only minimal dp staff being present at the central site and none elsewhere, the system had to be robust and capable of maintaining system data integrity irrespective of the processor link failures that commonly occur in a multi-machine communications network. The system also needed to provide automatic consolidation of data entered in each branch office into a central management information data base.

As a result of discussions between FM and Proactive Systems, extensive enhancements were made to BACKCHAT to support the above needs. For example the ability to replicate a data base with updates being entered on either copy was implemented (plus, of course, a way of stopping transactions echoing backwards and forwards for ever). In addition, because only certain information needed to be shared (with other information being maintained solely by a particular location) it was necessary to be able to select a subset of the data bases for replication - a selection by data set was used as a convienient way to split the application.

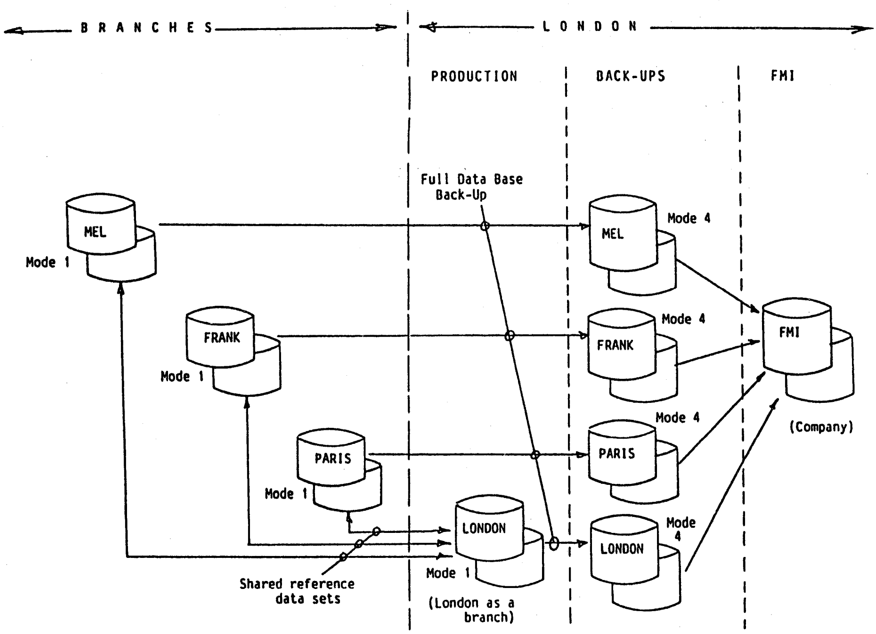
Note that because BACKCHAT accumulates transactions in its buffer files when a line failure occurs and will automatically catch up when the line is restored, operational recovery is straight forward. Also if a computer at one branch fails then this has minimal impact on the computers in the other locations.

Original Configuration

180 Data sets at each location
50 "Shared" data sets



Data base sharing

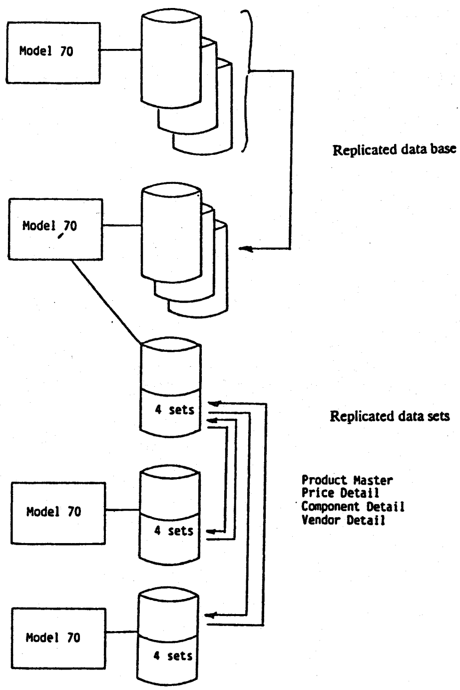


2. Epson America

REQUIREMENT: to spread workload and provide fault tolerance over multiple locally-connected Model 70s.

In this case for managerial reasons it was chosen to centralise the computer installation and support services. However to provide redundancy and because of the limitation on the power of any one HP3000 computer it was decided to spread the workload over several Model 70s. This involved both replication of a complete data base and selected replication of certain "reference" data sets (i.e. those that contain semi-static data referenced by many systems). However the latter can be updated on more than one of the computers so the system provides transparency of data location.

Configuration



Conclusion

I hope that I have covered some of the basics of distributed systems which is a very large topic if you wish to study it further. Also hopefully I have shown you how it is possible to build distributed systems based around IMAGE pending the arrival of other solutions based on SQL. It is certainly now practical to build high performance, distributed, transaction processing systems that embody data replication and data location transparency without writing a lot of application specific code.

References:

1. Aspects of Distributed Computer Systems, H.Lorlin, John Wiley 1980

2. Distributed DBMS: in search of wonder glue, E.D.Myers, Datamation, Vol 33 No.3

3. Relational Databases: State of the art report, D.A.Bell, Pergamon Infotech 1986

4. Design and Strategy for distributed data processing, J.Martin, Prentice Hall 1981

5. Recent Advances in distributed data base management, C. Mohan, IEEE Computer Soc. Press 1984

6. Distributed data bases, C.Delobel & W.Litwin, North Holland, 1980

7. Centralised and Distributed Data Bases, W.W. Chu and P.P.Chen, IEEE Computer Press, 1979

8. The Concept and Applications of Distributed Data Bases, E.J.Neuhold, Vienna HP User Conference Proceedings 1987.

MPE XL Switch Subsystem

by

R. Gregory Stephens
Hewlett-Packard Company

This paper addresses the subject of the MPE XL Switch Subsystem. The Switch Subsystem is that portion of the MPE XL operating system which allows a Native Mode program to call a procedure that resides in Compatibility Mode. It also allows a Compatibility Mode program to call a procedure that resides in Native Mode. At first this may sound like a rather esoteric feature. However, in situations where migration of an application to Native Mode is not a simple recompilation, the Switch Subsystem can provide a means of getting increased performance for incremental migration.

This paper will address several important aspects of using the Switch Subsystem, but it is by no means a substitute for the MPE XL Switch Programming Guide (p/n 32650-60030). In addition, it will not cover the subject in the depth that the accompanying presentation will. To be more specific, the paper will discuss a strategy for using the switch subsystem and writing switch stubs as well as addressing some of the potential pitfalls to watch for in writing switch stub routines. It does not provide a complete tutorial on how to write switch stubs nor how to use each of the switch intrinsics and their parameters (although the presentation will cover these subjects in more detail).

Compatibility Mode and Native Mode Addressing

There is a fundamental technical reason that we need the Switch Subsystem and that is the fact that addressing of data structures is completely different in Native Mode versus Compatibility Mode.

To provide object code compatibility between the Stack Architecture based MPE V HP 3000's and the new HP Precision Architecture MPE XL HP 3000's, the 'stack-3000' addressing used by the instructions in MPE V programs had to be supported on HPPA machines. This means that Compatibility Mode addressing has to be the same as the addressing on MPE V systems. So, in Compatibility Mode, the stack is still a 16 bit data structure that is addressed via registers like DL, DB, Q, S, and Z. Compatibility Mode programs are still segmented into 16k-word segments that are addressed via the P, PB,and PL registers.

In Native Mode the full features of the HP Precision Architecture are exposed. 32 and 64 bit addressing can be performed and the Native Mode stack and heap together, can theoretically be up to 1Gb. Code and data addressing conventions are also completely different from Compatibility Mode.

If the goal is to allow a Native Mode program to call a Compatibility Mode procedure, we have, by definition, a challenging problem of differing code and data addressing conventions to overcome. In Compatibility Mode, instructions still reference a 16 bit stack which is an entirely different data structure from the 32 bit Native Mode stack that is accessed by Native Mode code.

Phased Migration

Migration from MPE V to MPE XL offers a number of choices to the application developer. The first step in migrating to the 900 Series HP 3000 systems in most every situation is to restore an application and run it in Compatibility Mode with little or no changes to non-privileged mode programs. Even in situations where long-term execution of an application in Compatibility Mode may not be desired, simply running an application in Compatibility Mode requires no recompilation and makes this a natural first step to establish that the software and hardware configuration required to execute the application is in place before proceeding.

Having tested an application in Compatibility Mode, the next step is to determine which portions of the application should be recompiled into Native Mode. This may include all pieces of an application, none of the application, or any combination in between. (For those portions remaining in Compatibility Mode (CM), the Object Code Translator (OCT) provides a simple means of improving CPU performance of programs and SL's that will remain in CM.)

For programs that we want to be able to take advantage of NM performance or features (such as mapped files) we have to look closely at the language that they are written in and any dependencies that these programs may have on other code. The first question is whether or not the program itself was written in a language that is supported in Native Mode (as of this writing, Feb 1989, COBOL II, Pascal, Fortran 77, RPG, and HP C Native Mode compilers are all available from Hewlett-Packard, with HP Business Basic planned for the future rèlease. A Native Mode SPL compiler is available from Software Research Northwest).

If the program is written in a language that is supported in Native Mode, the next question is concerning dependencies that the program may have on other code. This may be procedure libraries that reside in RL's or SL's which the program is dependent upon. If so, we then have to look at the language this code is written in and the availability of source code for the procedures. We must also determine which other programs are dependent upon this procedure library or SL.

In many situations, the library routines have been written in SPL or they were supplied by another party and source code is not available. In these situations, the Switch Subsystem can act as the mediator between the code that you migrate to Native Mode and the code that remains in Compatibility Mode. Switch can also provide flexibility when migrating code from CM to NM by allowing the code to be migrated to NM in stages. When a number of programs are dependent upon the same library routines, a set of switch stubs allows these programs to continue to call the library routines from either mode as programs are migrated over time to NM.

Switch Intrinsics

The Switch Subsystem is called via 3 intrinsics. The 'HPSWITCHTOCM' intrinsic is used for Native Mode to Compatibility Mode switch calls. The 'HPSWTONMNAME' and 'HPSWTONMPLABEL' intrinsics make the Compatibility Mode to Native Mode procedure calls.

At this point it is worth emphasizing that switch calls are made on a procedure basis. If you are using the process handling intrinsics (CREATEPROCESS, CREATE, ACTIVATE, etc.) to spawn a son process and the process to be spawned is a program in a mode other than the currently executing mode, the switch subsystem is not needed since the creation of a process is essentially the same whether you type the :RUN command or use the process handling intrinsics. In both cases the MPE XL loader is invoked and it will determine which environment (CM or NM) the program needs to be loaded.

Call by Name or PLABEL

There are two ways to invoke a procedure on MPE V systems. The first and most obvious way is to call the procedure by its name. For instance, if we have a procedure called "JULIAN'TO'GREGORIAN'DATE" and we want to call it via our COBOL program we would write a statement such as:

```
CALL "JULIAN'TO'GREGORIAN'DATE" USING JULIAN-DATE-FIELD,
                                GIVING GREGORIAN-DATE-FIELD,
                                       STATUS.
```

There is another, less known way to invoke a procedure and that is to dynamically load and invoke the procedure by its PLABEL (or Procedure Address). To dynamically load and invoke this procedure on MPE V in SPL first requires a call to the LOADPROC intrinsic. This intrinsic loads the procedure into MPE's system tables and returns the PLABEL for that procedure.

The next step is to push all of the parameters onto the stack, push the PLABEL onto the stack and perform a PCAL 0 instruction which will take the PLABEL off the top of the stack and call that procedure.

This dynamic procedure calling is fine, but who cares? On MPE V the answer to that question was 'Not many people'. Its primary use was to allow you to determine, at run-time instead of compile-time, which procedure you want to call.

The reason it is relevant to a discussion of Switch is that we can now use this same capability to increase the performance of Switch calls in both directions. When a call to Switch is made using the procedure name, Switch, by default, dynamically loads the procedure, hashes the name to create an entry in a hash table for that procedure, and then invokes it. This means that the initial call to the procedure requires additional overhead, to search one or more SL's for the procedure to be called, while repeated calls only require the name to be hashed to arrive at the callee's PLABEL.

If we are going to invoke a procedure hundreds or thousands of times within a program, we can get even better performance by loading the procedure ourselves and saving the PLABEL. On subsequent calls to switch we can use the PLABEL and no loading or hashing takes place.

Switching to Compatibility Mode

There is one intrinsic provided that makes the Native Mode to Compatibility Mode switch whether we want to make the switch using the procedure name or PLABEL. This intrinsic is HPSWITCHTOCM. Because it requires us to pass explicit pointers (as opposed to parameters that are passed by reference, in which case the compiler implicitly generates and uses pointers on your behalf) it can only be called from HP Pascal/XL or HP C/XL. These are the only HP supported Native Mode languages that allow the programmer to generate and manipulate pointers.

To successfully use HPSWITCHTOCM requires a complete understanding of the procedure to be called. Some of the more obvious information that we have to supply to the switch intrinsic is the name of the procedure, the SL search path (equivalent to the ;LIB=G/P/S parameter on the :RUN command), the number of parameters to be passed, and whether we are calling a procedure or function.

For each of the parameters, we must provide the length of the parameter, whether the parameter is passed by value or reference. For parameters passed by reference we must also specify whether a byte or word address is expected by the callee. For performance reasons, which will be discussed in more detail during the presentation, we can specify that a reference parameter is used as input and or output to the callee.

Switch Stubs

If we look at the scenario mentioned earlier with a Native Mode COBOL program calling a Compatibility Mode SPL procedure, a good approach to structuring our calls to switch is to create a set of routines, called switch stubs. By creating our own library of Native Mode switch stub routines that mimic the actual Compatibility Mode library routines we want to call, we maintain a familiar structure and we isolate the switch stubs from the caller and callee code.

By taking this structured approach, we maintain flexibility in migrating the CM library in phases. This allows us to port the CM library to NM one routine at a time. As each routine is recompiled in Native Mode, its corresponding switch stub is deleted from the switch stub library. The new NM version of the routine is put in its place, while doing this we never had to modify or even recompile the calling COBOL program.

Switching to Native Mode

The concepts involved in switching from Compatibility Mode to Native Mode are similar to its NM to CM counterpart when it comes to the parameters that switch requires. The primary differences are a result of the fact that the Native Mode callee will have no problems addressing reference parameters that are on the Compatibility Mode stack or heap since NM code has the ability to use the full 32 bit addressing of the HPPA. As a result, CM to NM switching does not care about byte or word addresses or whether a reference parameter is input and/or output to the callee.

The other obvious difference is that there are two intrinsics supplied for CM to NM switches. Switches by name are made via 'HPSWTONMNAME' and switches by PLABEL are made via the intrinsic 'HPSWTONMPLABEL'. To initially load a Native Mode procedure and retrieve its PLABEL we use 'HPLOADNMPROC' before we call 'HPSWTONMPLABEL'.

Advanced Switch To NM Features

One of the first performance features that may be worth taking advantage of is the switch by PLABEL capability. To take advantage of this capability requires a call to 'HPLOADCMPROCEDURE' when making a NM-to-CM switch call. This intrinsic loads the procedure, like the original 'LOADPROC' intrinsic does. It also establishes an entry in switch's hash table as was discussed earlier. From that point on the 'HPSWITCHTOCM' intrinsic can be called using the PLABEL option to get better CPU performance. Because we called the 'HPLOADCMPROCEDURE', instead of 'LOADPROC', it also means that switch's hash table is loaded with an entry for this procedures PLABEL.

Because of the fundamental addressing restrictions that Compatibility Mode programs have, reference parameters that are passed from a Native Mode program to a Compatibility Mode procedure must be copied onto the CM Stack before the CM procedure can be invoked. On return from the CM procedure switch will copy the reference parameter back to the Native Mode data structure (normally the NM stack or heap) from which it came. Obviously, this could amount to a significant amount of overhead, imaging copying at 32k byte buffer 2 times for each call to a procedure that will be called thousands of times in a program.

To provide better performance in these situations, the HPSWITCHTOCM intrinsic has a couple of options. The first is the input/output copying option that was discussed earlier. For example, if a routine you are calling with a reference parameter does not require that reference parameter as input, you can specify it is 'output only' from the callee procedure. This tells switch not to copy the actual reference parameter in Native Mode over to the Compatibility Mode stack, saving us CPU time.

Another parameter which can provide increased performance for NM to CM switch calls is the 'method' parameter of HPSWITCHTOCM. This parameter allows you to specify that the callee routine is 'split stack callable'. The split stack feature on MPE V based HP 3000's allows a privileged program to point the DB register away from the stack to an extra data segment as a means of addressing a structure without having to copy it to the stack. When the split-stack option is used in the HPSWITCHTOCM call, it tells the Switch Subsystem that it doesn't have to copy a reference parameters from the Native Mode data structure to the CM stack. Instead switch creates a CM extra data segment that is 'wrapped-around' any reference parameters, switches DB to the newly created extra data segment and then invokes the callee.

It is extremely important to note that this capability REQUIRES PRIVILEGED MODE AND A THOROUGH UNDERSTANDING OF ADDRESSING AND THE 'STACK-3000' INSTRUCTION SET AND ADDRESSING. It also requires that the callee routine must have been written with the specific ability of being able to be called in split-stack mode. If you call a non-split-stack callable routine with DB split away from the stack, the results will at the least be unpredictable and cause data loss, program abort, or even system abort.

Switch Stub Pitfalls

One of the easiest ones to make and most difficult to duplicate is related to NM to CM switches with the input/output option specified incorrectly. On the surface, we might be writing a switch stub to call our FREAD-type procedure that resides in a Compatibility Mode SL. Recognizing that we could be reading data structures of up to 32 or 64k bytes and that our FREAD intrinsic does not need this potentially large buffer copied as input, we might specify that the 'buffer' reference parameter is 'output-only'. By doing this when we call our FREAD procedure, we have told switch that it doesn't have to copy the contents of our 32k byte buffer that is sitting in NM over to the CM stack. It only has to make room for a 32k byte buffer on the CM stack and copy that buffer from the CM stack to the NM data structure when our CM FREAD procedure has completed.

But what happens if we call our FREAD procedure and the procedure encounters some kind of normal (or abnormal) failure like EOF? In this case switch would have allocated space for this 32k buffer on the CM stack, called the FREAD procedure, and after completing (with errors), copied the CM buffer back to NM. Assuming that no data was successfully read by the procedure, switch will be copying whatever garbage was on the CM stack at the location of the 32k buffer it had allocated to the NM stack. This is different than how our FREAD worked on MPE V since our FREAD would probably have left the buffer untouched on a failure.

This may or may not be a problem. It is a problem if the calling program assumes that the buffer is unchanged on a failure (say EOF) condition and it accesses the buffer, which now contains garbage, based upon an assumption which is no longer valid!

Summary

The MPE XL Switch Subsystem can be an extremely valuable piece of a long term migration plan. It also requires a thorough understanding of the application program and libraries.

(apr '89)

THE TRUTH ABOUT MPE/XL DISC FILES

by Eugene Volokh
Vice President of Research & Development
VESOFT, Inc.
1135 S. Beverly Dr.
Los Angeles, CA 90035, USA
(213) 282-0420
Telex 9102409759 VESOFT
Fax (213) 785-9566

ABSTRACT

Several years ago, I wrote a paper called "The Truth About Disc Files". In it, I tried to describe some aspects of files that, I felt, inquiring minds wanted to know -- things like extent considerations, blocking, file locking, etc. Some of those things remained the same under MPE/XL's file system, but many have substantially changed; this paper will try to describe some of the key differences and similarities between the MPE/XL and MPE/V file systems, and explain some of their practical implications.

HOW FILES ARE STORED -- EXTENT CONSIDERATIONS

One of the key limitations of MPE/V was that you had to know a file's maximum size when you were building the file. If you guessed too low, your programs would eventually abort with an *END OF FILE* error; if you guessed too high, you could waste a lot of disc space.

Actually, technically speaking, you didn't **have** to know a file's true maximum size; you could always build the file with a very large file limit, e.g.

```
:BUILD MYFILE;DISC=1000000
```

and you'd be rather certain never to overflow it. The trouble, of course, is that the file's disc space was **not** allocated on a simple as-needed basis, but rather **one extent at a time**. Since a file by default had a maximum of 8 extents, the above :BUILD would build a file that was split into up to 8 chunks of contiguous disc space; the very act of building the file would allocate one chunk of this space, which would occupy 1,000,000 / 8 = 125,000 (contiguous!) sectors. (Remember that a sector is 256 bytes -- we used to say 128 words, but not any more; since a "word" means *2 bytes* on Classics but *4 bytes* on Spectrums, I will try to use "word" as infrequently as possible in this paper.) Even if you said

```
:BUILD MYFILE;DISC=1000000,32
```

to build the file with up to 32 extents, the file would initially be allocated with over 31,000 sectors. In other words, it wasn't so much selecting the right file limit that was the problem, but rather that selecting a file limit that was too high would cause prohibitive consumption of disc space.

This may seem somewhat nitpicky, but it is actually quite relevant to MPE/XL. **MPE/XL also requires you to specify the**

maximum size for a file. However -- and this is a big "however" -- **it lets you specify a very large file limit without using huge quantities of disc space.** If in MPE/XL you said

```
:BUILD MYFILE;DISC=1000000
```

the file would be allocated **2,048 sectors at a time**, even if this would require it to have almost 500 extents when full. Thus, you get the best of both worlds -- the file can grow to up to 1,000,000 records but will never have more than 2,047 sectors of wasted disc space. You'll find that MPE/XL often builds files (e.g. XL's built by LINKEDIT's -BUILDXL command) that have file limits of 4,096,000 -- more than they'll ever need, but what does it matter? In fact, the highest "maximum maximum file size" -- i.e., the highest (file limit * record size) value that you can have -- is 8,388,607 sectors.

One of the reasons why MPE/V had the 32-extent limit was that the disc addresses of all the extents were kept in the 256-byte file label. Since each disc address was 4 bytes, and a little bit less than 128 bytes were required for other file information (e.g. file name, creator id, file code, etc.), that left room for only about 32 extent pointers.

MPE/XL didn't make the same mistake of keeping extent pointers in a single fixed-size array. Instead, each file has a linked list of "extent descriptor blocks", each of which has 20 12-byte entries that point to the extents of the file. Thus, a 32-extent file on MPE/XL will have:

* **a file label,** which points to
* **an extent descriptor block,** which contains the disc addresses of 20 extents and also points to
* **a second extent descriptor block,** which contains the disc addresses of 12 extents.

Granted, this is 3 sectors (as opposed to the 1 sector, which is all that is needed for the file label on MPE/V), but think of the flexibility -- new extents can be added to the file with no difficulty. To avoid possible performance problems with access to files that have many extents, MPE/XL builds a special "extent descriptor B-tree" whenever you open a file; that way, it can very quickly find the address of an extent even in a many-hundred-extent file.

All right, you've said:

```
:BUILD MYFILE;DISC=1000000
```

and now you do a `:LISTF MYFILE,2`. What do you get?

```
ACCOUNT=  VESOFTD       GROUP=  WORK

FILENAME  CODE  ------------LOGICAL RECORD-----------   ----SPACE----
                  SIZE  TYP        EOF       LIMIT R/B  SECTORS #X MX

MYFILE            128W  FB           0     1000000   1        0  0  *
```

There are, obviously, three unusual things in this picture:

* The number of sectors allocated to the file is 0. If you recall, on MPE/V, even an empty file always has at least one sector allocated to it, and usually more. This is because on MPE/V, file labels were kept as the first sector of the first extent of the file, so each file always had to have at least one extent allocated to it. In MPE/XL, file labels are kept separately from the file data (in a special portion of the disc called the "file label table" -- extent descriptor blocks are also kept there), so no data extents are allocated and thus 0 sectors are actually allocated for the data. Of course, the file label still takes up 1 sector of space, but that doesn't get budgeted to the file in MPE/XL.

* The number of extents allocated to the file is 0. See above paragraph.

* The maximum number of extents for the file is "*". This means that the file was built **without** a maximum number of extents (though, as we'll explain later, even if it were built **with** a maximum number of extents, this number still wouldn't really be a maximum!). For extra credit, try to guess what an "*" in the "# of extents" (as opposed to "maximum extents") column means; we'll discuss that later.

Now, say that we write one record into this file and then do a :LISTF. We'd see:

```
ACCOUNT=  VESOFTD      GROUP=  WORK

FILENAME  CODE  ------------LOGICAL RECORD-----------   ----SPACE----
                  SIZE  TYP        EOF       LIMIT R/B  SECTORS #X MX

MYFILE            128W  FB           1     1000000   1     2048  1  *
```

One 2,048-sector extent has been allocated to this file -- it will be enough for the first 2,048 records to be written into MYFILE. When we try to write the 2,049th record, another 2,048-sector extent will be allocated, and so on. As we mentioned before, this means that no more than 2,047 sectors in this file will ever be wasted (allocated but unused).

This is quite nice, but if each such file wastes an average of 1,000 sectors (an average between those that waste 0 and those that waste all 2,047) and you have 2,000 such files, we're talking about 500 megabytes of wasted space -- about the size of one disc drive. Looking at it this way, saying that "at most 2,047 sectors can be wasted" is small comfort.

It would have been nice if we could build the file indicating what we'd like its extent size to be; we might then build our files with huge file limits but tell MPE/XL that they are to be allocated only, say, 512 or 256 sectors at a time.

Unfortunately, this is (to the best of my knowledge) not possible. MPE/XL determines the size of the extents that it will try to allocate for a file using a (rather bizarre) formula based on the maximum number of sectors the file can ever contain:

MAXSECT = (userlabelflimit*256 + recordsize*flimit) / (16*256) * 16
(i.e. the total number of sectors in the file's data portion, rounded up to the next multiple of 16 sectors)

If MAXSECT<=127	then DEFAULTEXTENTSIZE = MAXSECT rounded up to the next highest multiple of 16
If MAXSECT<=255	then DEFAULTEXTENTSIZE = MAXSECT rounded **down** to the next **lowest** multiple of 16
If MAXSECT<=4095	then DEFAULTEXTENSIZE = 128 sectors
If MAXSECT<=65535	then DEFAULTEXTENTSIZE = (MAXSECT/32) rounded down to the next lowest multiple of 16
If MAXSECT>=65536	then DEFAULTEXTENTSIZE = 2048 sectors

Don't ask me why this is the case -- it just is. (It certainly makes some sense for extent size to vary as a function of the file size, but I'm not sure why it varies exactly in this unusual way.)

Note, however, that one other great new feature of MPE/XL is that if it can't find extents of the size that it wants (e.g. there is no empty chunk of 2048 extents), it will just allocate smaller extents. MPE/XL will **not** report an "*OUT OF DISC SPACE*" condition unless there really isn't enough disc space for the file **or** there is enough disc space but it's reserved for virtual (transient) memory. Disc fragmentation does **not** appear to harm things except that it may make files be built with smaller extents.

So, this all shows (among other things) that any files with a maximum of 65,536 or more sectors will have space allocated for them in 2,048-sector chunks. If you have one of those files, how do you save the 1,000 sectors or so that will, on the average, be wasted?

On MPE/V, you could always "**squeeze**" the file (FCLOSE it with disposition 8 or MPEX %ALTFILE;SQUEEZE), which would set the file's file limit to be equal to its end of file and thus save the wasted space. Unfortunately, it would also prevent any more records from being added to the file (unless you rebuild it or %ALTFILE;FLIMIT= it).

MPE/XL has a very nice alternative to this that we call "trimming". It lets you tell MPE (using a new FCLOSE disposition) to **deallocate any unused space allocated to the file without changing the file's file limit.** In other words, this will save space **without** making files any less expandable; an operation such as MPEX's

```
%ALTFILE @.@.@;XLTRIM
```

can save you hundreds of thousands of sectors (it did for us and we only have a 925LX with two disc drives). Actually, before doing this, we called PICS and asked whether there were any files that should **not** be trimmed and they told us that they didn't know of any; I then did the trim of all the files in the system and nothing **seemed** to fail. Be warned, though -- it's certainly **possible** that some program (probably a heavily privileged one, since normal programs have no way of knowing whether a file has been trimmed or not) doesn't like its files trimmed.

How does this work? Well, say that you have a file with five 2,048-sector extents, the last of which contains only 537 actual sectors of data. When you tell MPE to trim the file, it will deallocate the last 1,504 sectors of the last extent, leaving it with only 544 sectors. (544 = 537 rounded up to the next highest multiple of 16; files are **always** allocated in multiples of 16 sectors.)

Now, the file has four 2,048-sector extents and a 544-sector extent. If you start adding more records to it, more 2,048-sector extents will be allocated to it; you may then again want to trim the file.

What makes this whole process work is that MPE/XL **allows you to have extents of different sizes.** If, like in MPE/V, all extents (except for the very last of the possible extents) had to be the same size, you wouldn't be able to throw away unallocated data because that would leave the last allocated extent with a different extent size. MPE/XL, however, is not bothered by this -- I've often seen files with many different sizes for many different extents. The "extent descriptor blocks" that I mentioned earlier actually contain several pieces of information for each extent:

* the **disc number** on which the extent resides (actually, the volume table index);
* the **starting sector address** of the extent;
* the **size** of the extent, in sectors;
* and, the **sector number of the first sector of data (relative to the start of the file) that resides in this extent.**

Actually, with a structure this flexible, it's even possible for records #0 to #999 to be located in the **second** extent of a file and record #1000 to #1999 to be located in the **first** extent! (Of course, when you read the file, the records will come out in the right order, but internally, inside the extent descriptor blocks, the extent information will be kept out of order.)

What are the **disadvantages** of trimming files? To the best of my knowledge, there are very few. Trimming files will increase disc space fragmentation, but it's not clear to me that this is a problem on MPE/XL, especially since MPE/XL seems to handle correctly situations where it can't find extents of the size that it wants (if necessary, it just allocates more smaller ones).

Trimming files **does** cause the file to have more extents, which **may** have a slightly adverse effect on performance. The file system usually tries to read 128 or more sectors (16,384 bytes) at a time, so if a file is all 16-sector extents (the smallest size possible), you will lose the advantages of these large I/Os since you'll never have 64 contiguous sectors. However, if a file has, say, 512-sector extents rather than 2,048-sector extents, this should cause minimal performance penalties (if any at all).

On the other hand, if you feel that a trimmed file isn't giving you the performance you'd like, you can copy its data into a new copy of the file, and all the extents will then be the same size (whatever the extent-size algorithm we showed above dictates). You can even build the new file with a higher file limit (to increase the extent size) and **then** trim the file to save as much space from the last extent as possible. MPEX users can do this by saying

```
%ALTFILE filename;FLIMIT=4096000;XLTRIM
```

-- this will rebuild the file to have all its extents be 2,048 sectors each except for the last one, which will only be as large as necessary. This will give you the maximum disc space savings as well as the maximum possible extent sizes (if that's what you want).

(Note that files with EOF = FLIMIT do not require trimming; the MPE/XL file system automatically allocates the last extent to be just large enough to fit all the records up to the FLIMIT, even if this makes the extent smaller than the other extents.)

In light of all this, why does MPE/XL still support the maximum number of extents parameter of the FOPEN intrinsic and the :BUILD command?

Well, compatibility is one reason. There are programs out there that might, for instance, do an FGETINFO or FFILEINFO of a file's maximum number of extents -- they should be able to get the value specified when the file was :BUILDed (:BUILt?). In fact, MPE goes so far as to return 8 as the maximum number of extents when you do an FFILEINFO for a file that was built **without** a maximum number of extents -- all this just to make sure that the program will get a value (though an incorrect one) that it will be able to handle.

Another reason involves moving files from MPE/V to MPE/XL and back. If you move a file from MPE/V to MPE/XL, it will have the same "maximum extents" value that it did on MPE/V (even if it will be ignored by MPE/XL). Then, if you move the file back to an MPE/V system, it will have the same maximum-extents value that it originally had. If the MPE/XL file had no maximum extents, MPE/V will select a "reasonable" value for this (based on the number of sectors the file uses).

Finally, the maximum number of extents is used (though in a very strange way) when you build a file specifying **both** the maximum number of extents **and** the number of extents to initially allocate. For instance, say that you enter

```
:BUILD MYFILE;DISC=100000,32,4
```

What do you suppose will happen? Here's what a :LISTF of the file will show:

```
ACCOUNT=  VESOFTD      GROUP=  WORK

FILENAME  CODE  ------------LOGICAL RECORD-----------  ----SPACE----
                  SIZE  TYP          EOF      LIMIT R/B  SECTORS #X MX

X                 128W  FB             0     100000   1    12512  1 32
```

The file was built with **1** extent of **12512** sectors! MPE/XL decided that what you wanted is a file with 4/32nd (= one eighth) of its space allocated, so it built you one like that, although with all that space allocated as one contiguous extent. From then on, the file will be allocated in normally-sized (in this case, 2,048-sector) chunks. Eventually, the file will need more than 32 extents (this file would, if full, need more than 40), and MPE/XL will just blithely ignore the maximum extents value and allocate as many as it needs. Thus, you'll often see files with more extents than the maximum (rather perplexing when you first see them).

Incidentally, to answer the question we asked earlier: if a file has 100 or more extents, MPE/XL will show an "*" in the "number of extents" column of a :LISTF ,2 listing. Getting the actual number of extents can prove difficult; the only ways I know of doing this are writing a program that opens the file and then calls FFILEINFO, or using MPEX's %LISTF ,2 (which always shows the actual number of extents).

Finally, remember that **IMAGE databases must still be fully allocated when they are built** -- I believe that IMAGE does this not because of any file system limitation but rather for data consistency's sake; it doesn't want to run out of disc space in the middle of a DBPUT. I have, however, heard a **hot rumor** that a future version of TurboIMAGE/XL will allow a detail dataset to be expanded when it runs out of space (so that you can initially allocate it with less space than it will eventually need); but (or so the rumor says), each dataset can only be expanded **once** in its life -- once it's expanded, it better not overflow again!

Seems bizarre, but that's what I've heard. Believe it or not.

HOW FILES ARE STORED -- BLOCKING CONSIDERATIONS

In discussions of MPE/V, much was said about blocks, blocking factors, and their effects on speed and disc space. Just when we had all taken the time and trouble to learn all of their intricacies, MPE/XL has made them (almost) completely irrelevant.

In MPE/XL, all physical disc I/O is done in multiples of one **page**, which is **4,096 bytes**. This is not to be confused with the **pages** that are **2,048 bytes** -- yes, that's right, there are **two** kinds of pages, each of different size, and both of which are called pages. One, which is 2,048 bytes long, is the unit in which the hardware sees the world; the other, which is 4,096 bytes long, is the unit used by the operating system from the memory manager on up (including the file system).

In any event, physical I/O is done some number of 4,096-byte pages at a time (it's good that it's a 4,096-byte page because you want to do I/Os in fairly large chunks). Since each 4,096-byte page consists of 16 256-byte sectors, a file is **always** allocated, read, and written in multiples of 16 sectors at a time.

Remember that the whole point of MPE/V **blocks** was that a block was the unit of file transfer for this particular file. This may have made sense in the very earliest HP 3000s, which often had as few as 128 Kilobytes of memory, and which couldn't afford huge chunks of this for file buffering.

However, since on MPE/XL file transfer is always done 4,096 bytes at a time, the concept of a 'block' becomes irrelevant. Each page has as many records in it as will fit; there are no inter-record gaps (as there used to be on MPE/V when the block size was not a multiple of one sector). In fact, records can even straddle pages -- if your file's records are 1,000 bytes long, then

* the first 4,096-byte page will have 4 full records and the first 96 bytes of the fifth record;
* the second 4,096-byte page will have the last 904 (1,000-96) bytes of the fifth record, the next 3 full records, and the next 192 bytes of the ninth record;
* and so on.

There's **never** any space wasted in a page (except, of course, in the allocated-but-not-written portion of the last extent) --

not because of bad blocking factors and not even because of records with odd record length. If you build an ASCII file with 1-byte records, exactly 4,096 of them will fit into each 4,096-byte page.

A curious thing, incidentally, is the lengths to which MPE/XL must go to make this efficient, reasonable, straightforward system compatible with MPE/V's baroque and inefficient mechanisms. If you read an odd-record-length file MR NOBUF, MPE/XL will actually **insert padding bytes** at the end of each record to be compatible with MPE/V; when do you an MPE/XL :STORE;TRANSPORT of a file whose blocks (in MPE/V) wouldn't be multiples of 256 bytes, MPE/XL will also insert padding at the end of each block to correspond to MPE/V's inefficient end-of-block padding.

The blocking factor of a file is, like the maximum number of extents of a file, specifiable but largely ignored. It's relevant only for compatibility, for transporting files to MPE/V machines, and for NOBUF file accesses, in which a program written for MPE/V would expect to get data in units of MPE/V blocks.

OTHER DISC SPACE CONSIDERATIONS

So, if MPE/XL can build large files without wasting space and do file blocking more efficiently and trim wasted space without changing file limits, one question remains: **Why does it use so much disc space?**

There is one philosophical explanation (it's somebody or other's Law, but I forgot whose): "*The amount of disc space required will increase until it meets and exceeds the amount of disc space available*". This is actually not just a facetious statement; as disc space use algorithms become more efficient and disc space becomes more plentiful, people will take advantage of this by building more and more files that are larger and larger. You'll get more bang for a buck's worth of disc space, but eventually you will exhaust it all the same.

There are, however, a few more pragmatic explanations:

* The operating system uses much more disc space than on MPE/V. The groups MPEXL.SYS and PUB.SYS use 570,000 sectors (150 megabytes!) on our 925/LX -- PUB.SYS on our MICRO/3000 uses only 100,000 sectors.

* Code -- programs and SLs -- uses a lot more space than it did before (this is actually a big part of the reason why the operating system uses more disc space).

  Why is this the case? Well, remember that all this "Reduced Instruction Set" means that it takes several RISC instructions to do the job of one Classic instruction. Thus, a program of 10,000 16-bit Classic instructions might be replaced by one of 50,000 32-bit RISC instructions -- a ten-fold increase.

  This is true of Native Mode code **and** of OCTCOMPed code. Compatibility Mode code still takes the same amount of space as it did under MPE/V.

* Although trimming files is **possible**, to the best of my knowledge, few things in MPE do it routinely. Compatibility mode USLs, it seems, are pretty substantial culprits (using far more space than they would if trimmed), and other files should probably be periodically trimmed, too.

Thus, our recommendations for saving disc space would be:

* **Purge old unused files.** This #1 space-saving feature from MPE/V days is still as important as ever on MPE/XL (and will probably be for a long time to come). Discs inevitably get filled up with junk, data that the owner no longer uses, no longer wants, and has probably already forgotten about; not only does it waste disc space, but it also makes your full backups take more time and more tapes. If you periodically archive and then purge all the files (except, say, IMAGE datasets) that haven't been accessed in 120 days, you will save **a lot** of a disc space with minimal user complaints.

* Trim files (e.g. using MPEX's %ALTFILE @.@.@;XLTRIM) periodically. As I mentioned before, trimming **seems** to be safe for all files in the system.

* Remember that native mode and OCTCOMPed program files are now **big** disc space hogs -- multiple unneeded copies of programs (which used to be rather harmless on MPE/V) may now substantially contribute to your disc space problems.

MAPPED FILES

Mapped files have been heralded (and correctly so) as a powerful and valuable new feature of MPE/XL. It has been discussed in a number of places, including chapter 11 of HP's "Accessing Files Programmer's Guide", and also, coincidentally, chapter 11 of SRN, Inc.'s excellent "Beyond RISC!" book. (I heartily recommend "Beyond RISC!" to anybody who's at all interested in Spectrums -- call SRN at (206) 935-3100).

At the RISC of beating a dead horse, I'd like to go over some of the key points of mapped files in this paper, too.

First of all, a "Mapped File" is actually not a type of file but rather a type of file access. Almost any file can be opened as a mapped file; once your program opens a file with the mapping option, **it will be able to access the file as if it were an array in its own data area.** Instead of accessing a file using FREAD and FWRITE (or the equivalent language constructs, such as PASCAL's READLN or WRITELN), you'll be able to access the data of the file **just as you'd access any array** (or record structure).

MPE/XL will, behind your back, realize that this isn't a "normal" array but is rather a mapped file; whenever you access a piece of this array, MPE/XL will, if necessary, go out to disc to get the appropriate data. (This is actually true for your stack, data segments, etc., as well, but it's especially important for mapped files.)

(NOT VERY IMPORTANT NOTE: Actually, the file system opens **all** files with mapped access for its own internal purposes; however, when I talk about "mapped file access", I refer to file access that is mapped from the user's point of view.)

Let's look at what might be the perfect application for mapped files -- keeping a large array of data that must survive from one execution of a program to another.

Say that you have a large number of payroll codes (numbered, say, from 0 to 99), each of which has various attributes (such as code name, pay scale, tax identifier, etc.) that your program must know about. Your program has to look payroll codes up in this file and extract the relevant data.

Without mapped files, here's what your program might look like (in PASCAL):

```
TYPE PAYROLL_CODE_REC = CRUNCHED RECORD
                        CODE_NAME: PACKED ARRAY [1..20] OF CHAR;
                        PAY_SCALE: INTEGER;
                        TAX_ID: INTEGER;
                         ...
                        END;
VAR PC_REC: PAYROLL_CODE_REC;
 ...
FNUM:=FOPEN (DATAFILE, 1 (* old file *));
 ...
FREADDIR (FNUM, PC_REC, SIZEOF(PC_REC), PCODE);
IF PCODE.TAX_ID=... THEN
   ...
```

With mapped files, you'd say:

```
TYPE PAYROLL_CODE_REC = CRUNCHED RECORD
                        CODE_NAME: PACKED ARRAY [1..20] OF CHAR;
                        PAY_SCALE: INTEGER;
                        TAX_ID: INTEGER;
                         ...
                        END;
     PAYROLL_CODE_REC_ARRAY = ARRAY [0..99] OF PAYROLL_CODE_REC;
VAR PC_FILE_PTR: ^PAYROLL_CODE_REC_ARRAY;
 ...
DOMAIN:=1 (* old file *);
HPFOPEN (FNUM, STATUS, 2, FILENAME, 3, DOMAIN, 18, PC_FILE_PTR);
 ...
IF PC_FILE_PTR^[PCODE].TAX_ID=... THEN
   ...
```

Instead of doing an FREADDIR (using PASCAL's READDIR statement), we directly access the file **as if it were an array.** The HPFOPEN call (more about its unusual calling sequence later) indicates that the file is to be opened "mapped" and that the pointer PC_FILE_PTR is set to point to its data; then, whenever we refer to PC_FILE_PTR^, we get access to the entire file as an array of records.

Why would we want to use mapped files? One reason is **convenience** -- in a situation like this one, it's easier (and makes more sense in light of the logic of the program) to view the file as an array rather than as a file. Instead of having to do a READDIR every time we want to get a record, we just access the record directly.

Another reason is **performance.** Avoiding the extra READDIRs not only makes the program smaller and cleaner, but also saves the CPU time that would otherwise be taken by each READ, READDIR, WRITE, or WRITEDIR. Each file system intrinsic (which are ultimately called READ, READDIR, WRITE, and WRITEDIR, and all the similar constructs in the other languages) has to do a lot of work finding control blocks, checking file types, etc., even before a disc I/O is actually done. This can take many thousands of instructions, amounting to up to a millisecond per call (or more). Access to a mapped file can take as little as one instruction -- one memory access.

As we will discuss later, mapped file access actually has some performance **penalties**, too, especially when we're doing sequential accesses to files that are not likely to be already in memory. It is actually quite possible with mapped file access to lose much more on disc I/O increases than you would gain on CPU time savings. However, if you're accessing files that are already likely to be in memory -- which often includes many heavily-accessed files -- mapped I/O can give you very large performance gains (again, more about this later).

Beyond convenience and optimization, I think that there are many more very interesting things that mapped files can let us do -- things that have rarely been contemplated in the past precisely because they were so difficult to do in the past. There is one idea that I have along these lines; I've never tried it in a production program, but I feel that it could very well prove quite useful.

One of the things that mapped files can give us is **shared variables.** By this I don't mean "global variables" that are shared among all the procedures in a program, but rather **variables that are shared among multiple programs and processes.**

For example, let's say that you have a program that runs in a job stream. The program might run for a long time, and you may want to check on its progress -- see which phase of processing it's in, what was the last record it processed, and so on.

With mapped files, you can do the following:

* Keep some crucial variables -- the current processing phase, the current record being processed, etc. -- in fields of a record structure. (This is a bit more complicated than having them all be separate variables, but not much.)

* Have the record structure be associated with a mapped file by HPFOPENing the file (with shared access) and using the pointer that HPFOPEN returns as a pointer to the record structure.

* Have another program that you can run online that will open the mapped file and print its contents for you.

Whenever the background program modifies one of the fields of this "mapped-file-resident" data structure, the field will be automatically updated in the file (even though this almost certainly require, on the average, far less than one disc I/O for each field modification). Then, the online program can at any time look at the contents of the file and tell you what's going on; and, if the batch program aborts, you'll be able to see where it was in its processing when it aborted (since the data is saved in the permanent mapped file).

This would also be an excellent tool if you'd like to write a debugger for some interpreter program that you have. As long as the interpreter keeps all its control variables in a "mapped-file-resident" area, then a debugger program (running in the same session or in a different one) can look at these variables and figure out exactly what the interpreter is doing. It can even change the variables, for instance setting some debugging flag, changing the value of a user variable, or whatever; and, if all the important data is actually kept in this file, it would permit "dump analysis" in case the program aborts, and even interruptions and restarts (since the entire state of the program would be automatically saved).

Another possible application is to have a program periodically (e.g. for every record that it processes) check a mapped-file-resident variable and terminate cleanly if it is set. Then, if we want to terminate all the processes running this program, we just set the variable, and all of them will stop. (Something like this could be done before with message files and soft interrupts, but it would require one record to be written to the message file for each process accessing it.)

Of course, this could all have been done before mapped files -- instead of accessing the mapped-file-resident variables directly we could just do FREADs or FWRITEs to read or write the appropriate record from the file. However, this would have been prohibitively expensive and clumsy -- imagine that you had to do an intrinsic call every time you wanted to access a particular variable; it would badly slow things down and make your program much more complicated. As I said, all of the above are relatively untested ideas, but I feel that much can be gained by doing something along those lines.

The really sad thing about mapped files -- something that I think is likely to drastically reduce their utility -- is that they can only be accessed from PASCAL/XL, C/XL, and SPLash!. FORTRAN/XL and COBOL/XL programs **cannot** access mapped files, not because of any file system limitation, but because those languages do not support pointers. In FORTRAN and COBOL, all variables are preallocated when you run the program or enter a procedure; to use mapped files, you have to be able to assign a particular address to a variable.

Actually, if you **really** wanted to use mapped files from FORTRAN or COBOL, you could write a PASCAL, C, or SPLash! procedure that lets you access pointers; however, this would most likely cancel any convenience advantages that mapped files can give you.

A few other notes about mapped files -- they're all documented in various places, but they're worth repeating:

* There are two ways of opening a file for mapped access -- "long mapped" and "short mapped". Long-mapped access lets you access any size file but requires you to use long (64-bit) pointers; in PASCAL, they have to be declared as $EXTNADDR$. Short-mapped access only lets you access a file of at most 4 megabytes; furthermore, you may have no more than 6 megabytes' worth of short-mapped files open for each process. On the other hand, short-mapped access lets you use 32-bit pointers, which are faster to operate with than the 64-bit ones.

* Because of the restriction on short-mapped file size and the fact that you can't open a file short-mapped if it's already opened by somebody else without mapping, your general-purpose programs (e.g. copiers, editors, etc.) should probably open files **long-mapped** rather than **short-mapped** -- it seems to be a more versatile, less restrictive access method.

* You can **not** access RIO, CIRcular, or MSG files as mapped files; you can access variable-record-length and KSAM files, but you'll see their internal structure (i.e. what you'd see if you read a variable file NOBUF or a KSAM file with COPY access) rather than their normal appearance.

  This may not seem to be such a big problem, and if often isn't; however, I've found that one of the most useful features of the MPE file system is its ability to subtitute one type of file for another -- for instance, give a message file to a program that expects input from a standard file, or a variable-record-length file to a program that expects input from a fixed-record-length file. This interchangeability will be **lost** for files that you open with mapped access.

* Remember that writing to a mapped file only writes the data; it does **not** increment the EOF. Even if you write data that ends up in record 1000 of the file, if the EOF is 200 it will stay 200. You have to do an FPOINT to the right record and then an FCONTROL mode 6 (as documented by the Accessing Files manual and by the Beyond RISC! book) to set the EOF to the right place.

  An interesting aspect of this is that the data that you write beyond the EOF will actually be written there and will remain readable the next time you open the file for mapped access. However, it will **not** be readable when you open the file normally, and will almost certainly disappear if the file is copied, :STOREd/:RESTOREd, or %ALTFILE;XLTRIMmed.

  Thus, if you write to a mapped file and forget to adjust the EOF, your programs might very well keep working just fine -- until the file is next :STOREd/:RESTOREd or %ALTFILE;XLTRIM. You can get some truly bizarre bugs this way.

  Don't even dare think that this is a useful feature and try to exploit it (e.g. to have some place to put "hidden data" that will appear not to actually be there)! Imagine trying to maintain or manage a system on which seemingly empty files were actually chock-full of data.

* If you have processes share a mapped file, you may have to do appropriate locking to prevent problems (especially if you have more than one person writing to the file). For instance, you might write all your programs so that they FLOCK the file before making any updates to it; unfortunately, this will make your program more complicated -- after all, the whole point was to treat the file data just as if it were normal program variables. Furthermore, the very fact that it's so easy to modify one of these shared variables (just assign to it or pass it as an output parameter to a procedure) may make it easier for you to forget to put in an FLOCK in the right place.

HOW THE FILE SYSTEM DOES I/O

One rule that we learned under MPE/V is: **always do disc I/Os in as large chunks as possible.** If a file has 256-byte records, don't read it from (or write it to) disc one record at a time; read it ten records at a time, or, even better, thirty or sixty at a time.

The reason for this was, of course, that as the transfer count (the number of words of data read or written) on a particular disc I/O increases, the time to do the disc I/O increases **much more slowly**. Thus, it might take you 30 milliseconds to read 256 bytes, but 100 milliseconds to read 8192 bytes; if you were planning to read those 8192 bytes anyway (and weren't just going to read one 256-byte record), you could read them ten times faster by reading them in one 8192-byte chunk than by reading them in 32 256-byte chunks. Furthermore, you'd incur the CPU overhead (which can be pretty substantial) of only one FREAD call rather than of 32 FREAD calls.

On MPE/V, the file system would always do disc I/Os in units of one **block**. The default **blocking factor** (the number of records per block) was usually **not** well-chosen by the operating system; for instance, any file whose record size was 65 words or more would, by default, have a blocking factor of 1. This might have made sense in the earliest HP 3000s (on which memory was a **very** scarce resource), but not on 8-Megabyte series 70s, which tended to end up being quite disc I/O-bound.

Thus, on MPE/V the recommendation was to **raise the blocking factors** of MPE (and KSAM) files that you frequently accessed, especially serially; this could save you a large fraction of the file's I/Os (increasing a blocking factor from 1 to 10 could cut by 90% the number of I/Os needed to read the file).

When disc caching was introduced, this became somewhat less important, since the file system would pre-read from 16 to 96 sectors (4K bytes to 24K bytes) whenever you'd do a serial disc I/O; thus, even a file with a low blocking factor could be read with relatively few disc I/Os. However, it still paid to have the blocking factor be high, since going to cache was still more expensive than getting the record from the file system buffer (though not as expensive as going to disc).

Finally, beyond increasing the blocking factor, it was often a good idea to read or write the file NOBUF (so that each FREAD returned an entire block) or MR NOBUF (so that each FREAD returned several blocks). Reading a file NOBUF caused you to do the same number of disc I/Os (since the file system also read the file a block at a time); however, you would save the CPU overhead of all those FREADs (which could be quite a lot). Reading a file MR NOBUF was even better, since it let you do even fewer disc I/Os (though increasing the blocking factor to a high enough value and then using plain NOBUF or even normal access could accomplish the same purpose).

The trouble with reading files NOBUF or MR NOBUF is that your program had to do its own "deblocking", i.e. it had to, by itself, separate each record in the block from the next -- not a very difficult task, but not a trivially easy one, either.

To summarize (again, remember that this is on MPE/V), here are the ways you might read a file of 1024 256-byte records (depending on the file's blocking factor and access method):

Blocking factor	Type of access	# of disc I/Os	# of FREAD calls
1	Normal	1024	1024
4	Normal	256	1024
16	Normal	64	1024
16	NOBUF	64	64
16	MR NOBUF (reading 8192 bytes at a time)	32	32

OK, enough re-cap. What's new in MPE/XL?

Well, the good news is that **all file system disc I/O is now done in units of several 4,096-byte pages**, often 8 pages (32,768 bytes), though the number seems to vary rather unpredictably. That's a lot of data (probably close to the optimum from the disc drive's point of view, since at some point the beneficial effects of reading larger and larger chunks of data will peter out), and it will **substantially** decrease the amount of disc I/O that will be done. Of course, what makes this possible is all those megabytes of memory that you had to buy to make your Spectrum run; they allow HP to go straight after performance optimization without having to optimize memory usage as well (or so we hope).

This means that blocking factors are now quite irrelevant to file performance (just as they are, as we mentioned before, irrelevant to disc space usage). You may set them high or set them low, but the file transfer unit will not change.

The bad news is that each FREAD and FWRITE call still takes a good deal of time, about .25 or so milliseconds running stand-alone on my 925/LX. (This may not seem like much, but remember that not everybody gets to use a Spectrum as a personal computer! On heavily-loaded systems, the FREADs and FWRITEs will take even longer to execute, and will adversely impact other users' response times.)

What can we do about all this file system CPU overhead? Well, we could access the files NOBUF or MR NOBUF -- the MR NOBUF would now be needed not so much to decrease disc I/Os (which, at one I/O per 16,384 bytes, can't really be decreased much further) as to decrease the number of file system calls.

Alternatively, we could access these files as **mapped files**. Once we open the file, we could then access all the data in the file using simple memory accesses -- when a disc I/O is required, it'll be done for us by the memory manager, but **no** file system overhead will be required!

The only problem -- and this is a really big one -- is that, together with the substantial CPU time savings that mapped files give us, they can also **substantially increase the amount of disc I/O that is done**. While the file system accesses the disc several 4,096-byte pages at a time (my observations showed me that it usually accesses 8 pages, or 32,768 bytes, in one shot), the memory manager (and thus mapped files) accesses the disc only **4,096** bytes at a time. Thus, while we can totally eliminate file system CPU overhead by using mapped files, we could at the same time **quadruple** the amount of disc I/O that needs to be done!

Now, as it happens, this disc I/O increase only becomes an issue if the file is not already in memory; to the extent that it is in memory (and many parts of your most heavily used files will be), the disc I/O size is irrelevant because no disc I/O will be needed. However, if a file is entirely or largely not in memory, you could suffer a **very** serious performance penalty by using mapped files.

To revisit our little table of the ways you can read a file of 1024 256-byte records, this time on MPE/XL (we'll assume that the file's blocking factor is 32 -- it's quite irrelevant except for NOBUF access):

Type of access	Maximum # of disc I/Os	# of FREAD calls
Normal	8	1024

NOBUF	8	64
MR NOBUF (reading 16384 bytes at a time)	8	16
Mapped	64	0

(This assumes that the file system reads 8 4,096-byte pages at a time, something that experiments on MPE/XL version 1.2 seem to indicate.)

Of course, the actual number of disc I/Os will vary depending on how much of the file is in memory.

Stan Sieler of Allegro Consultants (co-author of Beyond RISC! and one of the foremost experts on MPE/XL and the RISC architecture) ran some experiments that showed that a mapped read of a file that was 100% non-memory-resident took **more than 3 times the elapsed time** (though only about half the CPU time) of a file-system read; a mapped read of a 100%-memory-resident file took **less than 1/9th the elapsed and CPU time** of a file-system read.

The moral of the story:

* Blocking factors are no longer relevant to performance.
* NOBUF and MR NOBUF can still be a good idea.
* Mapped file access is much faster for memory-resident files, much slower for non-memory-resident files.
* Finally (as Stan Sieler discusses in his paper, and as we'll discuss more in the "NM FILES VS. CM FILES" chapter), KSAM access can be **faster** from **Compatibility Mode** than from Native Mode -- it's faster still from OCTCOMPed code.

Oh, yes, one other thing: FOPEN calls are **much** faster on MPE/XL than they were on MPE/V (they typically take from 25 to about 100 milliseconds running stand-alone on our 925/LX, compared to about 300 to about 500 milliseconds on a stand-alone Micro/3000). This may not seem like much, but this can be very important for programs that open some files, do a few checks, and then terminate (e.g. logon UDC security programs). These programs can now take a lot less time than before.

CM FILES VS. NM FILES

Every so often, you'll hear people talk about "CM" (Compatibility Mode) files and "NM" (Native Mode) files. There are a few things that are worth saying about this distinction.

The first thing that might come to mind is that a CM file is somehow accessible only from CM and an NM file only from NM. This is not so; both kinds of files are equally accessible from both modes (and, of course, from OCTCOMPed code, too); in fact, the access is completely transparent -- nothing behaves any differently (at least externally) from one mode to the other.

The distinction between CM files and NM files is purely internal. CM files are those **for which the internal file system code is implemented in CM**. KSAM files, message files, circular files, and RIO files -- the code that handles these files has simply never been rewritten by HP in PASCAL/XL; whenever you access these files, MPE/XL will execute the CM code that pertains to these files, even if this requires **switching** from NM to CM. NM files, of course, are those whose internal code is implemented in NM -- they include all the "vanilla" files, including both fixed and variable-record length files and

IMAGE databases.

The main way in which this CM/NM difference manifests itself is in speed of file access. As we said, if you try to access a CM file from NM (or an NM file from CM), the system will have to switch into the other mode in order to execute the appropriate file system code.

In addition to the switch from, say, your NM program to the CM file handling procedures, the CM file handling procedures will then have to switch to the NM internal file system procedures to do the actual file I/O. All these switches can take a non-trivial amount of time; for instance, it took a CM program more than 2.5 times longer to read a circular file than an otherwise identical non-circular file; however, even with this, the CM program ran 20% faster than an NM program reading the same circular file! A bizarre incident indeed -- NM code running slower than CM code.

This would not be that much of a problem if the only files that were slower in NM were message files, circular files, and RIO files -- after all, how much are these rather esoteric file types used in production? Unfortunately, the same thing applies to **KSAM files**, which can indeed often be quite performance-critical. My tests (and Stan Sieler's as well) showed that KSAM file accesses from NM were over 10% slower than from CM and about 20% slower than from OCTCOMPed code.

This might very well mean that KSAM users **ought not migrate their programs to Native Mode** for now (presumably, HP will come out with an NM KSAM soon). It seems that converting to NM will slow your KSAM file accesses down by about 20% (compared to OCTCOMPed code -- if you're still in CM, you should probably be OCTCOMPing all your code); you'll have to balance this against whatever performance improvement you expect to get on your other, non-KSAM-file-access code.

HPFOPEN

A number of the new features of the MPE/XL File System (including mapped file access and a few others that we'll talk more about shortly) have been implemented in the new HPFOPEN intrinsic, a successor to the old well-loved FOPEN.

Why a new intrinsic? Because the old FOPEN intrinsic, with its limited number of parameters (13 of them), just didn't have enough room for all the data that needed to be passed. By the time MPE/XL came around,

* 14 of the 16 foptions bits and 13 of the 16 aoptions bits were used up;
* The "device" parameter was actually used to pass no less than 4 different values (the device, the environment file, the tape density, and the ;VTERM parameter);
* The "forms message" parameter was used to pass 3 different values (the forms message, the tape label, and the KSAM file characteristics).

The MPE/V designers squeezed every last bit (almost) out of the FOPEN intrinsic because it was designed in an inherently non-expandable way; there was no way HP could have fit in the new parameters required to support the new features of the MPE/XL file system.

Much like the CREATEPROCESS intrinsic supplanted the CREATE intrinsic before it, HPFOPEN was designed to be a much more expandable (albeit, in some respects harder to use) version of FOPEN. The general calling sequence of HPFOPEN is

```
HPFOPEN (FNUM,          (* 32-bit integer, by reference *)
```

```
        STATUS,         (* 32-bit integer, by reference *)
        ITEMNUM1,       (* 32-bit integer, by value *)
        ITEM1,          (* by reference *)
         ...
        ITEMNUMn,       (* 32-bit integer, by value *)
        ITEMn);         (* by reference *)
```

HPFOPEN takes as input a list of item numbers and item values; it returns the file number and the error status. A typical call might be (naturally, we hope that you define constants for the item numbers -- 2, 3, 11, 12, 13, etc. -- and for the possible domain, access type, exclusive state, etc. values):

```
LOCK:=1;
DOMAIN:=1 (* old *);
ACCESS_TYPE:=4 (* input/output *);
FILENAME:='/MYFILE.MYGROUP.MYACCT/';
EXCLUSIVE:=3 (* shared *);
HPFOPEN (FNUM, STATUS, 2, FILENAME, 3, DOMAIN, 11, ACCESS_TYPE,
         12, LOCK, 13, EXCLUSIVE);
IF STATUS<>0 THEN
  PRINTFILEINFO (0);
```

The same call with the FOPEN intrinsic would be:

```
FILENAME:='MYFILE.MYGROUP.MYACCT ';
FNUM:=FOPEN (FILENAME, 1, OCTAL('344'));
IF CCODE<>2 (* condition code equal *) THEN
  PRINTFILEINFO (0);
```

As you see, the HPFOPEN intrinsic call is actually rather more verbose than the FOPEN intrinsic call, and may be argued to be harder to write, especially since you actually have to declare the variables DOMAIN, LOCK, ACCESS_TYPE, and EXCLUSIVE (since they, like all HPFOPEN item values, must be passed by reference). On the other hand, it does keep you from having to remember what all the positional parameters are (quick -- which FOPEN parameter is the file code?). More importantly, HPFOPEN lets you do things that FOPEN won't:

* Open a file for mapped access (items number 18 and 21).

* Open a file given the entire right-hand side of a file equation (item number 52). This way, you don't have to worry about all the other items or any "magic numbers" -- just say something like:

  ```
  FILEEQ:='%MYFILE.MYGROUP.MYACCT,OLD;ACC=INOUT;SHR;LOCK%';
  HPFOPEN (FNUM, STATUS, 52, FILEEQ);
  ```

  This can be a lot cleaner than the normal HPFOPEN approach, especially if all the parameters are constant (rather than having one be a variable, in which case you'd have to assemble the FILEEQ string using a STRWRITE or, in C, an *sprintf*). Unfortunately, not all file parameters are supported with this syntax -- exceptions include mapped files, user labels, disallowing file equations, and several others.

  Note that the value of FILEEQ started and ended with a "%"; it actually didn't matter what character it started and ended with as long as it was the same character. Rather than rely on terminators such as blank, semicolon, or whatever, HPFOPEN lets you specify your own string terminator as the first character in the string. In fact, almost all the HPFOPEN items that are strings (including the filename parameter itself) must be passed this way.

* Open a file as a new file and immediately save it as a permanent file (item number 3, value 4); this avoids the MPE/V headache of having an FOPEN succeed and then -- at the very end of the program -- having the FCLOSE fail because a file with this name already exists.

* Specify, when a file is opened, what disposition it is to be closed with (item number 50). In other words, if you open a file that you know should be purged when you're done with it, you can indicate this on the HPFOPEN call; then, even if the program aborts before FCLOSEing the file, the file will get deleted.

* And, a few other, less important things. In the future, though, new features that are added to the file system will be added through the HPFOPEN intrinsic, **not** through the already overloaded FOPEN, so this list will probably grow with time.

One other important point: how are you to interpret the STATUS variable that HPFOPEN returns to you? The manual tells you that the low-order 16 bits are always 143 (the code indicating that this is a File System error) and the high-order 16 bits are values from a 16-page table in the Intrinsics Manual. Naturally, rather than referring the user of your program to this manual, you really ought to format the message yourself, using the HPERRMSG intrinsic:

```
HPERRMSG (2, 0, 0, STATUS);
```

Easy enough to do, but I'll bet you that half the programs you run won't do this. If they don't and you have MPEX Version 2.2.11 or later, you can just say

```
%CALC WRITEMPEXLSTATUS (statusvalue)
```

and get the text of the error message. Of course, both the HPERRMSG call and the %CALC WRITEMPEXLSTATUS will work for all "MPE/XL-standard" 32-bit status results.

Finally, a few other interesting points:

* Whenever you create a file using HPFOPEN and do **not** specify a file limit, it will be built with a file limit of **8,388,607** records (not the measly little 1,023 that are the default on MPE/XL). This may be a good idea in theory, but in practice it means that you **must** close your files with disposition 16 (the "trimming" disposition) since otherwise your file will be allocated in chunks of 2,048 sectors each, so you could easily have some 2,048-sector files with one or two records.

* It has been said that HPFOPEN is not callable from CM programs. HPFOPEN is indeed an NM procedure, so it can't be called from CM programs as simply as, say, FOPEN can be; however, using HPSWTONMNAME (or HPLOADNMPROC and HPSWTONMPLABEL), one can relatively easily switch directly to HPFOPEN, passing to it whatever parameters you please. You don't have to write any native mode code to do this, nor do you have to put anything into any SLs or NLs -- it's a bit trickier than a direct call, but not by all that much.

  It is, however, true that there is no (documented) way of directly manipulating virtual pointers in CM, so mapped file access from CM is pretty much out.

ACCESSING FILES

While internal file structure and disc space considerations have changed dramatically with MPE/XL, the rules for accessing and sharing files have not (except for the addition of mapped files and the decrease in importance of NOBUF and MR NOBUF access). There's no reason to go into them in much detail now; I'll just go through a few of the key items that are worth repeating:

* If you want to have multiple writers appending to a (non-KSAM) file, use ;SHR;GMULTI;ACC=APPEND access. If you do this, you will **not** need to lock the file.

* If you want to have multiple writers doing any sort of writing other than appending, be sure that you **lock the file,** not just before the write but **before any read done in preparation for the write.** Thus, if a process needs to read a record, calculate a new value for a field in the record, and then write the record back, it must lock **before the read** and unlock **after the write**; otherwise, it risks the record being modified by somebody else between the read and the write and then having this other person's modifications wiped out.

* Attempts to lock a file (or a database) when you already have a file or database locked will normally fail with an FSERR 64. If you :PREP your program with ;CAP=MR, the attempt will succeed, but you stand the risk of causing a deadlock (which will still require a system reboot to resolve).

  If you must use ;CAP=MR, make sure that **all your multiple locks are acquired in the same order** -- if one program locks file A and then file B, **all** programs must lock those files in that order; otherwise, if any program locks file B and then file A, a deadlock becomes quite possible.

PROTECTING YOUR FILES AGAINST SYSTEM FAILURES

MPE/XL relies very heavily (even more so than MPE/V) on disc caching -- keeping as much disc data in memory as possible to speed up access to it. Unlike MPE/V, which, by default, only used this cache for reads and always did the writes to disc, MPE/XL caches writes, too; if you write a record to a file, **that record might not get written to disc for an indefinite amount of time.**

This has some substantial performance advantages (since a lot of disc I/O is avoided this way), but obviously puts your files very much at risk when the system crashes. KSAM files and IMAGE files seem to be protected by MPE/XL against loss of data at system failure time; unfortunately, **plain MPE files can very easily lose a lot of recently-written data when the system crashes.**

One of these forms of data loss could happen (and often did) on MPE/V -- when you're appending to an MPE file, the EOF pointer does not get updated on disc until the file is closed or a new extent is allocated. **Thus, the data that you append to an MPE file can get completely destroyed by a system failure because the EOF pointer did not get properly set.**

The solution to this problem, just as in MPE/V, is to do **FCONTROL mode 6s**, which post the EOF pointer to disc, as often as possible when you're appending to an MPE file. You might, for instance, do an FCONTROL mode 6 after every write, which will give you almost complete safety but also slow things down substantially; or, you could keep a counter, and do FCONTROL mode 6s every, say, five or ten records, thus minimizing your overhead while still protecting most of your data.

Unfortunately, on MPE/XL, there's more to it than this. **Any data that you write to a plain MPE file** -- even if you're not appending to it -- **might get lost in a system falure** because it may not get posted to disc until some time after you do the writes. On MPE/V, this possibility was limited to the data that was in your memory buffers (usually no more than about 2 blocks' worth of data); on MPE/XL, any data written since you opened the file could conceivably be lost.

For example, I ran a test with a file of 1000 records, each 256 bytes wide; I overwrote all 1000 records, kept the file opened, and re-booted the system. When the system came backup, only the first 768 of the new records were actually in the file; the remaining 232 records were still the **old** records from the time before I did my writes. (Note that 768 = 3*256; I'm not sure if there's any significance to this, but I suspect that there is.)

What can you do about this? Well, the simplest solution seems to be to call the **FSETMODE** intrinsic with the second parameter set to **2**. This means (according to the manual) "force your program to wait until the physical write operation is completed (the record is posted)", and this is what it seems to do. Of course, this causes each logical write to generate a physical I/O -- a great deal of overhead -- but it protects your data.

Alternatively, you can call FCONTROL mode 2 or FCONTROL mode 6 after each write or once every several writes (FCONTROL mode 2 is faster and may work well in cases where you're **not** appending and thus need not post the EOF); this is more work for you as a programmer than just calling FSETMODE, but it may be more efficient because you can do the FCONTROLs once every several records, thus decreasing the overhead of the extra disc I/O (but increasing the amount of data you may lose in case of a system failure).

A FEW WORDS ABOUT PERFORMANCE TESTS

The performance guidilines I've talked about (such as "FREADs of files that aren't in memory are faster than mapped file accesses" or "FCONTROLs mode 2 are faster than FCONTROLs mode 6") are strictly based on experience (my own or Stan Sieler's -- see his "MPE XL and Performance: Not Incompatible" paper). This experience may be inapplicable to your particular application, inapplicable to your version of the operating system, or perhaps just plain mistaken; I strongly encourage you to run your own performance tests to figure out how fast various file access methods work for **you**.

Unfortunately, file system performance measurement on MPE/XL is substantially more difficult than on MPE/V because of MPE/XL immense caching capabilities. It is almost guaranteed that, if you run a test twice in a row, you will get completely different results -- the first time your data was quite likely out on disc, but the second time it had just been read into memory and was therefore quite probably still in memory. Unlike MPE/V, there are no :STOPCACHE commands that you can use to make sure that this doesn't happen.

There are two key things you can do to **detect** possible bias due to a file's presence in memory and to **avoid** such bias:

* To find out how much of a file is in memory, do the following:
  - Go into DEBUG.
  - Enter "`MAP filename`" to open the file as mapped; this will output a line such as:

    ```
    1  MYFILE.MYGROUP.MYACCT  1234.0 Bytes = ...
    ```

  - The "1234.0" in the above line is the virtual memory address of the file -- type

```
=VAINFO(1234.0,"PAGES_IN_MEM")*#16,#
```

The value output will be the **number of sectors** of the file that are currently in memory (the #16 is there because there are 16 sectors per 4,096-byte page).

- Close the file by saying "UNMAP n", where n is the first number output by the MAP command (in this example, 1).

* Getting the file **out** of memory is a tougher proposition. My experience has been that the only way of doing this is to cause enough memory pressure to get the file's pages to be discarded (after they have, of course, been flushed to disc).

  One way of doing this is to read a very large file into memory; SL.PUB.SYS (22 megabytes on my system), NL.PUB.SYS (15 megabytes), and XL.PUB.SYS (6 megabytes) are good candidates. Just say:

  ```
  :FILE S=SL.PUB.SYS;LOCK
  :COPY *S,TESTFILE;YES
  :PURGE TESTFILE
  ```

  This will read all of SL.PUB.SYS into memory, which on my 925 LX is enough to flush any other files I may already have in memory. All you rich people out there with 128 megabytes of unused memory may need more than just this file, but you can always tell if the flushing succeeded by using DEBUG's VAINFO function discussed above -- if it tells you that your file has only 0 pages in memory, you know that you've flushed it out.

Given these precautions, you should be able to do your own performance tests (on an otherwise idle system, of course). Beware, though -- at least one key test I know of yielded **completely** different results on MPE/XL 1.1 and 1.2 -- much of the file system's performance characteristics seem to be quite MPE-version-dependent.

ODDS AND ENDS

Finally, a few miscellaneous futures which couldn't fit in anywhere else:

* DEBUG/XL's MAP command makes the debugger a powerful **data file editor**, even more convenient than the old DISKED on MPE/V. (Anything would be more convenient than DISKED. Do you remember how, when you asked for it to display octal and ASCII on the same line, it would display 8 words of the data in octal and then the first 8 **bytes** of the data in ASCII, completely ignoring the last 8 bytes?)

  In DEBUG/XL, you can say

  ```
  MAP filename WRITEACCESS
  ```

  DEBUG will output for you the "file index number" (used to close the file with the UNMAP command), the filename, the file's virtual address, and the file size, e.g.

  ```
  1    MYFILE.MYGROUP.MYACCT    1237.0    Bytes = 7560
  ```

  (For this example, we're assuming that you're in CM debug, so the numbers are output in octal; in NM debug, the output -- and default input -- would be in hex.) You can then display and modify data with addresses 1237.0 through 1237.7557 -- all the bytes in the file; thus, you could say

```
DV 1237.200,10
```

to display the 10 (octal) 32-bit words starting with byte 200 (octal) of the file -- the MV command will let you modify data. Note that DV and MV expect byte addresses, not record numbers and offsets within records; you have to do the calculation yourself (for instance, if each file's records are #256 [decimal] bytes long, record #10 occupies bytes #2560 through #2815).

The MAP command also provides you with one of the few ways to easily see (and edit) a file's user labels (:FCOPY, for instance, doesn't let you display their contents, and neither does the :PRINT command).

When MAP gives you an address whose second word is not 0 (1237.1400, for instance, rather than 1237.0), this means that the file has user labels (in this case, %1400/%400 = 3 labels). User label 0 starts at 1237.0, user label 1 starts at 1237.400, user label 2 starts at 1237.1000, and data record 0 starts at 1237.1400. (User labels are always %400 = #256 bytes long.)

Thus, you can use DV and MV to modify the file's user labels; also, remember that data record 0 now starts at a byte address other than 0 (in the example, %1400) -- keep this in mind when calculating the byte address of a particular byte in a particular record. If your file's records are #80 (= %120) bytes long, then, say, byte 6 of record 4 will be at location 1237.(%1400+6*%120+4).

* If you do a :DISCFREE A (which shows you how many free space chunks of each size there are on your disc), beware! You'll often see several large free chunks on LDEV 1 even though you're running out of disc space (or at least of contiguous disc space).

  On MPE/XL (unlike MPE/V), transient space (analogous to MPE/V's virtual memory) is treated as free disc space; however, at least 17% of the system disc (or more if you configure it that way) is reserved for transient space. Thus, you could have a huge chunk of free space on your system disc and still have it completely unusable for new disc files because it's reserved for transient space.

  :DISCFREE B tells you how much space is reserved for transient space, so its output shouldn't be too confusing; however, :DISCFREE A's output can be quite misleading if you don't keep the transient space issue in mind.

  This should probably not be overwhelmingly important, since contiguous space is less important on MPE/XL than on MPE/V, and you should therefore run :DISCFREE B more often than :DISCFREE A; however, I got bit by this thing myself when I was doing research for this paper, so I decided to mention it.

CONCLUSION

The MPE/XL file system is different from the MPE/V file system in many respects but is also similar to it in many respects. This paper was largely dedicated to the differences (since they're more interesting), but there are very many similarities as well, largely dictated by the requirement of **complete** (well, almost complete) compatibility -- a requirement that HP rigidly enforced on itself, and, I must say, very much lived up to.

Many of the old and unpleasant limitations of the MPE/V file system have been lifted; a few remain in place (such as the 3-level directory structure and a few other, relatively minor, problems); a few new ones have probably been added, but the user community hasn't discovered them yet and probably won't for some time. (Who would have thought, in 1972, that people would be running into the 2,097,120-sector limit on file size?)

Performance and disc space are still potential problems, and will be for a long time to come -- as long as CPU power and disc storage cost money.

I would like to thank Jason Goertz, Bob Green, Randy Medd, David Merit, Ross Scroggs, and, especially, Steve Cooper and Stan Sieler for their reviewing the paper and for their many excellent comments and suggestions. I would also like to refer the interested reader to Stan Sieler's "MPE XL and Performance: Not Incompatible" paper, published in the SCRUG 89 Proceedings, and, of course, to the "Beyond RISC! book" from Software Research Northwest (by S. Cooper, J. Goertz, S. Levine, J. Mosher, S. Sieler, and J. Van Damme, edited by W. Holt).

A Closer Look at IMAGE Paths

by David Merit
Bradmark Computer Systems, Inc.
100 Oceangate Avenue, Suite 505
Long Beach, CA 90802 U.S.A.
(213) 432-7713

Introduction

Paths: we know 'em, we love 'em; because as we are all painfully aware, the presence or absence of a path can mean the difference between a chained read that takes several seconds and a serial read that takes several hours.

But the existence of a path for a field being searched on does not necessarily guarantee good performance. Paths, like other IMAGE facilities, can be misassigned, underutilized, and inadequately maintained. This article takes a closer look at IMAGE paths, and answers some common questions about their use, benefit, cost, optimization, and maintenance.

(References to IMAGE in this article also apply to TurboIMAGE but not TurboIMAGE/XL, unless otherwise noted.)

"How many paths should I have for a dataset?"

Although IMAGE permits up to 16 paths to be related to a master or detail dataset, the number of paths that should be assigned for a particular dataset is very subjective. It is therefore always surprising to find HP3000 shops that place restrictions on the maximum number of paths that may be configured, as well as published recommendations limiting the number of paths to, say, two or three. In reality, even a single path would be inadvisable for some datasets; for others, 16 paths would be quite acceptable.

Several critical factors are involved in determining the worthiness of each path, which makes it impossible to give a good blanket recommendation about how many should be assigned. Really, you should assign as many paths as you need and can afford.

A path permits access to a detail dataset by a given field by maintaining a logical relationship between the detail dataset and a master dataset. Thus, each path for a detail set permits chained access by a different field.

To illustrate some important concepts about paths and their use, let's look at part of an order processing database. The INVOICE-LINES detail set has two paths, to INVOICE-MASTER and PART MASTER, which permit chained access by INVOICE-NUMBER and PART-NUMBER; the INVOICE-HEADERS detail set has two paths, to INVOICE-MASTER and CUSTOMER-MASTER, permitting chained access by INVOICE-NUMBER and CUSTOMER-ID; and the INVOICE-PAYMENTS detail set has a path to INVOICE-MASTER, permitting chained access by INVOICE-NUMBER.

Now, although the five paths shown in this example affect three detail sets and three masters, only the detail sets benefit from their existence. Appropriately, the detail sets also suffer most of the overhead required to maintain the chains along the paths, although a somewhat smaller price is paid when accessing the master sets.

Before examining the costs associated with paths, let's take a closer look at what benefits may be gained from their existence.

"How much does a path benefit ?"

Master sets do not benefit from paths: a master set with no paths or 16 paths can be accessed only by its search field. Each detail set path, however, permits access by a different search field, while the absence of a path for the field on which you are searching means a serial read (unless you are using a non-IMAGE indexing scheme, such as KSAM, Bradmark's SUPERDEX package, or DISC's OMNIDEX package).

A chained read along a path will almost always outperform a serial read of a detail set, although there are circumstances in which a serial read would be as fast as, if not faster than, reading along a chain. Of significance here is the number of chains for the path; that is, the number of unique search field values that appear in the detail set. If there are few (and therefore long) chains, it could take as many disk I/Os to read a chain as to read the entire dataset serially. The reasons for this involve the detail set's blocking factor, disc caching, and chain efficiency.

Blocking factor. The dataset blocking factor represents the number of entries contained in each disk block. Because IMAGE always reads an entire block (rather than just an entry contained in the block) into its internal buffers, many entries may be retrieved by IMAGE with a single read.

For example, the INVOICE-LINES dataset, which has a blocking factor of 20 and contains 20,000 entries, could be serially read with 1,000 reads, as shown:

	20,000	entries
/	20	blocking factor
=	1,000	number of blocks in dataset

Disc caching. Now, even though the dataset contains 1,000 blocks, a serial read would normally not require 1,000 disk I/Os to accomplish, since disc caching can have a major impact on the number of disk I/Os that would be required. Assuming that the dataset block size is 1024 words and system disc caching is configured with a random fetch quantum of 96 sectors, it would take only 84 disk I/Os to serially read the dataset, as shown:

	1024	block size, in words
/	128	number of words in sector
=	8	number of sectors
	96	random fetch quantum, in sectors
/	8	block size, in sectors
=	12	number of blocks read per disk I/O
	1,000	number of blocks in dataset
/	12	number of blocks read per disk I/O
=	84	disk I/Os to serially read dataset

At the conservative rate of 30 disk I/Os per second, it would take just under three seconds to read the dataset serially.

Chain efficiency. Let's compare the above with the number of I/Os and amount of time it would take to read a PART-NUMBER chain containing 500 entries. Taking disc caching into account, this could be anywhere from 3 disk I/Os (less than one second) to 500 I/Os (17 seconds) to read the chain. Wow! Why such a difference between these figures?

The first statistics represent the most efficient chain possible: excellent entry locality and all entries contiguous in chained order. This means that in reading chained by PART-NUMBER, all the line items referencing a particular part number would follow one another physically in the dataset. In this case, all entries are contained in 25 contiguous, sequential blocks in the dataset.

The second statistics reflect a very inefficient chain with very poor entry locality: only one entry with the designated part number per block spread throughout the dataset, with chain members located both physically before and after each other. In this case, the entries are contained in 500 blocks at locations throughout the dataset.

The second condition is probably closer to reality, as will be illustrated later.

More on disc caching. Not only is the locality of entries on a chain very significant toward the speed of chained access, but so is the physical direction in which the chain points. Disc caching can work either in favor of or against the chained read, depending on the relative physical locations of the entries on the chain. Since disc caching caches in a forward direction only, it is more efficient to read a chain whose entries physically follow one another in the dataset than one which points physically backwards or alternately in both directions.

Effectively, this means that serially reading a dataset using DBGET mode 2 (forward) is faster than mode 3 (backward) because disc caching reduces the number of disk I/Os when reading forward; however, this does not necessarily mean that DBGET mode 5 (forward chained) is more efficient than DBGET mode 6 (backward chained), as entries on chains very often do not follow each other physically.

These examples illustrate that a chained read along a path does not always outperform a serial read, and that the performance in reading a chain is highly dependent on disc caching and chain efficiency. This suggests that some maintenance is required to optimize the efficiency of a path, as will be discussed later.

"Why should paths be not used to support batch ?"

Normally, in considering the worthiness of a path, those that are used at non-critical times or infrequently should be questioned. If a path is used only once a week to support end-of-week processing, it may be unnecessary. Even if it takes much longer to read a dataset serially, if sufficient time exists to perform this task, it may be preferable to potentially degrading online performance with the overhead of a path if there is enough time to perform the serial read.

A serial read of a dataset can also be accomplished considerably faster by use of MR NOBUF (multi-block, unbuffered) reads, instead of than DBGET mode 2 or 3. Many sites have found Robelle's QUERY-like SUPRTOOL product, which performs MR NOBUF reads, to be a solution for speeding up serial reads. Some use Robelle's SPEED DEMON or Running Mate's IOMATE packages, which include MR NOBUF procedures that replace DBGET and may be called from programs, for performing faster serial (and even chained) DBGETs.

So if you can comfortably afford additional paths to speed batch processing, go ahead and use them. But be sure to consider the negative performance ramifications and potential disk space utilization, which we'll discuss now.

"How much does a path degrade detail dataset performance?"

Generally, any performance degradation caused by a path is experienced whenever an entry is DBPUT into or DBDELETEd from a detail dataset. Since each detail entry must be linked into a chain on every path when added and unlinked from every path when deleted, the amount of time required to complete a DBPUT or DBDELETE is greatly influenced by the number of paths.

(Some are under the misconception that if a search field value is blank or null, an entry is not chained on that path, but in this case, a null chain is created containing all these entries. In fact, some IMAGE users have been surprised to receive a "FULL CHAIN" error on performing a DBPUT with a null value, having already created a chain with the maximum of 65,535 entries on it, although TurboIMAGE alleviates this condition by supporting much longer chains.)

This amount of time required for a detail DBPUT or DBDELETE is determined by the number of disk I/Os, generally, three or four I/Os per path. This number could be higher for some paths, as well as particular entries due to various factors, including the efficiency of any sorted paths and the performance of related master sets.

If the related master is manual, the amount of time required to locate the chain head (equivalent to a DBGET mode 7) is endured for both DBPUT and DBDELETE. Master sets with low blocking factors and poor primary/secondary locality could be problems because IMAGE could take longer to locate the chain head. But even so, the difference in speed is usually negligible.

If the related master set is automatic and the chain head does not already exist, IMAGE internally performs a DBPUT to create it. If the automatic master has a clustering problem--a rare condition except with misused integer keys--the DBPUT could require many I/Os. When all detail entries are deleted, a DBDELETE of the chain head is internally performed.

If a detail path is sorted, it could take extra I/Os to DBPUT entries, depending on whether the entries are added in sorted (ascending) order, such as would be the case if the sort item is the current date and entries are added in chronological order. If entries are pre-sorted, there is no significant performance difference between a sorted path and a non-sorted path; otherwise, IMAGE may require substantially more time to add an entry, since it must read up the chain from the bottom to determine where to link each entry logically, and then update the pointers on the previous and next entry on the chain.

Another condition that could adversely affect the performance of a DBPUT is if an insufficient number of buffers are allocated for the database. IMAGE "previews" each DBPUT to make sure it can be completed successfully, checking such factors as whether space is available in the detail set, if the required chain heads exist in related masters and, if not--for automatic masters--if space exists to add a new master entry. In doing so, it fills its internal buffers with all the relevant data blocks before actually updating them and writing them back to disk; if sufficient buffers are not available, multiple buffer reading/writing operations must be performed, which can increase the number of I/Os significantly. Such a buffer supply crisis would not occur for a DBDELETE, since no preview is done.

Also when weighing the performance ramifications associated with DBPUTs and DBDELETEs against detail sets, be sure to consider the increase in time required for dataset maintenance. The performance of bulk DBPUTs (such as when loading the dataset via DICTDBL, DBLOAD, DBRECOV), DBDELETEs (such as when performing an extract/archive), and reorganization (such as with DBGENERAL or Adager) will be correspondingly worsened based on the number of paths.

Also remember that IMAGE does not permit the value of a "critical" field to be changed by DBUPDATE, so the existence of a path designates a field as a search field (and perhaps another as a sort field), and IMAGE will require a DBPUT and DBDELETE to change their values (not to mention any programming changes required).

Aside from the performance considerations associated with DBPUT and DBDELETE, a secondary potential performance problem is that the number of paths could affect the speed of serial reads, since the disk space utilized by the chain pointers (four words for each path) could cause the dataset to reside on more blocks and therefore require more disk I/Os to read. More on disk utilization in a moment.

"How much does a path degrade master dataset performance?"

As stated, paths have a minimal effect on master set performance--it is in details that the price is really paid.

Each entry in a master set contains for each path a discrete six-word area (five words for IMAGE) called the "chain head" area in which a count of the number of detail entries on the chain and a pointer to the first and last detail chain entry are maintained. When a master entry is added, these values are initialized to zero, and are updated whenever a DBPUT or DBDELETE (not a DBUPDATE) to a related detail dataset is performed.

Because the master entry is affected only as the result of activity in a related detail dataset, the price of the path is really assumed by the detail set. The number of paths related to a master set is therefore not terribly relevant to overall performance. There is little performance difference between a stand-alone master dataset (with no paths) and a master dataset with 16 paths. The speed of DBPUTs and DBDELETEs is effectively the same.

It could, however, take somewhat longer to read a master dataset serially with more paths because the disk space used by the chain head information could cause increase the number of dataset blocks and require more disk I/Os to read.

"How much disk space does a path require ?"

More relevant than the amount of disk space required by a path is the *additional* amount of disk required, since paths may require substantial disk space but no *additional* disk space.

Although a master path requires six words per entry (five for IMAGE) and a detail path requires four words per entry, figuring out the overall disk utilization for a dataset is not as simple as multiplying the number of words required by the path by the number of paths by the dataset capacity. Because IMAGE stores data in blocks that must be multiples of 128 words, there is almost always some unused space at the end of each block. This means that for some datasets, a path (or even multiple paths) may not require any more disk space than is currently used by the dataset, since sufficient unused space may already be available.

For example, a master dataset with one path, a data entry length of 104 words, and a block size of 1,024 words could accommodate two additional paths while requiring no additional disk space. Let's look at the math.

Currently, the master media entry length (data plus IMAGE structures) is 115 words:

	104	data entry length, in words
+	5	synonym chain area, in words (occurs in each master set)
+	6	path chain head area, in words
=	115	media entry length, in words

Based on the media entry length, IMAGE can fit eight entries into a disk block of 1,024 words, with 921 words utilized:

	1024	block length, in words
/	115	media entry length, in words
=	8	blocking factor (rounded down)
*	115	media entry length
+	1	bit map, in words (occurs at beginning of each block)
=	921	block utilization, in words

This results in 103 words per block unused:

	1024	block length, in words
-	921	block utilization, in words
=	103	residual space in block, in words

and therefore leaves enough room in each block to accommodate two additional paths per entry:

	6	path chain head area, in words
*	2	number of paths
*	8	blocking factor
=	96	disk space required for two master paths, in words

As shown, no additional disk space is required by the two new paths, and the residual space per block is almost eliminated:

	103	residual space in block, in words
-	96	disk space required for two paths, in words
=	7	residual space in block, in words

This, of course, does not mean that in other situations a path would not require a lot of disk space. Let's look at an example involving a detail dataset with two paths, a data entry length of 102 words, and a block size of 1,024 words.

Currently, the detail media entry length is 110 words:

	102	data entry length, in words
+	4	path 1 pointer area, in words
+	4	path 2 pointer area, in words
=	110	media entry length, in words

Based on the media entry length, IMAGE can fit nine entries into a disk block of 1,024 words, with 991 words utilized:

	1024	block length, in words
/	110	media entry length, in words
=	9	blocking factor (rounded down)
*	110	media entry length
+	1	bit map, in words (occurs at beginning of each block)
=	991	block utilization, in words

This results in 33 words per block unused:

	1024	block length, in words
-	991	block utilization, in words
=	33	residual space in block, in words

The residual space of 33 words does not permit any additional paths to be added, since another path would require 36 words per block:

	4	path pointer area, in words
*	9	blocking factor
=	36	disk space required for a detail path, in words

Therefore, the addition of a new path would increase the media entry length to 114 words and change the dataset characteristics as follows:

	1024	block length, in words
/	114	media entry length, in words
=	8	blocking factor (rounded down)
*	114	media entry length
+	1	bit map, in words (occurs at beginning of each block)
=	913	block utilization, in words

which results in 111 words per block unused:

	1024	block length, in words
-	913	block utilization, in words
=	111	residual space in block, in words

As you can see, the more residual space per block, the less efficient the blocking, and therefore the more blocks--and more disk space--used by the dataset. With a capacity of 100,000, the detail set in this example would require 44,448 sectors with two paths and 50,000 sectors with three paths, the additional path costing 5,552 sectors.

Also of interest here is that the resulting residual space of 111 words after adding the path can accommodate three additional paths, since they would require only 96 words in total.

These examples are extreme cases of very high and very low blocking efficiency using datasets that are blocked for the maximum blocking factors rather than disc savings. However, they are representative of many IMAGE datasets and illustrate the difficulty in determining the impact of a path on disc utilization without going through the math.

"Why are some paths in the same dataset faster to access than others ?"

Normally, access by one path in a detail dataset is more efficient than by its other paths. Additionally, some chains along a path are more efficient to access than other chains. To understand why, let's look at three different path scenarios and their performance implications.

Note that these examples assume that no or few deletions have been performed against the detail sets--a very important factor, as we'll soon see.

Naturally efficient paths. Chains along some paths are naturally inclined to have good data locality. For example, the INVOICE-NUMBER path into INVOICE-LINES (*Figure 1*) is naturally quite efficient because the line items for each invoice are added at the same time, and, with a single writer, would be contiguous; with multiple writers, they would be interspersed with line items for other invoices, but still physically close to one another.

Figure 1

INVOICE-NUMBER	PART-NUMBER
100	AAAA
101	BBBB
100	BBBB
100	CCCC
101	HHHH
100	DDDD
101	IIII
102	EEEE
101	JJJJ
100	EEEE
101	AAAA
101	DDDD
102	NNNN
102	OOOO
100	FFFF
101	KKKK
102	FFFF
101	LLLL
102	IIII
100	GGGG
103	FFFF
102	PPPP
102	DDDD
101	MMMM
103	MMMM
103	OOOO
101	CCCC
103	DDDD
102	BBBB
103	CCCC

When accessing these entries, both the dataset blocking factor and disc caching are significant in determining the performance of chained reads. With an average of 15 line items per invoice and a blocking factor of 20, all the line items could likely be accessed with a single read--even if fragmented due to multiple writers, the line items for a single invoice would likely be close enough that disc caching would take up the slack.

Naturally inefficient paths. Chains along other paths naturally have poor locality. For example, the PART-NUMBER path into the same INVOICE-LINES *(Figure 2)* detail set is naturally inefficient because a given part number occurs only once for each invoice, and some part numbers may appear infrequently.

Accessing one of these chains will likely require one disk I/O per entry, since neither the blocking factor nor disc caching would be sufficient to span the distance between entries. In fact, in this case, disc caching would impose an undesirable overhead since it would have to determine that the desired datablock was not already in cache before performing another disk I/O to read the block containing the next entry on the chain.

Single-entry-chain paths. Chains along some paths are very short, containing an average of only one detail entry, as shown in *Figure 2*. For example, the INVOICE-NUMBER path into the INVOICE-PAYMENTS dataset contains an average of one entry per chain, since only a deposit or multiple payments against a single invoice would result in multiple entries--both exceptional conditions.

Figure 2

INVOICE-MASTER

105
100
104
109
101
108
103
106
102
107

INVOICE-PAYMENT

100	...
101	...
102	...
103	...
104	...
105	...
106	...
107	...
108	...
109	...

Accessing such a single-entry chains will always require one disk I/O because that's all there is to read. In this case, the blocking factor is unimportant and, as before, disc caching is undesirable because it creates useless overhead.

In this same category of very short chains exists a very strange creature: the "master-detail"--a detail dataset that contains entries, such as customers or parts, that a purist would insist belong in a master dataset. After all, a master dataset is supposed to contain "entities" and a detail set "transactions", but because a master dataset can be searched by only one field, many database designers have chosen to create master-oriented detail sets with multiple paths.

"How do I maintain optimum performance for a path ?"

The three paths just shown and their performance implications are characteristic of most paths in IMAGE databases. Each type has its own recommended prescription for maintaining optimum performance.

As we've seen, the critical factors in the performance of chained access is minimizing disk I/Os, which are affected by the dataset blocking factor, disc caching configuration, and entry locality--all of which are adjustable.

Blocking factor. A dataset's blocking factor is based on the set's block size, which is dependent on the database's BLOCKMAX. While an in-depth discussion of database blocking is beyond the scope of this article, of concern are datasets with blocking factors that could be increased while retaining the same dataset block size or by increasing the dataset block size to equal the database BLOCKMAX. The latter case could require additional disk space.

A dataset reblock is a one-time operation that may be accomplished by database restructuring utilities (such as DBGENERAL and Adager) or by a DBUNLOAD/DBLOAD. You should read up on the subject before reblocking any sets.

Disc caching. IMAGE benefits only by the random fetch quantum setting of system disc caching--even when performing sequential (serial) reads. The higher the random fetch quantum, the more dataset blocks *forward* in the dataset are read from disk with each I/O.

As we've seen, disc caching sometimes imposes an undesirable overhead in chained access and in backward serial reads. Unfortunately, disc caching cannot be enabled or disabled for selected files; however, it can be turned on or off for particular disk drives. You may find it beneficial to move datasets which are impaired by disc caching to a disk drive for which caching is disabled. Other benefits may be gained by dynamically enabling and disabling disc caching or adjusting the random fetch quantum based on changes in IMAGE processing.

You should refer to additional sources for more information about disc caching, as well as disk controller caching.

Chain efficiency. This is most important element in maintaining optimum path efficiency, since it improves entry locality in a dataset, and even the highest possible blocking factor and random fetch quantum will not help if chain entries are scattered throughout the dataset. It is also the most difficult, since it requires potentially time-consuming maintenance on a regular basis. And it has two formidable enemies which can slowly fragment even the most efficient chains: time, and DBDELETE.

Time is the enemy for chains whose entries are created over time, such as along the PART-NUMBER path into INVOICE-LINES and CUSTOMER-ID path into INVOICE-HEADERS. Entries on both these paths must be forced into a beneficial order, which will deteriorate over time as new entries are added.

DBDELETE is the enemy for almost all paths, because in a detail set, IMAGE first reuses all locations made available by deletions before appending new entries. If the deletions were scattered throughout the dataset, the new entries that are DBPUT and take their locations will be also. What's more, IMAGE reuses the deleted locations in reverse order, meaning that if the deletions were performed using a forward serial read, new entries will be added in the opposite direction, and chain entries will thereby physically precede one another in the set. As we've seen, this counteracts the beneficial effects of disc caching.

Fortunately, there is a simple solution for these problems: flush the deleted entries and physically reorder the live entries to optimize entry locality for a path. This detail set "reorganization" or "repack" can be accomplished by utilities such as DBGENERAL and Adager, and by a chained DBUNLOAD/DBLOAD.

Before performing such a reorganization, though, it is desirable to get a true picture of how efficient a path is before jumping in and trying to improve its efficiency. Some tools that report on the efficiency of chains are DBGENERAL, Robelle's HOWMESSY, and the contributed DBLOADNG. All report a composite inefficiency statistic for each path, which reflects the number of block reads that were required versus the number of reads that would be necessary if entry locality were optimal for that path. The higher that statistic, the more can be gained from maintenance. Unfortunately, these utilities do not take the benefits (or penalties) of disc caching into account, but still provide a useful representation of a path's efficiency.

Now, don't be alarmed to see one path that is very efficient and others that are grossly inefficient: this is normal. After all, entry locality plays the major role in determining the efficiency of a path, and it is usually impossible to situate entries in a detail set in such a way to benefit all paths. For example, physically ordering the entries in the INVOICE-LINES detail set based on the chains for PART-NUMBER will speed access by part number, but will slow access by INVOICE-NUMBER by destroying entry locality for that path.

Therefore, in reorganizing a detail set any one path may be optimizing but all paths from that detail set must be considered, since what will benefit one path may harm another. Optimizing the PART-NUMBER path to INVOICE-LINES would probably not be a good idea because it would severely degrade retrieval by INVOICE-NUMBER, which is a more frequent and time-critical operation.

It is generally best to optimize the most frequently accessed path--unless the most frequently accessed path has chains containing only one or very few entries. There is little or no benefit to reordering entries for chains that contain a single entry--one disk I/O will still be required, regardless. Chains that contain two entries would then require one instead of two disk I/Os, which is hardly enough benefit to justify *not* optimizing another path, let alone the resources required for the reorganization.

Most detail set reorganization utilities enable you to specify the path by which to reorganize -- the path that will benefit most. However, a chained DBUNLOAD will always use the set's primary path; if using this method, make sure the appropriate path is assigned as primary.

If multiple paths are accessed with the same frequency, reorganize by the one with the longest chains, since the overall benefit will be greatest. Also, look for cases in which optimizing one path will benefit other paths at the same time. For example, let's say the INVOICE-HEADERS dataset also contained a path for ACCOUNT-REP, which would permit all the invoice headers for accounts handled by a particular account rep to be retrieved. Since the same account reps would be assigned to the same customers, optimizing the CUSTOMER-ID path would, as a byproduct, optimize the ACCOUNT-REP path somewhat. (Incidentally, the ACCOUNT-REP path was not configured because the small number of account reps would result in few chains, and a serial read would be more efficient than a chained read.)

Let's look at how detail set reorganizations could optimize the three paths described earlier, with the reorganization based on each path.

Efficiency along the "naturally efficient" INVOICE-NUMBER path could be improved, since this would cause entries for each chain to be contiguous--even if heavy deletions were done. This is the recommended path by which to reorganize (see *Figure 3*).

Figure 3

INVOICE-NUMBER	PART-NUMBER
100	AAAA
100	BBBB
100	CCCC
100	DDDD
100	EEEE
100	FFFF
100	GGGG
101	BBBB
101	HHHH
101	IIII
101	JJJJ
101	AAAA
101	DDDD
101	KKKK
101	LLLL
101	MMMM
101	CCCC
102	EEEE
102	NNNN
102	OOOO
102	FFFF
102	IIII
102	PPPP
102	DDDD
102	BBBB
103	FFFF
103	MMMM
103	OOOO
103	DDDD
103	CCCC

Efficiency along the "naturally inefficient" PART-NUMBER path could be improved more significantly by reordering entries along this path, since making all the entries for each part number contiguous would dramatically reduce I/Os and cause caching to be effective. Note, however, that doing so would seriously degrade performance on the INVOICE-NUMBER path because entry locality on its chains would be compromised (see *Figure 4*).

Figure 4

INVOICE-NUMBER	PART-NUMBER
100	AAAA
101	AAAA
100	BBBB
101	BBBB
102	BBBB
100	CCCC
101	CCCC
103	CCCC
100	DDDD
101	DDDD
102	DDDD
103	DDDD
100	EEEE
102	EEEE
100	FFFF
102	FFFF
103	FFFF
100	GGGG
101	HHHH
101	IIII
102	IIII
101	JJJJ
101	KKKK
101	LLLL
101	MMMM
103	MMMM
102	NNNN
102	OOOO
103	OOOO
102	PPPP

A reorganization of the path with "single-entry" chains would be of no benefit, since one disk I/O is spent on each detail entry regardless. An exception to this is if detail entries are reordered in the same sequence as the entries in the related master and are processed one-for-one (i.e. read a master, read its detail, read a master, read its detail, etc.), since this makes efficient use of caching.

USING USAGE TRENDS

TO MANAGE YOUR HP3000

COMPUTER SYSTEM

by

Brian Duncombe
c/o
Strategic Systems Inc.
11050 5th Avenue N.E., Suite 101
Seattle, WA 98125
(206) 362-2231

INTRODUCTION

Many of us at one time or another have been involved in some form of planning effort as part of managing an HP3000 shop. When this happens, we quickly realize that there is a wealth of information potentially available to us regarding the way we use our HP3000 computer systems. The problem in many cases is either that we are overwhelmed by the volume of that data or that we have not been collecting the data.

With the proper information available to us, we can determine where we have been in the past, where we are currently and with some gazing into the crystal ball, where we are going. Armed with this knowledge, we should be in a much better position to make superior decisions about hardware acquisitions, new applications, staffing of the operations area, software development, application tuning and general performance improvement.

This paper presents many of the potential sources of usage data, the considerations involved in collecting that data and the uses that the data can be put to if we chose to collect it.

The types of usage information that we can collect include:
- utilization levels of the various system resources
- availability of disc freespace
- system resource utilizations by program file
- system resource utilizations by user logon
- file accessing frequency

FREQUENCY OF SAMPLES

No matter what data we ultimately chose to collect and use as an aid to managing our computer systems, we are always faced with the question of how much detail to include in the data as well as how frequently the data sampling should occur. If we gather too much detail, we may use large volumes of mass storage to save the data and we may also complicate the extraction of meaningful information by overwhelming ourselves with details. If we sample infrequently, we save mass storage space but run the risk of gathering information that is statistically insignificant or misleading. On the other hand, if we gather our samples very frequently, we again use more mass storage space as well as potentially causing significant system loading just to collect the usage data. We must each come to our own conclusions about how we weigh the tradeoffs between these opposing factors of mass storage space and data collection overhead versus the granularity and detail of the information we require.

SYSTEM RESOURCE UTILIZATION

When we think of trends and utilizations, many of us focus on the system as a single entity. Our natural inclination is to view the system as a whole and deal with planning on a global level. In doing so, we normally focus on the "Three Bears of Performance" (CPU, main memory and disc accessing). It seems logical that if we are to collect information that will help us manage our systems, we should be collecting data that reflects how we are using these basic building blocks of the system. As we delve into the subject, we also discover that at the global system level, there are a number of other indicators of usage trends such as system idle time, the number of sessions logged on, job counts and job backlogs.

CPU ACTIVITY

Many people equate the computer system with the CPU. When we talk about hardware upgrades, we seem to automatically focus on the CPU even though it is not always the limiting factor in the performance of our system. If we could be sure that the CPU was being utilized at or near it's capacity and were able to measure this, we might feel more confident in recommending a change in the CPU hardware. If on the other hand, we can see that the CPU capacity is not a problem, we may avert the embarrassment of upgrading the CPU only to find out that it made little improvement to the system throughput as a whole.

The sources of data regarding the utilization of the CPU are varied and range from those that require manual data collection to those that are almost totally automated. The following are some of the sources of this CPU utilization information:

<u>The REPORT command.</u> By recording the connect and CPU used statistics from this command, we can get a rough idea of which accounts and groups are using the CPU the most. By plotting the sums of the net changes on a regular basis, we can derive patterns on a daily, weekly or monthly basis. Although this is not an accurate reflection of the CPU busy state, it can be used as a reliable "index" of CPU usage.

<u>Visual observation of the current instruction register (CIR).</u> This option although very crude is available to those of us who have a CPU that displays this register visually (Series II,III,/64,/68,/70). It involves periodically observing and recording how "bright or solid" the lights are being displayed. This subjective reading can be made at a number of fixed times during the day and recorded in some form of log book.

<u>Contributed monitoring tools.</u> A number of software tools have been written and contributed in some form to various contributed libraries or informally through Hewlett Packard. These tools include such programs as SOO, SOO5, POO, SURVEYOR and SCOUT. In order to collect data that reflects CPU utilization, you must either run the program continuously in batch mode and direct the output to a printer or disc file or else periodically run the program in an interactive mode and record the data in some form of log book. Most of these tools will break the CPU usage into a number of components reflecting such things as user program busy, system overhead and paused waiting for I/O.

<u>Supported monitoring tools.</u> Hewlett Packard as well as a number of third party software vendors sell and support software tools that will periodically sample the system and report on the CPU activity broken into components reflecting the various possible states of the CPU resource. Most of these tools will also run in batch mode and store the data into a logfile so that it can be programmatically reported on with a minimum of manual intervention. Some of these tools provide the data reduction and reporting of the information in tabular and/or graphical output formats.

SYSTEM RESOURCE UTILIZATION - cont'd

The following chart (Figure 1) shows the results of collecting CPU activity information over a period of 24 hours and then plotting the data at hourly intervals.

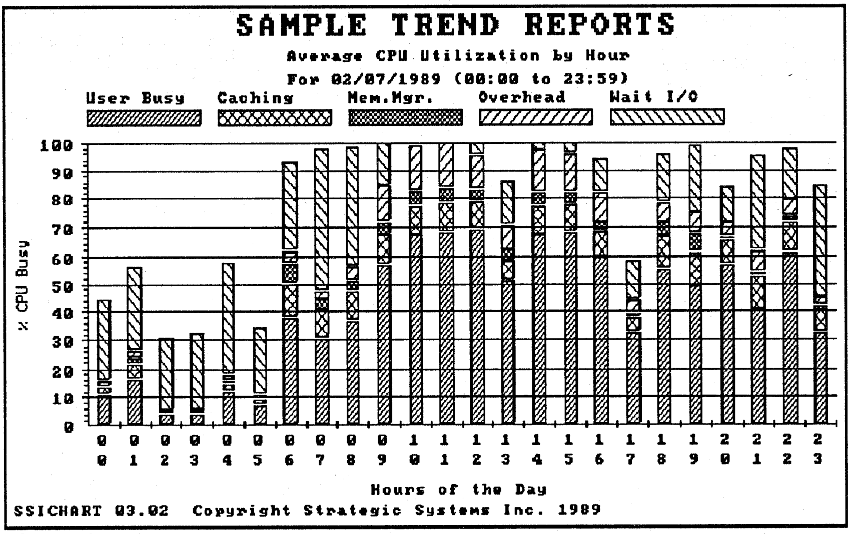


Figure 1

From this chart, we can make the following observations:

During the day shift (7:00AM-4:00PM) there is very little system idle time. The CPU has virtually no excess capacity during this period and in fact, it is fairly safe to assume that the CPU is actually overloaded much of the time.

There is a break in the CPU loading at lunch time and then again at approximately 6:00PM. This is consistent with the pattern of business. Everyone takes off for lunch and there is a period between the on-line workload of the daytime and the start of nightly batch processing. This shows some opportunity to shift workload to these time periods using techniques such as staggering the lunch break, running batch during the lunch break and beginning batch workload earlier in the afternoon if possible.

From midnight until 5:00AM we can see that very little activity is on the system. At the same time, there is a high percentage of time when the CPU is "Waiting for I/O". This might suggest that this time period has very little utilization or that tape handling is required and that the operator is not very quick to respond to requests. We can also see that there is no time during the period when the CPU is actually completely idle.

Figure 2 shows the data collected over a period of several months and plotted at monthly intervals.

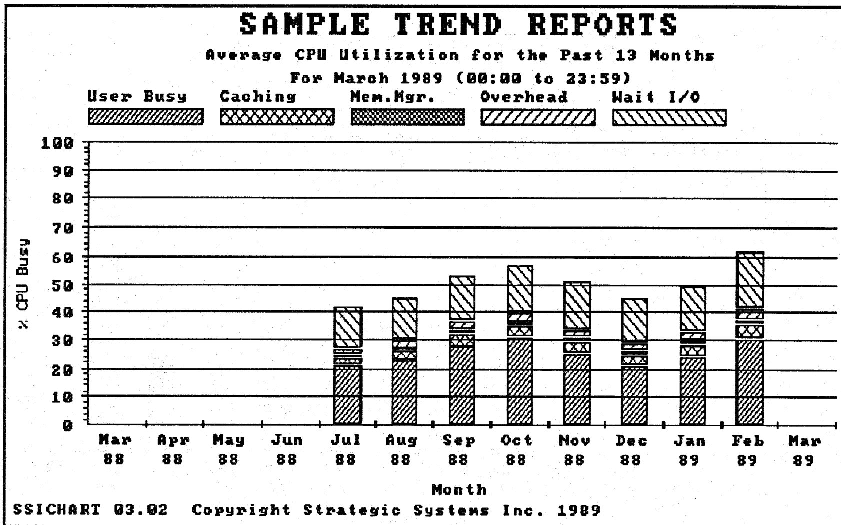


Figure 2

This chart gives us an indication of how the CPU resource has been utilized over the past months and provides some insight into what the utilization might be in the next few months. From this chart, the following observations can be made:

There was a slump in the usage during the Nov'88 and Dec'88 periods but the utilization levels have begun to increase again.

The general trend is an upward one that is approximately 10% every 3 months. Using a simple linear projection and assuming that the danger point is an average utilization level of about 80%, we might conclude that the system will be saturated in about Aug'89. This may very well be misleading since the critical time window is probably the day shift on-line activity and if we restricted the data reported to the 7:00AM through 5:00PM time period, the problem would probably appear much more critical.

Figure 3 shows essentially the same information as Figure 2 except that the time window has been restricted to 7:00AM-5:00PM.

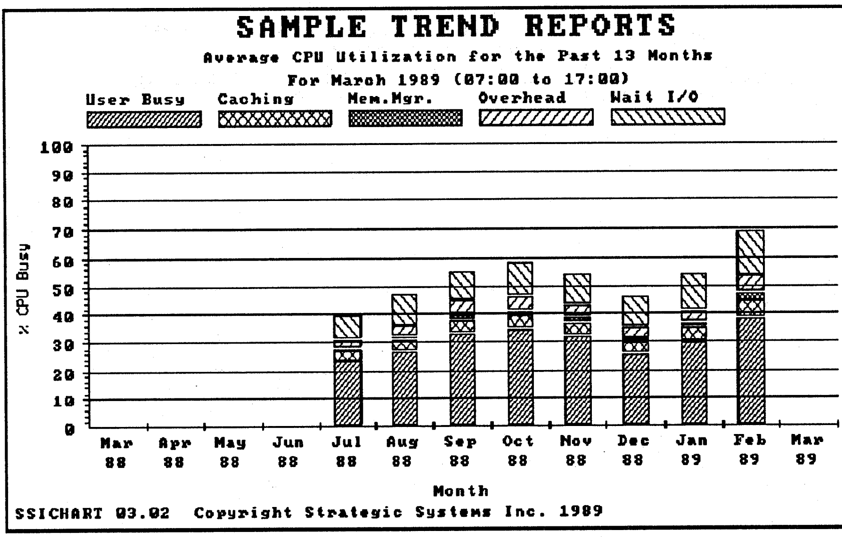


Figure 3

This chart gives us some idea of the differences between average trends for a full 24 hour time window and those restricted to the daytime hours. Comparing this chart to the previous one (Figure 2) we can see:

For the months of Jul'88, Aug'88 Sep'88 and Oct'88 the difference between the full 24 hour window and the 10 hour daytime window is not very significant.

As we compare the last four months, we can see that the averages are somewhat higher for the daytime window and that this becomes more pronounced in the last 2 or 3 months.

We might conclude that the trend to higher loading is more pronounced in the daytime hours and that the "critical" trend is that of CPU utilization increasing at a significantly greater rate than the 24 hour time window would suggest.

MAIN MEMORY UTILIZATION

This is one of the more difficult characteristics to measure. Historically, main memory has proven to be the most frustrating of the possible hardware upgrades. Often, there is no perceived improvement in the performance of the system after adding main memory resources. By collecting and tracking indicators of main memory shortages, you are in a good position to make a much better decision regarding this system resource.

The indicators of main memory shortage include CPU time spent executing memory manager code as well as the statistics regarding the frequency with which the memory manager cycles through main memory looking for space (clock cycle rate). For those of us using the MPE/XL operating system, the presence of high paging rates (in the thousands /second) combined with high clock cycle rates (greater than 0.02/second) is a fairly sure sign of main memory shortages. The sources of this information include the following:

Contributed monitoring tools. Several of the contributed tools will show you the percentage of the CPU activity devoted to the memory manager function (garbage collection and memory allocation). These can be sampled on a regular basis and plotted to provide trends.

Supported monitoring tools. Several of the tools sold by Hewlett Packard and other third party vendors provide both CPU activity in memory management related functions as well as Clock Cycle rates and Paging rates (MPE/XL only).

Figure 4 shows the data collected over a 24 hour period reflecting the average clock cycle rates at hourly intervals.

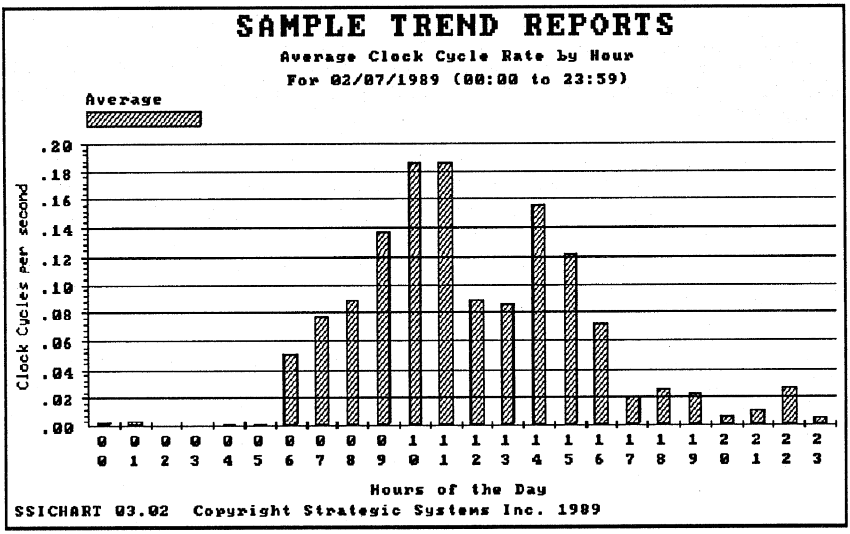


Figure 4

This chart gives us an insight into how the activities of the memory manager vary throughout a 24 hour period. Reviewing this information shows us the following:

The peaks and valleys are much more pronounced in this chart than they were in the CPU chart for the same time period (Figure 1).

The time period from 10:00-11:00 as well as the hour at 14:00 show that there is likely a shortage of main memory at these times although it is only the onset and not very severe. This could in fact be caused by MPE caching which under some circumstances tends to accentuate memory problems.

Outside of the interactive daily processing (6:00AM-4:00PM), there is virtually no memory manager activity.

Figure 5 presents the data collected over a period of months reflecting the average clock cycle rates at monthly intervals.

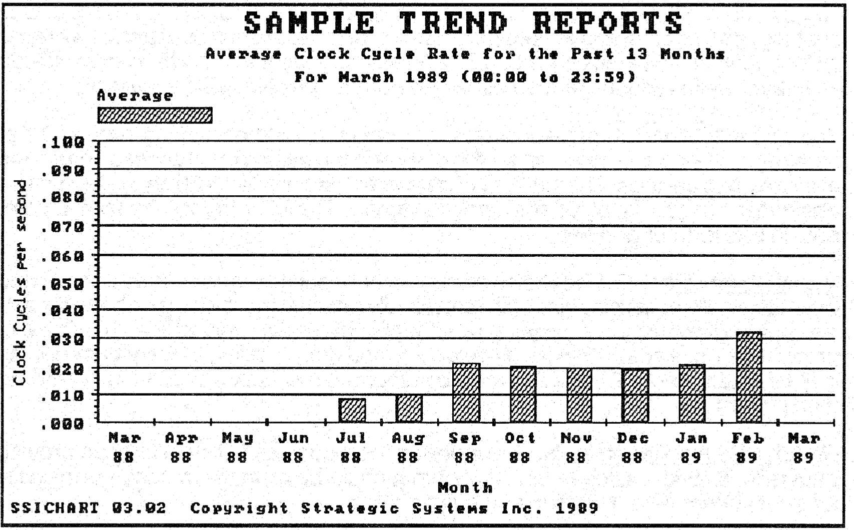


Figure 5

Looking at this data, we see a pattern similar to that for CPU activity collected for the same period (Figure 2). Again, we can conclude the following:

> The general trend is an upward one although there is a slump during the Oct'88 and Nov'88 time periods.
>
> If we accept 0.1 cycles/second as a threshold for the onset of main memory shortages when the data is averaged over long periods of time then this system is not near the point of having main memory shortages.

DISC SUBSYSTEM ACTIVITY

Perhaps one of the most important resources affecting the performance of an HP3000 computer system is that of the disc subsystem. Prior to the advent of disc caching, disc bottlenecks accounted for much of the slowness of systems. With the introduction of MPE disc caching, CPU and memory resources have been sacrificed in an attempt to reduce the requirement to perform physical disc accessing.

The key indicators of disc subsystem utilization include accessing rate (I/O per second), average queue length for the drive and the balancing of access across the available disc devices. The sources of data regarding disc activity range from simple observation to sophisticated software monitoring. The following sources provide this data in one form or another:

Visual Observation. For those disc drives that have access lights to indicate when the drive is actually accessing data, you can simply observe the flashing of the lights and attempt to guess at the accessing rate. While this seems very crude, in fact it can provide a simple scale (0=idle, 1=slow, 2=medium, 3=busy, 4=very busy) which is all you really need. Of course it must be recorded manually at regular intervals and then plotted.

Contributed monitoring tools. Unsupported tools such as SURVEYOR can provide data reflecting accessing rates. This output could be routed to a printer or disc file for subsequent data reduction and presentation.

Supported monitoring tools. Both Hewlett Packard and third party vendors sell software tools that not only are capable of collecting data about disc accessing but in most cases also provide a means to log the data into disc files for further processing. Some of these tools go as far as reducing and presenting this data summarized either in tabular or graphical format.

The following chart (Figure 6) shows a 24 hour period during which disc accessing data was collected.

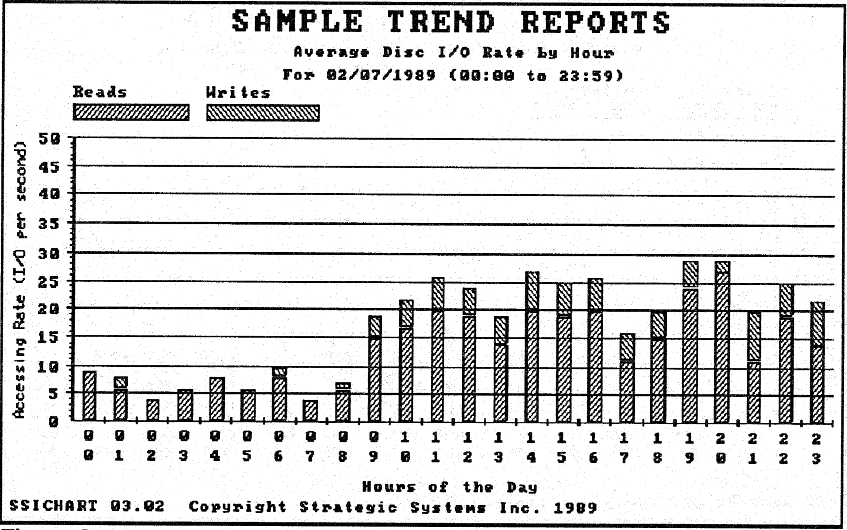


Figure 6

Reviewing this data, we can make the following observations:

The trends show the same patterns as those for the CPU information over the same time period.

Assuming that a single disc drive is capable of sustaining an accessing rate in the range of 20 accesses/second, the average accessing rates shown here would be a source of concern if there were only one or two disc drives.

The majority of physical accessing is in the form of reads rather than writes which would suggest that disc caching (both MPE and controller) should be expected to be quite effective.

Figure 7 shows disc accessing data that has been collected over a period of months with the data averaged at monthly intervals.

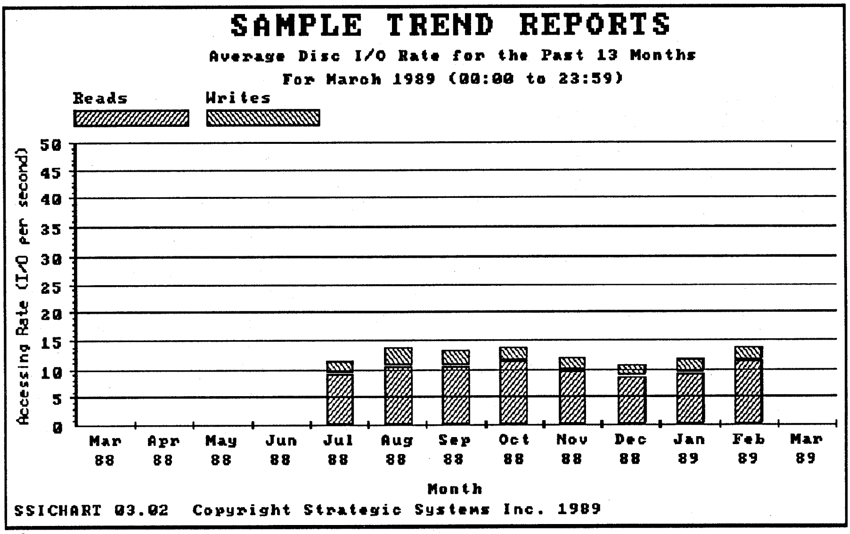


Figure 7

Looking at this data, we can make the following observations:

Like the CPU and memory manager charts, this one shows that there was a slump in activity during the fall of the year but that the trend is generally upward.

The extended time period indicates that an overwhelming proportion of the accessing is reads rather than writes.

RESPONSE TIME

For most HP3000 users, response time is the critical key indicator of how well the system is operating. If response times remain fairly good, then the management of the system will usually proceed in an orderly manner. If response times are poor, the system will usually be difficult to manage and you will likely spend a great deal of your time "fire fighting".

Measuring response time is a very difficult task. Most users consider response time to be the delay between the time they press the enter/return key and the time that they receive their "complete" response back. Any external software tools that attempt to measure response time must face the decision as to how they identify a transaction. They can simply consider all terminal read completions to be transactions in which case they have a simple task but will undoubtedly count more transactions than the users since many "logical transactions" consist of multiple terminal reads. On the other hand, it is possible to watch all activity on a terminal port and try to guess from the handshaking and delays between characters returned to the terminal and new data sent from the terminal when human input has been involved in the process although this imposes heavier loading on the system to do the processing and data gathering.

The following sources of response time can all be used to provide either a "real" response time or some relative index of response time:

Timing responses manually. This can be a very intimidating experience for the terminal operator. In my experience, it is very difficult to get reliable information this way especially using a stopwatch. If you develop the habit of periodically walking through the terminal user area at randomly selected times, you can covertly observe response and count to yourself to get a reasonable approximation of the response delays. As you might conclude, this is not a very acceptable way of regularly collecting data.

Supported software tools. As mentioned above, it is very difficult for generalized software to determine what a transaction is. These types of tools do however give repeatable results and for this reason can be used to develop "indexes" of response time that can be compared relative to one another. For most applications, this is probably the most practical way to collect response time data that can be used for trending purposes. Of course, you must be very careful not to present the response times collected as "real" response times since they will not likely resemble those measured by watching the terminal operators.

Instrumenting the application code. Where practical, this is the best method of measuring response time. It does however require planning during the design phase or at the very least going into existing application code to insert the collection routines.

The following chart provides an indication of the type of information that can be presented from response data collected over a period of time using generalized external collection tools. In this chart, the averaged data is presented at hourly intervals.

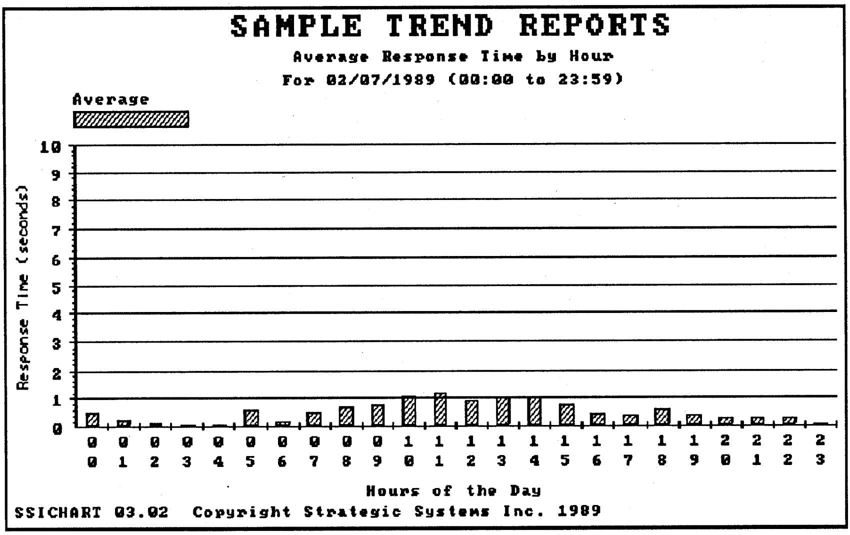


Figure 8

Looking at Figure 8, we can observe the following:

Again, the data shown here tracks the same patterns as those data reported for the other components of system utilization during the same time period.

On average, our response "index" shows a response time of less than one second most of the time. As mentioned previously, this probably bears little relationship to the response times that you would measure if you were to physically observe the transactions at the terminals.

Like the chart in Figure 8, the following chart shows response time data collected using generalized collection software.

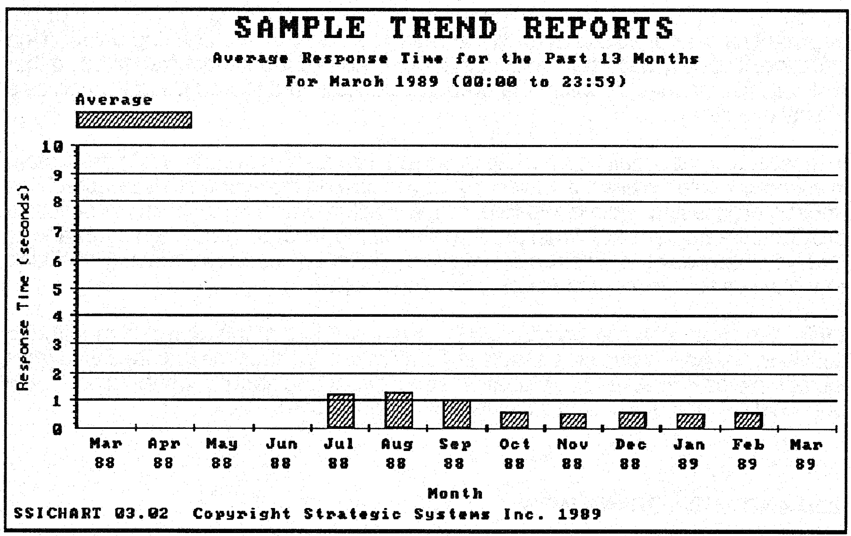


Figure 9

Looking at this chart, we can observe the following:

On average, our response "index" shows a response time of less than one second. Again, this is probably not the "real" response time experienced by the users.

While the other measures of system utilization show an increasing trend for the last few months, this chart shows that response time appears to be remaining somewhat constant.

SYSTEM IDLE TIME

One of the more common problems of system management is the case where suddenly one morning the overnight batch work has not completed and the online applications cannot be started until the batch work is finished. Looking back on the situation, it usually becomes evident that this is not a sudden occurrence but rather a slowly approaching problem that went unnoticed until it reared it's ugly head as a critical event.

In most cases, we require some time during our processing cycle in which the system is essentially idle. While this may seem to be allowing the system to be wasted ("an unused processing cycle is lost forever"), it is actually a form of insurance policy. It provides breathing room when an unexpected job must be run, when a job aborts and we must recover and re-run or simply as a buffer against the increasing workload that buys us time to increase our processing capacity.

While this "idle" time can be measured as a byproduct of the software monitoring activities mentioned earlier, a simple scheme of posting the time that the last nightly job completes is a perfectly adequate method of warning about impending problems provided we look at the data collected on a regular basis.

JOB AND SESSION COUNTS

A statistic that is simple to collect and yet may prove to be an excellent key indicator of how our system is being loaded is that of batch job counts and concurrent numbers of sessions. By tracking these gross indicators of workload, we can often see patterns of usage developing in their early stages.

The most simple measurements of job and session count can be made by recording the current job and session numbers as part of our regular weekly backup (and of course when we re-boot the system).

The more enlightening data regarding the number of concurrent jobs and sessions is somewhat more difficult to capture. In most cases, it requires that a "showjob status" be issued at regular intervals during the day and that the statistics be recorded in some form of log. Some of the third-party software tools log this information as part of their logging functions.

DISC FREESPACE

A resource that maintains a low profile in the trending and capacity planning arenas is that of available disc space. The performance experts will tell you in passing that it is good to have at least 15 or 20 percent of your disc drive capacity available as freespace so that the disc management components of MPE can work efficiently. While this is good and correct advice, it ignores the more severe performance problem of not being able to run an application at all simple due to not having sufficient disc space. Just as the available "idle" time on the system tends to creep away over time, so too does the available disc space while at the same time, most of us experience growing needs for space to accommodate the increase in business volume.

A simple plan that has a very good chance of warning us about disappearing disc space involves running FREE5 on a regular basis and logging the statistics in a summarized form so that they can be plotted over time to see trends. The combination of "total freespace" and "largest free area" should be sufficient to highlight disc usage trends. Armed with this information, we can plan for general cleanup activities as well as the installation of new disc drives before they are actually required and thus avoid a crisis situation.

PROGRAM FILE UTILIZATION

In attempting to manage an HP3000 computer system, a vital input to any planning activity is that of knowing which programs are being run the most and whether a particular program is being used more or less than it once was. The knowledge that a particular program or set of programs is being used more combined with an understanding of how that program affects the loading on the system can allow us to detect dangerous trends earlier than they might show up as part of the general utilization trends for the system. As an example, if a number of people begin to use HP3000 based graphics software on the system, they could impose a load on the system that increases much more rapidly than the general trends would indicate. This might mean that instead of reaching a decision point for action 6 months from now, you will be dragged into the decision much sooner.

This type of information also allows us to focus our tuning efforts on the programs that make up the major portion of the processing load. By doing this, we can maximize the effects of our work because we are improving busy programs instead of seldom run programs and any improvements that we make will be leveraged every time the program runs or that part of the code executes.

The following pie chart (Figure 10) shows the "top ten" accounts from which program files were run.

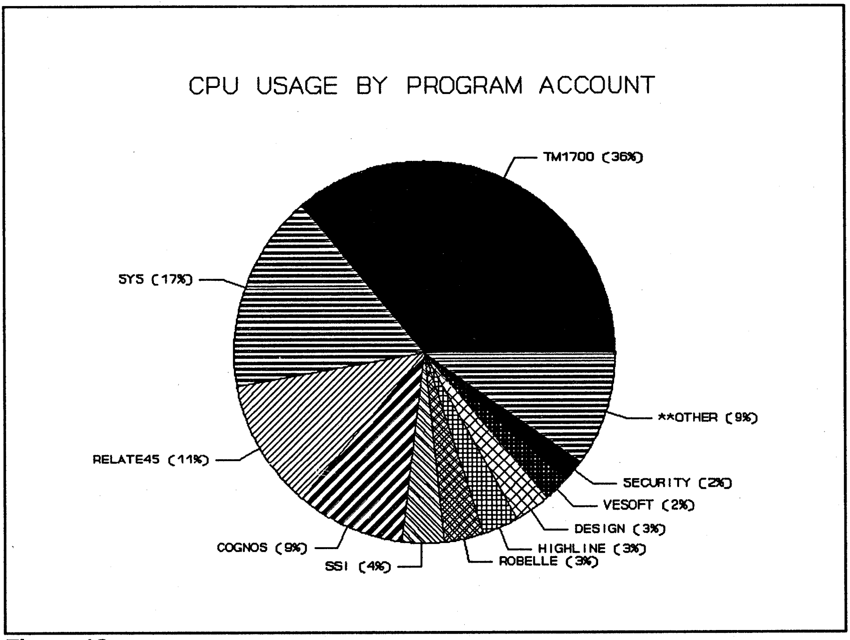


Figure 10

The following chart (Figure 11) shows the busiest program files on the system.

Figure 11

--------PROGRAM NAME--------			CPU
RIBS RBSEXEC	TM1700	851551.6	
RELATE	PUB	RELATE45	314200.2
CI			242201.8
STORE	PUB	SYS	231081.9
BUILDER	PUB	RELATE45	213456.1
QTP CURRENT	COGNOS	203430.6	
SIN15X1	PUB	TM1700	179015.7
QUIZ CURRENT	COGNOS	140545.7	
MPEX PUB	VESOFT	91286.2	
ACC6080	RBSEXEC	TM1700	90115.2
ACC5010	RBSEXEC	TM1700	85497.2
COBOLII	PUB	SYS	85144.4
STREAMX	PUB	SECURITY	75423.7
QEDIT	PUB	ROBELLE	66992.7
HOWMESSY	PUB	ROBELLE	65690.1
TOOLSET	PUB	SYS	61522.1
SEGSYM	PUB	SYS	60332.5
DBCLEAN	RBSEXEC	TM1700	59177.3
SEGPROC	PUB	SYS	55089.1
FCOPY	PUB	SYS	46077.1
ACMLOGBO	COPYLOG	SYS	44819.8
ACC4020	RBSEXEC	TM1700	42111.6
EDITOR	PUB	SYS	37083.2
ADAGER	PUB	REGO	35726.6
QUIZP	CURRENT	COGNOS	25202.7
QDESIGN	CURRENT	COGNOS	23827.4
QUICK	CURRENT	COGNOS	23172.9

Data reflecting the usage of particular programs or groups of programs can be collected from a number of sources including the following:

<u>MPE process termination logfile records.</u> Although the process termination logfile records indeed log the termination of a process, it is necessary to match these records to the most recent "file close" record to get the name of the program that was involved in the process. In spite of this minor programming problem, this is an excellent source of high-level information regarding the use of the various program files on the system. Because there is only one record logged for the entire duration of the life of the process, this data is of little value in determining the usage of programs with any time scale finer than 24 hours.

<u>Contributed software monitoring tools.</u> These tools sometimes offer a crude method for collecting data that reflects the usage of individual programs and their utilization of the key resources of the system. In general, the data must be collected manually and recorded for later analysis.

<u>Supported software monitoring tools.</u> Since most of these tools include a supported data logging facility, the analysis of program usage data can be almost completely automated. Many of the vendors include data reduction and reporting software as part of their products thus making the job even easier.

LOGON USER UTILIZATION

As well as knowing which programs are being used the most on your system, the knowledge of which users are using the system is an important trend to understand.

By detecting a user or group of users who are changing their system usage habits, you can often gain an early insight into what the future trends of the system usage will be. When you combine this knowledge of how they are changing their usage with an awareness for the magnitude of the system resource loading they normally cause, you can make a more informed estimate of how at least part of your user loading will change.

Once you can highlight changes in user activity, you can often investigate this to determine the reason that it is happening. In some cases, it might be that they have off-loaded a function from the HP3000 to PCs in which case you may be faced with requests for more or more powerful PCs, support for those PCs or a loss of auditability of the system that they are circumventing. In other cases, it might be dissatisfaction with the service provided in which case you can address the concerns before it is too late. In still other cases, it might be the first sign of a shift in the business plan of your company which someone neglected to provide as input to your planning activities.

Output from the REPORT command supplied within MPE can give you a general indication of user logon times by account and group although the actual user is not available. In many cases, this is enough to spot general shifts in the usage of the system. The following is a brief example of this output report:

```
:report @.@
ACCOUNT          FILESPACE-SECTORS     CPU-SECONDS       CONNECT-MINUTES
   /GROUP          COUNT     LIMIT    COUNT     LIMIT    COUNT     LIMIT
SYS                67127        **   346970        **   279258        **
   /CACHE             90        **        0        **        0        **
   /CHARSETS           0        **        0        **        0        **
   /DOC                0        **        0        **        0        **
   /FIGURE             0        **        0        **        0        **
   /HIST            4986        **    73278        **        0        **
   /OPERATOR          39        **     3070        **   125746        **
   /PUB            61985        **    46037        **   152852        **
   /UDC               27        **        0        **        0        **
   /USL                0        **        0        **        0        **


:report xxx.@
ACCOUNT          FILESPACE-SECTORS     CPU-SECONDS       CONNECT-MINUTES
   /GROUP          COUNT     LIMIT    COUNT     LIMIT    COUNT     LIMIT
GOFASTER           13276        **     2033        **      503        **
PCBACKUP               3        **    85842        **     8411        **
SSI                30604        **   602234        **   134129        **
SYS                67127        **   346970        **   279258        **
TECH                7256        **        0        **        0        **
TOOLBOX              656        **      467        **      524        **
TREND              30036        **   578779        **    80225        **
NO GROUPS FOUND IN GROUP-SET (CIWARN 432)
```

FILE ACCESSING FREQUENCY

An excellent indicator of many evolving trends in system usage can be developed from the frequency of accessing the various data files within your systems. By collecting this information on an on-going basis and then periodically reporting on the busiest files, you can often spot emerging trends.

In most cases, two parallel approaches should be used. The first one is to total the usage against each file individually and then sort and report on the busiest files. The number of files reported on can usually be reduced to the top 20 or so since this will normally include the key files within your organization. The second part of this two-fold approach is to identify the files that you feel are "key indicators" in your particular environment even though they may not be heavily used files and then track these files separately.

The frequency of accessing particular files can help in spotting changes in the usage of specific files which may reflect new uses for the data contained within those files. It also provides a focus for performance optimization since efforts directed towards reducing physical disc file accessing can be maximized when the energy expended is applied to the files with the highest accessing activity.

The INTEREX contributed library contains a program (FILERPT) that provides an excellent method of reporting on file usage. It relies on the availability of file close records collected by the MPE logging system. The following example is a brief example of the type of report that can easily be produced:

```
Sort on 1) # of RECORDS PROCESSED
        2) # of BLOCKS PROCESSED
        3) # of FCLOSES
Enter sort type (1, 2, 3): 2
What percentage should be printed? (100%) 1

BLOCKS PROCESSED   Report v2.0 (C) HEWLETT-PACKARD CO. 1980
TUE, MAY 30, 1989,  3:30 PM

FILE NAME                     TYPE  LDEV    REC COUNT    BLK COUNT      FCLOSE
                                                                         COUNT

 SORTSCR .PROD    .ANA          3     12*  18,661,433.  18,661,433.       343.
 SORTSCR .PROD    .FINEX        3      3*   7,128,384.   7,128,384.       223.
 SORTSCR .TECH    .INT          3     11*   3,360,561.   3,360,561.        46.
 ANA21   .PROD    .ANA          3     12*   2,108,367.   2,108,367.        66.
 ANA01   .PROD    .ANA          3     11    2,058,965.   2,058,965.       112.
 SORTSCR .PROD    .ANAPRO       3      1*   1,814,280.   1,814,280.       169.
 SORTSCR .PROD    .FILERS       3      1*   1,624,549.   1,624,549.       638.
 SORTSCR .PUB     .REP          3      2*   1,615,245.   1,615,245.        58.
 ANA08   .PROD    .ANA          3     12    1,484,544.   1,484,545.       100.
 FINEX07 .PROD    .FINEX        3      3    1,215,575.   1,215,575.        51.
 SORTSCR .PROD    .ANAMATE      3      1*     912,386.     912,386.        11.
 FINEX01 .PROD    .FINEX        3      3      856,845.     856,845.        62.
 INTKSAMK.TECH    .INT          3      3      818,956.     818,956.        43.
 FVANAUPF.PROD    .ANA          3     11      789,943.     789,943.       302.
 ANA28   .PROD    .ANA          3     11*     620,185.     620,185.        80.
 CODES02 .PROD    .UTIL         3      3      620,118.     620,118.       100.
 ANAMTE01.PROD    .ANAMATE      3     11      506,940.     506,940.         8.
 FILERS04.PROD    .FILERS       3      2      453,721.     453,721.        59.
 SORTSCR .CORP    .MGMT         3     11*     434,109.     434,109.         4.
         .PROD    .ANA          3     12*   2,969,867.     431,180.     1,283.
 SORTSCR .SUZIE   .FILERS       3      2*     426,527.     426,527.        45.
```

THE DANGER OF USING AVERAGES

As we have already seen in a several of the examples, data averaged over a period of time tends to quickly filter out the peaks and valleys of the data that would be collected if we were sampling on a moment by moment basis. Looking at data averaged for a 15 minute window does not tell us directly what the deviation was. Even if the average utilization level for a 15 minute period was 50%, there may be periods of a minute or more where the system is being severely overloaded and yet this is masked in the averaging. A good rule of thumb seems to be that if the average exceeds 75-80%, you can be quite sure that there were periods of overload. Unfortunately, the converse (ie. if average utilization is less than 70% then there probably wasn't any overloading) is not true.

Based upon these potential problems, you would be wise to take a close look at what the deviation is for the data that is being averaged. If you are looking at data averaged for a day, consider selecting only the data for the critical time window so that you see the relevant trend. If you are averaging for a month, consider whether you should be excluding weekend data as well as non-prime shift periods.

THE DANGER OF USING TRENDS

Just as it can be misleading to look at averages, blindly relying on trends based upon averages is also a dangerous practice. In this case, generalized trends can mask significant changes in some portion of your system utilization. When this happens, a rapidly increasing trend in a subset of the workload can go undetected for a period of time and get out of control before the general trends highlight it.

For many of us, the cycle of business activity follows a pattern that repeats itself on an annual cycle. If we were to blindly act based upon trends that we think are present based upon a few months data, we can often make the wrong decisions. We must always temper the trends we see with our knowledge of the business environment in which we operate. A sound knowledge of this is essential if we are to be successful in our planning activities.

In order to plan for future capacity requirements, you cannot safely rely on trends in system utilization by themselves. You must also look at trends that extent beyond the computer centre and include the strategic directions of the company in your planning. If your company is planning to acquire another company or to add more warehouses, retail stores or product lines, these decisions will undoubtedly cause increased requirements for data processing and yet will never appear in your system usage statistics until well after the fact.

SUMMARY

As we have seen, the sources of trending information are quite diversified. They range from visual observation to sophisticated software collection and reporting tools. Data is available in some form for virtually any type of information that you might find helpful in managing your particular HP3000 computer system(s). The opportunities to gain better insight into how your system is being used so that you are better equipped to plan for future needs are almost endless. By collecting and using some subset of this wealth of data, every one of us can improve the functioning of our computer systems.

When contemplating the use of trending information as an aid in the overall management of your HP3000 computer system, you should consider the following:

Decide how much time and effort you are willing to budget for this activity. Because the potential sources and uses of the information are almost endless, you must budget your efforts so that you avoid becoming too absorbed in the process to the detriment of your other activities.

Restrict the manual data collection effort to the minimum possible since a plan that involves regular manual intervention is much more likely to fail than one that is relatively automated.

Keep regular reporting procedures to a minimum. The objective is to do the minimum amount of reporting that will allow you to highlight emerging problems and then have sufficient additional raw data to go back and investigate the particular trend in more detail.

Identify the "key indicators" that reflect the business volumes for your organization and your specific circumstances. These may be computer related such as the usage of a program or set of programs. They may also be non-computer specific such as order volumes or average dollar value per order.

Determine the simplest method for collecting and tracking data that reflects the "key indicators" for your circumstances.

Determine what amount of detail you wish to collect in order to track general system usage. This may simply be the output from the "report" command or it may be an elaborate trending facility utilizing automated collection and graphical reporting capabilities.

Consider collecting more data than you currently see a need for. This is because circumstances change and it is often useful to be able to review the raw data that has been collected and never used. It is rather discomforting to discover some time later that by collecting data in the past, you could have gained a much better insight into an evolving problem. A good hedge against this dilemma is to enable process termination and file close log records within the MPE logging system and then periodically archive the logfiles to tape.

MPE XL and Performance : Not Incompatible

by

Stan Sieler

Allegro Consultants, Inc.
2055 Woodside Road, Suite 170
Redwood City, CA 94061
Voice: (415) 369-2303
Fax: (415) 369-2304

With the release of MPE XL on HPPA, many new features have arrived for the programmer. These include mapped files and a very large address space. One new feature overlooked by many is the RISC architecture. Although RISC means "reduced complexity", optimizing performance on RISC is paradoxically more complex than on the classic HP3000. This paper asks: "what can we do to maximize performance?" Some answers are presented, and particular attention is given to the characteristics of mapped files, the file system, and Native Mode versus Compatibility Mode.

1. Mapped Files

This section will introduce mapped files and discuss their performance characteristics.

1.1 Mapped File Introduction

From a programmer's viewpoint, MPE XL has two basic types of files: the ordinary, record-oriented files that have existed since the birth of MPE, and mapped files.

A mapped file is an MPE ordinary file that is going to be accessed via virtual memory loads and stores instead of via file system intrinsics. Instead of calling FOPEN, a programmer can call the new HPFOPEN intrinsic, and specify that a file is to be opened for "mapped" access. This will result in two pieces of information being returned to the program: a file number (like FOPEN would have returned), and a virtual memory address. The virtual memory address returned is the address of the first byte of data in the file. If the address is stored in a pointer, as shown in the following example, and the pointer is then "de-referenced", the first byte from the file is brought into memory.

HP Pascal/XL	SPLash!
var filedata : ^char;	virtual byte pointer filedata;
firstbyte : char;	byte firstbyte;
...	...
hpfopen (..., filedata, ...);	hpfopen (..., filedata, ...);
firstbyte := filedata^;	firstbyte := filedata;

Note: the above example was done with HP Pascal/XL, but most of the rest of the examples in this paper will be done in SPLash!, a native mode version of SPL/V, which allows easy manipulation of 32 bit and 64 bit virtual addresses. Mapped file access is also available in HP C/XL.

With the above fragment of code, let's look at fetching the first two 80-byte records.

```
byte array
   rec0'        (0 : 79),
   rec1'        (0 : 79);

move rec0' := filedata, (80);         ! get first 80 bytes
move rec1' := filedata (80), (80);    ! get second 80 bytes
```

If the file system had been used to access the first two records, as in:

```
fread (fid, rec0', -80);
fread (fid, rec1', -80);
```

then the total CPU utilized by the FREADs would be much greater than the CPU used by the two "move" statements.

1.2 How are Mapped Files Implemented?

In MPE XL, all files are stored on disc as an array of bytes. A file is called a "mapped file" if it happens to have been opened by a user who requested its virtual address be returned as a result of the HPFOPEN intrinsic. At the lowest level of MPE XL, ALL disc files are always opened as mapped files. Usually, we call a file a "mapped file" if we intend to access its data via virtual memory along with (or instead of via) the file system intrinsics.

Two aspects of disc files have changed from MPE V to MPE XL:

1) the file label is not stored as part of the file.

2) there is no wasted space between records or between blocks.

The first change is a decade overdue. The second change is a direct result of the virtual memory system of HPPA.

When any disc file is opened in MPE XL, a module called the "Virtual Space Manager" allocates a range of virtual addresses sufficient to cover the entire file. The process is called "mapping", as in: mapping the file into virtual memory. "Mapping" provides a one-to-one correspondence between a virtual memory address and a byte of disc data for every byte in the file.

If a program tries to use a virtual address that has been mapped onto a file to fetch a byte of data, the following is done by hardware:

1) Extract the upper 53 bits of the 64 bit virtual address, calling it the VPN (Virtual Page Number).

2) Is the virtual page "in" memory. (I.e.: is there a physical page of 2,048 bytes that has been assigned to that VPN?)

3) If yes, then using the bottom 11 bits (the page "offset") of the original 64 bit virtual address, index into the physical page, fetch the byte, and return.

4) If no, interrupt and ask the software to bring our page into physical memory.

5) When our page arrives in memory, our process will be restarted at step (1) above.

The above process can be phrased in a simpler manner:

If the virtual address is in real memory, fetch the data; otherwise do a "page fault" and swap the page into memory and then fetch the byte.

Note: this description of virtual memory is simplified, and omits features such as the Translation Lookaside Buffer (TLB).

Thus, to fetch the first byte of the 100th record of an 80-byte record file, we can simply take the virtual address of the first byte of the file, add 8000 to it, and then fetch a byte from that address. Sooner, or later, the byte will appear in the register that we asked it to be loaded into.

The detailed workings of virtual memory are quite complex, and beyond the scope of this paper. For now, let's just remember:

> When bytes of a file are accessed via a virtual address, the data is brought into memory as needed by the operating system via "page faults". Once a page is in memory, its data can be accessed at main-memory speeds. On a typical MPE XL machine, many millions of bytes of mapped files could be in memory all at the same time.

If anything is stored into the virtual address, the physical page is marked dirty. Dirty pages are eventually written out to disc, but this process might not occur for quite some time.

When we talk about a "page" in reference to the CPU hardware, we generally mean a "physical page" of 2,048 bytes. At most other times, "page" refers to a "logical page" (sometimes incorrectly called a "virtual page") of 4,096 bytes. When a logical page is brought into memory, it will occupy two consecutive physical pages.

1.3 Prefetch

"Prefetching" is the act of bringing more data from disc into memory than was immediately requested by a user, in an attempt to prevent a second disc read shortly after the first.

The disc caching code on MPE V had two "dials" the system manager could twist to control the amount of data prefetched. One dial to control the size of cache domains created for sequential disc reads, and another to control the size of domains created for random disc reads.

On MPE XL, the system manager has no such controls. Instead, the prefetch size is determined (at present) by one primary factor: what subsystem is asking for the data to be read from disc. If the request to read data from disc is from the memory manager (due to a page fault), one logical page is read. If the request is from the file system, several logical pages are read.

Clearly, this has enormous performance implications. Consider a program accessing a file of 256 byte records in a sequential manner. Assuming the file has about 90,000 records, and assuming that the file system requests 4 logical pages at a time, then the memory mapped access will have 5,625 page faults versus 1,406 for the file system accessor. (Remember: a logical page is 4,096 bytes, and a physical page is 2,048 bytes. Unless dealing with the lowest levels of MPE XL, we normally refer to logical pages.)

As a test of the above, a program was run that did a simple sequential read of the file SL.PUB.SYS (89,867 records of 256 bytes). This file takes about 22 megabytes of disc space. The following table show the CPU and Elapsed times required to read the file. In between each run, a separate 16 megabyte file was read in an attempt to flush as much of the SL.PUB.SYS file data from memory as possible (see the section: Measurement Problems).

The following table shows the time the test program needed to read SL.PUB.SYS. The test program was running in Native Mode.

SL.PUB.SYS sequential read (times in milliseconds)

CPU	Elapsed	Delta	Access Method
19686	146298	126612	Memory Mapped
35398	44361	8963	FreadDir
36590	44957	8367	Fread
39465	46802	7337	FreadDir & FreadSeek
48650	51949	3299	Fread & FreadSeek

The "Delta" column shows the amount of time the program was presumably waiting for the data to come from disc.

The "FreadDir" access method consisted of using the FREADDIR intrinsic with ascending record numbers, which results in reading exactly the same records as the FREAD intrinsic. The last two rows added a call to the FREADSEEK intrinsic in an attempt to have MPE XL prefetch data before it was read. For those two tests, FREADSEEK was called once every 4 reads, with a request to prefetch the fourth record following the current.

The implications:

1) Use sequential FREADDIR to sequentially read a file that is not already in memory (see note below);

2) Don't use FREADSEEK. At least in these tests, it never seems to help, and only costs extra CPU time.

Taking the first delta figure, 126,612, and guessing that we can do a disc read in 22.5 milliseconds, we get an estimate of 5,627 disc reads, which matches our prediction.

If we take the delta for the FREAD test, 8,367, and using the same estimate of 22.5 milliseconds per disc read, we see 372 disc reads. This implies that FREAD is prefetching in chunks of 15 or 16 logical pages, not the 4 originally assumed.

Note that with the FREAD & FREADSEEK test the delta was cut about in half, at the cost of greatly increased CPU time.

A second large file was tested, NL.PUB.SYS (64,275 records of 256 bytes each, 16 megabytes):

NL.PUB.SYS (sequential)

CPU	Elapsed	Delta	Access Method
11507	74920	63413	Memory Mapped
22109	26240	4131	FreadDir
23857	27364	3507	Fread
25735	28124	2389	FreadDir & FreadSeek
28887	31151	2264	Fread & FreadSeek

These results mirror those for reading SL.PUB.SYS.

1.4 Memory Resident Data

The previous section examined the performance of mapped files versus the file system for data that was out on disc. Frequently, the data for a file will happen to be resident in memory. This is the case when a file is accessed multiple times in a relatively short period. This section examines the performance of accessing file data that is already in memory. Using the same Native Mode program (an SPL/V program compiled with SPLash!), the file CATALOG.PUB.SYS was sequentially read. This file has 7040 records of 80 bytes each for a total of 0.5 megabytes.

CPU	Elapsed	Access Method
181	182	Mapped File
1660	1677	FreadDir
1678	1680	Fread
1959	1976	FreadDir & FreadSeek
1977	1994	Fread & FreadSeek

The file CATALOG was read once to bring it into memory. The time to do this is not reflected in the above table.

Note that the elapsed time is just slightly more than the CPU time. This is because the process is never paused to wait for disc I/O.

The implications:

1) If the file's data is likely to be in memory, use mapped file access!

2) FREADSEEK should not be used for files where the data is in memory already.

1.5 NM vs CM vs OCT

MPE XL can execute in any of three modes: Native Mode (executing RISC instructions), Compatibility Mode (emulating classic HP3000 CISC instructions), and a blend of the two produced by the Object Code Translator (OCT). Briefly, a Compatibility Mode (CM) program can be run through the OCT to produce a hybrid program file that contains the original CISC instructions as well as their translation into RISC instructions. OCT'ed programs must obey ALL the same restrictions as CM programs (e.g.: 16-bit wide stack of 65,535 bytes). (For more information on OCT, CM, and NM, the reader is directed to the book "Beyond RISC" from Software Research Northwest.)

The data in the preceding tests was obtained from a Native Mode program. This section examines the performance of the file system when called from the three types of program code: NM, OCT, and CM. As a reminder of what can be accomplished by what my partner, Steve Cooper, calls the "second migration", mapped file access is also shown in the table. The "second migration" is the process of adapting programs to take advantage of the new features in MPE XL. The "first migration" is the one HP talks about: porting a program to Native Mode (which usually means minimal changes).

The file CATALOG.PUB.SYS was sequentially read in the same manners as before, with the IDENTICAL program compiled in SPL/V (CM), run through the Object Code Translator (OCT), and compiled by SPLash! (NM). The following table shows the results:

CATALOG.PUB.SYS (times in milliseconds)

CPU	Elapsed	Mode	Access Method
181	182	NM	Mapped (requires NM)
1660	1677	NM	FreadDir
1678	1680	NM	Fread
1959	1976	NM	FreadDir & FreadSeek
1977	1994	NM	Fread & FreadSeek
3326	3343	OCT	FreadDir
3838	3854	OCT	Fread
4196	4214	CM	FreadDir
4850	4881	CM	Fread
5196	5216	OCT	FreadDir & FreadSeek
5670	5690	OCT	Fread & FreadSeek
6471	6493	CM	FreadDir & FreadSeek
7473	7493	CM	Fread & FreadSeek

The implications:

1) NM is far faster than CM or OCT.

2) Calling FREADSEEK from CM or OCT programs is even more of a penalty than calling it from NM programs.

3) FREADDIR is still slightly faster than FREAD.

The test program was produced from the source file "READER" with the following commands:

```
CM:   spl reader, $newpass, $null
      prep $oldpass, reader.cm

OCT:  octcomp reader.cm, readero.cm, , noovf

NM:   splasm reader
```

Note that the "noovf" option on the "octcomp" command tells the OCT that the program does not expect to generate arithmetic overflows and to optimize its translation with that in mind. This results in slightly faster OCT'ed programs.

The basic reason that the CM and OCT programs are so much slower is that simple disc files are handled by Native Mode portions of MPE XL. Some types of disc files are still handled by Compatibility Mode portions of MPE XL, ported from MPE V/E. These include message files, RIO files, Circular files, and KSAM files.

When a CM or OCT program calls the FREAD intrinsic to read a record from an ordinary disc file, the FREAD intrinsic must "switch" to Native Mode and call the Native Mode FREAD intrinsic. This switch is not inexpensive. OCT programs pay the same switch overhead as CM programs because they are still emulating the Classic instruction set, albeit faster than the emulator. NM programs (e.g.: HP Pascal/XL and SPLash!) are already in Native Mode when they call FREAD, so no switch is necessary.

The next test shows the results of serially reading a KSAM file of 1,000 80 byte records from NM, OCT, and CM programs. As in the CATALOG test, the file was brought into memory before the start of the test.

CPU	Elapsed	Mode	Access Method
2475	2494	OCT	Fread
2677	2696	CM	Fread
3239	3257	NM	Fread

Note that the FREAD intrinsic returns the records in key order, not the chronological order in which they were written.

Note that the FreadDir test was dropped. The FREADDIR intrinsic cannot be used on KSAM files.

The mapped file test was dropped because it reads the data in chronological order, not key order.

The implications:
If KSAM is being used heavily, don't migrate the programs into NM until a native mode version of KSAM is available (from HP or another vendor).

2. Memory & Disc Utilization

In MPE V, stacks were limited to a maximum of 65,535 bytes. In MPE XL, the limitation is 1 gigabyte (1,073,741,824 bytes). (This limit includes the CM stack & heap, the NM stack, the NM heap, and the XRT.)

In MPE V, if any part of the stack was in memory, then the entire stack was in memory. In MPE XL, only the logical pages recently referenced are likely to be in memory at any time. Additionally, only those pages that have EVER been referenced are allocated disc storage. As more and more stack/heap pages are touched, more and more pages are allocated on disc. This means that having an array of 1,000,000 bytes in SPLash! (or Pascal/XL, or any NM language) is not expensive...until you use it. A megabyte array will have 1 million bytes of virtual address assigned to it, but the disc storage will range from 0 to 256 logical pages!

Disc files are allocated storage exactly like the stack/heap: only those pages ever touched are allocated disc sectors. (Since extents may be allocated several logical pages at a time, some rounding-up does occur.) This means that it is feasible to have "sparse" files. For example, a file with 1 byte for every possible Social Security number would have a limit of 999,999,999 bytes. If a single write is done to record 2345, then a single extent will be allocated. A test done on MPE XL 1.1 resulted in an extent of 2,048 sectors being allocated. This does not mean that all future extents will be of equal size. Unfortunately, the programmer has no control over the extent size.

3. Data Alignment

On the Classic HP3000, the natural data alignment was 16 bits. With rare exceptions, 32-bit and 64-bit data could be placed at any 16-bit boundary with impunity and no performance ramifications.

On the HPPA HP3000s, the natural data alignment is 32 bits for 32-bit data, and (sometimes) 64-bits for 64-bit data. (The 64-bit alignment applies primarily to IEEE 64 bit floating point numbers.)

As a result, if code is ported from a CM language to its NM equivalent, one of two problems can result: program aborts (or other errors) due to misaligned data; or performance slowdowns.

Most NM compilers provide a means of specifying that certain variables are only 16-bit aligned. When this is done, then the compilers will typically emit 3 instructions to load a 32 bit variable instead of the 1 that would have been required if the variable was aligned on a 32-bit boundary. This is necessary because the RISC hardware does not allow the LDW (Load 32-bit Word) instruction to be given an address that is not a multiple of 4 bytes (32 bits). Instead, 2 LDH (Load 16-bits) instructions and one DEP (deposit) instruction must be used to build the 32 bit value in a register.

No performance data is shown here because the implications are clear from the instruction count: 1 versus 3.

4. SORT vs HPSORT

Compatibility Mode programs that call the SORT intrinsics still get the old sort package, running in OCT.

Native Mode programs have a choice of two intrinsics to do sorting: SORT and HPSORT. These two intrinsics are interfaces to a new sort package which runs in Native Mode. The native mode sort package lacks some of the features of the CM sort facility (e.g.: the ability to pass procedures to do the comparison), and has one additional wrinkle: sometimes it calls the CM sort to do the sort!

In its present incarnation, NM Sort will call CM Sort when it gets a "difficult" sort. This includes sorts that specify an alternate collating sequence.

Additionally, when NM Sort does stay in NM, it does NOT open a temporary file called SORTSCR. Instead, it uses two temporary files that are either nameless or have a name like HPSORT1 and HPSORT2 (?), depending on the release of MPE XL. This means that if a fairly simple sort is requested from a NM program, the programmer cannot point the sort scratch file to a disc drive he/she knows is separate from the input and output data.

In short, NM Sort is still evolving. Test runs should be made before converting to NM simply to call NM sort.

5. System Performance

The overall system performance can still be affected by proper tuning of the C, D, and E subqueues via the TUNE command.

The choice of disc drives for a file can also be controlled in the usual manner (e.g.: BUILD FOO;DEV=3). However, the number of extents cannot be easily controlled any more. The basic choice is one extent or many extents.

Main memory is vital to the performance of the system. Unlike MPE V, which tended to degrade slowly, MPE XL will suffer a very sharp drop in performance when not enough memory is available. Economize on everything else ... and buy memory.

A 950 (and 955) will support up to 256 megabytes (128 per memory controller). Three vendors offer memory for the machine: HP, Kelly Computer Systems (the first to put 256 megabytes in a user's computer), and EMC. Sites with Classic HP3000s may be interested in Kelly's RAMDISC for the 3000, which can be traded in on HPPA memory when needed.

6. NM vs. CM : Intrinsics

In an earlier section, we determined that some types of files are still implemented with Compatibility Mode code.

File system intrinsics are not the only ones that might actually be implemented in CM. The ASCII, BINARY, DASCII and DBINARY Native Mode intrinsics currently switch to CM to do their work. Although this may change in the future, the performance implications are still interesting today.

Porting a program into Native Mode may reveal other intrinsics that are still implemented in Compatibility Mode.

The following table shows the result of calling the ASCII intrinsic a large number of times from programs written in NM, OCT, and CM:

CPU	Elapsed	Mode
9051	9084	NM
11688	11728	OCT
12211	12252	CM

Although the Native Mode program was the fastest, it is by a very narrow margin.

The ASCII/BINARY/etc. intrinsics have always been a performance bottleneck on MPE V. They haven't changed in MPE XL. The following table shows the results of calling the ASCII intrinsic versus calling a "clone" of the intrinsic:

CPU	Elapsed	Mode	Procedure
457	471	NM	ASCII clone
9009	9040	NM	ASCII intrinsic

Similar savings can be obtained for BINARY, DASCII, DBINARY, and CTRANSLATE. Contact the author for NMOBJ files that can be used as replacement intrinsics.

7. Measurement Problems

Measuring performance on MPE XL is extremely difficult. Unlike MPE V, MPE XL provides no control over what disc data is (or is not) in memory. As a result, tests must be run multiple times with best-case (or average-case) times used.

The difficulty of measuring performance is at its worst when looking at disc I/O. The following is a partial list of features that would aid this type of analysis:

1) An intrinsic that will make free all pages of memory that are not marked memory-resident or locked.

2) An intrinsic that will force all pages that are dirty to disc.

3) An intrinsic that would return for a virtual address information like: size of object and number of logical pages currently in memory.

The first feature would allow the system to be returned to a known "blank slate" state, allowing repeatable performance testing.

Note: an intrinsic allows the system manager/performance tuner/ software developer the ability to exercise the above functions programmatically. This is clearly superior to simply having a command for two reasons:

1) A command can be written by the user which simply calls the intrinsic. The opposite is not inexpensively true.

2) Intrinsics are not as easy to abuse by the casual user.

One valuable tool used in this paper is DEBUG. Given a virtual address associated with a mapped file, the debugger can be used to determine the number of logical pages that are currently in memory. Assuming the file starts at virtual address $123.0, then the debugger command:

```
= vainfo ($123.0, "pages_in_mem"), #
```

will report (in decimal) the number of 4,096 byte logical pages that are currently in memory.

8. Conclusions

Obtaining optimum performance with MPE XL is more difficult than on MPE V ... there are more things to tune, with much less knowledge. Things to remember:

1) The amount of memory on the machine is critical;

2) Migration to Native Mode is important, but should not be done blindly. If an application is a heavy KSAM or message file user, do some timing tests first.

3) The "second migration" is more important ... it means taking advantage of the new features.

Perhaps, when MPE XL begins to stabilize, and third-party performance tools are developed and marketed, the folklore on how to maximize performance will begin to grow as it did under MPE V. In the meantime, keep the faith!

NOTE: All timings in this paper were obtained running under MPE XL 1.1. Initial testing on MPE XL 1.2 shows no major differences.

W * A * I * T

An important postscript ... on the next page!

Postscript

FREADSEEK has been given a bad name in this article. Well, like the "goto", it has its uses. Further testing (and a lot of thought) resulted in a modification to the test program that was reading SL.PUB.SYS with the results:

SL.PUB.SYS sequential read (times in milliseconds)

CPU	Elapsed	Delta	Access Method
15273	49525	34252	Memory Mapped & FreadSeek
19686	146298	126612	Memory Mapped
35398	44361	8963	FreadDir
35903	65936	30033	FreadDir & FreadSeek
36590	44957	8367	Fread
39769	64529	24760	Fread & FreadSeek

Notice the incredible change in the "Memory Mapped & FreadSeek" numbers. The crucial difference here is in the timing and quantity of calls to the FREADSEEK intrinsic. Earlier testing showed that the best case "throughput" for reading data with a mapped file (where the data was already memory resident) was about 3,111 bytes per millisecond. (Obtained from the memory-resident speed of reading CATALOG.PUB.SYS (80 * 7040 bytes) in 181 milliseconds.) Clearly, the any prefetch should be done far enough ahead of time that the data is in memory by the time it is needed. The above calculation showed that if we assume it takes 30 milliseconds to read data from disc, then it must be requested 30 * 3,111 bytes before it is needed.

The test program was adjusted to prefetch 128 records ahead (instead of 4 records ahead). The next round of timings showed a gain, but not as much as hoped for. Then, we realized that the prefetch was reading 8 logical pages. So, after processing 24 logical pages (100,000 bytes) the test program was prefetching 8 logical pages instead of 24! The program was modified again, to fetch 24 logical pages at a time, resulting in the times shown above.

Moral: prefetching via FREADSEEK is worth the time, but ONLY after careful analysis. Failure to prefetch at the right time, or not enough data, is worse than not prefetching at all.

Hewlett-Packard Fiber-Optic Link
A Performance Growth Path for MPE XL Systems

MELODY ARMSTRONG
HEWLETT-PACKARD COMPANY
P.O. BOX 39
BOISE, ID 83707

Introduction

The Hewlett-Packard Fiber-Optic Link, referred to as HP-FL, is a disk interface specifically designed for Precision Architecture Systems. HP-FL is based on fiber-optic technology and transmits data between the CPU and the disk drives via light pulses. With the advent of HP-FL, HP 3000 MPE XL systems now have an attractive alternative to the traditional HP-IB disk interface (Hewlett-Packard Interface Bus).

The advantages of HP-FL relative to HP-IB are numerous. First, up to eight HP 7936/37FL disks can be placed on a single HP-FL interface card while HP-IB is limited to six HP 7936/37 disks per HP-IB interface card. This means larger disk configurations can be achieved with HP-FL using fewer CPU I/O slots. Second, HP-FL supports fiber optic cable lengths up to 500 meters while HP-IB supports a maximum cable length of 15 meters. This allows HP-FL a higher degree of configuration flexibility because disks can be placed further away from the CPU. Third, the fiber-optic cable is immune to electromagnetic interference and does not emit radio frequency energy that might cause interference with other equipment. Fourth, HP-FL offers an improved data transfer rate relative to HP-IB, 5 megabytes per second versus 1 megabyte per second, respectively. In conjunction with protocol improvements, this increases the performance potential of HP-FL. Finally, the flexible HP-FL design provides new configuration opportunities. The fiber-optic interface is a high-speed peripheral network for shared devices and host interconnection. Multiple hosts as well as multiple disks can share this peripheral network (multiple-host configurations are not currently supported with today's MPE XL systems). HP-FL offers a growth path for the future and a platform for future mass storage solutions.

While many of the advantages of HP-FL are well known, one area that is not widely understood is the relationship between HP-FL and performance. From the beginning, performance has been a key design goal. The unique characteristics of HP-FL have made it necessary to incorporate a special feature set to ensure performance optimization. This feature set consists of several enhancements not previously available with HP-IB including a distinct method of managing channel utilization, transaction pipelining, command queuing and seek reordering. These enhancements complement the increased data transfer rate capabilities of HP-FL to maximize performance. Understanding the contribution each of these features makes to performance helps clarify the relationship between HP-FL and performance.

Channel Utilization

The channel is defined as the communication path between the system and the disk. The host and disks interact frequently during the processing of a disk transaction to transfer commands, user data and status reports. The method in which this interaction is accomplished has a big impact on performance. This is especially true in multiple disk configurations where several disks may require use of the channel simultaneously.

Early HP-IB disk implementations did not always offer the greatest efficiency in relation to channel utilization. As a result, several enhancements have been implemented over the years to optimize disk performance in association with channel utilization. Among these enhancements were buffer prefill, rotational position sensing and data transfer during seek on a write. These improvements have optimized the interaction between the disks and the channel and allow for greater efficiency in multiple disk configurations.

Although the HP-IB channel management techniques work well given the characteristics of the interface, they are not optimally suited for HP-FL. Unlike HP-IB, the transfer rate of the HP-FL channel is faster than that of the disk. In order to take advantage of the increased data transfer rate and provide maximum channel efficiency in multiple disk as well as multiple host configurations, a unique approach to channel utilization has been implemented. This method incorporates efficient resource management techniques that allow the disk and the channel to work independently of one another, such as 1) ensuring data will be available to transfer by the time the channel is acquired, 2) negotiating with the destination device to ensure the necessary resources are committed prior to moving data and 3) breaking large transfers that exceed the size of available resources into multiple data request/transmission blocks.

Multiple Data Blocks

With the current implementation of HP-FL, transfers that exceed the size of the disk's 32 kbyte internal buffer are broken into multiple data request/ transmission blocks. Thus, the disk is always capable of buffering a complete data block regardless of the total size of the transfer. If necessary, the disk's internal buffer is capable of managing two data request/transmission blocks simultaneously. That is, data associated with one block can be transferred from the buffer while data associated with another block is being accepted into the buffer. This prevents delays from occurring when multiple data blocks are used.

Figure 1 illustrates the multiple data block concept employed by HP-FL.

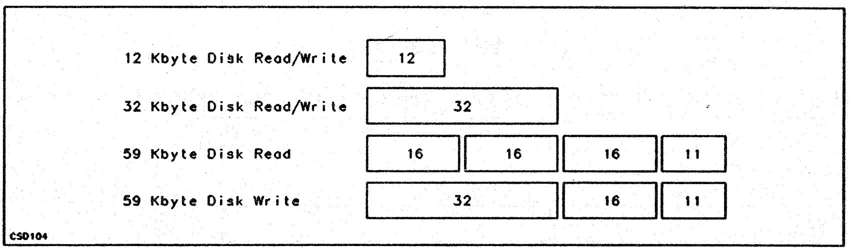


Figure 1. HP-FL Multiple Data Block Example

Disk reads in excess of 32 kbytes are separated into multiple 16 kbyte data blocks. Disk writes larger than 32 kbytes are broken into one 32 kbyte data block and subsequent 16 kbyte data blocks. Transfers that are equal to or less than 32 kbytes are transferred as a single data block.

The size of the data request/transmission blocks could change with future HP-FL implementations. The goal is to use block sizes that allow the disk to stay busy, while minimizing the overhead on the link. These goals are currently being met with the existing HP-FL implementation.

If it is necessary to break a data transfer into multiple data blocks, the channel is not held throughout the entire transfer. Instead, the channel is released in between each data block transmission. This means that each data block is treated as an independent transfer. As a result, it is possible for other disks or the host to acquire use of the channel in between data block transmissions.

Although it may seem inefficient to allow a single data transfer to be interrupted by another device, this scheme actually results in the fairest possible sharing of the channel by all devices. As a single entity each device may not always achieve the highest level of efficiency, but overall the efficiency of the disk network is improved.

The actual allocation of the channel is managed by hardware within the disk controller. For each group of disks attached to a system interface card, one disk is designated as the channel manager. This is determined by the device address and is typically device 0. A round robin priority scheme is used to allocate channel resources. This ensures that no host or device is starved.

Benefits

Figure 2 illustrates how the unique channel management techniques employed by HP-FL optimize channel utilization. This example compares the channel interaction that occurs with HP-FL during the processing of a 64 kbyte read to that of HP-IB (assuming average random seek).

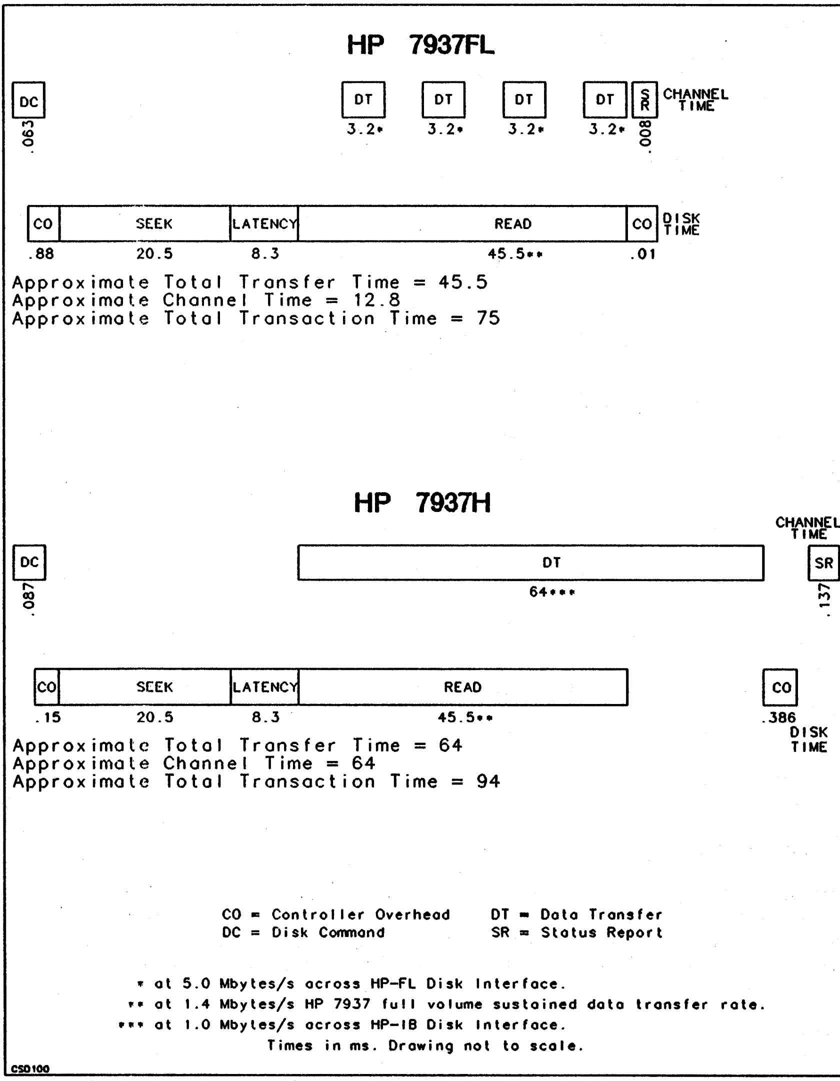


Figure 2. 64 Kbyte Read

One of the most noticeable differences between HP-IB and HP-FL is the usage of the channel during the data transfer. With HP-IB, the channel is requested .9 milliseconds before the target address is reached. Data is read from disk and transferred across the channel via the disk's internal buffer. The channel is held throughout the entire transfer regardless of total transfer size. If the transfer rate of the disk is slower than the channel, the channel may be forced to wait for the disk. Likewise, if the transfer rate of the channel is slower than the disk, the disk may be forced to wait for the channel, especially if the transfer exceeds the size of the disk's internal buffer. This means the maximum transfer rate is determined by the slowest entity, which in this example is HP-IB at 1 megabyte per second.

In comparison, the HP-FL disk ensures a full data block will be ready to transfer by the time the channel is acquired, so data can burst across the channel at the 5 megabyte per second data transfer rate. It accomplishes this by breaking the 64 kbyte transfer into four 16 kbyte data request/transmission blocks. The disk then buffers the initial data block into its internal buffer. While data is collecting in the buffer, the controller calculates how much time is required to complete the necessary resource negotiation with the host and how long it will take until the last byte for that block of data is read from disk. At the optimal time, the disk negotiates with the host for required resources and bursts the buffered data block across the channel at 5 megabytes per second. As soon as the data block is transferred, the channel is released. The process is then repeated for each data block. Since the drive's internal buffer is capable of managing two separate data blocks simultaneously, the disk can begin buffering a succeeding data block while the initial data block is being transferred.

Although data is transferred across the HP-FL channel at 5 megabytes per second, the time required to read data from disk must be factored into the total transfer time. This means that total transfer time is dependent upon the speed of the HP 7937 disk, which is typically 1.4 to 1.89 megabytes per second depending upon the number of head switches required. Based on a transfer rate of 1.4 megabytes per second, it takes approximately 45.5 milliseconds to transfer 64 kbytes with HP-FL and only 12.8 milliseconds of this is channel time. It takes approximately 64 milliseconds to complete the same transfer with HP-IB and a full 64 milliseconds of channel time is required. Not only does it take less time to complete the data transfer with HP-FL, but the availability of the channel is increased as well.

The exact transaction time savings associated with HP-FL varies from transaction to transaction and is dependent upon the length of the transfer as well as the transaction type. Typically, reads experience a higher savings than writes. This is because of the manner in which data is transferred during writes. Both HP-FL and HP-IB initiate the data transfer for a write as the seek begins. Depending upon the size of the transfer and the length of the seek and latency period, HP-IB may have enough time to buffer an adequate amount of data so the disk does not run out of data once the write begins. In such cases, total transaction time is more comparable to that of HP-FL. If there is not enough time to buffer an adequate amount of data, the HP-IB disk may be forced to wait for the channel. In order to compensate for these delays, the disk induces latencies which in turn increase total transaction time. In these situations, HP-FL will have some transfer time advantage. Either way, there is more channel time consumed with HP-IB than with HP-FL.

Transaction Pipelining

The HP 7937FL disks also provide transaction pipelining. This is a performance enhancement feature that maximizes disk throughput in I/O intensive environments by overlapping transaction processing. In other words, one transaction can begin before the previous one has finished. This offers an advantage relative to HP-IB where only one transaction can be processed at a time per drive.

A single HP-FL disk is capable of simultaneously managing multiple transactions. The disk controller has the ability to buffer a maximum of 14 disk commands at a single time (command queuing). This means up to 14 commands can be progressing through the decode process at one time. Furthermore, as many as two transactions can be simultaneously in the execution/report phase. This allows the disk actuator to be logically separated from the I/O channel, which permits the actuator to be dispatched to the next target address regardless of the delays associated with the channel. This enables transactions to be continually fed to the disk, thus reducing disk idle time and maximizing disk throughput especially during peak I/O periods. To fully understand the impact this has on performance, a closer look is needed.

Transaction Overlap

A typical disk transaction is comprised of three stages: the command phase, the execution phase and the report phase. The command phase is that portion of the disk transaction in which the disk controller receives and decodes a command. As soon as this is completed, the disk transaction enters the execution phase. During this stage, the disk mechanism performs the requested operation by mechanically positioning the heads over the designated location (seek and latency) and transferring the data via the channel. As soon as the transaction has been executed, it enters the report phase. During this phase, the disk controller conducts some cleanup and issues a status report to the CPU indicating successful completion of the transaction.

The HP-FL disks overlap transactions between command and execution phase and between report and execution phase. In addition, two transactions can overlap during execution phase as long as the total transfer size of each transaction does not exceed 16 kbytes. Transaction overlap allows for the masking of controller overhead during command decode and report phase. It also minimizes the effect of a busy channel on transaction throughput. Following is a description of how transaction overlap is achieved during the various phases.

Execution/Command Overlap

While the disk mechanism is executing one command, the controller can accept and decode other commands. As commands are received, they are queued and decoded one at a time in order of receipt. The decoded commands remain in the command queue until the disk mechanism is finished with the previous transaction. As soon as the mechanism is available, the controller is ready to immediately launch the next request. This enables the controller overhead associated with command decode for a transaction to be masked by the execution activities of the previous transaction.

Execution/Execution Overlap

It is possible for two transactions to be in execution phase at the same time. Naturally, the disk actuator cannot simultaneously perform the seek and latency for two transactions. Similarly, it is not possible for data to be transferred across the channel for two transactions at the same time. However, one transaction can be using the disk mechanism while another is using the channel to transfer data. This is only possible if the transfer size of each transaction is less than or equal to 16 kbytes. This is necessary to ensure that disk buffer resources can be fully committed for each transfer.

Execution/Report Overlap

While the disk controller is preparing to send the completion report for one disk command, the disk mechanism can begin executing the next command. This allows the controller overhead associated with the report status for a transaction to be masked by the execution activities of the succeeding transaction.

MPE XL Implementation

Although the HP-FL disks have the capability of simultaneously overlapping multiple disk transactions, the MPE XL operating system limits the transaction pipeline to a depth of two. This means a maximum of two disk transactions overlap at a given time per drive. The system maintains this control based on the number of commands queued within the disk. If the disk has a queue depth of two, the MPE XL driver does not initiate another disk command until the completion report for the previous transaction is received and the queue depth falls to one.

The MPE XL operating system limits the pipeline level in order to ensure that disk I/Os are conducted in the proper sequence. The nature of MPE XL applications is such that disk writes associated with a specific transaction must take place in the correct order to ensure data integrity. Although it may appear as though MPE XL is not getting maximum benefit from transaction pipelining, the largest incremental benefit is achieved when going from a pipeline of one transaction to two. Increasing the pipeline past two can offer some incremental benefit, but it is less than that realized by going from one to two.

Benefits

Figure 3 illustrates how transaction pipelining compares to the traditional method of processing transactions. This example provides a comparison between HP-FL and HP-IB when a read of 16 kbytes is immediately followed by a write of 16 kbytes (assuming average random seek for both transactions).

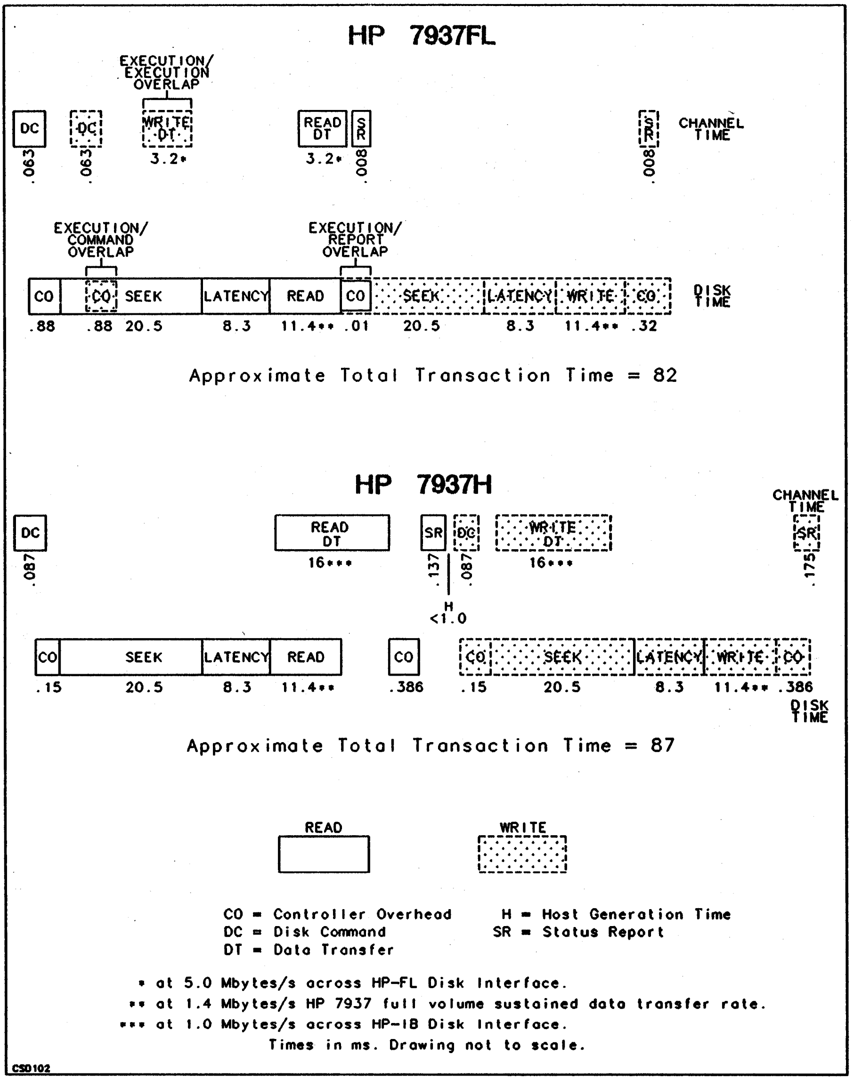


Figure 3. 16 Kbyte Read Followed by 16 Kbyte Write

Unlike HP-IB, the HP-FL disk can accept and decode the write command while the disk mechanism is performing the seek and latency for the read (execution/ command overlap). Although mechanical positioning cannot be initiated for the write while the disk mechanism is executing the read, the write data can be transferred to the disk's internal buffer (execution/execution overlap). Since the total transfer size of each transaction is 16 kbytes, there is adequate buffer space to accommodate the data associated with both transactions. As soon as the disk mechanism is available, the write command is immediately launched. While the mechanism is busy performing the seek and latency for the write, the disk controller prepares and issues the completion report for the read (execution/report overlap).

It takes the HP-FL disk approximately 82 milliseconds to complete the two transactions, whereas it takes the HP-IB disk approximately 87 milliseconds. The time savings provided by HP-FL is mainly attributed to reduced transfer time during the read transaction. There is also a small time savings associated with the masking of controller overhead and the elimination of host generation time.

The exact amount of time saved with transaction pipelining is dependent upon a number of factors including the sequence of transactions, the timing of I/Os and especially the availability of the channel. The previous example is representative of an environment in which channel availability is ideal. With multiple drives sharing the same channel, there is a higher probability of experiencing channel contention. In such instances, the benefit of transaction pipelining is more noticeable. This is because the disk actuator is not dependent upon channel resources. As a result, channel delays are less likely to impact disk throughput with HP-FL than with HP-IB.

Figure 4 illustrates the benefit of transaction pipelining when channel delays are experienced. As in the previous example, a comparison is shown between HP-FL and HP-IB when a read of 16 kbytes is immediately followed by a write of 16 kbytes (assuming average random seek for both transactions).

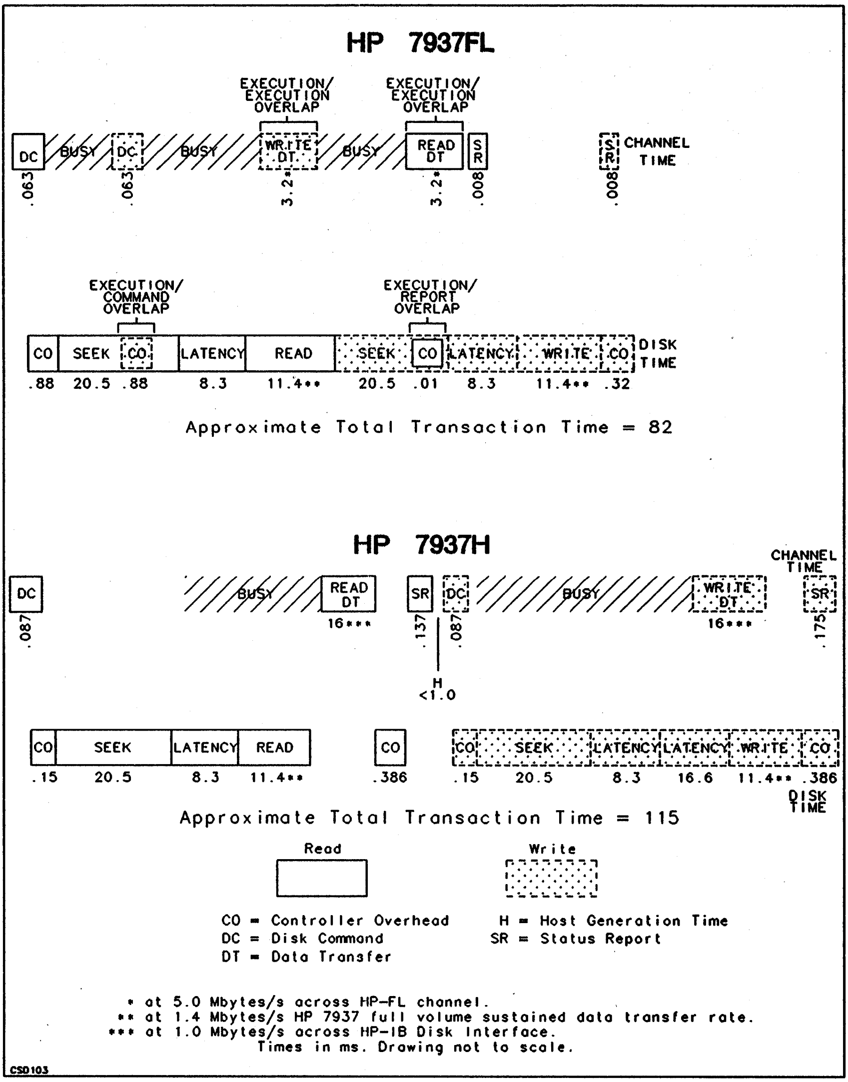


Figure 4. 16 Kbyte Read Followed by 16 Kbyte Write with Channel Delays

If channel delays are encountered with HP-FL, there is typically enough time for those activities that require use of the channel (the command transfer, the data transfer and the completion report) to take place later in the transaction process without impacting total transaction time. HP-IB does not offer this flexibility, because disk resources are tied to the channel. Therefore, there is a higher likelihood that channel delays will increase transaction time with HP-IB.

When compared to the previous example, the channel delays experienced with HP-IB increase total transaction time by greater than 25 milliseconds while total transaction time is unchanged for HP-FL. Although this example is very unfavorable toward HP-IB, it illustrates the benefit of HP-FL in multiple disk configurations.

Naturally, HP-FL is not completely immune to channel contention. If channel delays become excessive, the balance between channel and disk resources may be disrupted. When this occurs, the transaction pipeline becomes backed up and throughput is negatively impacted. However, the chances of this occurring with the present HP-FL implementation are negligible. The round robin channel allocation technique used by HP-FL ensures that channel resources are shared in the fairest possible manner. In addition, the transaction pipelining process has been specifically tuned for Precision Architecture systems to provide the highest level of efficiency, especially during I/O intensive periods.

Seek Reordering

The HP 7937FL disks also have the ability to reorder "Locate and Read" and "Locate and Write" commands. If multiple commands are queued in the command buffer, the disk controller reorders these commands so they execute in the most efficient order. The reordering scheme is based on seek distance and the length of time a command has waited in the queue. This method minimizes seek overhead associated with high traffic environments while ensuring no command is neglected. As a result, seek reordering helps level disk response time and increase disk throughput for burst activity.

The current MPE XL systems do not take advantage of seek reordering. The MPE XL operating system maintains control of the I/O sequence by limiting the pipeline depth to two. Thus, there is no opportunity for seek reordering at this time. However, seek reordering can make an important contribution in multi-host configurations where multiple systems pipeline transactions to each disk. In this environment, seek reordering will minimize seek time and increase the potential throughput of each HP 7937FL.

HP-FL and Physical Disk Performance

Hewlett-Packard uses the metric of I/Os per second to measure disk performance. I/Os per second is defined as the maximum number of disk transactions per second that a specific disk can perform at a given transfer size. Table 1 shows the disk transaction and I/Os per second figures for the HP 7937H and the HP 7937FL. A transfer size of 12 kbytes is shown because this is representative of the average MPE XL disk I/O. Since MPE XL also performs a number of very large transfers, a transfer size of 64 kbytes is also shown.

Table 1. HP 7937H and HP 7937FL Disk Read Throughput Figures

	12 kbytes		64 kbytes	
	HP 7937H	HP 7937FL	HP 7937H	HP 7937FL
Controller Overhead	1.0 ms	1.5 ms	1.0 ms	1.5 ms
Seek	20.5 ms	20.5 ms	20.5 ms	20.5 ms
Latency	8.3 ms	8.3 ms	8.3 ms	8.3 ms
Data Transfer Time (Based on 1.0 Mbytes/s Across HP-IB)	12.0 ms	-	64.0 ms	-
Data Transfer Time (Based on 1.4 Mbytes/s on HP 7937)	-	8.5 ms	-	45.5 ms
Total Access Time	41.8 ms	38.8 ms	93.8 ms	75.8 ms
I/Os Per Second (No Pipeline)	23.9	25.7	10.7	13.2
I/Os Per Second (Full Pipeline)	N/A	26.8	N/A	13.5

There is slightly more controller overhead associated with the HP 7937FL due to increased functionality. However, the ability to pipeline transactions effectively masks controller overhead on busy disks. The seek and latency time of the HP 7937FL is identical to that of the HP 7937H, because the same disk mechanism is used in both. The biggest contribution the fiber-optic link makes to physical disk performance is the reduction in data transfer time. This is because data transfer time is determined by the speed of the disk. As a result, a typical 12 kbyte MPE XL disk I/O executes in approximately 7 - 11% less time with HP-FL, depending upon whether the pipeline is full or not. This increases disk throughput by approximately 2 to 3 I/Os per second. A 64 kbyte disk I/O executes in approximately 19% - 21% less time with HP-FL and disk throughput is increased by nearly 3 I/Os per second.

It is important to keep in mind that disk transaction time and I/Os per second are used to compare the relative performance of one disk to another. These numbers represent fundamental disk performance without taking into consideration specific system attributes. Therefore, the benefit HP-FL has on system level performance cannot be extrapolated from these numbers.

HP-FL and MPE XL System Performance

The degree to which disk performance influences system level performance is determined in large part by the demands of the operating system and the user applications. Over the years, the disk I/O requirements of the classic MPE systems have evolved. Initially, these systems did not require a substantial amount of disk I/O. Therefore, CPU was often the performance bottleneck. As the hardware and software matured and the user and application base grew, disk I/O demands increased. Today, disk I/O is frequently a performance bottleneck for many classic MPE systems.

The knowledge gained from the classic MPE systems was incorporated into MPE XL operating system development. As a result, one of the MPE XL design goals was to improve system level performance by reducing disk I/O requirements. MPE XL accomplishes this through the use of mapped files, dynamic reads (prefetch algorithms), delayed and gathered writes, and a feature known as transaction management. These features take advantage of the large memory configurations available with HP-PA systems to effectively manage large amounts of data within main memory. As a result, MPE XL disk I/O characteristics differ considerably from those of MPE V. Not only does MPE XL require fewer disk I/Os to complete the same task, but disk I/Os tend to be larger and more bursty in nature.

Because of the reduced disk I/O requirements of MPE XL, the impact HP-FL has on the system level performance of today's typical MPE XL systems is small. In recent system benchmarks, HP-FL has exhibited a 0-8% increase in system throughput over HP-IB. Several of these benchmarks are summarized below.

On-line Interactive Applications

The first on-line interactive benchmark was a native mode COBOL manufacturing application. This application consisted of several modules typically found in manufacturing applications including purchase orders, work orders, labor processing, accounts payable, vendor maintenance, MRP, etc. It used a character-mode screen handling facility and the TurboImage data base management system. The data shown are representative of running the benchmark at 60 effective users with HP-IB and HP-FL disks.

The second on-line interactive application tested was a portion of an inventory management system which maintains orders and records inventory transactions into, out of, and within a stockroom. The majority of the software (COBOL, PASCAL) was migrated to native mode. Some SPL code was used in compatibility mode. Measurements were taken at 20, 30, 40 and 50 effective users with HP-IB and HP-FL disks.

Both benchmarks were conducted on an HP 3000 Series 950 running MPE XL 1.1 with 128 megabytes of main memory. The disk configurations were as follows:

Manufacturing Application

HP-IB: 6 HP 7937Hs, 3 per channel
HP-FL: 6 HP 7937FLs on 1 channel

Inventory Management Application

HP-IB: 5 HP 7937Hs on 2 channels (3:2)
HP-FL: 5 HP 7937FLs on 2 channels (3:2)

The results of the manufacturing benchmark are summarized in Table 2. The results of the inventory management benchmark are shown in Figures 5 and 6.

A *physical transaction* is defined as the amount of work the system does from the point at which the use hits the enter key or RETURN key to the next prompt for data by the system. A *logical transaction* is defined as the completion of a logical unit of work, i.e., create a work order or process a purchase order.

An *effective user* is a "heads down" user who takes no breaks while using the application. Effective users may have think times which represent the time between the system prompt and the user hitting the ENTER key (or RETURN). The think time includes the time required to enter data into the application.

Table 2. Manufacturing Application Benchmark Results

Disk	CPU % Busy	CPU % Pause	I/Os/ Sec	Response Time Mean	Response Time S.D.	Logical Throughput
HP-IB	86.0	10.9	22.8	.60	2.88	2,741
HP-FL	89.8	7.2	23.7	.54	2.32	2,796

Disk	Total Mbytes Transferred Reads	Total Mbytes Transferred Writes	Disk Queue Length % 1	Disk Queue Length % 2+
HP-IB	248	1,967	71.0	29.0
HP-FL	255	2,045	91.2	8.8

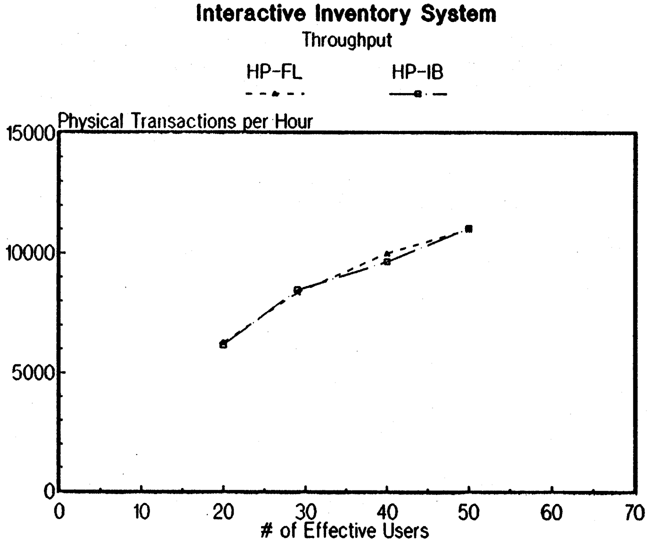


Figure 5.

Interactive Inventory System

Mean Response Time

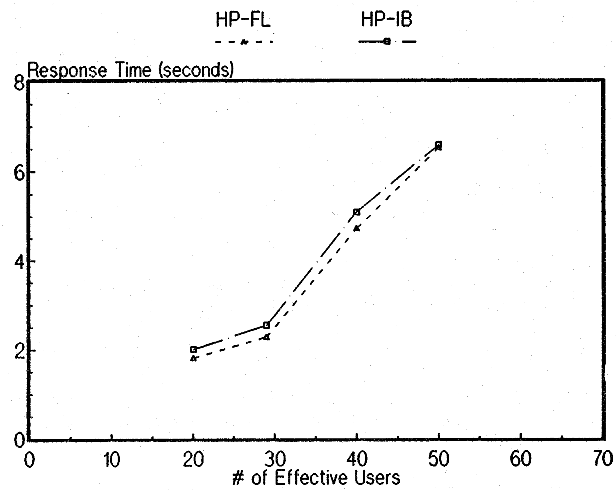


Figure 6.

HP-FL showed a 0-4% improvement in transaction throughput and an approximate 10% improvement in mean response time for both on-line interactive benchmarks. The reduction in response time, in conjunction with the significant reduction in disk queue lengths for the manufacturing test, is mainly attributed to transaction pipelining. Since MPE XL pipelines two transactions to the HP-FL disks, a great deal of disk request queuing is off-loaded from the system to the disk. This serves to reduce system level disk queue lengths. Transaction pipelining also helps even out response times during peak I/O periods.

It is important to note that memory size has an impact on the disk I/O rate of a system. Both on-line interactive benchmarks were conducted with large memory configurations. If a smaller memory size had been tested, the disk I/O rate may have been higher.

Batch Applications

Five different batch benchmarks were conducted as outlined below:

Benchmark 1

Native mode COBOL program which loads 50,000 records into a sequential flat file and then reads the entire file. This is done five times per job and four jobs are run simultaneously.

Benchmark 2

Single multi-part reporting job that extracts data from a large inventory data base using Robelle's SUPRTOOL. The job then processes and produces several reports based on the extracted item and purchase order data. Some COBOL programs are also used throughout the job. The COBOL and SUPRTOOL code is in compatibility mode.

Benchmark 3

Heavy reporting environment in which Business Report Writer XL is used to produce ten different financial reports from a TurboImage data base.

Benchmark 4

COBOL program that serially reads 570,000 entries of a TurboImage data set, and then serially reads 570,000 entries from a sequential MPE file.

Benchmark 5

Business Report Writer XL application that makes complex selections from a TurboImage data base. All entries which match a given selection criteria are sorted and reported into an output file.

All benchmarks were conducted on an HP 3000 Series 950 with the following configurations:

Benchmark 1 and 2

MPE XL (Release 1.1)
128 megabytes of main memory
HP-IB: 8 7937H drives, 4 per channel
HP-FL: 8 7937FL drives on 1 channel

Benchmark 3, 4 and 5

MPE XL (Release 1.1)
64 megabytes of main memory
HP-IB: 5 7937H drives on 2 channels (3:2)
HP-FL: 5 7937FL drives on 1 channel

The results of benchmark 1 and 2 are shown in Table 3. The results of benchmark 3 are summarized in Figure 7, while the results of benchmark 4 and 5 are depicted in Table 4.

Table 3. COBOL I/O and SUPRTOOL/COBOL Batch Report Results

Test	Disk	CPU % Busy	CPU % Pause	I/Os/ Sec	Elapsed Time CPU	Elapsed Time Wall
1	HP-IB	97.2	1.7	12.7	547s	10m
1	HP-FL	96.3	0.2	10.1	547s	10m
2	HP-IB	45.6	49.4	18.5	1896s	69m
2	HP-FL	54.5	44.9	19.8	-	66m

Test	Disk	Total Mbytes Transferred Reads	Total Mbytes Transferred Writes	Disk Queue Length % 1	Disk Queue Length % 2+
1	HP-IB	72	299	47.8	52.2
1	HP-FL	5	303	86.2	13.8
2	HP-IB	692	207	94.7	5.3
2	HP-FL	712	242	95.0	5.0

The performance of HP-FL and HP-IB in the first benchmark was very similar. HP-FL did provide a significant reduction in disk queue lengths, indicating the offloading of queuing from the system to the disk level.

In the second benchmark, HP-FL showed a 4.5% improvement in elapsed wall time. The system experienced a high percentage of pause for disk, primarily because the majority of code used for this benchmark runs in compatibility mode. Higher I/O levels would be achieved if the application was converted to native mode.

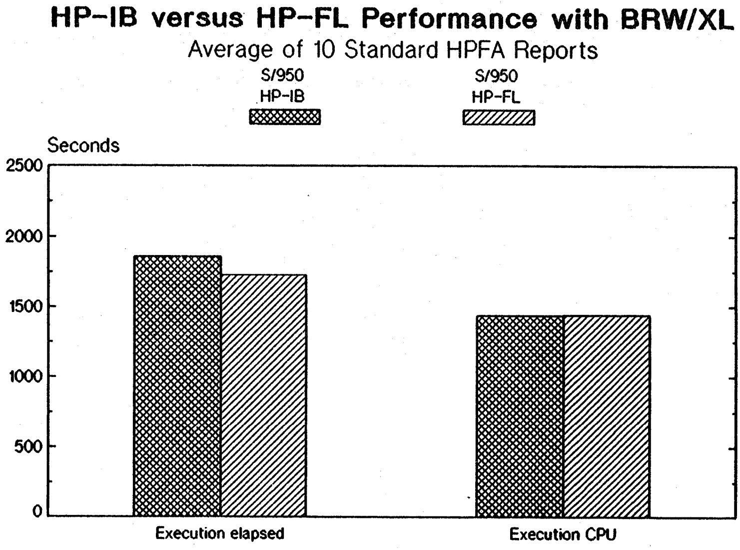


Figure 7.

The third benchmark showed, on the average, a 7% improvement in elapsed time for the HP-FL disk drives, whereas total CPU usage was nearly equivalent for both HP-IB and HP-FL. The total CPU utilization was 77% CPU busy with HP-IB and 83% CPU busy with HP-FL.

Table 4. COBOL and BRW Select Batch Results

Test		Disk	CPU	Elapsed
4	COBOL/T-Image	HP-IB	308s	5m
4	COBOL/T-Image	HP-FL	304s	5m
4	COBOL/MPE File	HP-IB	116s	4m
4	COBOL/MPE File	HP-FL	115s	4m
5	BRW Select	HP-IB	549s	699s
5	BRW Select	HP-FL	532s	645s

The performance of HP-IB and HP-FL was virtually identical in the fourth benchmark. The application used for this benchmark was not very I/O intensive. In the fifth benchmark, HP-FL exhibited a 8% improvement in elapsed wall time relative to HP-IB.

Benchmark Summary

The exact performance benefit derived from HP-FL is dependent upon the level of disk I/O generated by the application. These benchmarks did not generate significantly high I/O rates on the Series 950. Due to the success of the MPE XL operating system at reducing disk I/O, it is very difficult to find an application for which disk I/O is currently a performance bottleneck. Since the I/O system was not saturated, the full performance potential of HP-FL is not represented in these benchmarks. These tests do, however, give an indication of the impact HP-FL has on the system level performance of today's MPE XL systems.

To better illustrate the full performance potential of HP-FL, Figure 8 provides a comparison between the raw I/O capability of a single HP-IB drive and a single HP-FL drive. The data shown are representative of raw disk performance in an environment in which I/Os are generated at a rapid rate with very little system overhead.

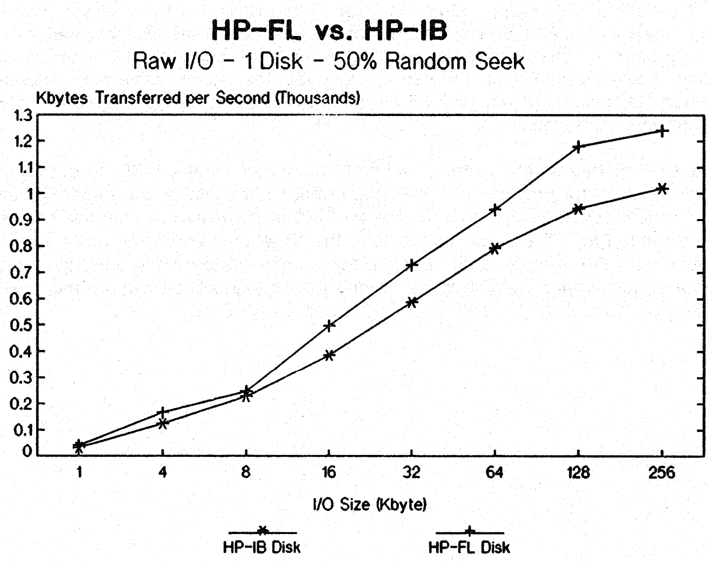


Figure 8.

It is evident that HP-FL offers a substantial performance advantage as the transfer size increases. At a transfer size of 8 kbytes and less there is only a small performance advantage over HP-IB. However, as the transfer size becomes larger, HP-FL outperforms HP-IB by a wider margin. This indicates that the biggest performance advantage with HP-FL is in environments where a substantial amount of disk I/O takes place and the majority of data transfers are large (greater than 8 kbytes).

Although Figure 8 gives a better indication of the performance potential of HP-FL, it is still a non-optimal view because the impact of better channel utilization is not fully represented. As more disks are placed on a single channel, the difference between HP-IB and HP-FL becomes even more dramatic. The HP-IB channel is already approaching its maximum burst rate while the HP-FL channel has 75% of its bandwidth remaining. This additional bandwidth will allow larger disk subsystems to increase their performance edge.

It is important to keep in mind that the MPE XL systems are just at the beginning of their life cycle. As the operating system matures and processing power increases, disk I/O demands are expected to increase. It is anticipated that increases will be seen in system I/O size as well as I/O traffic. This will place greater demands on the disk interface. HP-FL is the system interconnect that will meet the growing performance needs of these systems.

Summary

The Hewlett-Packard Fiber-Optic Link has clearly been designed to meet the growing performance needs of the HP-PA systems. All of the enhancements incorporated into the interface including increased transfer bandwidth, efficient channel utilization techniques, transaction pipelining and seek reordering work together to optimize performance. The unique characteristics of HP-FL make it an excellent solution for I/O intensive applications that perform large transfers.

The impact HP-FL has on the system level performance of today's MPE XL systems is typically small. This is not because of HP-FL performance limitations, but rather the excellent job MPE XL does reducing disk I/O. As the MPE XL systems evolve and disk I/O demands increase, the need for HP-FL will increase and the benefits will become more apparent. In combination with the numerous other advantages including large disk configurations, long cable lengths, few environmental concerns and new configuration opportunities, HP-FL is well positioned to meet the increasing storage needs of MPE XL systems.

HP's X.25 Private Packet Network - Statistical Reporting

Lisa Maldonado
Hewlett Packard Company
2000 S. Park Place
Atlanta, Georgia 30339

Networking, the buzz word of the 80's. Technological developments in the area of networking have skyrocketed over the past few years. For the international corporate world, distributed processing and shared resources are a requirement for optimum productivity and increased profit. The network becomes the "backbone", the artery by which all information and communication flows within a corporation's body. No matter what type of information is passed, what method or protocol is used, all networks must provide management and reporting capabilities. More specifically, these networks must provide for configuration and security changes, monitoring and tuning facilities, accounting or usage statistics gathering, and have expedient problem detection capabilities - all functions of a network management process.

If a network is to be managed effectively, it must provide a continuous record, of all network activities, that can be accessed and reviewed. Hewlett Packard's Private Packet Network is one network which provides this functionality. The accumulation of this data, and subsequent reporting of it, provides one of the prime tools for continuing decision making during the management process. Statistical reporting of network backbone utilization, error conditions, and individual access link usage gives the manager information necessary for growth planning, preventive maintenance, documentation, and trend analysis.

HP PPN uses X.25 packet switching technology, designed to provide highly reliable packet transmission and instantaneous call re-routing capabilities. The Hewlett Packard Private Packet Network management and control system allows on-line configuration management and network monitoring, including a process which records information such as network traffic, component activity, and link error data.

As important as the real-time display of information is in a Network Management System, the record of the network's configuration and it's activities are equally important. The HP PPN Network Control System, running on a HP 9000, uses two types of file structures for data storage. The first is a relational database management system. The second employs basic system file structures for circular logfiles. The databases store all of the network configuration information, comprised of database tables for each network component characteristic. For example, one table contains information pertaining to network addresses. Another contains X.25 frame and packet level parameters, and so on. The logfiles store historical accounting, security, statistical, and error information generated by the network on a periodic, configurable basis.

Wide area networks are not static entities. As new parts are deployed and existing pieces are modified it becomes extremely important to Network Managers to have up-to-date documentation reflecting the network's configuration. At Hewlett Packard, in working with our own internal Private Packet Network, we developed a procedure, accessing the Configuration Database to produce a current network configuration document. The configuration manual contains these elements:

- NCP parameters
- Operator types
- Event type definitions
- Node definition table
- Supervisory Network Control tables
- Backbone Link profile table
- Foreign network definitions
- Specific Node definitions
    - Node equipment configurations
    - Cluster operational values
    - Load Level groups
    - X.25 Interface profiles
- X.121 Address correlations

This report provides a tremendous resource for all network support personnel, including Network Managers, Operators, and Engineers . The Network Configuration Manual is a report that can be produced using the database information available.

The second type of data structure employed by HP PPN is the historical logfile. There are seven types of logfiles stored at the Network Communications Processor (NCP). They are:

Call Record Log - Call records are written when a call is initiated, at configurable time intervals while the call is active, and when the call is completed. It includes X.121 addresses, start and stop time, characters transmitted, and call route.

Event Log	- Various components, including the NCP, generate network events which are written in this logfile. They are also printed for a hard-copy record of network activity. This file records events such as a status change of components, access lines or backbone links.
System Statistics Log	- Network components transmit statistics records at configurable time intervals - usually every hour. Components recording information are the Network Communications Processor, Packet Switching Clusters, and the Auxiliary Service Processor. Link level and physical level errors, number of frames transmitted, and data packet sizes are some of the information recorded.
Pad Statistics Log	- This logfile contains responses from statistics poll requests to the HP2334/5 multiplexor, from the master NCP. It keeps a record of whether or not a given mux was responding, including a timestamp
Error Log	- Hardware and software errors detected by the NCP are contained in this file. The error log only records information about the NCP.
Trace Log	- This file contains information about software modules within the Network Control System. This logfile is used in conjunction with the Error Log, by HP personnel, for problem resolution.
Operator Log	- This logfile contains both Network Operations Console (NOC) commands and NCP console commands. Commands which modify the configuration or status are logged. This file provides an audit trail of network changes.

The logfiles used for Management Reporting are the call record log, the event log, and the statistics log.

The CALL RECORD LOG ...

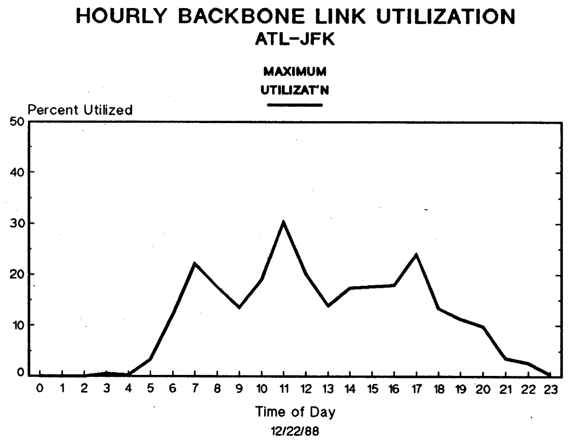
Contains a plethora of information for all calls in the network, including user and control systems calls. This provides the Network Manager information necessary to get a true picture of network usage. The call record provides call volume information, expressed in characters sent and received, between two ports. This data is the basis for providing meaningful reporting of traffic between two systems or users. Also, in each call record the route (which backbone facilities are used) that each call takes from source to destination is given. This data contributes to the calculation of bandwidth used on each backbone link.

An example of statistical information derived from the call record is the Hourly Backbone Link Utilization report. A detail report would provide a daily usage for a given backbone link, showing hourly call volume, as expressed in characters sent and received, for a 24 hour period; percentage of configured bandwidth is also given. This is calculated by dividing the number of characters transmitted by the capacity of the line:

(Baud/8 X seconds per hour X minutes per hour)

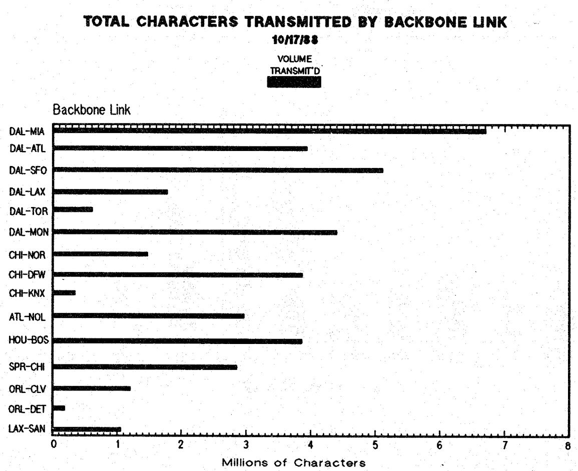
Example: 9600/8 x 60 x 60 = 4,320K char. capacity per hour.

Many variations of detail reports can be written. However, graphical representation makes it a bit more effective:



This display of information allows the Network Manager to quickly pinpoint time periods in which the amount of traffic may be exceeding a backbone link's optimum capacity.

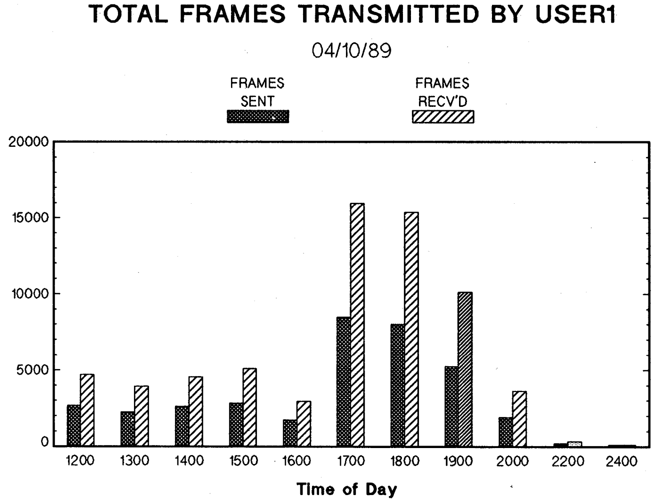
Another key graphical representation of information derived from the call record data might show a comparison of total traffic over all backbone segments:



In this graph, you would quickly see that some backbone links are more heavily utilized than others. You might question if perhaps calls could be re-routed for a more even distribution of call load. This graph highlights an uneven distribution among multiple links. The ability of a reporting system to present both detail and graphical call record data provides the tool for revealing potential problems, allowing changes to be made before the network performance is affected.

Traffic flow through an access link can be reported using data from the source and destination ports. End user accounting information can also be derived from the call record data. A company can choose to bill users based on total connect time, using start and end time, percent of bandwidth utilized, or total characters transmitted. (Public Data Networks currently use their equivalent of the call record for billing their customers.) Similarly, companies who have private networks, such as Hewlett Packard, choose to bill their internal "customers" as well. Therefore, for those HP PPN customers for whom billing is an issue, the call record log provides ample data necessary for full reporting of usage for accounting charge-backs.

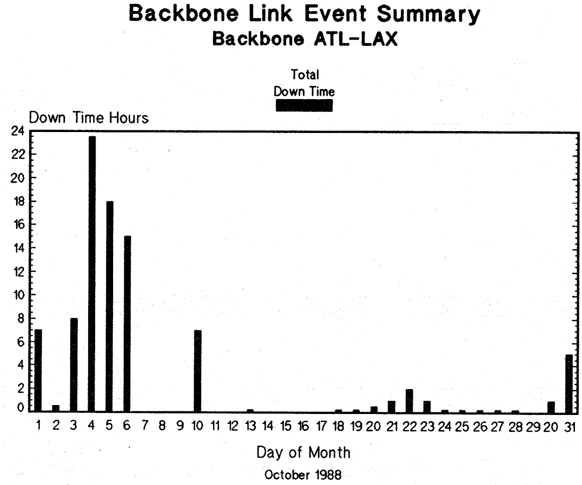
An example of usage data derived from an access link is shown in this Total Frames Transmitted graph:



The EVENT LOG ...

Records all significant network activity, as it occurs, and stores it at the NCP. Each network component is polled for status, and an event occurs when that status changes. Some events are more severe than others and require immediate attention. These events trigger an alarm on the Network Operator's Console (NOC) so that expedient trouble-shooting may begin. In a real-time situation the event provides the trigger for real-time action. In a static mode the event can trigger action of another kind. One example of an event to be monitored, over time, is what HP PPN calls a "Cluster Restart". This event, if it occurred once or twice does not constitute a problem. But, if through a weekly report of cluster events, for instance, you see that this component is getting many "cluster restart" messages per day, you would be able to identify that the cluster is having a problem. Network Managers also want to know how many times a component has gone down, and for how long. In the case of a backbone link this can be particularly important because these links are usually provided by a telephone company vendor. A large network may have several different line vendors. Therefore, a backbone link downtime summary could highlight which vendor's lines are having the most problems.

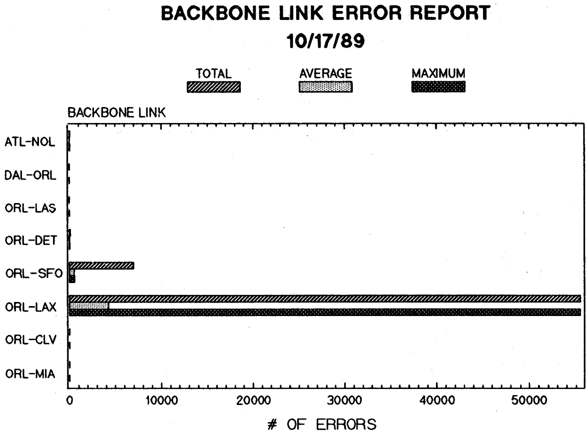
This graph shows a monthly summary of total downtime hours for a given backbone link:



The SYSTEM STATISTICS LOG...

Contains the most diverse data of all the logfiles. Statistics information is transmitted from each Packet Switching Cluster (PSC), providing information about the Line Interface Module (LIM). The Network Control System components, the Auxiliary Service Processor, and the Network Control Processor also report statistical information. The LIM statistics record is the one on which we will concentrate. Both physical level and link level statistics are given about each port on the LIM, including items such as number of CRC errors incurred, loss of carrier detected, and number of rejects transmitted. These types of errors are the ones that can cause degradation of network performance and should be reported regularly. Statistics are given for each port and they are further identified as backbone link or access link ports. A report that can be derived from this data is a Backbone Link Error Summary Report, showing the number of each physical and link level error reported in each hour of the day. For example, excessive CRC errors can slow down a user's response time because data frames require re-transmission when an error occurs. When a link is identified as having a high error rate, the Network Manager may infer that user calls should be re-routed through an alternate link or port, until the cause of the CRC errors can be corrected.

For example, a high CRC error rate link can be graphically displayed like this:



In addition to providing error counts, the Statistics Log also provides port usage information. Where the Call Record Log provides the number of data bytes/ characters transferred, the Statistics Log records the number of frames transmitted and a count of the X.25 data packets within specific size ranges. The count of frames sent and received identifies those ports that are most heavily used; an uneven distribution of user traffic can be determined from this. An average character count, within data packet size, can display what type of data, batch or interactive, is typically being transmitted by which user groups. An inference that the Network Manager can make is at what time of day the larger packets, batch traffic, are being transmitted, possibly impacting the flow of interactive traffic between users.

HP PPN provides for transfer of logfile data to an external CPU for offline processing. This connection is via a Local Area Network. There are many ways that user defined reports can be created. Statistics packages are available which can use the log data as input and produce summary, as well as detail, reports. Many of these packages provide attractive graphics presentation, for ease of management review. Fourth Generation Languages, such as HP BRW and the RAPID products, provide a versatile method of report writing, providing for timely modification. They can also create output files which easily become input to graphics applications. Or, for Network Managers who need detailed, numerical statistics, COBOL or PASCAL may be a more appropriate reporting tool.

Effective reporting, as shown, can provide usage and error statistics and documentation essential to quality Network Management. Intelligent decisions can then be made regarding future configurations and hardware acquisitions. Potential network problems can be avoided. Hewlett Packard's Private Packet Network provides a record of network activity, giving the manager the tools for configuration and security changes, network performance tuning, accounting data, and problem resolution.

Lisa Maldonado is currently an Analyst for Hewlett Packard's Customer Network Center, in Atlanta, Georgia; she has been with HP for 2 years. during this time, she has obtained extensive experience with HP's X.25 Private Packet Network, specifically in the area of statistical reporting. She has designed a reporting package for customers who have a HP PPN. Prior to coming to work for HP, Ms. Maldonado worked in the user community, for 6 years in the Systems Analysis and Design area.

Introduction to Capacity Planning

Dr. Arnold O. Allen
Performance Technology Center
Hewlett-Packard
Roseville, California

April 27, 1989

"I don't know what you mean by 'glory,' " Alice said.

Humpty Dumpty smiled contemptuously. "Of course you don't—til I tell you. I meant 'there's a nice knock-down argument for you!' "

"But 'glory' doesn't mean 'a nice knock-down argument,' " Alice objected.

"When I use a word, " Humpty Dumpty said, in a rather scornful tone, "it means just what I choose it to mean—neither more nor less."

"The question is," said Alice, "whether you can make words mean so many different things."

"The question is," said Humpty Dumpty, "which is to be master—that's all."

Lewis Carroll
Through The Looking Glass

In his classic paper Denning [7] Peter Denning claimed that, "We demand good performance but we do not have clear notions of what good performance is, or how to tell when it has been achieved." He said that the approach was SDRAWKCAB (backwards). A great deal of progress has been made since 1973. Many new tools have been developed to assist in performance management and capacity planning. Not only do we now have a much better understanding of what good performance means but Information Systems at some companies spell out in a Service Level Agreement a performance level that will be provided. Deer[11] describes Service Level Agreements as follows:

> Service level agreements are contracts between data processing and the end user that establish mutual responsibilities for service to be provided. Data processing is responsible for providing the following:
>
> - The agreed-upon service (response time, availability, etc.)
> - The measurement and reporting of the service provided.
>
> To receive the contracted service,

the end user agrees to certain volumes and mix of work. For example, a user would agree to:

- Provide input by a specific time of day
- Limit the number of terminals active at any one time
- Not exceed a given load level during a specific interval of time

If these and other stipulations are exceeded or not met, then service cannot be guaranteed and many, in fact, be degraded.

Capacity planning should not be confused with performance evaluation, performance management, or tuning. Performance management is sometimes used as a euphemism for tuning (by true believers in the theory enunciated above by Humpty Dumpty) but more often as an entity that includes capacity planning as a subset. Tony Engberg, the *Managing Performance* columnist for *The HP Chronicle* (see Engberg [12]) uses *Performance Management* as an umbrella term to include all aspects of computer performance.[1] He says:

> Performance management covers a significant portion of the world of performance. It is defined by the set of knowledge and techniques required to maintain and predict system performance levels within the context of your entity. This a broad statement, for the "system" could be a network incorporating the machines of multiple vendors, and your "entity" might range from a small business to a department within a large company to the capacity planning function of a corporation.
>
> Performance management can be broken into four categories: system management and operations, diagnosis, application optimization (including design and tuning) and capacity planning. Each is easily defined, at least intuitively. System management and operations encompass such areas as system tuning, optimal utilization and scheduling of existing resources, performance contracts (e. g., user response time contracts, resource utilization tracking and billing) and load balancing (across devices, SPUs or systems).
>
> Diagnosis deals with the evaluation of the causes of performance problems, such as degraded response time or fluctuations in throughput.
>
> Application optimization embraces system performance engineering (SPE) and tuning. An "application" can be considered equivalent to a "system"; concentration upon the design side of application

[1] Tony Engberg is also the manager of the Performance Technology Center for Hewlett-Packard in Roseville, California.

optimization is the single most effective means available for decreasing the time spent in reacting to performance problems.

Finally, capacity planning (CP) takes in the predictive side of system performance management. CP can be subdivided into two parts: one that focuses on the effects of changes to the current hardware and/or applications workloads, and one that attempts to predict what will be needed in the way of hardware and/or software in order to maintain acceptable performance levels as the business evolves.

My favorite definition of capacity planning is that given by N. C. Vince [10]. He says:

> Capacity planning is a means whereby one can achieve meaningful forward estimates of the resources needed, both hardware and software, relative to the demand expected to be imposed by the workload.

If you think this definition sounds like the sort of thing an English gentleman would say, you'd be right—Nick Vince *is* an English gentleman. Note that capacity planning has to do with the future, that is, not with immediate day to day activities but with what is going to happen in six months or a year from the present.

I believe there are seven aspects of a successful capacity planning program.

1. An orderly view of the system.

bottom line *n. The 24th line on a typical VDU, reserved for error messages. This convention is also used on balance sheets and other financial reports.*

Stan Kelly-Bootle
The Devil's DP Dictionary

I believe we should take an orderly, planned, approach to every endeavor and avoid being "crisis or event driven." Just as Wayne Dyer [13] says we should take responsibility for our own lives, I believe we should take responsibility for the orderly operation of our computer facilities. To accomplish this goal a carefully thought out plan is essential. Such a plan must have checkpoints and controls.

2. A statement of current workload.

"Cheshire Puss," she began, rather timidly, as she did not at all know whether it would like the name: however, it only grinned a little wider. "Come, it's pleased

so far, " thought Alice, and she went on. Would you tell me, please, which way I ought to go from here?"

"That depends a good deal on where you want to get to," said the Cat.

I don't much care where—" said Alice.

The it doesn't matter which way you go," said the Cat.

"—so long as I get somewhere, *" Alice added as an explanation.*

"Oh, you're sure to do that," said the Cat, "if you only walk long enough."

Lewis Carroll
Alice's Adventures in Wonderland

Before we can plan for the future we must understand clearly where we are today. As part of this effort workunits and workloads must be carefully defined in terms that are meaningful both to the end user and the capacity planner. Bronner [14] says

> A workunit is defined as an externally generated fixed quantity of work to be accomplished by the computing system (e. g., 1 batch job, 1 inquiry, 1 command). The workload is the number of workunits processed by the system during a specific period of time (e. g., 100 batch jobs/hour, 10 transactions/second, 2 commands/second). In any given installation it may be possible to define several types of workunits.

As Ferrari et al.[9] point out, the problem of workload characterization is one of the most difficult parts of successful capacity planning. Fortunately, tools and techniques for measuring and characterizing workloads are improving.

3. A measurement of present performance and resource consumption.

When you can measure what you are speaking about and express it in numbers you know something about it; but when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind.

Lord Kelvin

Devising a measurement strategy for assessing the actual performance and utilization of a computer system and its components is an important part of capacity planning. Users of Hewlett-Packard MPE V systems may use HP GLANCE/V [20], OPT/300 [21] or HP LaserRX [19] to measure the performance of their systems. Users of Hewlett-Packard MPE XL systems can use GLANCE/XL or HP LaserRX to find out what is happening performance wise with their systems. For UNIX systems see Glover et al. [22]

4. A projection of future workload(s).

Heaven from all creatures hides the book of Fate.

Alexander Pope

One of the major goals of capacity planning is to be able to install upgrades in hardware and software on a timely basis to avoid the "big surprise" of the sudden discovery of the gross lack of system capacity.[2] To avoid a sudden failure it is necessary to project future workload(s).

5. A prediction of expected performance.

Never make forecasts; especially about the future.

Samuel Goldwyn

To avoid the "big surprise" of (4), above, it is necessary to predict how the current system will perform with the predicted workload so it can be determined when upgrades to the system are necessary. The discipline necessary for making such predictions is *modeling.*

6. An evaluation of future configurations.

kludge *n. & v.trans., also called* **kluge** [Yiddish *klug* "smart."] **1** *n.* The programmers' vaseline. **2** *n.* A step in a STEPWISE REFINEMENT. **3** *n.* [From JARGON FILE] Something that works for the wrong reason. **4** *v.trans.* To evade the main issue by applying a kludge (to a problem). *See also* BUG; ONE-LINE PATCH; PTF.

Stan Kelly-Bootle
The Devil's DP Dictionary

For successful capacity planning it is necessary to be able to perform a performance evaluation of possible computer system configurations with the projected workload. This is another capacity planning function that requires modeling technology. Boyse and Warn [6] provide one of the first documentations of the successful use of analytic queueing theory models to evaluate the possible configuration changes to a computer system. They were able to evaluate configuration changes such as larger memory, faster I/O devices, and faster CPUs.

---

[2] Some would emulate Vince Lombardi's statement that "Winning is the *only* thing" to conclude that this is the only reason for doing capacity planning.

7. An ongoing management process.

Son, no matter how far you travel, or how smart you get, always remember this: Someday, somewhere, a guy is going to come to you and show you a nice brand-new deck of cards on which the seal is not yet broken, and this guy is going to offer to bet you that the Jack of Spades will jump out of this deck and squirt cider in your ear. But, son, do not bet this man, for as sure as you do, you are going to get an ear full of cider.

Damon Runyon

A philosophy that we strongly urge at the Hewlett-Packard Performance Technology Center is to "Keep on tracking." It is important to document all assumptions made in performance predictions. It is also important to regularly compare predictions to actuality and account for deviations. In this way we can improve our performance predictions—or find someone else to blame in case of failure. The key entity that is needed is a performance data base. Users of Hewlett-Packard MPE/V or MPE/XL systems can use HP LaserRX.

Figure 1 illustrates the ongoing process of computer capacity planning. The box labeled "Performance" illustrates the fact that we need to measure and analyze our system on a regular basis to determine what is happening and how performance and the workload is changing over time. (John Cunneen, a former colleague, called the methodology I am describing UMPV for Understand, Model, Predict, and Validate.) Certainly measurement and simple analysis is part of the understanding of a system but some kind of model is usually required to gain the level of understanding necessary for capacity planning. Of course we need not only to measure the performance of our systems but also to keep a data base of the history of each system.

The box labeled "Service Levels" reminds us that service levels are important. There has been some controversy over the alleged need for subsecond response time for interactive systems. The August 1988 issue of EDP Performance Review provides a good review of this subject. The compleat capacity planner will consider the needs and desires of the users in planning service level requirements. The users also need to be consulted to obtain growth projections and to provide feedback on how well the computer system(s) are satisfying their needs.

The service level requirements, together with data from the capacity planning data base are needed to drive the modeling effort. Modeling is such fun that we sometimes need to remind ourselves that it is not an end in itself but only a tool to use for the capacity planning effort. As outlined by Lazowska et al. [5], the first step in a modeling study is to validate the model. This is usually accomplished by: (1) Using the measured parameters from the actual computer system as parameters to set up the model. (2) Running the model and comparing the performance metrics from the model to those from the measured system. The model is considered validated if the results are sufficiently close. Lazowska et al. [5] state that device utilizations should agree within 5 to 10

percent while the response time discrepancy should not exceed 30 percent if queueing theory models are used for the modeling effort. Once the model has been validated it can be used to predict system performance with different configurations and workloads. We will discuss modeling in more detail below.
The "Management Decisions" box indicates some of the management decisions that can be made based on the use of the model. Thus decisions can be made on what growth projections to use, what configurations to investigate, and what service requirements must be met.
As we show in Figure 2, there is a spectrum of modeling techniques available for modeling computer systems. The simpler ones, such as rules of thumb, are easy to apply and require few resources. The complexity and resource requirements increase dramatically and nonlinearly from left to right. The rightmost technique, benchmarking, is extremely costly as well as very difficult to apply. However computer manufacturers *must* perform benchmark modeling as must some government agencies. For more about benchmarking see my paper Allen [3].
Almost everyone who uses complex modeling techniques such as queueing theory modeling and simulation uses rules of thumb as well. The best rules of thumb, of course, are ones that are developed at a particular installation by exercising a more sophisticated modeling technique. Some rules of thumb which have obvious applications to computer system performance evaluation are listed below.
Almost everyone involved with computers knows that Murphy's law in one or more of its forms applies. We state some of the most common versions.

MURPHY'S LAWS

1. In any field of scientific endeavor anything that can go wrong will go wrong.
2. Left to themselves, things always go from *bad* to *worse.*
3. If there is the possibility of several things going wrong, the one that will go wrong is the one that will do the most damage.
4. Nature always sides with the hidden flaw.
5. Mother nature is a bitch.
6. If everything seems to be going well, you have obviously overlooked something.

Anyone who has used a computer has probably verified all of the above versions of Murphy's law and will agree with the O'Flaherty's Corollary, as well.

O'FLAHERTY'S COROLLARY: MURPHY WAS AN OPTIMIST

The message we should get from studying Murphy's law is that we must be on the lookout for all the things that can go wrong in a computer system.
A truism due to Dr. Tom Bell that is often overlooked is that "all CPUs wait at the same speed."
A more conventional collection of rules of thumb are provided by Harry Zimmer [17]. Among them are:

1. The maximum number of active users on a system at any one time is equal to the number of user-id's divided by two.

2. Personal Computers are assumed to behave as terminals, but will consume 25% more than terminal in CPU utilization.

Linear projection is a very intuitive method of modeling. We all tend to think linearly. Many believe that they would be twice as happy if they had twice as much money. This is refuted by another who claims money does not make people happy. As proof he notes that a person with 9 million dollars is no happier than one with only 8 million. Daniel Seligman, a columnist for Fortune Magazine, claims is has been proven conclusively that money does increase happiness but that the formula is *not*

$$H = K \times M,$$

but rather

$$H = K \times M^{1/3},$$

where H is happiness, K is a constant and M is the amount of money that someone has.
A true linearist who is studying an interactive system with 100 users and measures the CPU utilization as 50% would immediately conclude that 200 users could be supported. The only problem with this is that the response time with 200 active users may lead to posthumous responses in many cases. However, linear extrapolation is not all bad. In fact a very large computer company, sometimes called Big Blue, has a capacity planning methodology called USAGE (for Understanding Your Application and Growth Environment) described by Cooper [8] which is, essentially, linear projection. Rules of thumb are used to ensure that response times do not get too far out of line.
Another very useful modeling technique is the back of the envelope method. For example, to see if a proposed computer system will handle the estimated I/O requirements, one merely estimates the I/O capacity of the new system by simple estimates. This important modeling technique is discussed in some detail in Allen [4], Paulos [15], and Harte [16]. My examples are all computer capacity related while Paulos discusses important real-life situations such as the probability of contracting AIDS or being on a hijacked airliner. Harte discusses environmental issues.
As we mentioned above, benchmarking is so difficult that it is not recommended for most computer installations. That means that simulation and queueing

theory modeling are the modeling techniques most used by most installations that have a serious modeling effort. In Figure 3 we indicate the tradeoffs in the two techniques. Most current modeling packages use analytic queueing theory models. For a more detailed discussion of the tradeoffs and of simulation modeling see my self-study course Allen [2].
Bronner [14] advocates the use of simple single-server queueing models to model the bottleneck areas of a computer system. The model most often used in these kinds of studies is the $M/M/1$ queueing model which was developed by A. K. Erlang in 1917 to study telephone systems (Erlang worked for a telephone company in Copenhagen, Denmark). This model and others developed by Erlang have been used successfully by telephone companies for years and by computer modelers for twenty years. Let us consider a simple queueing theory model.

In Figure 4 we illustrate the key elements of a queueing[3] system. A queueing system is a system with a service facility and customers who need the service that is provided by the facility. The "customers" in the queueing systems which are of interest to us include interactive requests for service, I/O requests, requests by jobs for CPU service, etc. In the service facility a "server" is an entity that can service a customer request. Thus a server could be a disc drive, a CPU, or the whole computer system. We assume there are one or more identical servers in the service facility. If all the servers are busy when a customer arrives, the customer must join a queue (waiting line) until a server is available.
Some key elements of a queueing system which have an impact upon the performance of the system include:

- the population or source of potential customers
- the arrival pattern of customers into the system
- the service pattern of the servers
- the number of servers
- the queue discipline

As mentioned above, the most widely used queueing model is the $M/M/1$ model. The notation for this model is due to David Kendall, the British queueing theorist. It means that the time between successive arrivals of customers follows an exponential distribution (the probability distribution that queueing theory thrives on), as does the service time of the single server (married servers are not allowed in Kendall's systems). The exponential distribution has pleasant mathematical properties and is ubiquitous in that mythical place often called the "real world."

[3]Queueing is the only word in the English language with five vowels in a row. Queueing theory aficionados deeply resent the cretins who use the obscene spelling "queuing" for this beautiful word.

The primary equations of the $M/M/1$ model are:

$$\rho = \lambda W_S,$$

where ρ is the server utilization or fraction of time the server is busy, λ is the average customer arrival rate, and W_s is the average service time. The average time a customer spends in the queue waiting for service is given by

$$W_q = \frac{\rho W_S}{1-\rho}.$$

The average time a customer spends in the queueing system ($= W_q + W_S$) is given by

$$W = \frac{W_S}{1-\rho}.$$

We are ready to consider an example of the use of this model. Suppose the analysts at Weirdo Engineers have modeled a bottleneck I/O system as an $M/M/1$ queueing system. They anticipate that, within three months, the system must support 1,200 I/O requests per minute with an average service time of 30 milliseconds. Then

$$\lambda = \frac{1200}{60} = 20 \quad \text{I/O requests per second,}$$

and thus

$$\rho = \lambda \times W_S = 20 \times 0.03 = 0.6.$$

Therefore

$$W = \frac{W_S}{1-\rho} = 75 \quad \text{milliseconds.}$$

Clearly, this load would be too high. Either a faster I/O system is needed or two of them is required.

In my talk I will give other examples of the use of queueing theory models and an example of a back of the envelope study.

References

[1] Arnold O. Allen, *Probability, Statistics, and Queueing Theory With Computer Science Applications*, Academic Press, Orlando, 1978.

[2] Arnold O. Allen, *Introduction to Computer Modeling*, Applied Computer Research, Phoenix, AZ, 1986. (This course uses my book and the book by Lazowska et al. [5] as textbooks. In addition to the textbooks a Study Guide and an HP11C calculator are included with the course.)

[3] Arnold O. Allen, "Benchmarking for the Beginner," *EDP Performance Review*, January 1989.

[4] Arnold O. Allen, "Back-of-the-envelope modeling," *EDP Performance Review*, July 1987.

[5] Edward D. Lazowska, John Zahorjan, and G. Scott Graham, *Quantitative System Performance: Computer System Analysis Using Queueing Network Models*, Prentice-Hall, Englewood Cliffs, n. J., 1984.

[6] John W. Boyse and David R. Warn, "A Straightforward Model for Computer Performance Prediction", *ACM Computing Surveys*, **7(2)**, June 1975.

[7] Peter J. Denning, "Why Our Approach to Performance Evaluation is SDRAWKCAB", *Performance Evaluation Review*, **2(3)**, September 1973.

[8] J. C. Cooper, "A Capacity Planning Methodology," *IBM Systems Journal*, **19(1)**, 1980.

[9] Domenico Ferrari, Giuseppe Serazzi, and Alessandro Zeigner, *Measurement and Tuning of Computer Systems*, Prentice-Hall, Englewood Cliffs, N. J., 1983.

[10] N. C. Vince, "Establishing a Capacity Planning Facility", *Computer Performance*, **1(1)**, June 1980, 41-48.

[11] David C. Deer, *Capacity Management Glossary*, National Advanced Systems, 1983.

[12] Tony Engberg, "Managing Performance," *The HP Chronicle*, November 1988.

[13] Wayne W. Dyer, *Your Erroneous Zones*, Funk & Wagnalls, New York, 1976

[14] LeeRoy Bronner, *An Introduction to Capacity Planning,* IBM Washington Systems Center Technical Bulletin, GG22-9001, January 1977.

[15] John Allen Paulos, *Innumeracy: Mathematical Illiteracy And Its Consequences*, Hill and Wang, New York, 1988.

[16] John Harte, *Consider a Spherical Cow: A Course In Environmental Problem Solving*, William Kaufmann, Los Altos, 1985.

[17] Harry Zimmer, "Rules of Thumb", CMG '87 Conference Proceedings.

[18] *The Economic Value of Rapid Response Time*, IBM GE20-0752.

[19] *HP LaserRX User's Manual*, Hewlett-Packard 50700-90001.

[20] *HP GLANCE User's Manual*, Hewlett-Packard 5959-9236.

[21] *OPT/3000 ON-LINE PERFORMANCE TOOL Reference Manual*, Hewlett-Packard 32238-90001.

[22] Dave Glover, Sherri Osborn, and Chuck Smith, *Designing Performance Tools for the Expanding UNIX Environment*, These Proceedings, 1989.

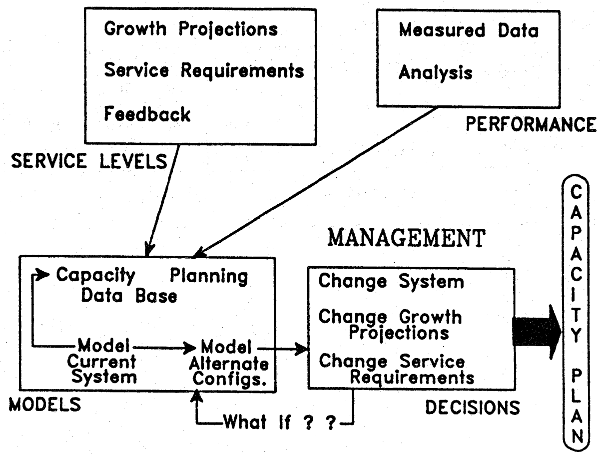


FIGURE 1. Capacity planning process

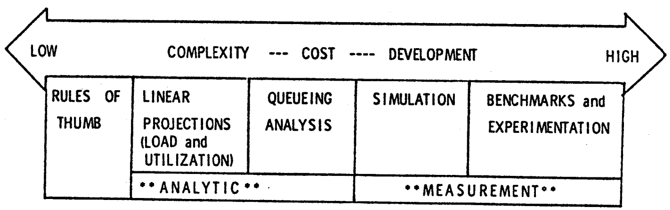


FIGURE 2. Spectrum of modeling techniques

Advantages	Disadvantages
Can model to arbitrary level of detail	Requires computer for even simple model
Can model extremely complex situations that are analytically intractable	Difficult to program
Provides estimates of distributions	Difficult to validate
Can study transient behavior of system	Requires statistical depth to design experiments and interpret results.
	Requires heavy computer and personnel resources

Pros and Cons of Simulation Modeling

Advantages	Disadvantages
Simple models can be calculated on programmable calculator	Cannot model logical decisions
Easy to set up model of computer system	Can only approximate complex systems
Product form models computed very fast	Cannot directly model scheduling
Many non product form models can be approximated	Cannot directly model simultaneous resource possession
	Cannot model some types of priorities

Pros and Cons of Analytic Queueing Theory Modeling

FIGURE 3. Tradoffs in modeling techniques

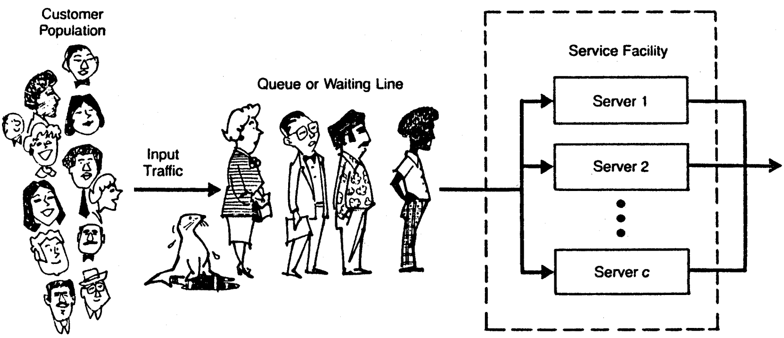


FIGURE 4. Elements of a queueing system

Security Assurance Review

By:
Shukri Abdallah
University of Maryland Baltimore County
5401 Wilkens Ave. Baltimore, MD. 21228
(301) 455-3838

In all the papers we've read about computer security, two points are stated over and over again:

1- The human element is the weakest link in computer security.
2- Most threats are internal.

To illustrate these points, we want to share with you a vulnerability that might exist on your system. When you install the HPOFFICE products, you include the account password, and a user password in some files that reside in the HPOFFICE account. Then you stream a job that compiles these files. One of these files is the standard message catalog for HPDRAW, SMCS108A.HPDRAW.HPOFFICE. The job file that compiles SMCS108A.HPDRAW.HPOFFICE is JMCJ108A.HPDRAW.HPOFFICE. For more information, see the COMMUNICATOR for UB-DELTA-2 pages 5-6 to 5-15.

* Step 1 - Log on to the system as an average user with AB, IA, ND, and SF capabilities.

* Step 2 - :LISTF @.HPDRAW.HPOFFICE,2

```
ACCOUNT=  HPOFFICE    GROUP=  HPDRAW

FILENAME       CODE        ------------LOGICAL    RECORD-----------
----SPACE----
                    SIZE  TYP        EOF       LIMIT R/B  SECTORS #X

SMCS108A             72B  FA         4596       4596  32     1305 13
```

* Step 3 - Using your favorite editor, text in the file.

```
:QEDIT

QEDIT. Copyright Robelle Consulting Ltd. 1977-1988.  Type ? for
help. (Version 3.7)


/T SMCS108A.HPDRAW.HPOFFICE
QEDITSCR
4596 lines in file
```

* Step 4 - Search for the word "HPOFFICE"

```
/F "HPOFFICE"
 4250     452 HPOFFICE  ROSTER                  X
```

* Step 5 - List the lines

```
/L 4239/
 4239     $ The following names indicates the user, group and
          account
 4240     $ where all the background jobs will be logging on to.
          The name
 4241     $ starts at column 1 and the corresponding password
          starts at
 4242 $ Column 11.
 4243 $              450 - User name and password
 4244 $              451 - Group name and password
 4245 $              452 - Account name and password
 4246 $
 4247 $   123456789012345678901234567890123456
 4248     450 MGR       DRAWING                 X
 4249     451 PUB       PUB                     X
 4250     452 HPOFFICE  ROSTER                  X
 4251 $
```

There you go! The password for the HPOFFICE account is ROSTER and the password for MGR.HPOFFICE is DRAWING. Now you can log on to the system as MGR.HPOFFICE and have OP capability. (See Figure 1)

To explain why you were able to read the file , see Figure 1 for account and group access control settings. For some reason, HP decided to set the account security to (R,W,A,L,X,S : ANY).

The human factor involved in this vulnerability is that the system manager neglected to do two things:

1- Eliminate the passwords from these files when the installation procedure is complete.

2- Change the user from MGR.HPOFFICE to GRAPHICS.HPOFFICE. The user GRAPHICS does not have OP capability.

* Step 6 - An average user logs on as MGR.HPOFFICE and causes some harm to the system.

Another way to achieve the same result, is to look for the compiled file which resides in the group PUB.SYS. This file is C01C108A.PUB.SYS. Issue the following FCOPY command:

```
:FCOPY FROM=C01C108A.PUB.SYS;TO=;CHAR;SUBSET=491:491
```

```
HP32212A.03.23 FILE COPIER (C) HEWLETT-PACKARD CO. 1984
C01C108A.PUB.SYS RECORD 491 (%753, #1EB)

00000:          XHARDWARE                                     XGALLERY
00044:            XMGR         DRAWING                      XPUB       PUB
00110:              XHPOFFICE   ROSTER                          X1
00154:                XO No KKA error File code
1 RECORD PROCESSED *** 0 ERRORS
End of Subsystem.
```

As you can see the passwords are stored in record 491 of the file.

To prevent anyone from logging onto the system as GRAPHICS.HPOFFICE, you have to install one security feature, personal passwords, on all users of the HPOFFICE account. Such a feature is available in third party security packages such as Security/3000, or OCS/Private, or SMASH.

This is but one example of potential vulnerabilities in a "standard" HP environment. To effectively secure the computing environment at your site, you have to develop a security plan. This plan should:

1- determine what needs to be secured.
2- strike a balance between securing the system, yet keep it accessible to the user community. That is to say, you don't install five passwords for a single user and expect the user to remember them all.
3- estimate the cost of implementing effective security relative to the worth of what you're trying to secure. In other words, you wouldn't spend ten thousand dollars to secure a PC that costs two thousand dollars.

The following issues are just some of the major concerns that a comprehensive plan should address:

01- Hardware security.
02- Special forms security.
03- Communications security.
04- Change CATALOG.PUB.SYS.
05- Job security.
06- Tape security.
07- Database security
08- Contingency planning.
09- Encryption.
10- Enhancing MPE security.
11- Third party packages.

1- Hardware Security:

- Keep your HP/3000 equipment in a secured area.

- Maintain a list of all the people who have access to the computer room.
- Install fire alarm and extinguishing systems.
- Keep two or three hand-carried fire extinguishers in the computer room in case of small containable fires.
- Install sensors under the raised floor to sense for water or flooding.
- Install a sensaphone that monitors the computer room for power interruptions, loud noise (i.e fire alarms), and overheating conditions. If any of these conditions occur, the sensaphone can dial any telephone number and leave a voice message describing the nature of the problem. The sensaphone can be purchased from RADIO SHACK.
- Install CALLBACK/3000 which has a similar function as the sensaphone. You can get CALLBACK/3000 from DESIGN/3000, INC. CALLBACK has the added capability of monitoring the successful completion or abnormal termination of any job.
- Install alarm systems on all doors that lead to the computer room.
- Ask the police or the security officers in your organization to conduct random checks on the computer room during off hours.
- Periodically test all your alarm systems.
- Ask the Fire Marshall to inspect your computer room for fire hazards.
- Keep your computer room clean.
- Make it a company policy that operators are not allowed to eat, drink, or smoke in the computer room.
- Lock all power closets that channel electricity to the computer room.
- Invest in some locks to fasten the terminals and PCs to the desks. This makes PC theft more difficult.
- Maintain an accurate inventory list of all the hardware you have in every building. This list becomes useful when you have a fire and the insurance company asks you for an estimate of the lost equipment.
- Keep this list up to date and remember that users keep moving equipment from one place to another.
- Use HPLIST or DBASE to help you track all this equipment.

2- Special Forms Security:

- Keep all your forms under lock and key.
- Track the usage of your forms. If you print payroll checks make sure that you have the needed controls to ensure that no checks are missing.
- Find a paper recycling company in your area and recycle all non-sensitive printouts. This should add some dollars to your budget.
- Shred all sensitive printouts that contain passwords or private information.
- Alert the user community to privacy laws. Encourage them to

recycle non-sensitive printouts, and shred all sensitive ones.

3- Communications Security:

- Determine port connectivity. That is, which ports are :

    - direct connect.
    - connected to modems/multiplexers.
    - connected to a data switch.
    - the dial in ports.
    - the dial out ports.
- Determine user connectivity. That is, which department uses what communication equipment? This exercise should help you trace the path the intruder followed to break into your system.
- Determine existing security features implemented on your communication equipment.

    - Does the communication equipment prompt the user for a password?
    - Do you have dial back modems?
    - If the dial in line drops, is the session aborting on your HP system?
    - Do you know who dials into your system?
    - When was the last time you changed passwords on the communication equipment?
    - Do you know all the sites that DS into your system?
    - Do you know all the users that DS out from your system to other systems?
- Install passwords on your communication equipment. Let the users supply a password before they see the colon prompt.
- Buy dial back modems.
- Buy modems with 2400 and 9600 baud. Usually these modems are more expensive than the 1200 baud modems. This will eliminate a large class of hackers.
- Test and ensure that if a line drops, the session is aborted (Ghost sessions).
- Frequently change passwords on communication devices.
- Maintain a list of who dials into your system.
- :DOWN and :UP devices as needed.
- Use the :ACCEPT and :REFUSE commands of MPE as needed.
- Identify users who DS into your system.
- Try to change the dial in telephone number on a regular basis if you can.
- Install TERMPASS/3000 of SECURITY/3000 on the direct connect lines. TERMPASS/3000 forces the user to supply a password associated with the LDEV.
- Use contributed library programs such as COMCHECK, and SECURE.
- Install LOGOFF/3000 of SECURITY/3000 or BOUNCER from the contributed library. These programs abort all inactive sessions.

4- Change CATALOG.PUB.SYS:

CATALOG.PUB.SYS is the file that contains the standard MPE error messages.

- Compromise user friendliness for security.
- Change all system error messages related to logging onto the system such as
  "EXPECTED HELLO, :JOB, :DATA"
  "EXPECTED [SESSION,] USER.ACCT,[GRP]
  to
  "INVALID"
- Some of the errors message that need to be changed are:
  1426, 1427, 1429, 1430, 1434, 1436, 1437, 1439, 3080, 3082
- stream the MINLOGON.JOB.SECURITY to change all logon errors in CATALOG.PUB.SYS.
- The procedure to change these messages is also outlined in the system managers manual.
- Make sure that you create a COLDLOAD TAPE that contains your modified catalog. Otherwise, when you COLDLOAD the system all your modifications will disappear.
- If you are a V-MIT user and have purchased HP MONITOR, then use SECCONF to enable the minimum logon interface.

5- Job Security:

- Issue Streams 10 only if you use jobs in your shop.
- Give BA capability to those who need it.
- Eliminate passwords from job streams.
- Don't leave third party installation jobs in PUB.SYS.
- Install STREAMX of SECURITY/3000.
- Buy a job management system (See last months article on job management systems).

6- Tape Security:

- Implement a tape management system to keep track of tape usage and life cycles.

- Safeguard the following tapes:

  - COLDLOAD tapes.
  - Disc Utility Subsystem (DUS)
  - Memory Dump tapes.
  - Zero Dump tapes and Daily Dump tapes.
  - Database logging tapes.

- Keep your old COLDLOAD tape for a period of time after your most recent MPE update.
- Don't mix old and new COLDLOAD tapes.
- Validate your stores and zero dumps especially before a reload.

- Erase tapes before discarding.
- Clean your tapes on a regular basis.
- Use TAPETEST.PUB.TELESUP to certify your tapes. If you get more than 10 tape errors, then throw the tape away.
- Clean the tape drives on a regular basis.
- Clean the heads of the tape drive before every zero dump.
- Keep the most recent zero dump at an offsite location in fire/water proof safes.
- Beware of users with OP capability. They can restore files from different accounts and groups into their own group using the LOCAL option.
- Beware of the fake restore. A fake restore is a program that simulates an MPE restore. The MPE store format is documented in the systems manager reference manual. Get a source listing of STAN, an INTEREX contributed library software.
- Make it a departmental policy that operators are the only ones to issue a restore command. If anyone else issues a restore command, then the operators should not reply to the request.

7- Database Security:

- Use Image and Turbo Image set level and item level security to your advantage. Turbo Image security is very effective.
- Enable logging on all your databases that are modified by the users.
- If you perform database logging to tape, then ensure that you use good quality tapes.
- If you log transactions to disc, ensure that the log file doesn't reside on the same disc as your database. This should minimize the risk in case of disk problems.
- Enable ILR and Rollback recovery.
- Design a recovery procedure before you find out that you have to recover and don't know how to do it.
- Use DBBEGIN and DBEND for logical transactions. This should help keep your database consistent at times of recovery.
- Set SUBSYSTEM to READ only or NONE depending on your application needs.
- Regularly perform capacity checks so that your programs don't abort because of full data sets.
- Run DBAUDIT/3000 by ROBELLE to generate audit reports from the log files.
- Run HOWMESSY/3000 by ROBELLE on your databases. HOWMESSY gives you an insight on tuning your databases.
- Investigate using the VESOFT VEOPEN routine in your programs. VEOPEN allows you to eliminate database passwords from your program files and enforces an access control file. The access control file contains restrictions by user ids, program files, access mode, etc.
- Build editing routines into your programs to ensure that all the data entered is good data. Remember that good data produces information.

8- Contingency Planning:

- Develop a comprehensive and structured approach to the possibility of disaster, then move on to other critical issues in your shop. Don't spend lots of time working on your disaster plan. At the same time do not be of the opinion that disasters don't happen.
- Identify threats that could cause harm to the continuity of operations at your site. A threat is any agent with the capability to reduce the effectiveness of the system, thereby negating management objectives. A fire is a threat.
- Define how vulnerable you are to the identified threats. A vulnerability is the opportunity available to a hostile entity to mount an attack. If operators smoke in the computer room and you don't have alarm or extinguishing systems, then you're vulnerable to the possibility of a fire in your computer room.

- Define what constitutes a disaster at your site. Is one day down time considered a disaster?
- Get some insurance.
- Identify critical applications.
- Find a backup site in case you need to relocate your operations to an alternate site.
- Write your disaster plan or buy one and tailor it to your installation.
- Train personnel and inform users in preparation for potential disasters.
- Test your disaster plan.

9- Encryption :

- Realize that there's a risk in encrypting your source code. Lose the key and lose the source.
- Keep track of your encryption keys.
- Don't make the encryption program available to anyone.
- Be aware of the possibility of internal personnel vengeance or retaliation.

10- Enhancing MPE security.

We feel that MPE security has been discussed extensively. Instead of reviewing it, we'd like to illustrate how you can enhance it.

To enhance MPE Security, there are certain basic things you can do:

- Select under Miscellaneous Configuration in the SYSDUMP dialogue, Change number of seconds to logon (MIN = 1, MAX= 500) to 90 seconds.
- Under I/O Configuration changes, Reply NO to DATA. This prevents people from using :DATA statements to logon to the HP system.
- Don't leave :STARTSESS with passwords in SYSSTART.PUB.SYS

without altering the security to
:ALTSEC SYSSTART.PUB.SYS;(R,W,A,L,X:CR).

- Change the home group of MANAGER.SYS to a different group than PUB. Every user on the system can read files in PUB.SYS.
- Keep PUB.SYS clean.
- Use a hardcopy console.
- Shred console printouts.
- Secure the files JOBACCT, JOBACCTB, JOBCUDC of BULDACCT program (a good contributed library program that recreates your accounting structure).
- Enable MPE logging.
- Make sure that log file numbers are consecutive. If a file is missing, find out why it is missing.
- Use SCANLOG.PUB.TELESUP to report all logon failures.
- Beware of AUTOALLOCATE. Consider this :

    - A program is prepped with PM capability.
    - It resides in an account with PM capability.
    - It resides in a group with PM capability.
    - Allocate the program.
    - Run the program. The program will run.
    - ALTACCT and exclude the PM capability.
    - Run the program. The program still runs although the account does not have PM capability.
    - Deallocate the program.
    - Run the program. It does not run.

  In other words, the loader is forced to satisfy the capability requirements only if the program is removed from the autoallocation table. This is an MPE bug.
- Lockword all MPE utilities.
- Eliminate all unnecessary accounts.
- Identify all third party accounts.
    - These accounts are read only accounts.
    - The system manager should be the only one to update these accounts.
    - Don't add users into these accounts.
    - Monitor account/group/user capabilities and access control settings.
    - Change default passwords on the account/users of these accounts.
    - Make sure that no PM files are released.
- Identify production accounts.
    - Keep track of new users added by account managers.
    - Ensure all accounts/users have passwords.
    - Frequently change account/user passwords.
    - Secure all user released files.

- Identify all users with SM, PM, OP capabilities using FINDCAP, a contributed program.
- Beware of the COMMAND intrinsic. If you disable a certain command using a UDC, the user can execute the command from

some subsystems. Consider the following example:

```
Stream !FN
Option LIST, NOHELP, NOBREAK
Comment not available

:RUN FCOPY.PUB.SYS
 > :STREAM MYJOB
  #J900
```

- Force users directly into applications using the LOGON option of UDCs.
- Buy a menu system and isolate your users from MPE.
- Beware of trojan horses. Do not load any program from outside your shop without looking at, and recompiling, the source code. After all, you don't know what kind of auditing and checking these programs are subjected to.

- Use third party security packages to :
  - Install terminal passwords.
  - Install a menu system and isolate access to MPE.
  - Enforce unique user ids. Use session name as part of the id.
  - Enforce passwords standards:
    - average users must have a password of at least five characters.
    - The system manager should have a password that is 8 characters long.
    - Force users not to use the same passwords over and over again.
    - Install password aging.
  - Enforce time of day, day of week, and terminal restrictions on critical applications.
  - Configure the number of responses or trials the user is allowed to attempt to logon.
  - Install a feature that displays the last date and time the user logged on, and number of failures since last logon.
  - Report violations to the console, system manager, and logfiles.
  - Frustrate intruders by downing ports or having a delay period if they fail to key in the correct password.
  - Install personal passwords.
  - Install a time out feature when entering personal passwords.
  - Logoff all inactive terminals.

We accumulated the above mentioned points through our years of experience and work in the HP environment. We hope that you find this information useful and helpful in securing your environment. Remember that people are the weakest link and that you have to strike a fine balance between effective security and providing the users with easy access to the information. If there are users in your organization who still post their passwords on the terminal, don't despair, but try to raise their level of awareness and outline their responsibilities. The road to effective security is a long one. Happy computing.

Figure. 1

```
:RUN LISTDIR5.PUB.SYS

>LISTACCT HPOFFICE
********************
ACCOUNT: HPOFFICE

DISC SPACE: 184793(S)             PASSWORD: **
CPU TIME: 128471(SEC)             LOC ATTR: %0
CONNECT TIME: 333(MIN)            SECURITY--READ:    ANY
DISC LIMIT: 400000(S)                       WRITE:   ANY
CPU LIMIT: UNLIMITED                        APPEND:  ANY
CONNECT LIMIT: UNLIMITED                    LOCK:    ANY
MAX PRI: 100                                EXECUTE: ANY
GRP INX PTR: %620
USR INX PTR: %621
CAP: AM,AL,GL,DI,OP,CV,UV,LG,NM,CS,ND,SF,IA,BA,PH,DS,MR


>LISTGROUP HPDRAW.HPOFFICE
********************
GROUP: HPDRAW.HPOFFICE

DISC SPACE: 1820(S)               PASSWORD: **
CPU TIME: 0(SEC)                  SECURITY--READ:    ANY
CONNECT TIME: 0(MIN)                        WRITE:   AL,GU
DISC LIMIT: UNLIMITED                       APPEND:  AL,GU
CPU LIMIT: UNLIMITED                        LOCK:    AL,GU
CONNECT LIMIT: UNLIMITED                    EXECUTE: ANY
FILE INX PTR: %13103                        SAVE:    AL,GU
MVTABX: %0                        PRIV VOL: NO
MOUNT REF CNT: 0
HOME VOL SET:
CAP: IA,BA,PH,DS,MR

>LISTUSER MGR.HPOFFICE
********************
USER: MGR.HPOFFICE

HOME GROUP: PUB          PASSWORD: **
MAX PRI: 150             LOC ATTR: %0
LOGON CNT: 1
CAP: AM,AL,GL,DI,OP,CV,UV,LG,NM,CS,ND,SF,IA,BA,PH,DS,MR

>E

END OF PROGRAM
```

Viruses — What They Are and How to Defend Against Them

Anita Doucet
Hewlett-Packard
19111 Pruneridge Avenue
Cupertino, California 95014

Abstract

As computers become more and more a part of our everyday lives, the threat of a system catching a virus becomes increasingly serious. The question is, how can we reduce the chances of being infected by a virus? In all commercial operating systems, there is a risk of being attacked by a computer virus that must be seriously considered and planned for by all Data Processing professionals.

This paper tries to educate you about what a virus is and the possible defense mechanisms that they can use against one in your computing environment. The different types of viruses are examined to see how they work, and how they can affect both program files and operating system functionality. We will look at what can be done to prevent the disclosure and corruption of data, and how to detect a virus attack should one penetrate your system's security. The discussion also recommends a few security policies that you can follow to protect your data processing investment.

What is a Virus?

Since the publicity involving the ARPAnet virus in November 1988, the user community has been more aware of computer security in general and computer viruses in particular. For most of us, there is often some confusion over what a virus really is. Most people are aware that a virus can be dangerous. Viruses can wipe data off a disk, corrupt files, and even alter executable code on a system. A virus can be so destructive that you may be forced to shut down your system to keep the infection from spreading. Though these characteristics are all associated with viruses, which attributes make a virus what it is? First let's look at the characteristics of some programs that are often confused with viruses.

Trojan Horses : The definition of a Trojan Horse has changed very little since the Greeks first presented their gift at the gates of Troy. A Trojan Horse program is a piece of software which has a known, desirable function. However, embedded within the code is an invisible function which gets executed unknown to the user. For example, a computer user may receive a utility program for which they have a legitimate use. When they execute the program, the hidden function uses their authorizations and performs services in their name that could cause a break in the system's security. The unsuspecting user is unaware that they may have opened a back door for the creator of the Trojan Horse. A back door is a piece of software that provides a way to get access to functionality provided by the software, without the normal security authorizations. In this fashion a virus also bears resemblance to a Trojan Horse.

Worms : A worm is a program which has the ability to relocate itself automatically. In most cases, a worm also has the ability to search out and access paths to other nodes of a system or network. It can then place copies of itself in these locations. The copying program then invokes the copy which then propagates itself somewhere else on the system or network. This reproductive process can continue until the worm program has virtually consumed all the disk space and processor time. The decrease in disk space and the increase in CPU utilization caused by the worm programs on a system or network may eventually result in the denial of services. This occurs because most of the systems' resources have been taken over to produce copies of the worm, and perform and other functions specified in the worms' programming.

Now, if worm and Trojan Horse programs aren't real viruses, what are? Dr. Frederick Cohen defines a computer virus as "a program that can infect other programs by modifying them to include a possibly evolved version of itself" [1]. The unique attribute of a virus is that it can attach itself to a program or other executable code. A virus resembles a worm because of its ability to reproduce itself. Unlike a worm, however, a virus will not invoke itself. Once attached to a host, a virus will simply wait until a user tries to execute the host code.

System Infiltration

A virus can spread through any system that shares, interprets, or retransmits information [1]. Research has shown that the systems most vulnerable to a viral attack are those with the greater numbers of users who have more contact with each other [4].

For a system to become initially infected, however, requires some form of external injection, and the injection is usually intentional. In most cases this attack is targeted at the masses by someone who wants publicity, or the thrill of knowing how much damage they can do. The more threatening attacks are targeted at a particular system or network, usually because someone wants unauthorized access to, or proprietary information from the system. The more subsystems and products on a system, the higher the chance of having a back door created or exposed as a result of a viral attack. The initial infection is most commonly created by a carrier [2] program. A carrier program can be produced in several variations, depending upon the intent of the creator. For example, the carrier could be a Trojan Horse, whose hidden function is to infect either a program, or the operating system itself. This is usually the method used for a mass attack, through a piece of contributed or public domain software. A carrier could also be a program whose only purpose is to infect the system. Then, once the initial infection occurs, the program may purge itself from the system, thus masking the original source of the infection.

How a Virus Works

Viruses spread throughout a system or network by using the access rights and permissions of each user that executes infected software. These access rights can be used in a viral attack to open up the system to unauthorized use.

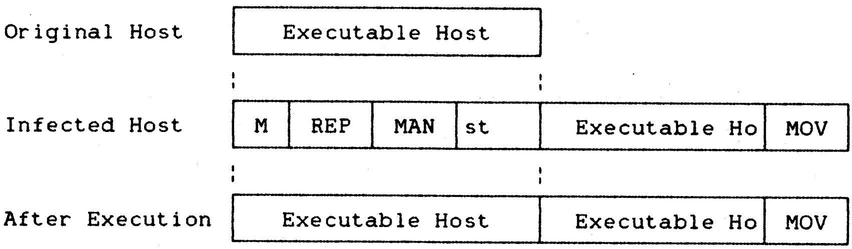
The code of a virus usually consists of three pieces, a marker (M), the reproductive routines (REP), and some extra manipulation functions (MAN) [2]. The marker is used by the virus' reproductive code to determine if a potential host has already been infected. This marker keeps the virus from wasting its time re-infecting a host. If a marker is present in a host candidate, the virus will ignore it and continue looking for an unmarked, or uninfected host. The function of the reproductive routines is to copy the virus and attach the copy to the host program. The manipulation routines perform additional functions, much like a Trojan Horse.

Most program or file level viruses are some variant of two basic forms, overwriting and non-overwriting viruses. An overwriting virus injects itself directly into a host by copying itself directly over the first few words of executable code. This usually destroys the original program so it can no longer execute properly.

```
Original Host  -->  |            Executable Host             |
                    :                                        :
Infected Host  -->  | M | REP | MAN |table Host              |
```

By placing the virus code over the front of the host program, the virus' instructions will be executed first the next time the program is invoked. Once the virus' reproduction and manipulation routines have completed, the original code following the virus tries to run. Of course, with the first few executable statements of the host missing, the program begins to print strange error messages, and will most likely abort. This type of virus is easily detected since the user is immediately aware that there is a problem. Of course, by then, another program has been infected.

The second form of virus is the non-overwriting variety. This type of infection will preserve the host program. It does this by writing an additional move (MOV) instruction into the host code [2]. First, the virus copies a portion of code of similar size to itself from the front of the host program to the end of the host. This copied code is then followed by the new move instruction. Next, the virus code copies itself over the start of the host program.



Again, by placing the virus code into the front of the host, the virus instructions will be executed first the next time the program is invoked. Once the virus' reproduction completes, the move instruction copies the first part of the host program back to its original position and executes a jump back to the original program start. The program begins to function normally. The user is unaware that anything unusual has occurred except perhaps a slight increase in the host's startup time. This type of virus is virtually undetectable except for the change in the size of the infected program.

At first an overwriting virus appears more dangerous than a non-overwriting. This is because the first type corrupts the host program while the second does not. However, the non-overwriting virus has the ability to remain in the system longer because its effects are not as immediately obvious. This makes the second form of virus potentially more hazardous, although it doesn't corrupt the host, it could leave the system open to intrusion and data corruption for years before ever being detected.

Operating system code is as susceptible to a viral attack as any user program. In fact, system code is most likely the real target if there is an intent to get sensitive information or system control.

Defenses against virus attacks can be divided into two categories : detection and prevention. Detection methods are aimed at alerting the user to the presence of an infection which may limit the scope of virus' spread. Preventive defenses try to inhibit or stop a virus' reproduction and infection completely.

Detection Mechanisms :

One indication of a possibly infected system is a noticeable drop in the expected performance of a program or system. One sign may be an overload or an unusually heavy use of the network or operating system services. Though these observations may be signs there is an attack on the system, there are several methods of detecting a viral infection that are more accurate.

Prefix, Postfix, and Pattern Matching Algorithms : These viral detection programs compare the first or last few words of an executable file. If several files in a group start or end with the identical instructions, the chances are high that a virus has infected those files. Think back to the description of how a virus propagates and you will understand why this detection mechanism works. A file infected with a virus will probably have the virus code copied into the front of the host, and may have the additional move instruction added to the end. However, this method won't find a virus that has been written into source code, and the checking programs should consider that some compilers place identical instructions into the beginning of the compiled code.

Source Code Analyzers : A code analyzer looks at the coding style(s) that a piece of source code is written in, and tries to classify its attributes. Like novelists, every programmer has their own style of coding, from how far to indent code to the conventions used in naming variables. A sophisticated code analyzer can examine these attributes and estimate the number of authors that wrote a particular piece of software by examining these attributes.

This method of detection is most useful if the analyzer is run against the source periodically during its lifetime. Each time the analysis is performed, the variations in coding styles and the estimated number of authors are recorded. Any deviation from the expected results is then investigated. For example, if only two engineers are permitted to access a source file, yet the analysis shows that there are three distinct coding styles in the source, there may be reason to believe that the file has been tampered with. The accuracy of the analysis depends largely on the sophistication of the analyzer. Also, this type of analyzer will only work on the source code, and would not detect any virus code that gets placed there by an unscrupulous author.

Length of Program/Code : Another method of detecting a virus is to keep a record of the original length of each executable file. At periodic intervals the current length of a program file is checked against its expected value. If there are any unexpected differences, the program can be isolated until it can be inspected. A more preventive method of using a file's length is to check it each time it is invoked. That way any infection can be found and dealt with before the program is run and the virus spreads further. Unfortunately, the cost of performing this check every time a program is invoked could be high. This would not work well in a software environment where the size of executables often changes.

Time, Date, and Checksum Values : Time and date stamps are used for virus detection in much the same way as the program length. First, the time and date of the last official modification is recorded. Each time the code is invoked, the current value(s) are checked against the recorded ones. If the values differ, the program is isolated until further investigation is made. However, this method makes the assumption that the virus does not or can not restore the file information to reflect the original, expected values.

Auditing : This detection method requires a record to be logged each time executable code is changed. The log records are then periodically reviewed to determine if any unofficial modifications have been made.

Prevention Mechanisms :

Protection methods aimed at keeping viruses off the system entirely are commonly more expensive to implement then detection mechanisms.

Access Control : One way to provide virus prevention is to implement an access control model. An access control policy places restrictions on who can have access to files on the system and what types of functions can be performed. A virus can only use the access permissions that are allowed to its host. The implementation of an access control policy will not allow the infected code to write into another executable on the system unless its host has the necessary permissions to do so. Though restricting the access path to executables won't stop most viruses, it may slow the spread of many viruses and decrease the amount of damage.

ROM : Another method of preventing a viral infection is to put all executable programs into ROM. Though this is an effective method of preventing a viral attack, it is also not practical. As with any program code, there will be the need to repair defects or implement new features and enhancements. Placing executable code into ROM makes replacing code too expensive and too slow for most installations [7].

Encryption : Encryption can protect the integrity of executable code. It can also provide a method of detecting the spread of a viral infection and can limit the amount of potential damage [8]. For encryption to work, executables must be stored on the system in the encrypted format. When the executable is invoked, it is run through a decryption algorithm. If the encrypted code had been tampered with, the deciphered executable will be unintelligible and will fail, preventing further viral spread or damage.

There are two key points to remember when protecting your system with encryption. First, this method of protection works on encrypted code. If the code is infected before the encryption, the virus can not be detected by this method and the virus would function as usual. Secondly, this encryption mechanism focuses protecting the encrypted object. It replies heavily on the underlying system security to ensure that the encryption/decryption keys and functions are protected from unauthorized access and modification.

Partitioning and Isolation : Partitioning a system or isolating the accesses of its users is more of a containment of a viral attack than it is a prevention of one. As stated earlier, the sharing, interpretation, and retransmission of information makes a system more open and vulnerable to attack. When the amount of contact or sharing between users is limited, the spread or scope of an infection is also limited. Partitioning involves setting up boundaries between sets or subsets of information, and protecting these boundaries from being crossed. The system then enforces the policy that no user can access more that one set of information at a time. That way, any infection is contained to a single set of data.

This type of isolation is often necessary for systems which store sensitive or classified data. In a commercial environment where we hope to benefit from the sharing of information, however, it may not be an effective method of protection.

The defense mechanisms listed here all incur some type of overhead. Most require additional services to be performed that will decrease system performance. A few require additional time for production, defect repair, or system administration. All require the time and money to implement and put them into effect.

Protection Policies :

There are several policies you can implement that will lower the risk of your system becoming the victim of a viral attack [5].

1. Educate your users to the damaging effects a viral attack can have on the system as well as to themselves and the company. Many users may not know how a virus appears and be able to identify peculiarities in the system's operation that indicate the possibility of an infection. The earlier a virus is detected, the less damage it can do.

2. Don't allow system users to install untested software

obtained from an unknown and possibly infected source. In their paper, "An Approach to Containing Computer Viruses", Maria Pozzo and Terence Gray propose that the level of risk associated with a piece of software corresponds inversely to its level of credibility. The following table shows their ratings for various sources of software :

Origin of Software	Credibility	User's Risk
User Files	Lowest	Highest
User Contributed Software	"	"
Software from Bulletin Board	"	"
Software from System Staff	"	"
Commercial Application S/W	"	"
Software from OS Vendor	Highest	Lowest

3. Don't connect your system to external bulletin boards where users will want to retrieve games or utility programs. As stated above, these sources are potentially hazardous.

4. Store new software on a stand alone system before placing it on a network or timeshare system. Doing this will allow you to observe how it affects the system. If the software is infected with a virus, the problem will become obvious before too long, and you will have saved yourself having to repair damage on a system that is networked or under heavy use.

5. Keep sensitive information and important programs under restricted access. By, placing more restrictive permissions on these files you will help to slow the potential damage to them, or possibly prevent it should the virus be detected early.

6. Assign only those capabilities and permissions to users which are necessary. Again, a virus uses the permissions of the user executing the infected program. Allowing a user to have management or privileged capabilities when they aren't necessary is a bad security policy that opens the door for a virus to infiltrate the system faster.

7. Use and enforce security features (such as access control and auditing) that you have available on your system to provide a robust and secure environment.

Some of these policies may not be productive in certain environments. However, for those who can implement them, they should help to reduce the risk of your system becoming infected, or reduce the damage should an attack occur.

Conclusion

There are so many variations of viruses that defending against them is a difficult proposition. A defense to one strain may not provide any defense to another. Each viral attack is different and more variations are constantly being created, as are new defenses to combat them.

The descriptions in this paper of how viruses work are very basic, even primitive in their implementation. In today's sophisticated computing environment, many viruses could be produced which are much more elaborate, less open to detection, and potentially more destructive then those examined here. For every new attack there will hopefully be a new mechanism to defend against it.

Your best defense is to find a combination of detection and prevention mechanisms that maximize the protection for your own environment. By implementing a procedural security policy that reinforces virus protection defenses, you will have a workable solution to protect against a viral attack. In addition, staying abreast of viral attacks against your computer system by reading journal articles, will keep you in a position to make a positive stand against any dangerous viruses lurking on the horizon.

References

[1] Cohen, Fred : "Computer Viruses: Theory and Experiments". Computers & Security, Volume 6, Number 1, February 1987, pp. 22-35.

[2] Burger, Ralf : "Computer Viruses: A High-Tech Disease". Grand Rapids MI.: Abacus, 1988.

[3] Bloom, Steven G. : "System Security - As Soon As I Can Find The Time". INTERACT, September 1988, pp. 24-27.

[4] Murray, W. : "The Application of Epidemology to Computer Viruses". Computers & Security, Volume 7, Number 2, April 1988, pp. 133-145.

[5] Highland, Harold J. : "Virus Defense Alert". Computers & Security, Volume 7, Number 2, April 1988, pp. 156, 158.

[6] Young, Catherine L. : "Taxonomy of Computer Defense Mechanisms". 10th National Computer Security Conference Proceedings, September 1987. National Computer Security Center.

[7] Davis, F.G.F. : "Recovering From a Computer Virus". Department of Computer Science, Wyoming University, Laramie, Wyoming. Journal of System Software, Volume 7, Number 4, December 1987, pp. 253-258.

[8] Gray, Terence E. and Pozzo, Maria M. : "An Approach to Containing Computer Viruses". Computers & Security, Volume 6, Number 4, August 1987, pp.321-331.

[9] Highland, Harold J. : "Computer Viruses and Sudden Death" Computers & Security, Volume 6, Number 1, February 1987, pp. 8-10.

[10] Highland, Harold J. : "Anatomy of a Virus Attack". Computers & Security, Volume 7, Number 2, April 1988, pp. 145-150.

[11] Department of Defense Trusted Computer System Evaluation Criteria". Department of Defense, National Computer Security Center, December 1985. DOD 5200.28-STD.

[12] Lai, N. and Gray, T.E. : "Strengthening Discretionary Access Controls to Inhibit Trojan Horses and Computer Viruses". USENIX Conference Proceedings, Summer 1988, pp. 275-286.

Trusted Systems: Features and Certification of a Secure Operating System

John Maus
Hewlett-Packard Company
N. 1225 Argonne Road
Spokane, WA 99212

I. History

In October, 1967 a task force of the Defense Science Board was formed and chartered with finding ways that classified information could be protected in computerized systems. They issued a report entitled "Security Controls for Computer Systems" in February of 1970. In response to recommendations in this report the Department of Defense (DOD) issued DOD Directive 5200.28 in 1972 as a way to establish uniform DOD policy for protecting classified information.

At that point, ARPA (Advanced Research Projects Agency) and other agencies began working on actual implementations of these recommendations. This research continued through the mid-70's.

Then, in 1977 the DOD Computer Security Initiative was started. Concurrently, the National Bureau of Standards (NBS) began to address issues involving building, evaluating and auditing secure computer systems. They sponsored one workshop in 1977 and one in 1978.

Using recommendations from the second workshop, the MITRE Corporation began to develop a set of criteria to enable one to assess the degree of trust that could be placed in a computer system that was processing classified information. Their work has been critiqued by many agencies, universities and computer vendors.

In January of 1981 the National Computer Security Center (NCSC) was formed to expand upon this earlier work. A major goal of NCSC is to foster wide availability of secure systems. In December of 1985 the NCSC issued a report which has become known as the "orange book". It is entitled "Department of Defense Trusted Computer System Evaluation Criteria". Copies are available from the Office of Standards and Products, NCSC, Fort Meade, MD 20755-6000, Attention, Chief, Computer Security Standards. The provisions of the document apply to all DOD component agencies.

II. Objectives of the Criteria

The criteria presented in the trusted systems document were designed with three major objectives in mind:

A. To provide guidance to manufacturers of computer equipment so that the hardware and software developed would contain the features needed to protect sensitive information.

B. To provide users of computer systems a means or metric for assessing the degree of trust they could place in specific equipment.

C. Provide a basis for acquisitions of new equipment.

The document sets forth two distinct sets of requirements. The first are specific security feature requirements. These are the requirements that must be met by the operating systems of general purpose processors. The second are assurance requirements. These requirements are applicable to the full range of computer processors from dedicated to general purpose.

III. Fundamental Security Features

The basic assumption that NCSC made was that a secure computer system should control access to information so that only properly authorized individuals or processes could gain access to it. They proposed six fundamental requirements from which they developed a security classification scheme and a large number of detailed requirements. These six fundamental features are:

A. Policy
   1. Security Policy - "There must be an explicit and well-defined security policy enforced by the system".
   2. Marking - "Access control labels must be associated with objects".

B. Accountability
   1. Identification - "Individual subjects must be identified".
   2. Accountability - "Audit information must be selectively kept and protected so that actions affecting security can be traced to the responsible party".

C. Assurance
   1. Assurance - "The computer system must contain hardware/software mechanisms that can be independently evaluated to provide sufficient assurance that the system enforces" the above four requirements.
   2. Continuous Protection - "The trusted mechanisms that enforce these basic requirements must be continuously protected against tampering and/or unauthorized changes".

IV. The Classification Scheme

NCSC created four "divisions" of trusted computer systems. The division specifies the general security rating of a system. Some of the divisions are further sub-divided into "classes". Each division and each class within a division signifies a more secure system. The divisions and their associated classes are as follows. They are listed in order of increasing security provisions:

1. Division D - "Minimal Protection"
   Any system that is evaluated but fails to meet any of the higher requirements gets this rating.

2. Division C - "Discretionary Protection"
   Systems rated within this division provide need-to-know protection and auditability.
   - Class C1 - "Discretionary Security Protection"
   - Class C2 - "Controlled Access Protection"

3. Division B - "Mandatory Protection"
   The concept of a "sensitivity label" is a key feature of systems rated in this division.
   - Class B1 - "Labeled Security Protection"
   - Class B2 - "Structured Protection"
   - Class B3 - "Security Domains"

4. Division A - "Verified Protection"
   Formal security verification methods are used to verify protection mechanisms in systems with this rating. Extensive documentation is also required.
   - Class A1 - "Verified Design"

V. Specific Security Features

The NCSC report delineates twenty-seven specific security features. As the system security class increases from C1 through A1, more and more of these features are required to be present. Table 1 shows the relationship between the security features and the class rating. In other words, in order for a system to achieve a specific class rating, it must contain all the features mandated for that class.

These specific security features have been grouped into four categories: 1) Security Policy, 2) Accountability, 3) Assurance, and 4) Documentation. What follows is a brief description of each security feature:

1. Audit - The system must be able to create, maintain and protect an audit trail of the accesses to the objects it protects.
2. Configuration Management - When the operating system code is modified a system must be in place that ensures that the design specifications, source code and documentation all agree for the version be generated.
3. Covert Channel Analysis - A covert channel is any communication path that can be used to obtain data in such a fashion as to circumvent the system's security features.
4. Design Documentation - Documentation from the vendor that describes the operation and use of the security features.
5. Design Specification and Verification - An informal or formal model of the security policy supported by the operating system.
6. Device Labels - A method of attaching a security level to a physical device so the operating system can enforce constraints on it.

Trusted Computer System Evaluation Criteria
Summary Chart

Security Class -->	C1	C2	B1	B2	B3	A1
I. Security Policy						
Discretionary Access Control	YS	YS	SA	SA	YS	SA
Object Reuse	NO	YS	SA	SA	SA	SA
Labels	NO	NO	YS	YS	SA	SA
Label Integrity	NO	NO	YS	YS	SA	SA
Exportation of Labeled Information	NO	NO	YS	SA	SA	SA
Exportation to Multilevel Devices	NO	NO	YS	SA	SA	SA
Exportation to Single-Level Devices	NO	NO	YS	SA	SA	SA
Labeling Human-Readable Output	NO	NO	YS	SA	SA	SA
Mandatory Access Control	NO	NO	YS	YS	SA	SA
Subject Sensitivity Labels	NO	NO	NO	YS	SA	SA
Device Labels	NO	NO	NO	YS	SA	SA
II. Accountability						
Identification & Authentication	YS	YS	YS	SA	SA	SA
Audit	NO	YS	YS	YS	YS	SA
Trusted Path	NO	NO	NO	YS	YS	SA
III. Assurance						
System Architecture	YS	YS	YS	YS	YS	SA
System Integrity	YS	SA	SA	SA	SA	SA
Security Testing	YS	YS	YS	YS	YS	YS
Design Specification & Verification	NO	NO	YS	YS	YS	YS
Covert Channel Analysis	NO	NO	NO	YS	YS	YS
Trusted Facility Management	NO	NO	NO	YS	YS	SA
Configuration Management	NO	NO	NO	YS	SA	YS
Trusted Recovery	NO	NO	NO	NO	YS	SA
Trusted Distribution	NO	NO	NO	NO	NO	YS
IV. Documentation						
Security Features User's Guide	YS	SA	SA	SA	SA	SA
Trusted Facility Manual	YS	YS	YS	YS	YS	SA
Test Documentation	YS	SA	SA	YS	SA	YS
Design Documentation	YS	SA	YS	YS	YS	YS

Legend: SA = Same requirement as previous lower class
YS = Yes, required for this class (either same requirement as previous class or enhanced requirement)
NO = No, not required for this class

Table 1

7. Discretionary Access Control - A method whereby access must be controlled between named objects and named users. This control must be possible to the level of a single user.
8. Exportation of Labeled Information - Each I/O device must be defined as either handling information of strictly one security level or of handling information of multiple levels.
9. Exportation to Multilevel Devices - If a device is to handle information of differing security levels, a security level label (called a sensitivity label) must remain attached to the information.
10. Exportation to Single-Level devices - Since this device handles only information of a specific security level the sensitivity label is not required to be maintained.
11. Identification and Authorization - The procedure whereby users identify themselves to the system and it authenticates their identity.
12. Label Integrity - If data is exported the exported sensitivity label must be identical to the internal label.
13. Labeling Human-Readable Output - The printing routines must label the printed output with sensitivity label information.
14. Labels - The system must maintain security level information (the sensitivity label) for each object of data. When importing new data the system must receive the security level information from an authorized user.
15. Mandatory Access Control - The operating system must enforce a mandatory access control over all subjects and objects and the resulting access may need to be determined by a combination of security schemes.
16. Object Reuse - If an item is released back to the system then current access controls must be reset.
17. Security Features User's Guide - A vendor-supplied reference manual describing the security features of the system.
18. Security Testing - Tests must be performed to ensure the security features work as described. Test plans are outlined for each division.
19. Subject Sensitivity Labels - A terminal user must be notified during an interactive session if his/her security level changes.
20. System Architecture - The operating system must maintain an execution domain of its own (e.g., privileged mode) and must provide process isolation via distinct address spaces.
21. System Integrity - Hardware and/or software diagnostics must be provided to ensure the proper operation of the system.
22. Test Documentation - The vendor must provide the evaluator with a test plan describing how the system was tested.
23. Trusted Distribution - A method must exist to ensure that the copy of the operating system that is delivered to a site is identical to the master copy.
24. Trusted Facility Management - The duties of a security administrator must be defined. An audit trail must be produced showing changes in the administrator.
25. Trusted Facility Manual - A reference manual describing the role of the system administrator must be supplied.
26. Trusted Path - The communication path that a user uses to login must be secure.

27. Trusted Recovery - After a system failure it must be possible to recover the system without any security compromise.

The foregoing is not meant to be an exact description of each feature but rather to give the general idea behind the term. As is shown in Table 1 no class except A1 requires the presence of all the features. Once a feature is required by a class it is also required by each higher class. However, the specifics of the feature are sometimes changed/enhanced from class to class. To get the exact specification of a feature for a particular class please refer to the original NCSC document.

VI. A Specific Implementation of Class C2

With the arrival of Hewlett-Packard's MPE V Operating System, version V-Delta-4 (G.C3.04) Hewlett-Packard customers were able to obtain an operating system evaluated by NCSC as a Division C Class C2 Trusted Computer System. This class requires eleven of the twenty-seven security features identified by NCSC.

In what follows we will see what specific MPE features implement the required NCSC feature. Note that some required security features are only obtained with the proper V-Delta-4 version running in combination with the product known as HPSecurity Monitor.

A. Security Policy
   1. Discretionary Access Control - This has been implemented by means of Access Control Definitions (ACD's). ACD's can be associated with files or devices by means of the :ALTSEC command. Access modes can be granted to lists of users or down to the level of a single user.
   2. Object Reuse - If an ACD is removed from a file or device then the normal MPE security features are used. The ACD is stored in a "pseudo-extent" and not in the file label. The extent location is not stored in the extent map.

B. Accountability
   1. Identification and Authorization - This is provided by the :HELLO command and passwords. The user identification is also stored as part of the "time stamp" in logfile records.
   2. Audit - New logfile record types were implemented (record types 38-42). Assurance of auditability is provided by an option in HPSecurity Monitor. If the system is unable to log events then all users are aborted.

C. Assurance
   1. System Architecture - MPE V can operate in privileged mode.
   2. System Integrity - Diagnostics are provided to ensure the correct functioning of the hardware and firmware.
   3. Security Testing - Extensive testing was done by Hewlett-Packard before release.

D. Documentation
   1. Security features User's Guide - This is provided by the reference manual "MPE V/E Security and Accounting Structure User's Guide", P/N

32033-90136.

2. Trusted Facility Manual - Same as above.
3. Test Documentation - Provided by Hewlett-Packard to the NCSC evaluation team.
4. Design Documentation - Provided by Hewlett-Packard to the NCSC evaluation team.

VII. Evaluation and Certification

The NCSC has established a "Commercial Product Evaluation Process" by means of which it can evaluate and rate computer systems against the aforementioned criteria. This evaluation process has three steps:

1) Preliminary Product Evaluation - In this step the NCSC team and the vendor open a dialogue about a proposed vendor product. After discussion, NCSC tells the vendor what rating might be expected if the product were formally evaluated.
2) Formal Product Evaluation - This is a formal evaluation of the product by the NCSC team and results in the product being given its security class rating.
3) Evaluated Products List - After the formal evaluation the product is then added the the official list.

The vendor has to expend a certain amount of resources to produce a product of a certain rating. If the rating to be obtained is lower than the one the vendor intended, then the vendor will either have to expend more resources or abandon the effort. Because of the openness of the dialogue either the vendor or NCSC can decide to stop the process at any point. It should be noted that the NCSC signs a non-disclosure agreement with the vendor at the beginning of the process.

Just because a product is on the Evaluated Products List does not mean that it is "certified". A system can only be certified when put into to use in a specific application environment and only after the NCSC team performs an evaluation on the system in that environment.

VIII. The Evaluation Process of Hewlett-Packard's MPE V

Hewlett-Packard contacted the NCSC for a product evaluation of V-Delta-4 in April of 1987. NCSC has two companies under contract to provide evaluations besides itself. They are the MITRE Corporation on the East Coast and the Aerospace Corporation on the West Coast. A team from the Aerospace Corporation was chosen to perform the evaluation.

The team acted as a consultant to Hewlett-Packard during this phase. They did not see or exercise any code. They prepared an Initial Product Assessment Report (IPAR) based on documentation and discussions with Hewlett-Packard. The Technical Review Board (TRB) of the NCSC reviewed the IPAR. The TRB and Hewlett-Packard agreed to go forward for a formal evaluation. The team then actually began to test the code and added tests of their own. Hewlett-Packard gave the team formal training on the product during this phase.

At this point the team wrote an addendum to the IPAR stating what needed to be tested. The IPAR was again reviewed by the TRB and the team was given the o.k. to actually perform the tests. The team then wrote the test evaluation results. These were reviewed by the TRB and V-Delta-4 was awarded the C2 rating and added to the Evaluated Products List. The IPAR was turned into a formal report and kept by the NCSC. The time required to perform Hewlett-Packard's evaluation was the shortest of any vendor.

IX. Conclusion

The NCSC document "Trusted Computer System Evaluation Criteria" sets forth a trusted computer system classification scheme whereby the higher the rating given to a system the more confidence a user can have that it is properly safeguarding sensitive information. While computer systems acquired by DOD agencies must be rated (i.e., appear on the Evaluated Products List) for the type of environment they will be used in, civilian companies may want to employ rated systems as well. The use of a rated system by any civilian company will give that company a higher degree of confidence over the protection of their data than they would have had they employed a non-rated system. For Hewlett-Packard customers those running the G.C3.04 version of V-Delta-4 MPE V in combination with the product HPSecurity Monitor will be employing a system evaluated at the Class C2 level.

Manage Your Multiple Data Centers - Don't Kill the Messenger

Betsy Leight
Operations Control Systems
560 San Antonio Road
Palo Alto, California 94306
(415)493-4122

INTRODUCTION

The efficiency and security of data processing operations is a major concern to corporate data processing managers.

To assess their data center's performance, management often conducts a review of the following areas: standards and procedures, operational work flow and control, scheduling, data security and access control, equipment utilization, and environment.

If the data center management and staff understand the concerns of upper management and the information that is needed, the operations review can proceed more smoothly, and the results can be more beneficial to the entire organization.

Corporate direction for data center management focuses on control issues, automation procedures and the gray area in between. The data center manager reviews the operations and interprets corporate objectives to the operations staff. Because this article focuses on the concerns often overlooked by the data center manager, it can be a useful check list for improving daily operations as well as preparing for a data center operations review.

STANDARDS AND PROCEDURES

Data center managers should verify that standards and procedures exist and are enforced. These written rules are the controls; they should include:

* Ensure proper timing in running programs and jobstreams.

* Insert changes into production runs; enter run dates.

* Use correct data for programs; access correct data files.

* Protect data and programs from accidental or intentional destruction.

* Specify methods of physically moving input and output.

* Schedule work and getting work rerun in the event of errors.

* Keep records of work performed and session logons.

* Determine and record sufficient resources for the work.

* Perform maintenance and general housekeeping associated with the operation of the data center.

The data center manager should ensure that formal standards exist for systems development and maintenance, program and system testing, file conversion, program and system change control, library operations, computer operations and documentation.

For each aspect of standards and procedures, your installation can implement procedures using automation software as shown in the following chart.

Task Description	Controlled Operation	Automated Controls	Controlled Automation	Automated Operation
Run Programs	User Task	User with Scheduler	User who Schedules	Scheduling Software
Change Jobstreams	User Task	User with Editor	User who Edits Jobs	Job Change Software
Verify Data File	User Task	User with File Scan	User who Scans Files	File Scan Software
Monitor Jobs	User Task	User with Jobstream Monitor	User who Monitors Jobstreams	Jobstream Monitor Software
Control Master Files	User Task	User with File Copy	User who Copies files	File XFER Software

OPERATIONAL WORK FLOW AND CONTROLS

The data center manager should investigate specific items in this area, including whether:

* Enter input data from other departments completely and on time.

* Track job accounting and session logon information.

* Notify appropriate personnel in case of production processing error.

* Document batch processing errors and logon violations.

* Accumulate error statistics.

* Follow up on all errors so that they do not recur.

* Distribute all reports to the proper user departments.

* Establish procedures to control the distribution of sensitive output.

* Dispose of confidential reports when they are no longer required.

The data center manager should also confirm that downtime is reported and statistics compiled. A log of late reports and jobs should be maintained.

There should be a formal communications channel between data center operations and other departments; operational tips and other advice should be passed to all operators.

All problems encountered at the computer, as well as any action taken to prevent their recurrence, must be documented. Operators must also receive feedback on reported problems.

The data center manager scrutinizes output report distribution and disposal and determines whether:

* All reports have been distributed to the proper user departments.

* Procedures exist to control the distribution of sensitive output.

* Procedures exist for disposing of dated confidential reports.

Finally, the data center manager should ensure that jobstream run instructions are kept up to date.

SCHEDULING

Efficient and effective scheduling is extremely important in providing a high level of reliability and predictability to data center operations. The data processing manager should determine whether:

* Daily processing activities are scheduled and a daily contingency schedule is maintained.

* Actual run times are recorded for batch programs and jobstreams.

* Data is used to calculate expected run times for a given day.

* Expected run times are compared with actual execution time to ensure that processes have not been terminated abnormally.

* Unscheduled runs are supported by a work request or other written authorization. Schedule deviations should be documented and followed up on by a supervisor.

* User-submitted jobs are recorded to allow forecasting of future schedules, resource requirements, and special processing considerations.

* All jobs are submitted through or controlled by data center operations.

Scheduling software enforces controls and totally automates the data center operations. Manpower reductions can result depending on implementation. A chart on scheduling appears below:

Task Description	Controlled Operations	Automated Controls	Controlled Automation	Automated Operation
Create Schedule	User Task	User with Streamer	User who Schedules	Scheduler Software
Monitor Run Times	User Task	User with Job Log	User who Logs Jobs	Monitor Software
Job History Report	User Task	User with Job Log Data Base	User who Compares Job Logs	Data Base Report Generator

DATA SECURITY AND ACCESS CONTROL

Data base and master file information should be protected from unauthorized access or loss. Employees must be instructed about their responsibilities concerning confidential information. Management should periodically review and update controls and security provisions relating to data.

Live production programs should be physically separated from development programs. The staff should be prohibited from running test programs against live files, and operations personnel should be denied access to sensitive data files.

Secured file management is not limited to source and object control. Data center managers should ensure that procedures have been established for:

* Transferring programs from development to production.

* Approving program library changes.

* Testing changes to programs before transference to the production libraries.

* Updating production documentation after changes.

To maintain security, operators should be prohibited from renaming or transferring programs without supervisory approval. Internal labels must be used from all data and program files.

Passwords and lockwords should be used to protect accounts, users, data files and port access. Passwords, lockwords, dates and constants should be introduced at run time, eliminating the need to hard-code sensitive data in jobstreams.

Data security and access control software can bring automation to the data center; with automation you can expect a more efficient system operation as summarized in the chart below.

Task Description	Automated Controls	Controlled Automation	Automated Operation
Separate development and production areas	User who moves files	User with file mover	File Librarian Software
Restrict live file access	User who locks files	User with lockwords	Protected File sets to User Sets
Approval pre-step to Production Move	User who moves files	User with file mover	Automated File move after Approval
Project/memo notes Related to Changes	User who completes forms	User with text editor software	Online dialogue requesting memo text at save time

EQUIPMENT UTILIZATION AND EFFICIENCY

Once it has been determined that the entire data processing department is following a properly implemented set of standards and procedures, the data center manager should review equipment utilization.

The data center manager should collect raw data from the system log files in order to report the following information:

* How much machine time is spent on reruns?
* How can the reruns be best analyzed?
* Which jobs are especially susceptible to reruns?

With reported resource utilization information, the data center manager should check that the full multiprogramming capability of the system is being used. It then follows that multiple jobstreams should run concurrently, if there are no data file bottlenecks.

The data center manager then reviews whether many jobs can be restarted without rerunning the entire job. Jobstep tracking and restart software should be implemented for efficient data center operations.

ENVIRONMENT

The data center manager should review the work space to ensure that it is adequate for the number of employees. The environment should be neat, and supplies should be easy to locate.

Auxiliary items located outside the computer room, such as bursters and de-collators, should be accessible for the flow of work in the department. Tapes, discs and other storage media should be stored in a closed, fire-protected, limited-access area.

RECOMMENDED COURSE OF ACTION

The data center manager should make the organization aware that the following steps can enhance the operations review:

* Determine whether certain jobs are especially susceptible to reruns.

* Provide the data center management with as much information as possible.

* Implement software systems that leave clearly defined audit trails.

* Keep accurate records, log files and file history information.

* Maintain formal written standards and procedures.

* Implement an effective data security system and access control facility.

Following the data center manager's recommendations and procedures for operations can yield an efficient, secure and automated data center.

OPERATIONS MANAGEMENT IN A MULTI-HP3000 ENVIRONMENT

Roberto Drassinower
Teresa Brzozowski
Carolian Systems International Inc.
3397 American Drive, #5
Mississauga, Ontario
L4V 1T8 CANADA

This paper discusses centralization and automation in a multi-HP3000 environment. It offers suggestions on how to plan for and introduce network-wide operations control, thereby making large shops more manageable and efficient. This paper deals with centralization and automation as two distinct issues, considering first where centralization is required, and only then implementing the automation process.

INTRODUCTION: THE OPERATIONS OPTIMIZATION PROJECT

In recent years, multi-system installations have become more common in the HP3000 community. While the advantages of both distributed and local network systems are often obvious, the challenges associated with their management as one coherent unit are all too often underestimated. Since each machine is typically monitored and controlled through its independent console, an overall perspective of network-wide operations is difficult, if not impossible, to obtain. Yet sensitivity to global issues and centralized decision-making are essential to the efficient management of large scale integrated systems.

The problem of centralized control over multiple 3000s is vastly complicated when a central site has to support remote sites. Often a lack of adequate operations staff limits control over production and severely impairs the flow of system status information back to head office. Similarly, the sheer size and complexity of local multi-machine installations can also render traditional operations management techniques ineffective.

The possibilities for streamlining networked operations do exist in an HP3000 environment, and provide several advantages over traditional operations procedures. A well-planned centralization project will allow you to develop the tools with which you can centralize your information and which will give your existing operations staff greater control over distributed systems from one location. All of this can be achieved through a program which takes into account important hardware, software and network issues, specifically in the areas of automation, integration and compatibility.

The resulting centralization of information not only makes your operations more efficient; it also provides management with procedures for production. Such policies inevitably lead to improvements in overall system throughput and higher system utilization. Moreover, continually reviewing operations management systems guarantees the optimal use of existing resources as installations grow.

What is needed, and sometimes missing, is a firm commitment to an ongoing, long-term operations optimization project. Although this will divert resources away from satisfying some demands of the data processing department in the short term, failure to review procedures may significantly erode its operating efficiency and ability to service the company in the long term. Since the results of your labour will be a long term solution, you can implement distinct phases without incurring prohibitive expenses, while at the same time remaining confident of a proper design and direction. In the long run, it will be well worth the effort. In tackling the issue of centralizing and automating your DP shop, it is important to understand fully how your operations work and, more importantly, how you would like your shop to work. Therefore, I would like to review some of the basic concepts about the operations environment before continuing.

Today, HP3000s are controlled by an operator through the systems' local consoles. This system console is the lifeline to the HP3000, since all operations status messages pass through it. It is a hub where all problems can be defined in two categories: regular, day-to-day operations messages (such as tape requests), and messages that report an error condition to which someone should be alerted.

In this scenario, it is assumed that trained operators are constantly monitoring the console screen so that they can respond to the requirements of the message immediately. This also assumes that the operators are trained in a variety of areas, so that they can handle job management issues, system security policies, internal system management, data comm, and so on. In short, good operators have to specialize in many different areas -- they must be a jack of all trades. Traditionally, the operators must also contend with a variety of messages which are irrelevant to their primary tasks.

For example, if the operator is mainly concerned with executing and monitoring an installation's production schedule, then that person will only want to view job completion and error messages from among all activity passing through the console. He or she would not be interested in logon messages, HPDesk messages, data comm messages and so on. The result is an operator who must act as a human filter. As the site becomes more active or complex, the operator's job becomes increasingly more difficult as critical information scrolls off a busy console and disappears. Then the operator is faced with paging through hard-copy $STDLISTs.

Now imagine this task in a multi-machine environment. Busy shops might need an operator per system in order to keep operations running smoothly. Such a setup, however, makes centralization difficult since each console is physically separate from the next. Therefore, messages are received by operators based on the physical, rather than on the logical structure of the operation. Because messages coming from multiple machines cannot be grouped logically, you must have several operators monitoring separate machines, looking for and responding to only those messages which are relevant to them. This often results in wasted time for a number of operators since they are only concerned with a small percentage of the total messages. In addition, this can lead to a duplication of tasks since two operators may be performing the exact same function on totally separate machines. If, however, messages could be grouped logically, an operator could receive all messages relevant to his or her task from all the machines in the network. With such a setup in place, the operators' tasks could easily be organized according to function rather than to the physical location of the machine, and such centralization of resources and information would thereby result in the optimization of your operations practices.

THE FIRST STEP TO IMPROVEMENT

Throughout this discussion I will be making general recommendations which, when grouped together, provide the basis for a long-term operations review plan. The first of these is as fundamental as it is self-evident: Commit resources to examining existing operations practises through a long-term operations optimization program. This must be an on-going project so that your operations optimization can accommodate changes in demands.

As I alluded to earlier, the most substantial obstacle this kind of project must contend with is the difficult task of providing operations staff with both the information and control required to efficiently oversee multiple systems. In order to overcome these obstacles, it is crucial that you determine the nature of these needs precisely so that they can be satisfied as efficiently as possible.

As you may have guessed, the issues that must be confronted during such a study are not unlike those a systems analyst must wrestle with when designing an order entry system. For example, this individual would be concerned with the kind of system access data entry personnel require and the content of management reports. Quite clearly, different individuals often will require highly specific information which is of little consequence to others. On the other hand, recognizing common needs is just as important to the design of an integrated system.

In designing operations management systems, much the same is true. To truly centralize your information, operators will need access to the consoles of all systems. In addition, particular operators may only be interested in a subset of all console messages. For example, an operator charged with servicing tape drives will have a keen interest in tape mount requests, but will not be so concerned about interactive users logging on and off. A security specialist, on the other hand, should be aware of these logons, and will also want to monitor remote users logging on through dial-up modems. A tape librarian will profit from reports regarding the frequency in which particular tapes are used, and from knowing when these tapes are needed during the production schedule. By monitoring console messages based on function, a shop would be better able to optimize its resources through specialization and centralized control.

To design a proper operations management system, you will need to identify the flow of information within your company. You should ask yourself what types of messages pass through your console on a day-to-day basis? What kind of information flows from your users to your operations staff and technical support? Such analysis will help you to identify the information which is critical to your operations, and the person who should receive these messages. It is only when you have a good understanding of your environment that you should move towards the type of centralized operation referred to earlier.

WHY CENTRALIZE?

For the same reasons that you want to divide console messages into logical categories for your operators, you should aim to structure your DP department in a manner that reflects the logical structure of your installation. In multi-machine environments, machines are rarely isolated from one another with respect to your company's production. Most shops network their machines in order to ease the transfer of information. There are usually many dependencies among machines for production, so the independent management of each HP3000 is an inefficient way to control your DP environment. It would be better to have centralized control and management of multiple systems so that all information can be easily related. The result is that DP managers can make better management decisions and increase overall network uptime because they are working in a more controlled environment.

Some of you probably have, or are already taking steps towards centralization today. Batch scheduling software is a good example of this. Third-party vendors such as OCS and Unison offer utilities which allow you to do sophisticated batch job scheduling and which give you the ability to have cross-machine job dependencies. By knowing the status of jobs on all of your systems at one central site, you can truly run a distributed network efficiently. When provided with up-to-date information about the status of all your systems and by having all production scheduling information routed back to a central site, you can schedule jobs throughout a network and load balance across a number of systems. This means a maximization of the use of your resources by having all systems available from one central location.

Centralization also has implications on system security. By having control over all consoles, you are reducing the number of access paths to your systems; that is, fewer people need the high level capabilities required for operating your machines. Secondly, since all console messages are centralized, it is easier to watch for unauthorized access (i.e. logon violations can be grouped together for easier identification.)

Centralization of your consoles will also allow you to centralize your MIS staff -- both operators and specialists. If all machines can be accessed and full console control is available from one central location, then you don't need to have duplication of staff at remote sites. Because commands can be issued centrally, your resources and expertise can be at one site.

TOWARDS AUTOMATION

Once centralization has been achieved, you can begin to consider automation. It is important to note, however, that automation is a relative term. It should be viewed as an effort to reduce human intervention without necessarily eliminating it. So, the real issue is not whether to automate, but how much to automate. Significant gains in reduced intervention can be attained without substantial costs associated with automation and, in particular, artificial intelligence applications. It is also important to remember that automation comes with time; until you know exactly how and what to automate, you should not attempt to do it.

Before implementing an automation strategy, you need to know exactly what to automate. The key ingredient for such an analysis is information. A good way to gather this information is to log all your operations activity -- that is, the console traffic over a period of time. Next, analyze it to see what can be automated by considering the types of responses required for the messages coming in. Once this has been determined, you can design a plan and only then begin to implement procedures. Undoubtedly, you would have gone through a similar process in implementing a job scheduling package.

A job scheduling package is a good example of an automated process that shops "grow into" over a period of time. In many cases, system managers start out by using the MPE :STREAM JOBxxx;AT=time command to run jobs at night unattended. Running jobs overnight is often the solution to completing processing without affecting performance for on-line users. However, where job dependencies exist and jobs have to be run sequentially, using the STREAM command may not be enough. With STREAM you would have to guess when job A will complete so that you can stream job B to run once A has finished. If your predictions are off, jobs could be colliding or, alternatively, you could have a lot of idle time on your system.

At this point, a shop considers a job scheduling package to automate all overnight procedures. Once the scheduling package is installed, it is up to you to carefully plan your production so that processing continues properly. Typically, when people learn that the package is up and running, they will find more and more jobs that can be scheduled into it. Anything that can be automated will be automated over time. Most shops find that automating job scheduling makes life easier for operators as there is less of a need for human intervention. The result is an increase in overall productivity of machines and staff since available processing time has been optimized, and your operations staff's time has been freed for other tasks.

In automating your entire operation, the process for implementation is much the same: review what you are currently doing, analyze what can be automated, plan for implementation, and then implement. The process does not stop there, however. Just as jobs are continually added to a scheduling package, operations tasks should be continually evaluated as candidates for automation. After the initial implementation, your focus becomes one of review and analysis of tasks to be automated.

WHERE SHOULD ANALYSIS START?

Listing from your console will give you an accurate picture of what happens on a system from an operational point of view. From a log file, you need to identify which mechanical tasks are possible to automate. In many cases, either HP or a third party offers a way of automating a task to varying degrees (as in the example of batch job scheduling). Currently, there is a wide variety of software packages that can be purchased to help you automate: automatic error detection, high-speed backup, database manipulation, spooling, performance analysis, and so on. An in-depth study of console activity and your overall operational goals will help you to decide which products may be of benefit to your shop. The key to these tools when moving towards automation is that they are programmable, where expert systems and artificial intelligence may be closest to complete automation.

Of all the messages which are sent to the system console, those that require human intervention are perhaps the most critical. These could range anywhere from a reply to a tape request to correcting an error in a job to restarting a failed system. In each case, someone has to be made aware of the situation; otherwise, processing will not continue as planned. Rather than automating these tasks, you could put an alert mechanism in place. Depending on how critical each message is, the appropriate reporting mechanism could require a console command to be issued, a message to be sent to a specialist, or a programmer on call to be contacted. Through an alert mechanism, you will know how much human interaction is necessary and where. This is invaluable information for your operations automation plan since it can help you to determine your requirements for staffing and software tools.

Simply stated, the rationale for analyzing your console activity is to know exactly what to automate and how. This helps you make sure that all bases are covered when moving to automation. The more information you have about your console activity, the more situations you can account for. Again, going back to the job scheduling example, if you know what the various types of possible errors are, then you can set up a number of contingency plans to compensate for those errors. Another example of this is the automation of procedures relating to system failures.
Through analysis of your console activity, you will know the most common reasons for system downtime, and you can then set up procedures to restart your system, tailoring the restart to the type of failure.

Also remember that automation is something that should evolve as you learn more about your system and your operation. In the early stages, the mechanism you use to alert personnel to critical situations may simply be a beeping message at the console. Later, it may grow to include a voice messaging system that automatically dials a programmer on call. To handle the problem of loading tapes and replying to requests, some companies are going so far as to get robots to mount tapes and then to program a reply which will execute automatically.

Even for less critical activity, it is a good idea to log information so that you can monitor the demand for certain types of tasks. Again, this will also help your specialists complete their jobs more efficiently. Consider the example of system performance. Both Hewlett-Packard and Carolian Systems are taking steps towards making it easier for users to centralize and automate performance analysis through the creation of an interface between a performance tool and the console. HP has chosen to tie OPT/3000 to their recently announced OpenView product, thereby allowing performance-related information to come back to a central site. Carolian has added a module to SYSVIEW so that the program can run in a "monitor" mode. In this mode, information about unusual performance situations gets passed back to a central console where a performance specialist can then monitor and respond to these messages.

IN SUMMARY

In order to manage multi-HP3000 sites effectively, centralization and automation of console activity are necessary. In order to develop an effective operations optimization strategy, however, careful analysis and logging of console operations is required. It is only once your objectives have been defined that you can begin to implement your plan. Such long term planning will be well worth the effort since centralization of all your HP3000s will mean increased control over your entire operation, optimization of both your hardware and human resources, and greater opportunity for standardization across your organization.

Once centralization has been put in place, a plan for automation should begin. Automation should be implemented over time so that you can automate as many different tasks as possible. Remember that any tool you choose to help you automate should be flexible, programmable and able to grow with you. Once again, the benefits of a well-planned automation program will become obvious as it frees up time for your operators, allowing them to concentrate on the tasks critical to your operation. As utilities become more and more sophisticated, you may find it possible to program even these tasks.

Reading through any HP publication, you will find that there is a growing need for standardization and efficient use of both hardware and human resources. It is this need which is driving users and vendors alike to search for and implement solutions which can incorporate both the control of centralization and the power of automation in their operations optimization projects.

Using the HP 3000: What I Wish I Had Checked Into Sooner

David L. Largent
Data Processing Manager
N. G. Gilbert Corporation
P. O. Box 1032
Muncie, IN 47308-1032
(317) 284-4461

If you are like me, you tend to avoid some things in life as long as you can. For some reason, writing a will for yourself and going to the dentist are not on most people's priority list. This avoidance of certain aspects of life carried over to my HP 3000 as well. And, as is usually the case, once I finally got around to investigating the very thing I had been avoiding for so long, I found not only was it not too bad an experience, but it ended up being downright useful! This paper presents the topics I wish I had checked into sooner that made my life with the HP 3000 easier and more productive.

"Freeware".

I will talk about two categories: the INTEREX CSL tapes, and the TELESUP account. Neither of these are really free (few things in life actually are), but both are relatively inexpensive. To obtain a CSL tape, you (or your company) must have a site membership with INTEREX. The TELESUP account comes with HP support -- which of course is associated with a fee. Now for some specifics.

An INTEREX Contributed Software Library (CSL) tape is a marvelous storehouse of treasures. All one has to do is read through the index to find at least a dozen programs that can be beneficial to an installation. INTEREX produces an updated version of this tape yearly. Take the time to investigate this!

The TELESUP account, just like the INTEREX CSL, contains many treasures. This account is provided by Hewlett-Packard and is installed on most systems so your HP SE and CE have access to some very useful utility programs. Take a look; you will likely find some use for them yourself!

A few of the programs I have found useful from the INTEREX CSL tapes and the TELESUP account include the following.

AMORT	Creates loan amortization schedules. Found in the CSL.
BACKUP	Full and partial SYSDUMP manager. Found in the CSL.
BULDACCT	Re-creates the accounting structure. Found in the CSL and the TELESUP account.
DBCHGCAP	Expands/contracts IMAGE and TurboIMAGE datasets. Found in the CSL.

DBLOADNG	Provides statistics about a data base. Found in the CSL.
LIST	General purpose ASCII/binary file lister. Found in the CSL and the TELESUP account.
LOGAUDIT	System logfile analyzer/lister. Found in the CSL and the TELESUP account.
LOGERR	Analyzes/lists system logfile I/O records. Found in the CSL and the TELESUP account.
PRIME	Prime number generator for IMAGE masters. Found in the CSL.
PSCREEN	Copy screen contents to a file or printer. Found in the CSL and the TELESUP account.
QUAD	A quick "full-screen" editor. Found in the CSL.
SYSINFO	System configuration lister. Found in the CSL and the TELESUP account.
TAPETEST	Magnetic tape testing and certification. Found in the CSL and the TELESUP account.
TUNER	System table usage/capacity reporting. Found in the CSL and the TELESUP account.
UDCLST	UDC listing/re-creation program. Found in the CSL.
VALIDATE	Verify SYSDUMP/STORE tape sets. Found in the CSL and the TELESUP account.
Games	Various and sundry games of varying sophistication. Found in the CSL.

I do not want to take the time or energy to explain how each of these programs work, but rather want to bring them (and many more like them) to your attention. Again, take a look at these sources of programs. I guarantee you will find at least one program that will pay for your access to them.

QUAD.

This is one program that I will highlight from the CSL. Are you tired of EDIT/3000, but do not have the money to buy one of those fancy editors? Give QUAD a try. It is a versatile editor whose main virtue is the ability to text even the largest files instantaneously. As a result, it is ideal for listing or searching files and for making simple changes. However, it also has sufficient power to handle most editing needs.

QUAD's features include (as of the January 1988 version):

* The ability to undo any or all editing changes.
* Limited full-screen editing capability.
* Maintenance of multiple versions of the file being edited.
* Fast (sometimes instantaneous) keeping of files.
* The ability to show and keep only the changes to a file.
* The ability to compile programs and execute MPE commands.
* A modify command that allows multiple changes on a single line.
* An extensive help facility.

QUAD is not a true full-screen editor, but it does come very close. And the price is right! Check it out.

Loading Terminal Function Keys.

Make use of the function keys on your terminal. They can be a real time/keystroke saver since they can provide you with one or two keystroke execution of commands. It may seem a little intimidating at first to load them with the information you want, but once you do it a time or two, you will kick yourself for not learning how sooner!

There are three ways to load your terminal function keys with information: manually, programmatically, or with a UDC. Locate and read the reference manual for your terminal to find out how to load them manually or programmatically. To load them with a UDC, you actually need to set up three UDCs: two UDCs that need to be defined once -- probably as system level UDCs, and one (or more) additional UDC(s) defined that use(s) the first two. The two system level UDCs should be defined as (the characters <esc> should be read and typed as the escape key):

```
SFK Key=1;Attr=0;Head1="        ";Head2="        ";Length=40;&
    Function="  "
Option List
Comment <esc>&f!"Attr"a!"Key"k!"Length"L!Function<esc>M<esc>A
Comment <esc>&f!"Key"k16d0L!Head1!Head2<esc>M<esc>A
**
UserKeys
Option List
Comment <esc>&jB<esc>M<esc>A
**
```

The SFK UDC accepts the information for one function key and "loads" that information by causing it to be displayed on the terminal with the COMMENT command. Be careful when you type this in: upper and lower case makes a difference in how it will execute! The KEY parameter signifies which function key (1 through 8). The ATTR parameter indicates what should happen when the function key is pressed:

0 = (Normal) The defined string is displayed. To execute it, the user must press the RETURN key.
1 = (Local Only) The defined string is displayed; however, it can not be executed.
2 = (Transmit) The defined string is displayed and immediately executed.

The HEAD1 and HEAD2 parameters provide values to be placed in the labels on the screen (only for terminals that can "label" the function keys). LENGTH indicates how many characters are in the function string. And finally, the FUNCTION parameter provides the actual character string to be "loaded" into the function key.

I have used the COMMENT command twice in this UDC because we have a mixture of terminals and not all of them have the capability of labelling the function keys. The first (longer) COMMENT command will work on all the terminals and will cause all the information to be loaded except for the function key labels on the screen. The second COMMENT command provides the additional label information to those terminals that can accept it (the older model terminals just ignore it). It must be split in two steps; if combined into one, the older model terminals will not have any of the information loaded into the function keys. If all of your terminals have the function key labelling capability, you may combine them into one COMMENT. On the other hand, if none of your terminals have this capability, the second COMMENT could be left out and the HEAD1 and HEAD2 parameters could be eliminated.

The USERKEYS UDC simply causes the function key labels to be displayed on the terminal screen (if they are not already). Again, this is ignored by the older model terminals, but is needed for the newer ones. The "<esc>M<esc>A" sequence in both UDCs effectively "erases" the comments from the screen as the UDC executes.

Here is an example of what the third UDC might look like:

```
SetMainKeys
SFK 1,0,"          "," QUERY  ",,"Run QUERY.PUB.SYS"
SFK 2,0,"          "," DBUTIL ",,"Run DBUTIL.PUB.SYS"
SFK 3,2,"          ","SHOWJOB ",,"ShowJob"
SFK 4,0,"          ","FORMSPEC",,"Run FORMSPEC.PUB.SYS"
SFK 5,0,"          "," EDITOR ",,"Editor"
SFK 6,0,"          "," SPOOK  ",,"Run SPOOK5.PUB.SYS"
SFK 7,0,"          ","SEGMENTR",,"Run SEGMENTER,PUB.SYS"
SFK 8,0,"PrepSave","Program ",,"Preps "
UserKeys
**
```

You may have any number of UDCs set up like the SETMAINKEYS UDC -- each of them causing different information to be loaded into the function keys. With a little bit of thought and creativity, you can even develop a "menu" system with only UDCs and your terminal function keys.

JCWs.

Job control words are very seldom understood or used by new (and often experienced) users. A JCW is MPE's way of permitting programs and commands to communicate with each other within a given job or session, and thus provides you with a tremendous amount of power to control the "logic" of a batch job, or even a UDC. JCWs are unsigned integer variables used at the operating system level with values ranging from zero through 65,535. Each JCW has a name and can be set and/or interrogated either by MPE commands or programs.

By testing JCWs against specific values, the user can "program" conditional statements that take action(s) based on the results of the test. JCWs can be set to predetermined values to indicate completion of steps within a procedure, or they can be checked to determine if certain events (usually errors) have occurred within MPE. Take the time to learn about JCWs; it will be time well spent!

PROTOS. (Fourth Generation Programming Language)

I know I won't make many friends from all of the other "camps", but for our installation, PROTOS, from PROTOS Software Company, has proven to be an extremely flexible and useful tool for application development. It has the advantage of generating a complete, structured, COBOL II program resulting in compiled code (as opposed to interpreted) which means that the application typically runs faster than it would if developed with other fourth generation languages. PROTOS has handled every task I have ever needed to do. If you are looking for an application development tool, check it out; it will probably fit your needs as well!

RUGs.

If you are not already involved in a regional users group, search one out and become involved. Start a RUG if one does not exist in your area. RUGs are an extremely good source of information about your HP 3000. By talking with other users, you will find you are not alone, and you can often solve each others problems! Everyone has something to offer to (and receive from) a users group. Step forward and volunteer to do something -- your RUG leader will be eternally grateful to you!

INTEREX Conferences.

Just like RUGs, but on a much grander scale. The quality of speakers tends to be very high, and the amount of information that can be gleaned in one week is tremendous. Also, having easy access to vendors in the exhibit area makes your comparison shopping much easier. Do not pass up these golden opportunities. This is a chance to make lots of contacts and maybe even have a little fun!

Present a Paper at a Conference.

Hear me out on this one. Before you give me some excuse like, "I don't know enough," let me remind you that your employer thought you knew enough to give you a job. Virtually all of you have enough knowledge about <u>some</u> topic to teach someone else something new. In the process of preparing for the presentation, you might even learn a little more. Besides, it is fun and rewarding!

Write an Article.

If I can not convince you to present a paper, then at least consider jotting down your thoughts and ideas and sending them off to one of the publications for their consideration. It is not as difficult as it seems. You do not have to be an "English major" to write an article. All of the publications have editors that are very willing to help you smooth out some of the rough edges of your article. This is a great way to gain some recognition from your peers (how many people do you know that have had something published?), receive a monetary reward from the publisher, and get a tremendous ego boost. Who knows, you may even become famous!

Closing Thoughts.

So there are the areas I avoided, that I wish I had checked into sooner. Any of them sound familiar? All of them were areas that I thought were too complicated, or for some reason, just never quite got around to investigating. When I did finally investigate, every one of them resulted in higher productivity, provided an easier way of accomplishing something, or, improved my self esteem. Tell you what: You make some (or all) of these your top priority, and I will go to the dentist... next year!

"HAPPILY EVER AFTER: EDP AND THE BIG BAD AUDITOR"

Michael Marcotte
University of Oklahoma Foundation, Inc.
100 Timberdell Road, Norman, OK 73019
(405) 321-1174

Introduction

An EDP audit is a bit like the fable "The Three Little Pigs". There is one EDP shop made of straw by the laziest of the little pigs, which the big, bad auditor quickly finds insecure. Next, there's an EDP shop made of sticks, which under most circumstances would hold up fine, but still can't withstand an unforeseen occurrence. And lastly, there's the brick EDP shop, which the industrious little pig took great pains to build, and aside from serving its primary purpose, it doesn't leak, keeps out intruders, has much fewer bugs, and for all the huffing and puffing of the big, bad auditor, it still withstands the assault.

This paper looks at the EDP audit from the viewpoint of the auditor, as a starting place for examining several tools and techniques, specifically within an HP3000 environment, that can be used to address many of the risk areas examined during the audit. The paper addresses primarily the review of EDP performed in conjunction with an annual financial audit, rather than an EDP audit conducted as an in-depth review of current systems status by a consulting firm, though many concepts will still be relevant.

Specific programs and utilities are discussed which may already be a part of your systems software (but which may not be activated), as well as features of application software (but the auditor didn't specifically ask, so you didn't mention), CSL contributions, Telesup account utility programs and some relevant third party utilities and software packages. Also discussed are certain easily implemented practices and techniques that can improve your ability to provide references for implemented controls, and better understanding on the part of the auditors of the full range of security and controls within your systems.

This paper does not propose to address the design for a "foolproof" system of security or EDP controls, though both areas are discussed in relation to an EDP audit. Nor will every control and tool discussed herein be applicable to every size shop, and/or situation. Every company has its own unique set of internal policies, procedures, and mix of hardware and software and must therefore tailor the approach(es) discussed here into some subset or expanded set of controls, tools and practices that will adequately address those issues and identified risks most likely to be included in an EDP audit of a company of your own particular size, nature and composition. Depending upon the particular environment of the reader, some of the controls and solutions presented may require additional procedures or processes in order to sufficiently address a certain area of risk.

The EDP Audit

An EDP audit may be many things to many people. To you, it may be an annually recurring source of migraines, or a security/controls/backup/recovery questionnaire that goes on forever and ever. To your boss, it may be your job evaluation. To best understand the purpose of an EDP audit and therefore be best prepared to survive one, it is helpful to try to view the objective of an EDP audit from the standpoint of the auditor.

The auditing principles and standards that are employed in reviewing manual records remain applicable to an audit of records maintained via a computer system; it is primarily the audit approach and testing procedures that will differ. The EDP environment may have a sizable impact upon the amount of risk determined to exist. The auditor therefore will need to assess the extent to which EDP contributes to the preparation of financial statements, and to what extent the controls present in the system may be relied upon.

Generally speaking, when the audit is conducted in conjunction with the annual review of the financial statements, the auditors are primarily looking to obtain a certain level of comfort with the security, automated controls and EDP policies and procedures present in the system. The audit's objective is not in this case to verify, with complete assurance, the accuracy or sufficiency of the automated controls, but rather to measure the impact of EDP upon overall and/or specific risk assessment. This type review is searching for any lack of accountability, areas that could be suggested for improvement, and for any gaping holes in automated controls, security, recovery, etc.

An EDP audit conducted independently from a financial audit usually has a different objective, a specifically defined scope, and while conducted in substantially more detail, will have been contractually arranged for the purpose of reviewing certain areas that are of concern to client management. The emphasis of the independent EDP audit will therefore vary from case to case. The prevailing philosophies also appear to vary somewhat among accounting firms as to the extent of the reliance to be placed on computerized controls versus managerial practices and procedures.

It is worth stressing that regardless what you the System Manager/Security and Controls Manager/Accounting Applications Manager think is the true mark of a secure system with strong data integrity and strong controls, it may not be the same as what the auditor thinks. Sometimes as systems personnel it may be hard to see the value of procedures which hamper throughput or exist in place of a more polished technical solution. Management may be more prepared to accept such trade-offs in the interest of protecting assets, especially when the procedures have been recommended by the external auditors. The opinion of a large CPA firm can carry a lot of weight.

At past INTEREX conferences, I have listened to several related seminars and to attendees expressing varying opinions as to the relative technical merits or disadvantages of different control approaches over one another and how to know which method was best from an audit standpoint. In all but an extremely rare instance a "typical" auditor would have had difficulty in following the discussions, except at a very general, or conceptual level, just as most EDP personnel could not, in the vast majority of cases, knowledgeably discuss the latest FASBs or audit methodology. In the case of a special EDP review by an external consultant, obviously the auditor would, or should have an EDP background, if the end objective is to resolve existing system problems or to suggest EDP improvements.

This is not to say that a CPA firm's financial auditors are generally unqualified to conduct an EDP audit, but rather that their orientation is explicitly different. These auditors, practically speaking, do not need a great deal of technical system expertise, and it would be uncommon to find that the auditing firm had spent a great deal of money and time attempting to train their staff in the technical aspects of a certain brand of hardware or software when their clients' systems vary so widely. The exception might be where the auditors included on the audit team one or more persons from their MIS consulting practice. Even such individuals would rarely have been trained or experienced in much depth on the HP3000, given the few consultants who would have been involved in the technical aspects of the security and controls design for this particular machine, the particularly high turnover of personnel from such firms, and the tendency of the Big Eight firms to primarily utilize recent college graduates at the audit staff level.

It is not, strictly speaking, the job of the financial auditor to understand or to evaluate, in any great detail, the technical features of the various controls. That is **your** job. The auditor should, on the other hand, have a basic working knowledge of computers and computer-based transaction processing, as well as some preparedness for testing the internal system controls. When the EDP audit is performed in conjunction with the annual audit of a firm's financial statements, the job of the auditor is to examine, on a test basis, evidence <u>including the computing environment</u> to determine whether the financial statements fairly represent the status of the company, and that the company is utilizing generally accepted accounting principles to arrive at the presentation of those statements. The audit does not, or should not, delve into the technological merits of one technique over another, and can not encompass a 100% compliance test of all data, or even sample **each** and **every** system control feature. Even an EDP audit conducted independently relies upon a sampling of the various controls utilized within the system, and a random, testing of those controls.

During an audit, the auditor must find with reasonable assurance that such controls do, in fact exist, and that they appear sufficient to maintain the integrity of the data contained in and presented by the system, and that these controls are being enforced. Once this is understood, it should become a more simple task for EDP to demonstrate the appropriate controls that match the risk areas identified by the audit team. It should also be understood that the EDP audit is not necessarily entirely oriented toward the detection of lack of accounting controls, but may in fact provide an increased level of assurance that the system fills in certain gaps in manual controls, while at the same time providing a means for greater audit efficiency.

A Few Words About Risk

The reference to risk has already been made a number of times. The concept of risk is employed by the auditor to define an area or a situation within the client company where a potential exists for loss of assets or income due to any number of reasons, including negligence, fraud, malice, lack of information or use of inaccurate information, human error, or even natural disaster.

The EDP audit specifically reviews the information processing function of a company to assess the levels of risk associated with the various system components, including personnel, and to determine the extent to which existing controls satisfactorily eliminate or limit such risks.

The auditing profession subcategorizes risk into three areas: 1) inherent risk, or the risk that accounting information may be susceptible to material misstatement, 2) control risk, the risk that internal controls will not detect or correct such misstatements on a timely basis, and 3) detection risk, which is the risk that the audit may not uncover such misstatements. For those unfamiliar with accounting terminology, material misstatement might very loosely be defined here as "nontrivial error", and be grouped as either unintentional errors occurring somewhere within the processes leading up to the inaccurate preparation of the financial statements by the client, or 2) fraud or accounting irregularities. EDP is examined to the extent that preparation of the statements is dependent upon the system, and therefore might be the source of the error(s), and for the purpose of examining whether the system could be the means by which fraud or an irregularity might be perpetrated and concealed.

The audit, in reviewing the existence of or potential for unintentional errors will concentrate on proper initiation, uniform processing, staff skill levels and internal processing controls over normal and exception transactions. In reviewing for the potential for, or existence of fraud, intentional misrepresentation, or other irregularities, the auditors will look for proper authorization procedures, supervision, division of various system responsibilities, audit trails and controls for detection of improper system activity.

Risks are identified by the auditors at varying levels of severity. As should be expected, more effort is made during the audit in verifying the presence and adequacy of the controls over the areas evaluated as posing the more serious risks.

In several instance the auditors will view risk as actually being reduced by the presence of automation. Most errors in a mature system are of the human type, and the reduction of human intervention may actually enhance the reliability of the overall system. Also, where the auditors have become comfortable with the reliability and integrity of the system, less detailed testing may be required, if it can be shown that no modifications have been made since a prior audit.

Survival

Talk to the Auditor:

It is a good practice to have the System Manager establish an open line of dialogue with the audit manager and audit senior assigned to your company. I was told by one auditor that if any one point came across from this article, they hoped it would be the importance of communication. Offer your assistance in assuring the audit team access to you, your staff and the reports that the auditors may need in the course of their work. Do this sincerely. It is highly preferable to be somewhat smothered by questions for a couple of weeks out of the year by the auditors, than similarly questioned after-the-fact by management regarding "system deficiencies" even if you feel that you have a good defense.

Make certain that you have the Audit Manager's agreement that any identified EDP deficiencies are communicated to you before being distributed to top management. This will not only give you the opportunity to prove otherwise, but also allow time to identify and and possibly implement/demonstrate a solution, i.e.- "It has already been taken care of". Another advantage here is the chance to review and possibly affect the ultimate wording of any risks or deficiencies that you are unable to disprove, or find a solution for. The auditor may not be aware of the effect that a certain phrasing has upon other readers.

You may have found yourself in the situation in the past of having a new set of personnel on the audit team each year. This tends to increase the amount of effort that you must expend in "training" these people each year, explaining the same EDP concepts, and control mechanisms each time to a new person(s). You may find it helpful to comment to the Audit Manager how much smoother the audit seems to go when they have a person or two repeat from year to year. If this doesn't work, at least ask the auditors to document particularly cumbersome techniques and explanations within their work papers for the benefit of their successors the following year.

Some of the Big Eight promote their services via the concept of a "client service team", i.e.- staff members from the audit, tax and management information consulting divisions, assigned to a team that supports the client in a more balanced sense, pooling their combined knowledge to the greater benefit of the client. If such is the case, hold them to their word. Ask them to help educate the audit staff with some of the explanations, especially if their MIS consultants helped select or develop the hardware or software, or helped design the system controls.

A word of warning about open dialogue with the auditor: Don't talk **too** much. A response to a hypothetical question about your abilities to circumvent a control if you really wanted to, may come back to haunt you. Don't boast of capabilities, or of controls that you aren't prepared to back up. If your security has holes in it that would allow someone with your abilities to breach it, spend time prior to the audit looking for a way to fix them, rather than boasting about your prowess. You may know that you would never be tempted to do anything fraudulent, but the auditor doesn't care. An opportunity for fraud or a breach of security to exist is still an area of risk.

You obviously should not lie about the ironclad-ness of a certain control. There are, of course, some who can still break through many of the controls discussed later in this paper. The important thing is that you try to have enough security in place that **one person** would not have the capability or knowledge to breach security and cover all his/her tracks, or be able to detect all of the controls that may not be advertised by word-of-mouth.

What can you do to prepare?

Some accounting firms may take a generic approach and look at the same EDP issues for all sizes and type of firms, even those with broadly differing sizes, styles, and personnel organizations. Most of the Big Eight accounting firms now use a preliminary questionnaire, completed by the EDP manager, or by both he and the Audit manager, and then used to manually select the appropriate detail checklists, or as input to a software program that will generate a more detailed questionnaire tailored according to the afore-mentioned factors, as well for the specific vendor and sophistication of the system hardware, software, DBMS, etc. The more sophisticated the EDP structure and computer system, the more EDP is felt to impact the financial audit due to the client's increased reliance upon the computer system, and therefore the heavier EDP will be tested to verify such reliance is warranted.

It is a good idea to make a copy of your completed questionnaires and to retain them for reference for the following year. Look at the old questionnaires well in advance of the expected arrival of the audit team in order to allow time to update any controls, documentation, passwords, etc., that may have grown somewhat out of date. In fact, it is good EDP management at any rate to use such a questionnaire on at least a semi-annual basis to assure yourself that such controls are being kept up to date. If you have to answer "no" on a questionnaire to whether a specific control exist, add a footnote that explains that you are addressing that point via an alternate control, or why the control is not as important at your type, size company, etc.

Specifics:

* <u>EDP Organization</u> - Perhaps the best EDP control of all is segregation of duties within the EDP department. The audit team will want to see an organizational chart that shows the various reporting relationships and this segregation of duties, so create one and keep it up to date. Specifically, the job of security administration should be separated from programming, file maintenance, operators, etc. The EDP staff should not have duties in other departments. Similarly, systems programming should be segregated from application programming. Obviously, a small company cannot perform these levels of segregation. Security administration should still be handled by a different individual than the programmer. If only one EDP person exists and handles all aspects of system support, including security, then some other level of higher management such as the Director of Finance, or the Controller should be required to authorize all programming changes, and provide other types of supervision. Such supervision should be evidenced in your documentation by written approvals, memorandums, etc. Written authorizations should apply within larger EDP staffs as well. Regardless of the objections posed frequently that suggest such procedures are cumbersome and delay implementation, waste management time, and wind up as a rubber stamp process, where no one is truly reviewing what they are approving, such authorization is an important control. Lazy implementation of such controls are the fault of management, and will probably eventually eat their lunch!

Another factor which should be considered when assessing the importance of segregating EDP duties is the extent to which the auditor feels he or she may rely upon many of the other controls. The extent that the other controls prevent or detect any possible irregularities involving EDP personnel is, in the first place, dependent upon the fact that the establishment, maintenance and monitoring of those controls are not all performed by the same individual(s).

Security - Security is a frequent topic of papers and presentations within and outside of INTEREX. Certainly those papers will address this risk area in much more depth, than I intend to, in this particular paper.

Obviously, the auditor will want to be assured that passwords and lockwords are used at the various levels within the MPE account structure. Certain fields or files may also be encrypted. The auditor will invariably ask about the frequency of password changes. An auditor with more familiarity of the HP3000 will also inquire about the passwords on such accounts as Telesup, often left unchanged, and therefore open season for anyone trying to breach your system. The same auditor may have some knowledge of the dangers of privileged mode. You will want to have educated yourself on those same dangers, and taken the relevant precautions. The INTEREX proceedings are often a good source for this type of information, and may also serve as documentation of the solutions you have implemented. Auditors love documentation. It is not necessary to recreate the wheel when supplying such documentation.

A good example of this are the MPE manuals. Don't overlook the documentation provided in the MPE Account Structure and Security Users Guide contained in Binder 1 of your MPE manuals. The descriptions contained therein should provide in more than sufficient detail, for the auditors, a view of the additional security often not mentioned during the audit, but nevertheless provided via user, group and account capabilities, and read/write/execute access restrictions.

On MPE/V, LISTDIR5 used with LISTUSER, LISTGROUP, etc. can be used to interactively demonstrate to the auditor these various capabilities, and your implementation of them across various accounts, beyond the simple password security provided by MPE. Though LISTDIR5 apparently is no longer around once on the Spectrum series, similar abilities exist.

Other third party software is also available, such as VeSoft's SECURITY package, for the shops with security requirements exceeding those provided by MPE. Don't overlook your own built-in controls such as tailored menus, or different UDC files for different classes of users, or custom software that in whole, or even in part, verify access authorization against an internal table or master file.

Remote access should be carefully supervised. Attempts to break into computer systems for fun are seemingly a great enticement to certain college students and hackers; sort of like state-of-the-art graffiti. Modem ports and remote terminals may be DOWNed after hours, or modems with a callback feature can help prevent this sort of unauthorized access. Again, there are several articles on security that offer many ideas on this specific topic.

There is such an abundance of security features readily available on your system or from TELESUP or the CSL, that you should be able to even overwhelm the auditor with your controls, if you can in fact demonstrate that you have implemented them. Passing an EDP audit may be one thing, ensuring that all these controls are sufficient to, in real life, secure your system may be another. In fact, the audit may be the much easier of the two. Certainly, however, the manager who uses past audits as an independent source for identification of potential weaknesses (rather than a annual nuisance), who looks for respective solutions, and who stringently enforces those controls boasted of during the audit, will be better prepared to preserve the security and integrity of his or her company's data.

System Modifications/Enhancements/Version Control - Some sort of record should exist that tracks all programming changes, for both application software and system software. Several software packages exist that can automate this task, both on the PC, and the HP3000. Alternately a manual system can be employed without becoming overly voluminous if the company is small or has a fairly stable system with little new development.

Appendix B includes a sample of the type information that should be recorded, as well as some sample logs. Such systems can also be used to record any abnormal types of non-programming modifications such as Editor changes to production files, restores of accidentally deleted source or object code from backup tapes, etc. At a minimum, such systems should record the date the change was made, name and number of the program modified/added, the name of the persons requesting the change, implementing the change, and approving it, a brief description of the enhancement or modification, and some explanation as to why the change was required. Version control may be implemented in many different manners, depending on the standards used in your own particular shop. At a minimum, the program's source code should include a comments section used as a history of changes made to the program. These comments should reflect the date, author and brief nature of the change. Some method of comment should be used within the source code as well to flag added/modified lines of code, and to tie them back to the descriptive history section. A example of one method is shown in Appendix A. Some of the newer CASE software now on the market incorporates all or most of the above features into their product. Auditors thrive on documented proof of the various controls in place within a system, so the methods described above are useful in demonstrating that changes are authorized, documented, and traceable. It then becomes easier to identify any programming changes done out of malice, fraud or to shortcut other system controls.

An example of one utility that enhances the above processes, is VeSoft's MPEX Listf,3 which may be used to list all files that are object code or source code, with the respective dates created, date last modified, and date last restored. This listing makes a useful report to provide to auditors requesting a summary of all program IDs modified since the prior year's audit. Critics will no doubt point out that it doesn't take much of an expert to temporarily change the system date while a program is being modified, in order to give the appearance that a change was effected on an earlier date, but it similarly it shouldn't take much of an expert security administrator to keep this from happening, either: Restrict access to CLKPROG and similar programs via passwords, remove them, or place them in a group or account outside the programmer's access. Add other detection-oriented controls that only top system management and the security administrator are made aware of. Let the auditors know you are running background software that the programmers don't know about. Enforce passwords, use lockwords, control the use of privileged mode, segregate duties. The auditors are primarily testing the fact that you are tracking program changes, and have implemented and are enforcing controls to detect non-authorized changes. They realize that in the short time they have, they are not able to prove the absolute security of all controls or even a single control in all cases.

Use a separate account or different CPU for development work. Good EDP management means having adequate testing procedures, standards, and resources to make sure that an addition, enhancement or modification is thoroughly tested before coming anywhere near the actual production files. These standards and procedures should be well documented and made available for the auditors' inspection on request. Also, the separation of development work, via a test account or processor, provides a means for and helps further enforce the segregation of programmer duties from other positions. If you need a source for programming and testing standards, the respective sections of the EDP audit questionnaire can even be one initial step.

<u>User Involvement</u> - Be prepared to offer evidence that users have been actively involved in the design/selection of software. Auditors want to be assured that the system is providing results that the end users find dependable, before they are likely to put the accounting firm's own approval on the system's dependability. Again, written user sign-offs in development project documentation is the best proof of this. Also, a lack of user involvement is likely to produce poor oral confirmation from the users concerning the EDP department's effectiveness and the system during the auditors fact-finding sessions. Disenchanted users are likely to both misunderstand and misrepresent the controls and reliability of the system.

<u>Backup/Recovery</u> - The auditors will want to know if you have a regular system backup schedule, the retention period for stored data, and offsite storage of backup tapes. The answers to all of these questions should be easy. Most knowledgeable sources recommend a full system backup at least once a week, with daily partial backups storing files updated since the last full backup. Many systems have built-in or customized checkpoints where additional backups are performed of certain files immediately prior to critical processes such as check writing, or posting to the general ledger. There doesn't appear to be any uniform wisdom as to how long regular backup tapes ought to be stored; some shops says 6 months, others only 1 month. Partial backup tapes are frequently rotated on a two to three week basis. Critical backups such as Month-end, and Year-end/Pre-close tapes should be held for much longer periods of time. The "correct" retention period is correct only to the extent that its long enough to enable the specific company in question to reasonably recover or retrieve data when the need arises. It will depend on the particular nature of each company, and may be affected by other related controls.

Offsite storage is available in most larger cities from vendors specializing in such services, but no one should opt to ignore this important safety feature. Even the most remote, and smallest of shops can at least store tapes offsite at the system manager's home. EDP audits are also increasingly looking for the existence of documented disaster recovery plans. Many articles have been published expressly for the HP community on how to develop your own disaster recovery plan. The process need not be costly nor excessively burdensome. The EDP audit should only be a secondary motivation for developing this life(job-)saver. Recently there has been more and more attention being given to the issue of whether the responsible system manager might even be held legally liable for failing to provide such a precautionary plan. Like backup tapes, at least one copy of the recovery plan should be stored offsite. EDP auditors may also look for documented recovery procedures for more temporary system failures, such as CPU, data base, KSAM or workstation failure in mid-process, rejected transactions, program aborts, etc. Many such recovery routines already exist in the Hewlett-Packard or software vendors manuals and need not be duplicated. It is good, however, to reference the manual and section where such procedures are found in your own documentation. One excellent index for this cross-referencing is the copy you will have made of last year's EDP audit questionnaire.

The Telesup account contains several useful utilities that are available to you and which ought to be referenced as additional controls/safeguards to your backup/recovery procedures. BULDACCT, for instance, is a job that would use your most recent full backup tape, to extract account structure data and then build three jobs that would, in turn, re-create your accounting structure.

VALIDATE is another Telesup program that many of you already use, and which should be referenced as a tool available for verification of the usefulness of backup tapes. Similarly GETFILE is a program useful for retrieving files from backup tapes which may have been partially damaged, or become otherwise uncooperative. TAPLIS, also in the Telesup account, and STAN, from the INTEREX CSL are tools for replacing lost or destroyed listings of the contents of a magnetic tape.

Physical security - Another area of attention during the EDP audit is the physical safeguards assigned to the actual equipment, the computer room terminals, tapes, and documentation. The auditor is looking for assurance that only authorized personnel have access to the computer room, usually controlled via locks and direct supervision, and the other system resources. Special locking devices exist for keyboards, PCs and printers, but most often the placement of such devices where they may be directly supervised by management and authorized personnel is sufficient for the auditor. In the case of the largest companies, it may be necessary to maintain a system of badges and sign-ins to adequately ascertain who should be in a certain area with access to terminals, PCs and printed output. As in previous instances, such procedures should be documented to best satisfy the auditors. System documentation such as control memorandums, data layouts, source listings, etc. should be located in the office of the person(s) responsible for those areas, and be secured during off hours.

Processing controls - This area of the EDP audit addresses the adequacy of the internal software safeguards. The auditors look for examples of input controls, such as interactive validation, cross-referencing of data items for reasonableness, upper and lower limits, and automated totals that are provided by the system for reconciliation against manually-calculated totals. The opportunity for the clerical errors normally associated with manual handling is mostly eliminated when transactions are uniformly processed by the computer, therefore a greater reliance may be placed upon results so processed, when it may be shown that all transactions are subjected to the the same processing controls.

Batch totals such as record count and total dollar amount are among these types of controls. It is not a bad idea to consult the audit firm for an opinion of the controls that should be included in projects that are in progress, or for assistance in evaluating the controls to be sought in new packaged software that you may be considering. A little pride of ownership in the design process by the auditors could go a long way, not to mention the comfort factor that the auditors would experience with an application for which they had helped design accounting controls. Before/after image logging, and date and time stamping at a transaction level are desirable where practical. Even if impractical on a broad basis in the system, it is helpful in the audit process if you can point out such features on the very critical files/datasets. Output controls should be in place to establish and monitor distribution. As with all of the above, written design standards, as well as documentation of the procedures controlling input, processing and output, will shorten the time spent in demonstrating controls.

Lest the continued emphasis on written documentation of controls sound like a waste of time, consider the effect of continuously being able to refer the auditor to your standards, security/control procedures and recovery plans, i.e.- "Didn't you read our controls documentation!"

Other MPE, CSL and Third Party Tools - These controls fall under the category of things the auditor may not specifically be looking for, but which may provide alternate or better controls against the same risks. A couple of utilities in the INTEREX Contributed Software Library that are great EDP control tools are ENFORCER and BOUNCER5.

ENFORCER is a small data base tool that allows the implementation of increased security via a specific or set of users/accounts and/or device ID. For instance, you can establish an additional password for your modem devices, still another for a certain group of users, user-specific logon messages, etc. One of ENFORCER's drawbacks is the increased delay which it causes when in effect. ENFORCER is activated via the OPTION LOGON feature in a UDC file set for an account, user, system, etc. One method around this if the additional security is desired only in off duty hours, is to activate a different UDC at 5 p.m., then reactivate the original at 7:30 or 8:00 a.m. There are undoubtedly many more ways that involve more creative or more sophisticated touches to accomplish the same objectives, given that you have personnel capable of developing them.

BOUNCER5 is a tool which can be used to bump users off the system when there has been no activity from that terminal for a specified length of time. OCS offers the same feature in their software, as does VeSoft's SECURITY. These tools offer the safeguard that an active session is not left unattended and tempting to unauthorized passers-by for any length of time.

Computer Consultants and Service Center, Inc. (CCSC) offers an inexpensive product called RAS/3000 which actually is a resource accounting package used for chargeback purposes. The package dumps MPE log files into a data base which is then accessible for activity reports by job, user, account, etc. The reports may be used to scan for unusual activity within certain accounts or group or by specific users or between certain hours. The package also comes with a neat little utility called DISPATCH which works similar to OCS' scheduler to allow you to schedule and launch jobs at certain times in the evening, on let's say every Friday, or the first day of the month, or nearly any date/time combination. CCSC lets you keep DISPATCH when you demo RAS/3000 whether you buy the package or not (check with the vendor to see if this offer still exists). Those who like to mess around with MPE and UDCs can always set up their own delayed scheduling system using the MPE STREAM command along with the AT/DAY/DATE/IN parameters. Such delayed launch capabilities allow added controls that also address EDP audit questions. For instance, you can DOWN all remote terminals at a certain hour and on weekends/holidays without operator intervention. You could switch to an abbreviated system-wide UDC, or one that displays a more limited menu than during normal working hours, etc. JOBQUEUE, within the Tech account of the INTEREX Contributed Software Library is another batch job scheduler.

CONSLOG and CONSLOGX are other CSL programs that provide increased controls and address EDP audit questions. CONSLOGX, in particular, provides considerable flexibility in reporting a hard copy of all console messages (provided that Console logging is switched on in your system logging table). CONSLOGX also allows you to print only those messages containing a certain string, such as "INVALID" or "VIOLATION", for a quick review of potential security violations which may have occurred overnight, or over the weekend, for example.

The TELESUP account also contains programs (LOGUTIL, LOGSNAP, et al) that can be used to provide hardcopy console logs. SCANLOG and SCANUSER in the Tech account of the CSL are similar programs, that like CONSLOGX, allow selective scanning of logfiles for specific user activity, file closes, etc. These tools are useful for providing a means of identifying of at least some measure of the activity that might have occurred in the event of a successful breach of your security. In terms of the audit, you have all these tools to help keep unauthorized persons off the system or out of specific accounts, but in the event that a loophole could be found, you also have all these tools to help identify what a perpetrator would have accessed while on the system.

Other programs in the CSL and from third party vendors offer "compare" features similar to FCOPY's. These can be referenced as tools for detecting changes to source or object code as well as production accounting or master files, if a violation of security or accounting controls is suspected. Before/after image logging or reporting should be referenced as a means of control over this risk area, as well.

Another control that may not be specifically asked about is the ability to keep the users out of the command level of MPE. One method of doing this is via the OPTION LOGON,NOBREAK command in the respective UDC to move a user straight into a menu program or master program run from the UDC following this option, then to exit via BYE still within the UDC. RPG users should also investigate PROCMON on later versions of MPE/V.

Note that while some of these tools add security and controls, the presence of others such as SUPERZAP, or GOD may be viewed by the auditor as a threat to system security themselves due to the capability to accidentally or malevolently wipe out massive numbers of files with a single command. Such tools should be restricted to the accounts/groups of responsible individuals with a real need for their use. An additional control over their usage would be to add lockwords.

System Methodologies - System methodologies are useful in providing structure to the various planning, design and implementation phases of system projects. Auditors look for the presence and use of such methodologies for evidence of control/coordination over the multi-user development processes on-going in such projects. The absence of such methodologies or the ignoring of existing methodologies can signal the presence of additional control weaknesses to the EDP auditor as well. Some experts are beginning to indicate that methodologies are perhaps becoming less necessary given the trends toward CASE products and end-user computing, and the right combination of tools is more important than the presence and use of a systems methodology. Auditors however tend to trust documented procedures and methodologies, and therefore are likely to be seen as a plus. In fact, one CASE product offered by a Big Eight firm itself is heavily integrated with an automated version of that firm's own proprietary systems development methodology.

Some of the steps which exist in certain methodologies are obviously more useful, and more practical at larger companies, than they are at smaller firms. For example, prioritizing system projects is meaningless if there are no projects pending. Other tasks, though at first glance appearing to be intended for large companies, such as the development of a Systems Plan, still have considerable value for the smaller firm, though the total output of such a plan is likely to be much, much more abbreviated and entirely able to be accomplished in-house. Since the EDP audit will concern itself to some degree with the ability of the system to keep pace with evolving business requirements, a documented Systems Plan, can demonstrate to the EDP auditor the extent to which the EDP department is addressing any obsolescence of hardware, software or personnel education which might affect the overall dependability of the system.

<u>Source Documentation</u> - The EDP department, user departments and the Auditing firm must communicate sufficiently between one another, that each is aware of what if any source documentation is disposable after it is entered into the system. Most source documentation, even data entry input forms require some period of retention. Other source documents, such as invoices, revenue distribution stubs and others may require a substantial retention period, a matter of policy for accounting management rather than EDP. EDP staff and management should be careful at any rate, to avoid contributing to the mistaken idea that the computer is going to replace all of the paper, lest the auditors pick up on a misunderstanding, and incorrectly report that critical source documentation is being destroyed, once the data is on the computer.

Summary:

You can only accomplish that which you are given the **tools** to do, and that which you have the **inclination** to do. The first case may be partially out of your control, though certainly there are several inexpensive and some contributed tools that can greatly assist you. The second case is totally in your hands.

Think ahead, design programs, systems with an eye to security and controls, and with future access/security possibilities in mind.

You may find that a timely spoken word about a security or control feature (that has been bugging you, and of which you are concerned may come up as an audit point in the approaching fiscal year audit), may be all that is needed to get the green light for acquiring the tool that provides the solution.

Auditors sometime feel that they must provide added value to the audit via suggestions for improvement. Be prepared for some such points. Suggest alternate wording if theirs is offensive. They will generally work with you if you have worked with them during the audit.

Lastly, an EDP risk is cited in the auditors "Suggestions for Improvement" is not necessarily likely to adversely affect a favorable opinion by the auditing firm, as long as the audit turned up otherwise sound accounting practices and procedures, and other strong EDP controls. A risk is not a condemnation, and a suggestion does not necessarily indicate a deficiency. Nor is it practical to assume that all suggestions are feasible and must therefore be implemented. In such cases, it is mostly important that you, and top management understand the spirit in which such suggestions have been made, and that nothing more is expected than that management understand the risks, and evaluate the extent that it may be practical and/or desirable to remove these risks.

The EDP audit does not have to be a bad experience. With proper understanding and communication on both sides, EDP and the big, bad auditor can live happily ever after.

Acknowledgments:

I am grateful to individuals at the firms of Arthur Andersen & Co. and Peat, Marwick, Main & Co. for insights offered during the preparation of this paper.

APPENDIX A

```
$CONTROL MAP,LINES=60,QUOTE=',USLINIT,LIST                             CD540
H********************************************************************   CD540
H*             CONTROL RECORD SPECIFICATIONS                            CD540
H*******************************************************************    CD540
HDUMPFILE     N                                    X1                   CD540
F********************************************************************   CD540
F*       FILE SPECIFICATIONS                                        *   CD540
F********************************************************************   CD540
F*                                                                      CD540
F*       PROGRAM -  CD540 PRINT I.R.S. 1099'S                           CD540
F*                                                                      CD540
F*                                                                      CD540
F********************************************************************   CD540
F*       PURPOSE -  THIS PROGRAM READS THE SORTED CHECK REGISTER        CD540
F*         FILE AND PRINTS A 1099 FORM FOR EACH PAYEE. THE 1099'S       CD540
F*         PRINTED ARE 1099-DIV, 1099-INT, AND 1099-MISC.               CD540
F*                                                                      CD540
F*                                                                      CD540
F*                                                                      CD540
F*.                                                                     CD540
F********************************************************************   CD540
F*       REVISION LOG                                                   CD540
F********************************************************************   CD540
F*       NUMBER    DATE                  DESCRIPTION                    CD540
F*         1      2/13/86       ADD LOGIC TO PRINT 1099 BASED ON        CD5401
F*                               IRS 1099 DOLLAR LIMIT.                 CD5401
F*                                                                      CD540
F*         2      1/30/87       MLM-FIX BUG IN MOD 1 ABOVE; REMOVE      CD5402
F*                                   IND. #20 IN LINES ADDED 2/13/86    CD5402
F*                                                                      CD540
F*         3     12/07/87       MLM-ADD CHANGES FOR 1987 DIV. 1099      CD5403
F*                                                                      CD540
F*         4     12/11/87       MLM-CHANGE EDIT CODE J TO L FOR IRS     CD5404
F*                                                                      CD540
F*         5      1/12/88       MLM- ADD LOGIC FOR IRS MAGNETIC MEDIA   CD5405
F*                                   REPORTING.(1099-MISC ONLY)         CD5405
F*         6      1/27/88       MLM- ALLOW M1 AMOUNT TO PRINT BUT PLACE    #6
F*                                   ON LINE 2; ALSO OUTPUT TO CD5401.  CD5406
F*         7      3/01/88       MLM- ADD PRINT OF PAYEE ID IN OPTIONAL     #7
F*                                   BLOCK ON FORM 1099.                CD5407
F*         8     10/19/88       MLM- ADD 1988 IRS CHANGES                  #8
F*                                                                      CD540
F********************************************************************   CD540
F*       INDICATOR SUMMARY                                              CD540
F********************************************************************   CD540
F*                                                                      CD540
F* 01-                                 51-                              CD540
F* 02-                                 52-                              CD540
```

.
.
.

```
CL4 80      TOTLM     ADD  TOTLM3    TOTLM  112                         CD5401
CL4 80      TOTLM     ADD  TOTLM4    TOTLM  112                         CD5401
CL4 80      TOTLM     ADD  TOTLM5    TOTLM  112                         CD5401
CL4 80      TOTLM     ADD  TOTLM6    TOTLM  112                         CD5401
CL4 80      TOTLM     ADD  TOTLM7    TOTLM  112                         CD5401
CL4 80      TOTLM     COMP 599.99                  59                   CD5401
CL4 80 59   MCOUNT    ADD  1         MCOUNT                             CD5405
CL4 80 59   TOTLM1    ADD  MTOT1     MTOT1                              CD5406
CL4 80 59   TOTLM3    ADD  MTOT3     MTOT3                              CD5405
CL4 80 59   TOTLM7    ADD  MTOT7     MTOT7                              CD5405
CL4 80 59             Z-ADDTOTLM1    MAMT1  102                         CD5406
```

APPENDIX B - Figure 1

System Enhancement\Modification Request

Index: ____________

Reference: (yymm-nn)

Prepared By: ______________________________ Reviewed By: ______________________________

Department: ______________ Date: ________ Date: ______________________________

Type of Request: (Check One) ○ Error ○ Enhancement/Modification ○ New Program

---

Requestor's Priority: Required ○ Important ○ Desirable ○

Date Required: ____________

Description of Error or Request:

Reason required:

Estimated Effort in Manhours:

Analysis/Design: ____________ Programming/Implementation: ______________

Programs/files Modified or Created:

Supporting Documentation: Yes_____ # Pages_____ NO______

Recommendation: Accept___ Defer___ Reject___

Comment: ____

____

Signature: ______________________________________

Date of Final Disposition: ______________

SYSTEM REQUEST LOG

PAGE:____

System Request ID # (yymm-nn)	Brief Description of Request	Date Submitted	Status (Reject) (Compl.) (WiP)	Date Final Disposition

APPENDIX B - Figure 3

SYSTEM LOG: PROGRAMMING CHANGES AND ADDITIONS

INDEX

ITEM

PAGE

PROGRAM ID	DATE INTO PRODUCTION	SYSTEM REQUEST ID	(Check One) NEW	ERROR	CHANGE	COMPLEXITY (E,M,D)	PROGRAMMER	DESCRIPTION/COMMENTS

APPENDIX C

PRIMARY AREAS OF REVIEW FOR AN EDP AUDIT

1. EDP Organization

   a) levels of supervision
   b) personnel qualifications
   c) job descriptions and organization charts
   d) policy and procedures manuals
   e) segregation of duties
      . programming
      . testing
      . operations
      . security
      . data administration
      . data entry
      . management

2. Security

   a) passwords
   b) read/write protection
   c) restricted functional access
   d) restricted physical access
   e) supervision
   f) testing (re: viruses, Trojan horses, timebombs)
   g) equipment location
   h) personnel identification
   i) privileged mode, backdoors, "super" tools

3. Development

   a) new applications
   b) major enhancements or modifications
   c) version control
   d) standards and methodology
   e) documentation
   f) authorization/approval
   h) management/user involvement
   i) testing/conversion procedures

4. Maintenance

   a) authorization/approval
   b) documentation
   c) testing/conversion procedures

d) standards and methodology
e) restricted access to source code and program documentation
f) controls over systems software changes

5. Operations

a) documentation of procedures
b) restricted access to operator functions
c) supervision
d) hardware/systems software error detection

6. Backup/Recovery

a) disaster recovery plan
b) less serious failure recovery procedures
   . hardware
   . software processes
c) backup schedule
d) critical processes special backups
e) documentation

7. Data entry

a) input authorization
b) batch controls
c) on-line error validation and masterfile verification
d) reasonableness checks
e) processing controls
f) output reconciliation
g) audit trails
h) restricted access/segregation of duties
i) user procedures

8. Output distribution

9. Internal Audit procedures/involvement

a) development
b) testing

System Management on the Fly II

Eric S. Messelt
Microwave Applications Group
3030 Industrial Parkway
Santa Maria, Ca. 93455
(805) 928-5711

It had all the ingredients for a Data Processing disaster. A fresh-outa-college programmer is suddenly awarded the high visibility position of System Manager. Imagine yourself a young kid entrusted with the system integrity for a multi-million dollar organization! You barely know how to code COBOL, yet now you have responsibility for system tables! The slightest misstep will be visible to everyone from the customers to the Board of Directors. Your job description reads like Superman, but you feel like Jimmy Olsen. What would **you** do if you were the young programmer in question? How would **you** cope? Where would **you** start? Maybe you don't start. Then the question becomes: where do you go to get help with your resume'?

I was the inexperienced programmer and have some thoughts gleaned from that experience.

We were expecting our System Manager to be leaving eventually. So we took steps to prepare. Management chose a "Backup" - yours truly - and I went to Hewlett-Packard's System Manager course. One week after completing it I found myself the System Manager for real! I hadn't counted on getting the real responsibility so soon - I was expecting a nice, leisurely apprenticeship with the System Manager acting as the Kindly Old Mentor. In the back of my mind, I knew that I never would really be the System Manager - I was way too inexperienced. Management would put out the want-ad and someone would be hired to do the job right while I just held the place together. Even if I messed up like the Sorcerer's Apprentice, the Sorcerer would still be around to get me out of real trouble.

Instead, management said they had no other plans: I was it. Myself as Luke Skywaker would not have an Obi-Wan or Yoda to teach me the ways of the Jedi Knights!

The Question: how does a junior programmer (who is <u>not</u> a Computer Science major) cope with the new demands for a System Manager's expertise and get himself up to speed in a short time? <u>System Management on the Fly</u> will answer that by relating how someone can smoothly launch into the System Manager function. The "- On the Fly" part acknowledges that this poor character must learn system management quickly and effectively in a production environment where you must not get lost or bogged down; get real work done; and able to deal with changing circumstances even though you may lack confidence.

What is a System Manager?

In HP3000 installations from time immemorial, there has existed the Creature known as "The System Manager". This is an artificial role created by Hewlett-Packard to provide System, Technical, and Vendor support. In a sense, they have pieced together a monster.

In the first area of support, System; the System Manager acts as a kind of "Techno-Bureaucrat" of system resources. He doles out accounts, groups, users, capacities, access, and file space and then collects information about the various resources. In the second role, Technical; the System Manager is the installation 'guru', an Enlightened Being who understands system tables, intrinsics, is conversant in three programming languages (including SPL), and can patch object code with the disk editor while simultaneously playing three games of "Adventure"! The ability to bend steel in his bare hands is usually not expected and is left to the Data Processing Manager. The last role, Vendor support; is like unto a phone-exchange made of flesh and blood. It exists so when problems or issues occur (for either party), HP can have one source of contact with the installation rather than deal with a whole pantheon of users. One person schedules updates and maintenance; one person to call the Response Center; etc.

In some installations, the System Manager title can be next to meaningless. I was not blessed with that situation. In my organization, the System Manager was a nearly full-time function with its own job description. I was a "real" System Manager and when things went wrong, people pointed their fingers at me. It was a bit threatening and I felt stress constantly. Always there was the nagging doubt: did I forget something? Is there a safer or more efficient way to do this? What will I say when somebody discovers that I'm really not competent to do this job? I had a real confidence problem; wouldn't you?

IMMEDIATE CONCERNS

So there you are, sitting at your desk with a new nameplate giving your title as "System Manager" and maybe feeling some uncertainties. You know what a System Manager is expected to be; what do you do first? The way to confidence is to know a few things beforehand about the situation and then know what needs to be done. That's what we're going to do. First, give you a generalized view of your situation and what you should be concerned about. Second, give you some concrete steps to take to deal with those situations. Let's start -

Realize that you do have a honeymoon period where most folks will leave you alone to get your bearings. You may even think that you don't to have much to do the first few days. That's what I thought. Don't believe it! You have plenty to do. Your first tasks will be to insure that the installation is running well enough that you don't get bitten by some obvious but overlooked item. Give yourself a time-frame of a week to take care of these immediate concerns:

Password Security

First priority: make sure the system is secure. First step: get THE password passed on to you, the new System Manager. The next step applies to protecting your old System Manager. If a person COULD abuse the system, then they are suspect if anything goes wrong regardless of their character or intentions. Therefore as soon as the previous System Manager is relieved of his responsibility, restrict his old access. This policy applies whether the System Manager was dismissed, quit in a peeve, or it was just time to move on.

Make sure that the user MANAGER.SYS is passworded and that password is appropriately guarded. For example, never written unless secured for an emergency and changed periodically. Maybe not so obvious are users who were created with SYSTEM MANAGER (SM) capability. This can happen for a variety of reasons. Maybe the previous System Manager wanted to use his own name for logging on rather than having to spell out "M-A-N-A-G-E-R". Or there is an odd utility or jobstream that requires SM capability and that user was created. Whatever, you must find out those extra users (use LISTDIR or BACCT from the Contributed Library Tape), weed them out, and change their passwords. You must also limit access to any jobstream files that contain these passwords so curiosity seekers won't be able to look at them. As before, change these passwords simultaneously when you change MANAGER.SYS. You can find a useful utility to do this in the contributed library tape called CHGPASS which changes passwords of all jobstreams in the logon group. Or you can commit to using STREAMX or other utility that streams jobs without requiring embedded passwords.

Be sure not to forget those ubiquitous accounts such as TELESUP, SUPPORT, HP@ and other users in SYS.

Here's what I did: when our System Manager left the department, I was the last person to shake his hand. I said goodbye, watched him walk out, turned around, logged on to MANAGER.SYS and changed the password. I then wrote it down, put it in an envelope that couldn't be looked through, licked the seal and signed my name over the flap. I gave the envelope to the department manager for safe keeping. One of the first things I did right.

There are other passwords to be changed than just the Big One. Major user and account passwords must be changed. Certainly the users with ACCOUNT MANAGER capability. The procedure is the same. Use LISTDIR or BACCT to find the users with AM capability, weed them out (or change their capability), use the :ALTUSER command on all the affected users. Then use the CHGPASS utility to catch the jobstreams. Same goes for changing account passwords. Log on as MANAGER.SYS and use the :ALTACCT accountname; PASS=whatever. Remember that some applications have security features of their own that may require you to duplicate the change of user, group, and account passwords. Look out for application MGR and account passwords in the application's security section. What **you must know** is that YOU, as the System Manager, are responsible for the system's security. You have the authority to make it secure. Note that the courts will look to you to determine if computer security was adequate and appropriate if unauthorized access becomes a legal issue.

Got that done? On to the next concern:

<u>Tools</u>

The second immediate task is to make sure all the System Manager "Tools" are available. This does not only mean screwdriver, pliers and breakout box! Also make sure you have a complete set of system tapes: Coldload (or SLT for MPE/XL), DUS, and current full-system backup. Make sure you have a set of manuals current to your version of MPE; especially <u>System Operation and Resource Management</u>, <u>MPE Commands</u>, <u>MPE System Utilities</u>, and <u>MPE File System</u>. An often forgotten set of tools are manuals for the pieces of hardware that you're responsible for: terminals, printers, modems, plotters, and other data communications equipment.

I was not kidding about the hardware side of that either; a System Manager ought to have a small toolbox. Include a small flatblade screwdriver for RS232 cables, a pair of needlenose pliers, a set of normal flat and Phillips screwdrivers, D-25 (RS232 cable) gender changers, and a RS232 breakout box. If you support PC's, include a diskette that contains several utilities that you use frequently. If you don't have the tools you think you'll need, now is the time to ask for them - during your honeymoon period.

In this, I was fortunate. The D.P. Manager asked me to inventory our 'tools' and make some recommendations for purchase. I had seen that the old System Manager was constantly tracking down data communications problems so I made out a purchase requisition for a breakout box. I also got hold of a small fishing tackle box to keep all the hardware tools together instead of the paper box they'd previously been in. You need to look around; what do you need to get?

Got all your tools together? Great! Let's push on:

<u>System Backup - a review</u>

Make your next immediate priority to insure that you back up your system at least once a day. This doesn't have to mean the WHOLE system, however. A Partial (:SYSDUMP with the DUMP DATE? as the last full backup date or the :PARTBACKUP command) is sufficient. However the WHOLE Full System backup should be done at least once a week. If this has not been the practice, make it a consistent practice now. Also if that is the case, note that you're biggest problem will be in scheduling - the hour of the day for the daily backup and the day of the week for the full system backup. There is always someone who'll be inconvenienced by the downtime that a backup creates. Don't let them deter you - be reasonable but get the backups scheduled and done.

If you are not doing regular backups, you're certainly not doing regular database backups using DBSTORE. If you value your database applications - and especially if you are doing IMAGE logging - I'd highly recommend you start a separate routine of backing up your major application databases. Perhaps you could do this in the middle of the week between full system backups. Granted that HP hasn't made DBSTORE of multiple databases easy, but that will be a poor excuse when you need them and you don't have them.

In addition, consider your protection of the backups. Think about offsite storage. Offsite storage can mean a special warehouse or the System Manager's apartment. If you can't do offsite storage, give some thought to fire and water protection as in a fire resistant safe.

You might also want to back up all your source code to tape occasionally, store that onsite and make a copy for offsite storage as well.

All these backups provide protection from disaster as well as being classic site management concepts. What I mean by "classic site management concepts" is that everyone who talks about this says the same thing - do your backups. Nonetheless, there is a reason they all say that. I was <u>un</u>pleasantly surprised at how many files got **accidentally** purged while I was getting oriented. Those frequent backups came in handy!

Orientation

So far, you've been a busy beaver running around your own department, now you can venture into the outside world a little. Write, or have written, a memo about the new personnel change and ask your users/customers for cooperation in the change. You should then take an unhurried walk around the facility. Introduce yourself, see the lay of the land, listen to your new customers and take lots of notes. Even the most cynical customer wants some assurance that this change will not shake up his world. Be reassuring, confident, and considerate - but don't promise anything! You can be sympathetic to the raw treatment someone once got from Data Processing but don't put yourself in a box by making uninformed commitments.

I remember after making a job change that I was sitting in the coffee room talking about nothing when one of the guys started ranting on the insulting lack of support his department had got from my predecessor. Out of the blue I was the recipient of this man's wrath! He wanted to know if I was going to be another snake-souled devil like the one who'd come before me! I was so shocked that I had no choice but to be diplomatic. I told him that my predecessor was history and that I am a different person. I also reminded him that I couldn't make any commitments but I am definitely interested in helping him in any way I could. That seemed to be what he wanted to hear. As it turned out, I was able to be more helpful than my predecessor but mostly it was because I was willing to listen to the man when he went off on one of the now familiar tirades.

Get back to your desk and write out a list of what you learned and things that will need attention. Prioritize those items and then discuss them with your management.

A good next step is to orient yourself with Hewlett-Packard. Call your Sales Representative. Inform him of the personnel change and make an appointment for an "Account Review" to get more information and a free lunch. Ask that your System Engineer and Customer Engineer be in attendance. Don't <u>schedule</u> the meeting time until after you complete your "Immediate Concerns" phase. Do <u>prepare</u> for the meeting beforehand by formulating questions and issues you want discussed. You might want to have them brief you, from their perspective, on your installation - its history, current situation, and future needs. It can be eye-opening to hear what HP might have to say about their experience with your organization. Ask practical questions. When is the next hardware preventive maintenance? Is there a new MPE update coming that would interest you? Don't be afraid to dream with them a little -any interesting products coming for release?

<u>Physical Inventory</u>

This is the last priority in the "Immediate Concerns" section, but don't think that this is less important. Goof up on this now and you may find yourself facing embarrassing questions later.

One of your primary responsibilities as a System Manager is for the equipment. You need to know what your organization has, where it is, if it is under some maintenance agreement, etc. This means you'll need to "square the books" and make sure nothing has been 'lost'. Get whatever exists in the way of an equipment inventory and <u>verify it yourself</u>. You'll probably find the list badly put together so just use it as a starting point to create your own. If there is no list, inform management of the deficiency and then create it. A suggested format is discussed later. Also remember that your Customer Engineer has a list of all equipment covered by a maintenance contract. You would also want to make sure that all the pieces you want covered are actually on a contract.

Go through the whole facility and look for serial numbers for each piece. If there are pieces offsite go to the holder and get the serial numbers checked. I was nasty enough to demand that the holder physically bring the pieces in! You should have an accurate equipment inventory completed, printed and distributed to appropriate parties within a month after becoming the System Manager.

Another part of physical inventory is to check the data communications cabling, patching methods, and configurations. Know where things are and how they fit together. Documentation of this will also be discussed below.

STABILIZING INFLUENCES

After you've satisfied yourself that the installation is running in a basically sound manner, you can now focus on increasing the scope of your orientation. This is where you'll be practicing System Management "on the fly". The key to getting up to speed is systematic orientation. You've just spent the last week making sure nothing is going to bite you from behind. Now you can start soaking in some of the broader details of your responsibilities.

Rule Number 1: <u>Don't be in any big hurry to make changes.</u> If things are running smoothly, don't change anything. If you start making changes when things are perceived as going smoothly, you'll transmit a bad message. You want changes that you make to be seen as real improvements, not just different ways of doing the same thing. However, if things are a real wreck, get your Immediate Concerns phase finished within a week and then make decisive changes that show your new direction.

Rule Number 2: Don't change anything until you've tested it nine ways 'til Tuesday. Make sure its implementation is seamless and stable. You do not want little mistakes distracting you while you're trying to learn the big picture. This is very important when you're new to the position.

This was something I wish someone had told me. I was so eager to show myself as being active and on top of everything that I made foolish mistakes and did some real sloppy work. "If it's not broken, don't fix it" didn't register with me. One nightmare I remember was deciding that I wanted to clean out the PUB.SYS and other groups of various trash files that had accumulated. I did call my SE to find out what files ought to be in PUB.SYS and did make an effort to determine the files that I did purge were unused. But. Certain routine jobs that had performed faithfully for two years without complaint stopped working. Omygod - I'd purged the file it needed! This started happening more often than my good will could match. It got to the point that whenever any job bombed, the operator and programmers would first look at me and ask, "Do you know anything about this?" It took a long time to live that one down.

I hope that your environment is already smooth and stable. Here's what you should be doing next:

Look for Stretching Seams

Identify all the production IMAGE databases on the system. Look at the capacities for the datasets. Any trouble areas? Identify the sets, what their new capacities should be and expand them. Caution! If you're going to use DBUNLOAD, DBSCHEMA and DBUNLOAD: first, be sure to do a DBSTORE on the database before you do the others. Second, **verify** the current schema against the existing database. Look for any extra dataitems or datasets that show up in QUERY but aren't in the schema file. Look at the capacities and make sure they are nearly the same.

I had several nightmares where I changed the schema to expand a dataset only to find that my predecessor had used a demo database manager package to expand some dataset without bothering to update the schema! Without knowing this I recreated the database and that modified dataset (according to the obsoleted schema) was now too small for all the entries that it had really contained before.

Along the same lines, check the account structure for accounts or groups that were given limited file space. Are any in danger of running out? Find out the reason for the limit and if it is still valid, warn the user to do some housekeeping.

Review General System Documentation

The first productive task a System Manager should do is updating the various Operations and System documentation. There are benefits from doing a review and update right away. First, if you decide that you want out after a month and quit, the updating still needs to be done and there will be an even less experienced person to sort through it. Next, just the act of sorting through such documentation will serve to provide an automatic training manual for you. You, a relative novice, has to understand all that it says to be able to update it. Since the documentation was initially written by an expert, you are now in the best position to know how it should read to a novice. Thirdly, as a System Manager, you may be heavily involved in the documentation cycle. This experience will help you to more clearly understand and appreciate its purpose. Last, the task is relatively unhurried yet important task. Thus you can be productive as a System Manager immediately.

If you spot an inconsistency, correct the documentation now! A rule during your orientation: <u>If you learn about a thing, write it down!</u> Of course you'll also want to review system management history such as purchases, letters and memos, and system trouble from the "Gold Notebook".

I remember my first week as System Manager. I was reviewing the documentation for recovery from a System Failure. I noted a couple of inaccuracies and ambiguities so updated the documentation file. Not two hours after I'd printed off the new version and put it in the Operations Notebook, we had a system failure. My first one! I immediately knew what to do, where to go for instructions and the system came up right away. I really looked like I knew what I was doing - no mean feat for someone who was previously a Junior Programmer! Of course that experience motivated me to continue reviewing and updating whatever documentation I could find.

<u>Regular Staff Meetings</u>

Another stabilizing influence (in addition to your documentation review and other forms of orientation) can be weekly staff meetings - if you don't already do that. This area may not seem appropriate under how to survive a personnel change, but it can make a difference. Here's the suggestion: every Monday afternoon after the "Monday Morning Rush", all the department staff assemble to 'plan' for the upcoming week. This break acts as a breather for the staff from the first taste of the week's action, pause for this meeting to get perspective, and then go right into the work for the week. It is a chance to rise above the urgent and consider the important.

Is this kind of meeting "too formal" for your installation? My response is two-fold: First, I find there is a lot of irrational resistance to formal communication - as if our dignity as human beings relies on complete spontaneity. What is supposed to be an attitude of informality or bureaucracy-bashing I often see worked out to be a more insidious form of office politics. Secondly, there are many things that can be said in a formal communication situation that somehow aren't brought up in casual conversation. If you want real communication within your department, this kind of meeting can bring those things out.

Granted, you may not have the influence to bring this about by just wanting it to happen. However, perhaps your management will see the advantage this could have in accelerating your orientation. These kinds of meetings can have a major impact on your new job. The role of the System Manager is in the nether world between programming, operations, and administration. Since most anything done on the machine can in some way impact the System Manager's responsibilities, most anything could be of interest to you. A formalized "staff meeting" helps the new System Manager to hear what people are interested in.

In any case, if your management doesn't like the idea, you'll need to develop some other form of regular communication with others in the department. Maybe you can develop a schedule of appointments with each of the folks in the staff. Whatever it takes, keep communication open between yourself and others in the department.

<u>Technical Support</u>

Very rarely can a new System Manager handle all the technical problems with applications, data base management, the operating system, etc. Some sort of technical assistance is useful or your tenure as the System Manager will not be stable. You'll find yourself constantly fighting your own ignorance.

Now this is a tough subject for me. It strikes right at the heart of my self-confidence. On one hand, I came into the position shaking in my boots for fear of making some terrible mistake or being shown to be some kind of a fraud. But on the other hand, my pride doesn't like to admit that I need help when I'm in trouble! I know that isn't rational, but I didn't claim my head was straight on this. If you have similar feelings, here's what to do: swallow your pride and get the help. I'm paid to solve problems quickly and efficiently. Ego gratification should come only after I've done what I am paid for. Get some help so you can solve problem quickly and efficiently.

Not only that, petition management to purchase HP's top-of-the-line support. Once you've gotten your "system legs", you can tell management the good news that you can lower monthly expenses by going to a lower level of support! But if you work it the other way, starting with a lower level and finding that it would've been nice to have a System Engineer at your beck and call; it will be near impossible (and **more** embarrassing) to petition management for more expensive support.

The first choice would be Hewlett Packard's AMS or "Teamline" support. This is excellent for the immature installation. If yours is a small or medium shop of average expertise, purchasing Response Center support is an excellent insurance premium for a new System Manager. The Response Center will not answer a question immediately but does return your call within a reasonable time. While you are learning the system, the Response Center can serve as a big brother to help you through rough spots. This gives you a sense of security and frees you from much anxiety therefore <u>allowing you to do a better job</u>. Also I found that using the

Response Center eventually creates more self-sufficiency. As you place calls and get answers, you begin to see how HP has organized the system or put together the manuals. You'll be able to find your own answers.

I must've placed a million calls to the Response Center in the first month. I was glad they were around and certainly felt we got our money's worth.

Support from HP is the first choice, but if your installation doesn't want it, then you will need to find other options. A local users group may be able to help in suggesting a particular resource at a near by installation. Of course, you may not get the best service due to that other installation's own time and resource constraints. That is, if you ask for charity, you might not get it right away and maybe not at all. Another possibility would be a third party vendor - perhaps the people you did that custom software of yours. If you find your choices are limited, contact your HP Sales office and let them help you find an answer.

Of course, the first option is to try to solve the problem yourself. You learn through experience and become more self-sufficient. However there may come a time when you can't even understand the problem and you need help. It would be a wise thing to make arrangements ahead of time.

System Manager Specific Documentation

The next stabilizing influence you can contribute to the installation is getting your documentation cleaned up. **Good documentation compresses experience into a novice.** You are a novice. Given the turnover which most shops are saddled with; accurate, reliable and useful documentation will go a long way in increasing an installation's productivity as well as preparing for personnel turnover. Getting your documentation straight is a significant service you can perform.

Alright, we need documentation, but how much and what kind? Know that there are two types of documentation - the first is the technical documentation which is dusted off when a problem occurs and is used for trouble shooting. This would be before-the-fact documentation. The second type is more on-going and is more properly called 'records': lists of equipment, configurations, account structure, etc.

There are be several records used specifically by the System Manager that can be very useful. Not only are such things helpful for the future when the crises hit; but the exercise of digging out the data is another orientation to the installation. If the previous System Manager didn't create these documents, you can do so right away.

The System Manager is usually given responsibility for hardware: maintenance scheduling, tracking down data communications problems, calling the vendor for repairs. There are a couple of records that can be kept to manage the hardware end of things.

First and obvious should be an absolutely current copy of device and system configurations. Use the SYSDUMP $NULL or the SYSINFO utility from the Contributed Library. Create this document and get it printed off. When <u>anything</u> changes, make a new copy - right along with the new Coldload tape or SLT.

For data communications, I have found that records of cable use (tying logical device numbers to cable assignments and building locations) can be useful when you are tracking down a difficult data communications problem. I write down the LDEV number, the cable number, patching data, and show the building location (with the user's name) that the connection terminates - such as a wall plug.

Remember that we'd discussed inventory above? Another useful document is an Inventory of serial numbers (and company asset ID numbers) for each piece of Data Processing equipment and where they are located. My report shows the location (ie: ACCTG), model description (ie: HP2392), the type of equipment (ie: TERMINAL), the serial number, the organization's asset ID, and then we include the equipment's valuation. Valuation does not reflect the usual System Manager's stated technical duties; but when insurance is discussed for D.P. equipment the accountants seem to gravitate to the System Manager since he has (and rightly should!) the most accurate listing of computer equipment. So be prepared and have equipment valuation data.

Start collecting data on disc space. Before the days of HPTREND, us old-timers would collect our own data using the FREE2/5 utility. That is still a good idea. It is run once a week to track your disc's health. Once every six months via HPTREND is too long to stay properly informed of resource use to take corrective action. The same goes for CPU time, Connect time, and file space usage for each account - another easy report using the :REPORT command output (:REPORT X.@ [assuming you have no groups called "X" on the system], and then :RESETACCT @, CPU and :RESETACCT @, CONNECT). Do this once a month.

However, HPTREND is mighty useful. It first verifies what you've been seeing with your own reports. It helps to give you the Big Picture. It also reports on load management which I can't do on a home-brewed basis. And lastly, it presents resource data in a format that is understandable (with some of your tutoring) to management.

On the software side, the System Manager should be conversant with all the applications on the system. First he has to know what's on the system. It is very embarrassing when your top management receives the above resource usage reports and asks, "What the blazes is the ABND12X account why is it taking so much disc space?" - and you don't know! Believe me, I know what I'm talking about here. Get that information before he asks. Create a document that summarizes all the accounts on the system - gives their names, a description, and who the main users are then distribute that to the appropriate management. Try to categorize them according to purpose, users, or source. Find out what is on your machine.

Another "soft" reference is a listing of all users, groups and accounts and their capabilities and access. The first part of this can be got by using the Contributed Software program BACCT. It automatically prints off this listing. Such a report can be useful when you're trying to figure out why a user can't seem to access a program or file. I was constantly having trouble with that problem and when I got that printed off, it was so much easier to look at the report than mess with LISTDIR. Find out who is on your machine.

Remember - keep the number of reports to a minimum to avoid being swamped with having to examine each one!

Now we get to the "System Manager's Cookbook". This is a document containing information useful only to a System Manager. This should include specific instructions on how to solve problems that have plagued the installation before and procedures that may not be easily found or are nonexistent in other documentation. The idea is a description of specialized routine procedures and helpful tips or tricks. It also can serve as a history giving explanations of why certain things are now done in an unusual or non-standard way.

This updated System Manager's Cookbook (if it exists) should be one of the first documents the new System Manager should read - and it should be the previous System Manager's own personal copy since there may be a wealth of information penciled in the margins! A few years back, Biola University contributed its own departmental standards called STNDRD@ in a Contributed Library tape. Their System Manager's Cookbook was located within their operations section. You'll see that your own version of it can be a useful reference document for the new System Manager.

After reviewing the Cookbook the new person might, from their new perspective, see some inaccuracies or ambiguities. Get into a good habit: correct your documentation now! **Right now!** Once corrected, print off a new copy and toss the old.

Let me stress that you commit to documenting your work. The biggest gripe I've heard from other new System Managers is that they found the shop they came into undocumented and therefore they

1) either spent time in documenting that could have been used for other tasks or 2) spent much time and heartache tracking down a problem that could've been fixed in five minutes if someone had only left a note about it.

The flip side is that this document isn't just for when you leave your position to someone new. It is for you to use ten months from now when that same sticky problem comes up again and you remember the problem, but you can't remember the fix.

This underscores the importance of documenting the fix I make today. I have started an "Expert System" for such things. This was mentioned in the press a couple of years back. What I did was to create a Personal Card File application on my HP150 that holds information on problems and their resolutions. The card format includes a keyword field, date of problem, name of victim, description of the problem and information on the fix. This Expert file doesn't take the place of the Cookbook since the Cookbook covers general and routine reference material while the Expert file covers specific and maybe onetime fixes.

Here's another way: for those with Response Center Support - there is no excuse for you not to use the "Response Center Inquiry Sheet". Photocopy a bunch of those sheets off and use them as gotcha/fix documentation. Before you place the call, fill out the sheet. Be careful to describe the problem in detail. Then, when you have the fix, write down the fix (again, in detail!) in the Resolution section. When the problem comes up again, you'll probably remember that you called HP about it, then you can look through your old calls and find what happened before.

Now let's work on making you more of "The Expert" than you are now.

<u>Things **You Must Know** to be a System Manager</u>

The answer is easy: the contents of the <u>System Operation and Resource Reference Manual</u>! But realistically, be sure you are comfortable with the concepts below. If you haven't done so in a while, get out the manuals and read the sections pertaining to:

* System operation
  - how to start and stop your system, how to back up the system and the options (when, how often, what method)
  - how spooling works and the use of SPOOK
  - know about system (and, if appropriate, database/application) logging.
  - know about all security - hardware and software based

* Your hardware - how a terminal works, how to operate every single piece of equipment in the computer room, know simple maintenance and repair.

* Get real familiar with MPE - know the commands and what they do. There may be some that you'll never use, but understand what they are about so you'll be able to know what can and can not be done.

* In the same vein, get real familiar with the various utilities: EDITOR, FCOPY, SORT/MERGE, QUERY as well as FREE5, LISTDIR5, LISTLOG5, and LISTEQ5. If you can, also get to know VINIT, SADUTIL/RECOVER5, etc.

* You'll never succeed on an HP3000 without complete, intuitive understanding of users, files, groups, and accounts and how they all fit together. This is why LISTDIR5 is so important to know how to use. Know the NEW- and ALT- -USER, -GROUP, and -ACCT commands.

* Know about user/program capabilities and access. This can be real confusing to the neophyte. Remember the BACCT report mentioned above?

* Know how to use your third party or Contributed utilities such as MPEX, QEDIT, ADAGER, et al. Learn and use these. They were created with helping you in mind.

FUTURE DIRECTIONS

Ok, so you now are familiar with the basics and you know all about the current situation in your installation. Things are running stable and smooth. Can things be improved? Let's see -

Performance Review

You can look for improvement first in the computer room. Arrange with your account System Engineer to meet and run some performance benchmarks on your system. Check system table settings, TUNE command options, ALLOCATED programs; look for your system being I/O, Memory, or CPU bound; determine if certain applications cause different problems. Discuss the results. Get some facts together and discuss his specific suggestions for improvement. Get dollars assigned to those suggestions and write up your conclusions for your management.

For do-it-yourselfers, HP has recently released two performance packages: HPGlance and HPLaserRx. Both show promise; however, HPGlance is the easier to use and to buy. Otherwise, you could go with OPT or the "Poor Man's OPT", SURVEYOR.

Care of Disc Space

Run a job every month during the middle of the night that does the following:

```
:FREE2 [or FREE5]
:VINIT
>COND 1
>COND 2   [this assumes you have two disc drives...]
>COND 1
>COND 2
>EXIT
:FREE2
```

- this job takes care of disc fragmentation. Running the FREE utility before and after shows you if the CONDense did any good - you can actually gain back some 'lost' sectors. Repeating the COND command for each disc device is recommended for badly fragmented discs since (last I heard) the COND command may not get all the fragmentation on the first pass. If you have a lot of drives, you may want to spread this out over the month by doing a couple of drives a night.

You'll also need to schedule down time to periodically do a COOLSTART and RECOVER LOST DISC SPACE? Yes. This will take about ten minutes per disc and cleans out dirty spool and temporary files that MPE can accumulate.

Look for accounts on the system that are deadwood. By now you ought to be <u>certain</u> which ones those are. To get a certain knowledge, do a STORE to $NULL with a DATE option set to some access date and be sure to include the SHOW option. Store the offending accounts off to tape (make two copies?) and purge off the accounts.

<u>Disaster Recovery Plan</u>

Hardly anybody wants to deal with possible disasters. If there is a Disaster Recovery Plan - review, update, and then memorize it! Put it on your calender to hold a Plan test.
If there isn't, put it on the list of things that must be addressed by you and Management very soon.

I remember a test run where I simulated a bomb threat to the Data Processing Department. The only people in on the fact that it was a test was myself, the chief of Security and my roommates. My roommates dummied up a beautiful looking fake bomb and we experimented in finding and gaining access to the department. Then on the day of the test, we put in a call through the receptionist stating that a bomb had been placed in the computer room. The security folks burst into the department and herded us out. They then swept the room but couldn't find the box. It was very realistic - you could see the sweat on the faces of these guys as they started to consider that this bomb could go off <u>before</u> the stated time. It was a rich experience for the Security chief as he was able to make judgments on his operation.

After the cat was let out of the bag - that this was a test, only a test - we all sat down and evaluated our performance. What an eye opener! To say that some procedures were changed is an understatement. By the way, I was the one who was the terrorist's "inside man". I put the bomb in the CPU box - lots of empty space in there - and none of the security guys even thought of looking inside the computer.

Transaction Logging

Consider the value of your database applications. I hardly ever have had a problem with the "physical integrity" of any of mine. But I have had some times where I was glad I could roll back some logical transactions. Give some good, hard thought to implementing database transaction logging. It is cheap insurance and could save a lot of heartache if something physically or logically goes wrong. Those lucky ducks with the 900 series machines need not worry about physical integrity, Transaction Manager works transparently like ILR on the Classic machines.

Continuing Education

This area of improvement has to do with you! It is assumed throughout the paper that you've made use of HP's System Manager course (or its equivalent). You may wish to consider an appropriate time to take the follow-on course - "Advanced System Management". I found it to be an interesting and useful course - especially since it had been six years since I'd taken the original System Management class. I suggest waiting until you've been in the position a year before you sign up.

However the course that I found the most interesting from HP was the Application and Design course. I recommend it highly for mid-level programmers and System Managers to understand how your HP3000 is really behaving underneath all that cabinet work.

Let me put in a plug for your Local Users Group. Get involved to the point of at least attending meetings. Not only will you meet others of your ilk, but you'll have another means of unhurried technical support. I can never give any good solid reasons why User Groups are useful. I only know that every time I go to a meeting, I learn something that is new, interesting and useful.

Also don't forget various journals and periodicals: INTERACT, The Chronicle, HP Professional, and others. Another plug - Ed Sharpe's First and Best On-Line HP Users Group. Ed Sharpe has a electronic bulletin board running out of Phoenix, Arizona that serves as another clearing house of ideas, tips, and also some interesting gossip. Use of the board is free, though I'm sure Ed would welcome contributions.

CONCLUSION

What we've done is to outline a plan of action for the newly anointed System Manager. It is possible to hit the ground running - knowing what to do, in what order, with a clear vision of what you want to do next. This builds confidence in yourself and as others see your clear headed approach to your new responsibility, they will grow in their reliance on you.

Nobody is saying that adjusting to a new situation is easy - just that it doesn't have to be a horror movie. You need to know where you are. You need to know where you want to go. Then you take the steps necessary to get from where you are to where you're going. I hope this paper helps you find out where you are by first, stressing systematic orientation; and secondly, that it helps you know where you want to go by giving a realistic plan. You can succeed!

THE CARE AND FEEDING OF MULTIPLE HP3000'S IN A NETWORKED ENVIRONMENT

CHRISTINE DALE
KAISER FOUNDATION HEALTH PLAN OF COLORADO
2045 FRANKLIN ST.
DENVER, CO 80205
(303) 861-3229

It was a dark and stormy night. RRR-I-N-GGGG. I look at the alarm clock. It is 2:00 A.M. The adrenaline is pumping through my veins. I answer the phone. The voice on the other end says, "It's Steve, the operator. We have a LAN problem with the 950E." I say, "OK, tell me what happened". And so begins the nightlife of a Technical Support person at Kaiser Permanente.

KAISER PERMANENTE - THE COMPANY

Kaiser Permanente, Colorado Region, is a part of a national Health Maintenance Organization serving nearly six million people. Kaiser Permanente opened to community enrollment in 1945 by Henry J. Kaiser to fill a public need to access quality health care at costs that families with average incomes could afford. The Colorado Region has been in the Denver area for 20 years serving 217,000 people in 11 medical offices and a Special Care Center.

THE INFORMATION SERVICES DEPARTMENT

The Information Services Department (ISD) is made up of five groups each headed by a Manager/Supervisor. The groups are Operations, Telecommunications, Production Control, Systems and Programming, and Technical Support.

OPERATIONS

Operations is the "keeper of the hardware". They run all the jobs according to a schedule and perform full backups Monday thru Friday. They also monitor and maintain the data communications network including terminals, multiplexors and modems.

Starting with one HP3000 Series I in 1978, the hardware has grown considerably. The present configuration consists of three HP3000, Series 950's; three HP3000, Series 70's; two HP3000, Series 37's; about 800 terminals, and 24 gigabytes of disc storage. There is also a 2680A laser printer utilizing the Forms Design software. NS/3000 is used for communicating between computers. We also have a Data General system running a Pharmacy System. This is presently a stand-alone system and we use "sneaker net" for data exchange from the HP to Data General (using tape drives).

The computer room operates with staff 24 hours-a-day, 5 1/2 days-a-week. In the fall of 1989 we will move to a new 3500 square foot data center which could be a topic of a another paper.

TELECOMMUNICATIONS

Telecommunications installs and maintains the voice/data hardware consisting of leased lines and T1 lines. We are converting this year entirely to T1 lines. They also have responsibility for telephone operations and are installing a voice mail system this year.

PRODUCTION CONTROL

Production Control is the "keeper of the production environment". Data entry is done using DE3000 software from Infocraft on a departmental Series 37. All production and test runs are scheduled using Maestro from Unison Software. They perform distribution control using Spoolmate from Unison Software. A source code maintenance system called Librarian from OCS is being installed this year. They also perform disc space management. The production environment runs 24 hours-a-day, 7 days-a-week with someone on-call via a pager.

The three 950's are running an Appointment Scheduling System. One Series 70 is running a Membership System and some Financial Systems. Another Series 70 is running a Laboratory System and the last Series 70 is the development machine and also runs Financial Systems. The applications running the Membership and Appointment Scheduling Systems were developed in-house. The Membership System is the basis for information kept on all the members. It is written in Cobol. The Appointment Scheduling System is written in InfoCentre's 4th generation language, Speedware. The applications also uses Omnidex from Dynamic Information Systems Corporation for keyword retrieval and Netbase from Quest Software for remote database access. The Financial Systems are a combination of 3rd party software and in-house software. Nearly 100% of the new development is written in Speedware.

SYSTEMS AND PROGRAMMING

Systems and Programming is responsible for design, coding and implementing new systems and enhancing existing systems. The department subscribes to a Systems Development Methodology to implement projects and a Project Control system for monitoring projects. Major projects use PC-based software packages such as Excelerator from Index Technology Corp and Harvard Project Manager from Software Publishing Corporation.

TECHNICAL SUPPORT

Technical Support is the "keeper of the operating environment". This is the group I belong to. We install, configure and maintain operating systems and new and existing 3rd party software. We do performance analysis and capacity planning with software products such as OPT, Glance, and LaserRX with the help of our HP Systems Engineer. We are also responsible for the design, maintenance and testing of the Disaster Recovery Plan. System security is maintained using Security/3000 from VESOFT.

We provide consulting services to other ISD areas in hardware and software evaluation and selection. We always have a Technical Support person on user task forces for system design and implementation - such as: Chart Tracking System, Medical Transcription System, Marketing System, and Time and Attendance System - a few systems being implemented this year. We also have someone on-call 24 hours-a-day, 7 days-a-week via a pager.

Other hardware vendors will be entering the Kaiser Permanente environment this year. This is due to implementing a Common Systems project for all Kaiser Permanente regions nationwide. We are installing systems on DEC, Wang, and IBM.

There are about 250 personal computer users doing office automation functions using word processing, spreadsheet and database software and Reflection from Walker Richer and Quinn for terminal emulation to HP3000's. In addition to the PC's, there are about 30 HPWORD, HPLIST and HPDRAW users on the HP3000's with the executive staff having their own Series 37. Electronic mail has 150 users and is being implemented in phases. The objective is to hook up all 230 physicians and many of the 1650 administrative support staff to electronic mail in the future.

A new area for us is PC LAN's. We are evaluating LAN's and will be selecting one in the near future.

PROJECT IMPLEMENTATION

Now that I've described what we do, I'll tell you how we do it. With the amount of new systems added in the past 4 years, the staffing levels have increased substantially. Managing this growth has required flexibility and determination to bring organization to project implementation. There is always more work to do than people to do it. Priorities are put on every task and every project. Because there is so much to do, it is hard to resist the temptation of skipping the "grunt" work and move on to the next task or project. I always say the opera isn't over until the fat lady sings.

Each year, the Information Services Manager publishes ISD's goals and objectives. The projects on this list are approved by an ISD Steering Committee made up of executive level managers. Other projects throughout the year get added to this list, also subject to ISD Steering Committee approval.

The task force for each project is made up of people from each user area, a systems analyst, a programming team leader, a technical support person, and a telecommunications analyst, if necessary. Project plans are published and everyone has to commit to meeting the schedule.

When projects are implemented, we often choose to pilot major applications in a small medical office or for a select group of people. For example, the executive staff was selected for the electronic mail pilot. After several months of monitoring a pilot, it is then evaluated. If the system meets the goals and objectives stated in the proposal, the rest of the project is planned and costed out for phased implementation.

The Appointment Scheduling System is another example of a pilot implementation followed by phased implementation. We started with one medical office three years ago and in two years had all 11 medical offices and the Special Care Center online. During this time, we added two more Series 70's, and then migrated from the 70's to 950's and added one more 950. With each step of the way, we did a performance analysis and capacity plan for the next phase of implementation. There were also modifications made to the application software to improve performance.

OPERATING ENVIRONMENT

The Technical Support group is responsible for the operating environment. When a new operating system or patch comes in, we put the user at risk. In some instances, depending on the system, we could affect all 800 users. Before we apply any changes to any system, we thoroughly review the materials and what is affected when we make the change. We update one system and allow it to run for one week without any problems before we update any other system. We do updates on Thursday nights to minimize the impact on our users. We do updates only after a full backup which means coming in at 2:00 A.M. We do reloads and extensive database changes on the weekends.

The "update only on Thursday" rule applies to 3rd party software updates as well. One machine is updated first, unless we HAVE to update across-the-board. We also sometimes get different results on the 70's versus the 950's. For all updates, we follow a test script for all affected applications. All CPU's running an integrated application must be idle. For example, the Appointment Scheduling System uses 4 of our CPU's. Coordination with the production schedule and backups is essential.

Anytime we schedule any updates, we broadcast the schedule via electronic mail to Operations, Production Control and all Systems and Programming people who are responsible for applications. If, for any reason the update doesn't work, or the testing fails, we have a documented plan for recovery. Since we are allowed only a certain amount of time to perform the update, test and recover if necessary, you can see that we don't have much room for error or failure. All the experience we have received has made us excellent planners and communicators, especially with the 950 migration.

The migration from the 70's to 950's started in January 1989 by attending MPE/XL classes and taking our Appointment Scheduling System to the HP migration center. Our application ran with very few changes and after several performance tests, came up in April on a pre-production release of the 1.0 operating system. We initially encountered a problem with the number of processes filling a table and after receiving a patch from HP's Lab we were able to add more users to the 950.

We spent many hours and iterations updating the operating system and applying patches and saw many improvements with each update. We are planning on going to native mode this summer when the native mode Speedware and Omnidex products are released. We also have an order in to update the 950's to 955's.

MANAGING PROBLEMS

Until just recently, anyone with a problem or a question could come to any one of us in Technical Support and expect immediate service. Being service minded, we did our best to respond. We realized that we were not only crisis driven, but also interrupt driven. We were becoming frustrated at not getting our projects completed by the deadlines. We were also becoming short-tempered and could see ourselves on the road to burnout.

Last year, we made a 24 hour-a-day, 7 day-a-week commitment to being on-call for emergencies. We rotate the pager each week among three people. This on-call process has worked fine for the non-working hours, but did not work during the day. Again, anyone with a problem could call his or her favorite

Tech Support person.

We considered using a dispatch service to direct all the calls to the on-call person, but discarded that in favor of a logging system, which any one of us can direct the call to the right person. We keep a log book which tracks all problem calls and requests for service. A goal of ours this year is to have the log book online. When a call comes in to any of us, we become the dispatch and put a priority on the call. We have 3 responses to a call. Priority 1 means immediate response with a maximum response of 15 minutes. Priority 2 means up to 4 hours response. Priority 3 means response within 24 hours. When the call is logged, we can determine if the call needs to go immediately to the on-call person or through normal channels. If we can't schedule the fires, at least we can schedule the fire fighters.

We keep track of all open items and can easily determine if we are meeting our deadlines. We publish statistics the time spent working on all our activities. This will become even more important when we start adding more systems this year and plan for staffing requirements.

The Office Automation/Personal Computer (OA/PC) subgroup of Technical Support has been using a Helpline for the past year. This is one phone number which any OA/PC user can call between 8:30 A.M. and 5:15 P.M. All calls are logged and categorized by type of problem or request. All time is logged and statistics are produced monthly. This tracking has justified our need to expand our staff based on the growing use of this service. I have attended management meetings where the users have raved over the service they were getting.

CONCLUSION

I feel that with the amount of growth that Kaiser Permanente has experienced, we've learned a lot about how to keep it all together. I keep a "to do" list which can be prioritized in three categories: 1) The things I know I'll get done, 2) The things I'll get done someday and 3) The things I'd love to do, but will probably not get to. In the past 4 years, I have been able to move items up from one group to another. This is the fun and challenge of an ever-changing and ever-growing environment.

Oh, by the way, the calls in the middle of the night have been reduced drastically over the past 6 months. I believe this is due to better organization and planning in all the areas of the department.

My First Job as a System Manager - Where Do I Start?

Kathleen Dowling
Allied Signal / Garrett Engine Division
M/S 48-4 / 301-1CB
Phoenix, Arizona 85034
(602) 231 - 3295

The realization of your first System Manager's job can be quite a harrowing experience.

Having survived the trauma of the hiring interview and having convinced my prospective employer of my capabilities, I now arrived at my 'new station in life'.

Unfortunately, I was completely new to my present responsibilities and my predecessor had left me with an unexpected gift -- an undocumented system.

After starting my new job I decided to make up a task list of things to get done and assigned priorities to each individual task. I utilized the Task Planning / Assignment (TPF1) form.

Documenting the system was very high on the list. However, I could not afford the luxury of becoming completely consumed by any of these tasks, I still had to survive in a real on-line day-to-day production environment.

I, therefore, came up with the idea of a 'system manager's documentation workbook.' The workbook consists of various forms that document everything from the accounting structure to database capacities. In retrospect, the forms were created as I developed and as new problems or problem areas developed.

I wanted to design the documentation so that any non-technical person, with the desire to do so, could attain a very generalized understanding of our H/P structure. Therefore, the forms have been kept uncomplicated intentionally.

(TPF1)

TASK PLANNING PLANNING / ASSIGNMENT			
TASK NO.	DESCRIPTION	ASSIGNMENT	PRIORITY
PROJECT NAME		DATE	PAGE

The Accounting Structure (ASF1) documents the users, groups, accounts, passwords, the date and any comments. I initially documented every single account, group, and user.

The User Identification form (UIF1) was used to substantiate the validity of the H/P 3000 users. Every account manager was asked to sign off on all the users residing in their particular account. Through this process, I was able to obtain a 'true' list of current users. From the initial 500+ logon-ids possible, a whopping 400+ were eventually deleted from the system.

This documentation proved very useful to our internal auditors and helped support the auditor's recommendations. As non-active logon-ids were deleted from the system, the deletion date was recorded on the ASF1.

Additionally, a Computer Profile Request form (PX4417-1) was created to record new logon requests. This form has to be filled out by the account manager who in turn forwards it to our Security Administrator for approval. After approval the source document is permanently filed. If the request requires the system managers' involvement, they become involved, otherwise, the account manager proceeds accordingly.

The Capabilities (CF1) form was used to record all users, groups, and accounts, their capabilities and the date. By positioning the more critical capabilities, such as SM, PM, OP to the rightside, I could, at a glance, pinpoint possible problem areas.

I now felt I had a pretty good idea of who was using the H/P system and what security restrictions were imposed upon them. My next area of concern was how was the user community was attached to the system.

This concern created the H/P Equipment-User List (EUL1). All logical devices, their description, location, contact person, phone, communication line number, communication device and date were recorded.

FORWARD TO: Security Administrator
48-4/553D

COMPUTER PROFILE REQUEST

LAST NAME ______ FIRST NAME ______ MIDDLE INITIAL ______

EMPLOYEE BADGE # ______ DIVISION ______ SUB-CONTRACT ___ Yes ___ No

DEPT. # ______ MAIL STOP ______ COMPLETE PHONE EXT. ______

FUNCTION(S) REQUESTED	ADD TO PROFILE	DELETE FROM PROFILE
1. HEWLETT PACKARD ACCESS		

(UIF1)

ACCT	GROUP	USERID

SIGNATURE: ___

APPROVED BY: ___

DO NOT WRITE BE

COMPLETED BY:

USER IDENTIFICATION FORM

Account	Group	Userid	Person's Name	Phone	Location

(ASF1)

ACCOUNTING STRUCTURE

User	Group	Acct	Acct Password	Group Password	User Password	Date	Comments

(CF1)

CAPABILITIES																						
User	Group	Acct	IA	BA	AL	GL	DI	DS	CV	UV	NA	NM	CS	ND	SF	MR	PH	AM	PM	OP	SM	Date

(EUL1)

H/P EQUIPMENT - USER LIST							
LDev#	Description	Location	Contact	Phone	Comm Line #	Comm Dev	Date

This information helped to familiarize myself with the equipment attached to the system, the location of the equipment and the person who was the primary user of the equipment.

Outgrowths of the EUL1 were the H/P Datacomm Info-Device (DCID1), H/P Datacomm Info Mux/Modem (DCIM1), and the Multiplexor Settings (MSF1), forms.

The DCID1 was helpful in documenting internal asset numbers assigned to each piece of equipment and in documenting which multiplexor network the equipment was attached to.

The DCIM1 was used to record the device numbers, multiplexor network and any special settings required.

The MSF1 contained all settings for the local and remote modem and multiplexors.

Right about now this probably seems like overkill but if you've ever spent any time tracking datacomm problems, you know how much time can be wasted by trying to guess what the original configuration settings were and what settings you've already tested.

In my case I work with a communication group -- who never document anything. So to help me out I created these forms and requested that they fill them out whenever they make any equipment changes. Then at least a well documented audit trail is established.

These three datacomm-equipment forms were then utilized by myself to design and create a Datacomm database system that has the current equipment documented and the capability to print various informational reports.

However, for certain situations all this information is too cumbersome for the task at hand. For instance, if the system manager receives a call about a particular piece of equipment having problems; they may only need to know, at a glance, how the equipment is connected to the computer.

(MSF1)

Multiplexor Settings

Mux Descr.	Channel	L-Speed R-Speed	L-Flow R-Flow	L-Data Parity R-Data Parity	Priority	Date	LDev	Description

(DCIM1)

HP DATACOM INFO- MUX/MODEM

LDev #	Mux #	Channel #	Mux Descr.	Circuit #	Circuit Supplier	Modem Supplier	Post Location	Serial Settings

(DCID1)

HP DATACOM INFO - DEVICE

LDev #	Descr.	Bldg. Loc.	Contact	Phone	Asset #	Serial #	Special Settings	Mux	Channel

Date ________ initials ________

Diagrams 1, 2, and 3 give a pictorial look of what you will be dealing with. For my purposes, I recorded descriptions to the left of Ldevs and people's names and telephone extensions to the right. The modem and multiplexor networks were recorded with a viable LADA circuit number. To the upper right hand corner I documented the building location.

As all our H/P 3000 users are remote, the diagrams help me communicate on a more personal level with the users. Users feel much more secure when you are able to say, "Bill, let me make sure I understand your problem. You're in the 101 building and you're having response problems with your 2397A".

Another form that I have found very useful is the Communications Log (CLF1). The minute I answer a problem call or place a problem call, I pull out my handy Communication Log book and record the date, person I talked to, call back number, problem or PICs number, a description of the problem, and the problem resolution with a date-time stamp.

This audit trail establishes a 'psuedo recall' for me when problems reoccur. I'm able to say "Jane, is this the same problem or symptoms you were experiencing in June?" In addition, the form helps me fill out my weekly status report for my supervisor. I am able to jar my memory as to how much time was spent resolving certain issues.

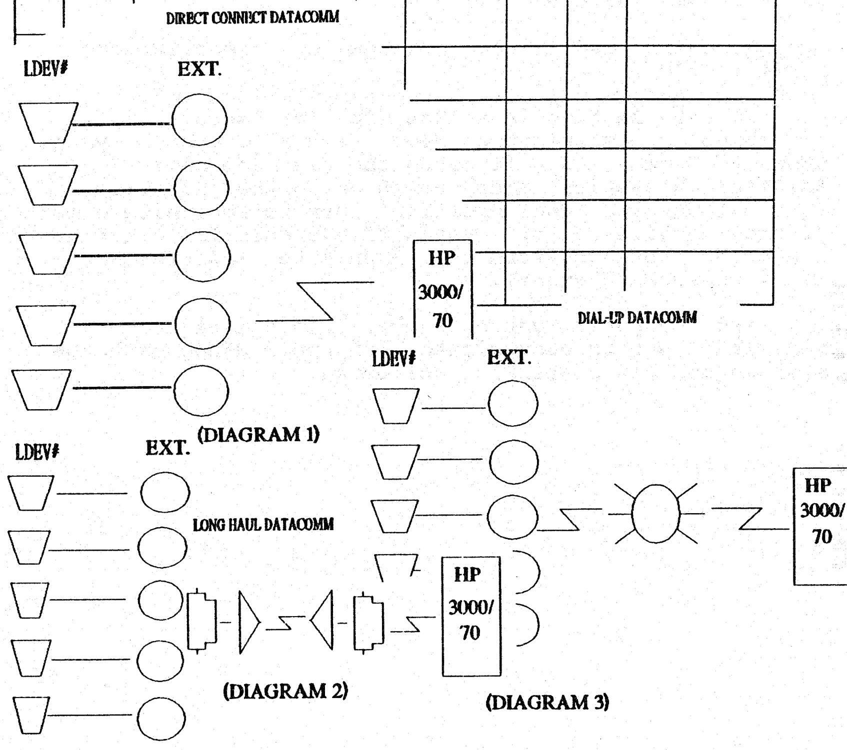
Last but not least is the area of databases. The Database Capacity form (DCF1), was used to document what the status of the databases were. The form includes what database, where it resides, data set names and the before and after capacities. Initially I only recorded the original database capacities. I then wrote a program using 'dbinfo' which gave me a report giving the database name, data set names, current capacity, current entry count, and a percentage used figure. Any data sets using 90% or more were flagged.

(CLF1)

COMMUNICATION LOG				
Date	Person Called	Phone	Problem	Resolution

(DCF1)

DATABASE CAPACITIES					
Database Name :	Group:		Account:		Date:
Dataset	Type	Before / After Capacity	Done by	Drive	Additional Operations



(DIAGRAM 1)

(DIAGRAM 2)

(DIAGRAM 3)

By monitoring the database report for two to three weeks, I was able to make a decision to decrease all the current dataset capacities (in some cases as much as a 50% reduction). Now, I do caution you that if you make such a decision, it may not be easily reversible. As everyone knows unloading and reloading databases is a very time-consuming process. However, there are several third-party vendor products that help to ease this burden.

I also utilize this report to pinpoint datasets that might require attention, datasets that are currently at 90% or above 90% capacity.

Also included in the workbook is a miscellaneous forms section.

In this section is a 'Hot Tips for Users', 'Messages', 'Communication Memo', and a 'Computer Systems Summary' form. I usually send the user community a 'helpful suggestion' every month using the 'Hot Tips' form. I use the 'Communication' form to communicate more 'formal' notices (i.e. system file archivals), whereas, I use the 'Message' form to communicate 'less-formal' (i.e. password changes).

The 'Computer Systems Summary' is one of the vehicles I use to communicate with upper management the systems monthly statistics collected.

FLASH ! FLASH ! FLASH !

Messages
from
Kathleen Dowling

---

hot tips for H/P users

Instead
of
the

My First Job as a System Manager - Where Do I Start?
4504-11

Communication Memo

from Kathleen Dowling

EVERYTHING
IS
RELATIVE

My First Job as a System Manager - Where Do I Start?
4504-12

Computer Systems Summary for 1989

System: HP-3000 (MPE5/VMIT)	Jan	Feb	Mar	Apr	May
CPU Busy - Percent (a)					
Availability					
Scheduled Hours					
Availability Percent (b)					
Down Time (Hours)					
Number of Incidents					
Activity (Total)					
Transactions per Month					
Transaction per Hour (a)					
Response Time (c)					
Total CPU Hours					
Workdays per Month					

a - Prime Time: Monday - Friday, 8: AM to 5:00 PM
b - Percent of Scheduled Time
c - Maximum Response Time for 90% of Transactions in seconds

The last section of the workbook has some JCL listings. The JCL was developed to fill a need one of our divisions had. This particular division had no on-site "H/P" support person.

Let me set the stage--I got a panicky telephone call from this division saying the databases were full and could I 'walk' someone through the procedures over the phone. I opted to go on-site and get a first-hand look at the H/P system. The system had been neglected for quite sometime. Naturally, I spent the whole day doing database work and cleaning up some potential problem areas.

From this experience grew the idea of creating some JCL that could be used if the same situtation occurred again. The JCL had to be almost completely automated with a message interface between the system and the person running it.

The first JCL List is able to do a store, unload, purge, create and load of a database. A person need only 1) change the job statement, if applicable, 2) change the database schema, and 3) have two scratch tapes available before executing this JCL.

JCL LIST 1

```
1  !JOB DATABASE,MANAGER.SYS;OUTCLASS=LP,1,1;PRI=CS
2  !COMMENT  IF NOT SYS DATABASE CHANGE TO user.account
3  !TELLOP  CHANGE SCHEMA BEFORE STARTING THIS JOB!
4  !TELLOP  HANG A SCRATCH TAPE
5  !TELLOP  LABEL TAPE SYS DATABASE DBSTORE
6  !TELLOP  RETAIN THE TAPE FOR TWO WEEKS
8  !RUN DBSTORE.PUB.SYS
9  SYS
10 !FILE DBUNLOAD;DEV=TAPE
11 !TELLOP  HANG ANOTHER SCRATCH TAPE
12 !TELLOP  LABEL TAPE SYS DATABASE UNLOAD
13 !TELLOP  RETAIN TAPE FOR TWO WEEKS
14 !RUN DBUNLOAD.PUB.SYS
15 SYS
16 !RUN DBUTIL.PUB.SYS
17 PURGE SYS
18 EXIT
19 !FILE DBSTEXT=SCHEMA
20 !FILE DBSLIST;DEV=LP
21 !COMMENT  RESTART POINT, DELETE LINES 2-18 KEEP THE
22 !COMMENT  FILE UNDER A NEW NAME AND STREAM THAT FILE
24 !RUN DBSCHEMA.PUB.SYS;PARM=3
25 !IF JCW<>OK
26 !   CIERROR<>0
27 !THEN
28 !   TELLOP  ERROR IN THE SCHEMA, FIX AND RESTART JOB
30 !   ABORT
31 !ELSE
31 !CONTINUE
32 !RUN DBUTIL.PUB.SYS
33 CREATE SYS
34 EXIT
35 !TELLOP  PUT SYS DATABASE UNLOAD TAPE BACK ONLINE
37 !FILE DBLOAD;DEV=TAPE
38 !RUN DBLOAD.PUB.SYS
39 SYS
40 !IF JCW=OK
41 !    CIERROR=0
42 !THEN
43 !   TELLOP  DATABASE UNLOAD/LOAD SUCCESSFUL
44 !ELSE
45 !   TELLOP PROBLEM WITH LOAD, CHECK ERROR AND RESTART
47 !EOJ
```

The second JCL List is setup for running a daily partial backup with a listing, a validation of the backup tape just written to, purging of all deferred spoolfiles and restarting our IBM-HP SNA/NRJE Link. The only steps to follow before running are 1) logoff all applicable jobs and sessions and 2) have a scratch tape available.

JCL LIST 2

```
!JOB DLYBACK,OPERATOR.SYS;OUTCLASS=LP,1,1
!FILE BACKUP;DEV=TAPE
!FILE SYSLIST;DEV=LP
!TELLOP  DO NOT LOGOFF NRJEMON AND SYSPLAN JOBS!!
!TELLOP  DO NOT LOGOFF SCHEDULED JOBS!!
!TELLOP  LOGOFF ALL OTHER JOBS RUNNING ON SYSTEM!!
!COMMENT THIS JOB STREAM BACKS UP ONLY USER FILES
!COMMENT WHICH HAVE BEEN MODIFIED ON OR AFTER DATE
!COMMENT SPECIFIED IN STORE COMMAND.
!PARTBACKUP *BACKUP,*SYSLIST
!IF JCW=OK THEN
!   TELLOP DLYBACK COMPLETED SUCCESSFULLY
!ELSE
!   TELLOP ERROR--RERUN DAILY
!ENDIF
!TELLOP  PLEASE PUT BACKUP TAPE ONLINE
!RUN VALIDATE
N
N
Y
N
!RUN FLUSHER5
PRI=1
PURGE
OK
EXIT
:SNASTART
:NRJESTART
!EOJ
```

JCL List 3 is setup to run the weekly backup with a listing, validation of the backup tapes just written to, purging all deferred spoolfiles, condensing disc drives and restarting our IBM-HP SNA/NRJE Link. The only steps to follow before running are 1) logoff all applicable jobs and sessions and 2) have several scratch tapes available.

JCL LIST 3

```
!JOB FULLBACK,OPERATOR.SYS;OUTCLASS=LP,1,1
!FILE DUMP;DEV=TAPE
!FILE LIST;DEV=LP
!TELLOP  DO NOT LOGOFF SCHEDULED JOBS!!
!TELLOP  LOGOFF ANY JOBS RUNNING-OTHER THAN OPERATOR!
!COMMENT THIS JOB STREAM BACKUPS THE MPE SYSTEM,
!COMMENT ACCOUNTING STRUCTURE, AND ALL USER FILES
!COMMENT IN THE SYSTEM--THIS IS A WEEKLY RUN--
!FULLBACKUP *DUMP,*LIST
!IF JCW=OK THEN
!    TELLOP  FULLBACK COMPLETED SUCCESSFULLY
!ELSE
!   TELLOP ERROR--RERUN WEEKLY
!ENDIF
!TELLOP PUT BACKUP TAPE 1 ON TAPE DRIVE-PLACE ONLINE
!RUN VALIDATE
N
N
Y
N
!RUN FLUSHER5
PRI=1
PURGE
OK
EXIT
!TELLOP  DISC DRIVES ARE CONDENSING--DON'T INTERRUPT
:RUN PVINIT.PUB
COND 1
COND 2
COND 3
COND 1
COND 2
COND 3
EXIT
:SNASTART
:NRJESTART
```

JCL List 4 is setup to run on demand. This job will print out a list of files not accessed after a particular data. This job is great to get a listing of files that meet archival criteria. I currently have our H/P setup for quarterly archivals. Before running the job 1) change the date 2) change the output class if necessary.

JCL LIST 4

```
1  !JOB FLABLIST, MANAGER.SYS;OUTCLASS=LP,1,1
2  !COMMENT THIS PROGRAM LISTS FILES NOT ACCESSED
3  !COMMENT  SINCE A DATE
4  !COMMENT  CHANGE LINE 8 TO DEV=LP,8,1 IF YOU WANT
5  !COMMENT  OUTPUT TO PRINT
6  !COMMENT  CHANGE LINE 11 TO THE LAST ACCESS DATE YOU
7  !COMMENT  WANT
8  !FILE FLIST;DEV=LP,1,1
9  !RUN FLABUTIL.PRV.TELESUP
10 !16
11 09/08/88
12 @.@.@
13 !EOJ
```

This JCL takes into account that you have the 'CSL' program FLABUTIL. If not, I'm sure you'll be able to acquire a copy from your local SE. After running this report, I scan the output and then create an indirect file that is in turn used in my STORE job stream. I do this because I don't necessarily want everything that met the date criteria to be archived.

Lastly, is JCL number 5, which is setup to run a broadcast message to all terminal Ldevs currently assigned on the system. The first file 'Mess2' consists of a UDC 'Sendmessage' assigned to Ldev 21. This UDC when executed will FCOPY the 'Message1' file to LDEV 21.

A person need only 1) create a Message1 file 2) create the Mess2 file 3) insert applicable Ldevs in Mess2 4) copy as many repeated message statements as you need. I found this UDC to be very useful when I needed to notify the users that the system was no longer down and was available for use. This may not seem like an insurmountable task, however, when you're new on the job and you're not sure who the users are, this method sure beats picking up the phone and calling everyone.

Message1 File

```
1  THE SYSTEM IS AVAILABLE FOR USE.
```

Mess2 File

```
1  SENDMESSAGE
2  MESSAGE 21
3  MESSAGE 22
4  *****
5  MESSAGE !PARM
6  FILE TO;DEV=!PARM
7  FCOPY FROM=MESSAGE1;TO=*TO
8  CONTINUE
9  *****
```

Then issue the SetCatlog command

:SETCATALOG MESS2

To execute just issue

:SENDMESSAGE

I'm sure many newly arrived system managers as well as seasoned system managers are saying my job isn't to do paperwork - my job is to manage the system resources. I say you can only manage a system when you know what you have and where you are going with that system. This paperwork, once completed has afforded me the freedom to work on other challenging projects and yet be able to 'jump back' into the H/P world when I need to. Having done all this front-end documentation work, I am better able to fulfill my responsibility as a System Manager, and at the same time, give my company and myself a solid base to pass on to any future successors.

At the end of the day, I can go home with 'true' peace of mind, my head is full of exciting new opportunities -- not unnecessary information that can be retrieved by picking up a manual and looking it up.

Make your job and your career a lot easier to manage -- get the 'FTD' habit.

System Management on a Budget

JC Strote - KVIE Channel 6
PO Box 6, Sacramento 95812-0006
(916) 929-5843

This paper is for the small shop (I mean small single machine) with minimal staff (less than 5 people) with an even smaller budget. That is to say that those of you with more than one machine and full time operator's whose secondary responsibility isn't data entry and part time receptionist, may not find something that will help you in the presentation. But larger shops with, larger staffs have a different idea of what a "ON A BUDGET" means.

For the last 5 years I've been working for a Non-Profit Public television station where we measure income in $15.00 donations. Television equipment is notoriously expensive and our primary business is putting a quality signal on the air. Therefore when it comes to MIS equipment, software or staff, MIS is usually at the end of a long line. I've made some progress in convincing management that some purchased tools are required but at one time or another I've done without them. I'm not a programmer, (as long as 4th GL's don't count), so I can't sit down and write a system utility when I need one, nor would I want to write one if I could. In a small shop you usually don't have time, your too busy being a jack of all trades.

There are some tools that I believe are essential and that if nothing else I think you should go the mat to acquire them. I'll list them in priority order.

1. Equipment Maintenance Support - If it matters at all to your company that when you hit return a friendly colon pop's up on the screen then make sure you have maintenance on your machine. If 4 hour response is too expensive then go to 24 hour response but carry some kind of regular maintenance. Time and material just doesn't cut it if your down and HP says sorry all CE's are unavailable. There are a number of third party maintenance firms around that will beat HP's maintenance cost by 20% to 30%, check them out. I've been using a third party firm for about 3 years now, and have experienced nothing but great service and the cost of that service for 4 hour response is close to 30% less than HP.

2. Software Maintenance Support - I have FOS support as a minimum level of support. My site changes between ASM support and RCS support. I went on RCS support when the cost of having that SE at the end of the phone meant that I'd lose an operator. But when we upgraded our 5 year old Series 30 to a Series 48 we went back on ASM support because we wanted the additional support an Account SE could provide. Now I'm back on RCS support because of budget cuts last year.

   The cost of that SE is about $350 a month for a Series 48 but I've found that if you spend the time to train your SE, that on occasion, the payoff is worth the fight in the budget committee meeting. When I say train your SE, make sure that you control the agenda of any account visits that occur. If you've got your act together you don't need HP to come in and find out if you do regular backups and if you have a disaster recovery plan. If they insist on filling out that little form with all the questions their district manager requires them to turn in, ask them to give you a blank one, fill it in and copy it for future meetings. This will save time in you meetings for the important things, like capacity planning, performance questions and tuning, and any program changes your planning and want to check out with an HP consultant. Make sure that you have their office number handy, just in case PIC's answer isn't completely understandable.

AMS also provides HPTREND which is an added tool for capacity planning. It also assists in identifying problem areas. The quarterly HPTREND reports help you to show your management what's going on and why you need that extra printer, or disc drive.

Another benefit we found with AMS was the installed base seminars that HP has periodically. Although I don't get to pick the topics, I have the opportunity to attend or send staff to some pretty good seminars at no additional cost.

4. Other Purchased software maintenance - I try to budget to keep any purchased software under maintenance. Since my organization is small I need the help of outside consultants to assist me. I can't afford to have a word processing, data base, 4th GL expert on staff I rely on those people who sell me products to provide that expertise. Make sure that when you purchase software that you test not only the software but the phone in consulting support of the vendor from whom your purchasing the software. No matter how easy and error free their software is, your going to have a problem. You need to feel confident that the person on the other end of the phone can answer your questions and provide the assistance you need.

   I also try not to abuse the phone in consulting services. I check the manual first and where possible attempt to solve the problem as indicated. I want to make sure that there isn't a sign over their desk with my name on it indicating 'BORN STUPID'. On the other hand I believe in the 15 minute rule. If you can't find the answer in the manual wi thin 15 minutes then call support, they should have done a better job on the manual.

4. INTEREX SITE Membership - The CSL is still the cheapest utility software on the market today. I made joining INTEREX a term of employment when I came to Channel 6. Besides the CSL, you get INTERACT, one of the best sources of technical information around. I read it cover to cover every month. In addition there are the discounts to the International conferences and membership in the Regional Users group in your area.

JC Strote - KVIE Channel 6
PO Box 6, Sacramento 95812-0006
(916) 929-5843

I'm active in both my local Users group and in Interex as well as software SIG's of the purchased software we use. The pay back is tremendous not only to me but to the company. My phone book contains the names and numbers of the GURU's, some of them even know me by first name. I attend all the local meetings at no cost to the company. I've gotten discounts to conferences by giving papers and INTEREX committee participation. And the technical knowledge and product awareness is much greater due to this participation.

Because of the contacts that I've made through the users group, I've been able to suggest technical presentations on topics that apply to my shop directly. I've worked with HP in focus groups to determine what HP user needs are. I know one of our RUG board members gently guide the rest of us to plan a meeting that included a cocktail party and a presentation on a topic he had been trying to sell to management for 6 months. He got his CFO to come because of the type of meeting planned and bingo sold his idea. If you aren't active, get active!!

4. Chronicle Subscription - Best source of HP news around. I actually like the chronicle better than Interact but if I could only have one I'd choose INTEREX because of the other benefits. There are other HP publications that if you have the money, they provide some value but for someone on a limited budget the Chronicle and Interact provide the best information, in my opinion.

   HP professional is usually free and has a number of worthwhile articles. I also get a couple of other industry magazines but never seem to have the time to read them. These are usually the ones that send me a free subscriptions if I just fill out their little card allowing them to flood my mail box with junk mail. I almost always fill out survey cards and answer phone surveys (although I beginning to question the wisdom of this).

5. Database Management Tool - For years we did without a Data Base management tool. But because so much of our data is stored in Image data bases and these data base have grown in size and complexity over the years, I felt that this was a very essential piece of software.

   There are Contributed Software Library alternatives to purchased software and they do work. They are not as easy to use, and sometimes the performance isn't as good, phone in consulting is usually nonexistent but it sure beats an unload reload.

   Also check your purchased application software utilities. We found a nifty little program in our accounting software's utility program group. It gives us the capacity, disk utilization and percentage of available entries in a neat little report that we run nightly.

6. Backup Utility - We purchased a backup utility about 3 years ago. This was a low cost alternative to a 6250 tape drive and a 3rd shift operator. The software uses less tape than HP Store, provides for deferred backup capabilities, and has some real nice features that weren't available from standard HP Backup programs.

There are other utilities that people will tell you they can't live without like MPEX, Security software, Fast copy Utilities, Spooling Utilities, Performance monitors and the list goes on. And there's a reason to purchase each and every one of these types of software. But I've found that in many cases there are alternative software programs available in the CSL, again not as easy to use, or supported at the end of a phone line but useable. If you have the money to buy system management tools by all means buy them. They will certainly make your time more productive and system more efficient. If you do buy them be sure you use them, buying a product you don't have time to use is not the best use of your company's dollars.

There's software on my capital budget that's been there for all 5 of my years at Channel 6, but when it get's down to making the decisions between systems management software and user software my users win more often than not. I just work longer hours and get more imaginative with my job streams.

Disc Management

I've read all the articles I can on disc management. I choose the technical sessions at these conferences that talk about disc management. I still have an IO problem that doesn't improve. There have been a rash of articles on disc management recently, in Interact, Chronicle and Supergroup. I'm so intimidated by these people who write them that I'm not even going to try to compete.

We run LOSTDISC (from the CSL) and FREE2 daily on our system to monitor our disc usage. We use disc caching. We use DIRK (Tech Account CSL) to identify and purge files that our users have abandon or forgotten. We run the data base capacity checker we found in our accounting software weekly. We do regular backups, we do occasional reloads (about once every 3 months). We reorganize our data set details regularly. And when we have time we try those tips that everyone writes about in those articles.

A well managed and well performing disc drive requires lots of time and evaluation and in a small shop time is usually not available and expertise for the evaluation is not always on the payroll. So you work at it whenever you can and you try whatever you can.

Performance Tuning

Again there are much more qualified people than I putting out articles and presentations on performance tuning. I use caching, I sort of understand it. I spend time trying to identify program HOGs and getting rid of program HOG's or at least off loading them to night time hours.

Actually the very next systems tool I plan to buy is a performance monitor, it's either in by FY 91 or FY92 budget because we need help. But mostly what happens is someone calls me up and says why is the system so slow and if it really is slow (sometimes the users exaggerate)I spend an hour or so trying to figure out what were doing differently that is slowing the system down. Usually I do a show job and find the accounting office running a GL or payroll calculation but sometimes I actually find something I can do something about.

About all I can tell you is read what all those experts have to say then try some of the stuff they tell you to do. The more you have time for the better your machine will perform (sometimes).

Tape Library

I wanted to talk about tape libraries in a system management presentation for years. Everyone focuses in on the biggies like performance and disc management and figure every one knows about and maintains a tape library for the backups. But that's not always the case. When I first came to Channel 6, I found that there was no tape numbering system, and there was no tape log. If you wanted to find a tape you searched through the tape racks, which were stored in date order, until you found the tape you wanted. It took me a year to convince the operator that we really should have some kind of tape numbering system and file retention system.

You really don't have to have a complex automated system you can use a fairly simple manual system for logging tapes. Those tapes are your companies history, in some cases they are worth their weight in gold, so keeping track of them is very important. Make sure that you retain the correct versions for auditors and others. Make sure that what you do maintain is readable.

For years we used "used" tapes. The theory was that we couldn't afford to purchase new tapes so we purchased used tapes. We did double backups so that if the tape was bad we could reload, which took double the operator time. We finally got smart and bought good quality tapes, used the CSL program TAPETEST to test our tapes before we use them. And always verify our backup tapes. It has helped alot in our backups . We have fewer problem tapes and the backup goes alot quicker and we rarely have problems reading the tapes back.

System Management on a Budget
JC Strote - KVIE Channel 6
PO Box 6, Sacramento 95812-0006
(916) 929-5843

Backup Strategies

We use a rather different backup strategy. It's not one I really recommend but I bring it up to show you that there are alternatives to the normal HP backup strategy. HP recommends that you do a full backup once a week and then a partial each day going back to the full. The trouble is that this strategy doesn't work very well for data bases because all the data sets need to be in sync with one another even if some of them rarely change. As the capacity and number of data bases grew on our system, we began running out of operator time and tapes to do backup the traditional way. We went through a couple of variations and finally found one that works for us.

We do a full once a week (ours is on Sunday). We dump all of our system except data bases to tape on our full. We dump the data base files to a separate set. This makes them easier to use for recovery and helps with file retention is the library, we don't have to save the entire sysdump if the accounting office closes the GL. We use Image logging for all our data base files. We eliminate the data base files from our partials, only backing up non data base files that have been changed since the day before our last full. Image Log files are dumped to a separate tape twice a day. In the morning when we do our partials and at the end of the day (6PM). If we have a problems then we can reload our database files from the full and used image recovery to recover the data base. Not perfect but without this kind of strategy I would have had to hire a full time 3rd shift operator just to do backup. It takes a little longer to bring our users up after a problem but considering the infrequency of problems that require a data base reload and recovery, this has worked well.

Summary

System management covers so much besides what's listed above and you have to determine how much you and your staff can do. What's the cost to the company if you don't accomplish everything. What is acceptable at my organization, like 10 second response time (sometimes) is not acceptable at yours. A company that is unwilling or unable to purchase the equipment, software or staff needed may have to put up with the down side of less efficient performance.

One of the pitfalls you need to try to avoid is becoming a miracle worker. If the only way the GL meets the schedule for the finance meeting is for you to work to midnight every month then it's time to stop meeting the schedule. This is still a very hard lesson for me the learn. What brought the lesson home to me in sharp focus was an incident the I call the 'Series 48 Incident'.

System Management on a Budget

JC Strote - KVIE Channel 6
PO Box 6, Sacramento 95812-0006
(916) 929-5843

I had been using the chinese water torture method on my Manager for about 6 months explaining how overloaded our Series 30 was, why it was so slow. But all the time I was meeting all the deadlines, writing all the reports, making my users happy a clams. I did this by putting in 12 hour days, always working weekends. dialing in from home to make sure that the critical jobs ran correctly. Although, he understood the need, he didn't see the need. Then the membership department (the ones that raise the money) wanted to load last years data base to run some reports they had forgotten. I told them they would have to be offline for a day while we backed up their current base, restored their old base, ran the reports, then reloaded the current base. Normally I would have done this on a week end but I'd had lunch that week with Dave Moraio and Dave had cautioned me about miracle working. The next thing I knew the Membership manager and her Division manager had taken my boss to lunch. He approached my office (a rare occurrence) and informed me that we needed a new system because Membership was unable to have two years worth of information on the system at the same time. I told him what a great idea he had and promptly called HP.

It wasn't quite that easy but if MIS is the only one putting in the late nights and weekends and everyone else thinks that SOP, you need to do something to break the cycle. Try saying no!!

Hopefully you've picked up a tip or two from this presentation and if you have one to share with me I hope you will. System Management on a limited budget doesn't have to be bad management or even inefficient management. There are tools and tricks you can use to help you. I think the biggest problem with operating with a limited budget is that you try to do the most you can with the least resources available. This occasionally leads to problems that only $$$ can cure.

SUPPORTING THE PART-TIME SYSTEM OPERATOR

Bob Gordon
Southwestern Bell Telephone Company
Employees' Benefits Department
One Bell Plaza, Room 905
P.O. Box 655521
Dallas, Texas 75265-5521

In many of today's businesses, the computer system is now located right in the production office. These locations no longer consider the computer a sacred artifact that must be carefully watched and guarded by teams of full-time System Operators. Instead, computer systems are now viewed as tools of the office. Like the secretary's typewriter, is can be banged around, sat upon and used as a resting place for the boss's coffee cup. With this mentality comes the idea that the computer system should take care of itself and thus needs minimal attention. System Operators who work in this environment must contend with working part-time on the computer system and part or full-time on other job assignments.

My department at Southwestern Bell Telephone Company is a good example of this type of computer operation. I am in the Employees' Benefits Department and our users work 8am to 5pm, Monday through Friday, with a few ambitious souls working before 8, after 5 or on weekends. However, changing needs of the business dictate that the system must be available from 6:00am until 12:00 midnight, seven days a week.

As the System Manager for the State of Texas, I oversee three such locations: Dallas, Houston and San Antonio. Each office has appointed two of their regular benefit employees to act as part-time System Operators. Since they are actually benefit employees, they do not report to me for job assignments, evaluations or salary treatment. I must rely on them to take the time away from their benefit jobs to perform all the necessary system related tasks. I must also rely upon their supervisors to make allowances in their benefits jobs so they have time to perform these tasks. Since their primary job relates to employee benefits, these part-time System Operators, as a rule, do not come in early, stay late or work weekends.

In the Employees' Benefits Department we run only the "canned" HP software products of HPDESKMANAGER, HPWORD and HPLISTKEEPER. QUERY, IMAGE and FORTRAN are all foreign terms to our users. I am not saying that our shop couldn't do more, better or faster if we knew more about these and other products; we simply do not have the resources to investigate, learn or obtain them.

We have no "developmental" shop. In fact, in Texas, we have no programmers. Ours is a production only, or "on-line" operation. While our people are at work they need the system for the full 9 hours. Since very little of the benefit job remains a manual process, when the computer is down, all benefit employees are severely hampered.

I am the only full-time computer person on the payroll, and as such, it is necessary for me to make the computer related tasks as "user friendly" as possible for the part-time System Operators. To accomplish this, I have to view the support I give with two different, distinct goals in mind. On the one hand, I have to approach the technical aspects of making the system itself more "self-regulating". On the other hand, I have to be mindful of the human resource issues involved in supporting personnel who I have to rely upon for proper system operation but for which I have no responsibility or authority.

Making the part-time System Operators' jobs more "user friendly" has been one of my primary concern for the past 24 months. I have been able to achieve my technical and human resource goals in a number of ways:

BACKUPS ARE COPS DISGUISED AS MUGGERS

From the first day on our HP3000, we have performed daily partial backups and weekly full backups. We also ran MAILMAINT on HPDESKMANAGER daily which performed a separate backup of the HPDESK data bases. Following each complete backup, all tapes were then validated via the VALIDATE utility in the TELESUP account.

The time needed to complete these processes has always been our greatest cause of system unavailability which, in turn, is directly related to lost user productivity. Since we do not have the "luxury" of having a System Operator come in early or stay late to perform backups, our systems remained unavailable daily until 10:30. On Thursdays, our full backup day, this unavailability stretched out until 1:30 in the afternoon.

Don't get me wrong: I am NOT in favor of performing backups any less than daily. In the event of an emergency, I know I will need those backups just like I might need a "cop." However, those same backups were "mugging" us every day of the week. We had to find some way to lock up the "mugger" while leaving the "cop" standing guard.

At the beginning, I made three fundamental decisions: First, our partial backups would only be for the past 24 hours, not back to the last full backup; Secondly, we would set the tape drive to auto-reply. Both these decisions were made to reduce the time and attention required by the part-time System Operators.

The third decision was that my ultimate goal would be to reduce to zero our system's unavailability due to backups and to reduce the part-time System Operators' involvement to an average of 15 minutes per tape.

To reduce the part-time System Operators' involvement in backups, I created two UDCs, PARTBU and FULLBU, that started a series of jobs that took the system down and started the backup. This eliminated much of the time required for the Operators to find their notes and to type in the necessary commands. Also, after they had mounted and un-mounted enough tapes to complete the backup and the MAILMAINT, the jobs then restarted the system and began the tape validation program. This process reduced our down time somewhat but not enough. The system still remained down until our System Operators had changed enough tapes.

I then wrote another job which started the backup process at 6:00am each morning. By the time the part-time System Operators reported to work at 8:00am, the system was already shutdown, the backup started and the first tape complete. This again reduced our down time, however, the total amount of down time still hinged on the time it took the Operators to change the tapes.

Time for another fundamental decision: Eliminate the tape validation process. This would not reduce our down time any further but it would drastically reduce the time the part-time System Operators were involved with the system.

Now, you will hear many, very knowledgeable experts explain in great detail why you should NOT eliminate this process. The purpose of tape validation is to call your attention to a bad tape or bad data so you may correct the problem prior to the possibility of having to reload your system from it.

Let me, however, explain why I made this decision. Given our 8 to 5 working environment, we cannot afford the user time that is necessary to shut the system down and rerun the backup if a tape validates as bad. In the interactive shop, the longer the system is down for backups the less time it is up for the users. Since we could not afford the user down time, I decided that it made little difference in the event of a reload whether or not we knew in advance we had a bad tape or bad data.

To somewhat compensate for this lack of validation, the Operators review the twice weekly Predictive Support printouts very carefully for tape or tape drive problems.

By this time, the part-time System Operators' and system down time was at a respectable 1-1/2 hours for partial backups and 3 hours for full backups. Good, but not good enough; about half of our down time was still due to the system backup while the other half was due to the HPDESK backup.

At this point our SE provided us with a job that updated the access date of all HPDESK data bases to the current date. This then allowed for the combination of the HPDESK backup with the system backup, thus eliminating one of the two store-to-tape processes. This change saved two 1600' tapes and 30 minutes of down time every day. Our systems were now up by 9:00 each day and by 10:30 on Thursday.

To help achieve my ultimate goal of zero backup down time I decided to investigate our first third-party software package. BACKPACK by TYMLABS, INC., of Austin, Texas, now allows us to run un-attended backups at 3:00am and for the system to be available for the users by 6:00am every day.

For those of you unfamiliar with BACKPACK, it performs data compression prior to storing files. BACKPACK also has a "defer" mode which allows for unassisted backups. After compression, it starts storing files to a tape that was mounted previously. When that tape is filled, it writes the balance of the backup to a temporary disc file. After this process is complete, BACKPACK allows subsequent jobs to be streamed. Then, at the System Operator's leisure, tapes are changed and BACKPACK automatically writes the temporary disc file to tape.

BACKPACK has also reaped us two other important rewards. First, it reduced our tape usage by 50%. Where a full backup used to take 18 tapes, it now takes 9. Less tapes also means less time spent by the part-time System Operators in changing tapes. This time was also reduced by 50%. Secondly, BACKPACK has a great tape error recovery process built in. Prior to BACKPACK, we were experiencing tape failures which "crashed" the backup at least once a week. After switching to BACKPACK, not a single backup "crash" has occurred in 18 months.

I am sure there are other third party backup programs available which will perform the same or similar backup functions. It will be up to you to meet with the various vendors here to determine which would work best in your shop.

Through the combined efforts of our jobs and TYMLABS' software we have almost achieved my ultimate goal. In 1988, we had less than 0.1% down time due to backups and the System Operators' involvement was less than 30 minutes per day for partial backups and less than 2 hours for a full backup.

UDCs: TOOLS OF THE TRADE

Just as "Greg Shorthand" is a secretary's tool for streamlining their job, UDCs are the System Manager's tools for making system tasks more user friendly for the System Operators. We have divided our UDC files into four groups: System, User Logon, Operator and Manager.

The System UDC file is set at the system level and checks logon day-of-week and time-of-day. Depending upon the day and time, two security programs from the Contributed Software Library (CSL) are run. One, PORTPASS, provides port password protection for those terminals not physically located in a secured office and for the dial-up modem port. The second, SECURITY, allows the users to establish individual, personalized questions which must be answered between the MPE user password and the HPDESKMANAGER signon.

Our primary User Logon UDC file is set at the user level for every normal user. Its purpose is to prevent normal users from getting to a MPE colon prompt outside of HPDESKMANAGER. With the NOBREAK option in effect, user are automatically loaded into HPDESK at logon and logged off MPE when they exit HPDESK.

A separate User Logon UDC file is set at the user level for each part-time System Operator. This one allows them to "break" out of HPDESKMANAGER if they have the need and also runs the ALLOW program in the TELESUP account which gives them system console capabilities without being at the console.

The Operator UDC file is activated at the user level for OPERATOR.SYS and for each System Operator. By doing this, they are able to perform many system tasks at their normal work station instead of having to stop their current work, walk to the computer room and perform these tasks at the console.

The Operator UDC file contains many unnecessary (but nice to have) abbreviations, e.g. SJ (SHOWJOB); SJJ (SHOWJOB JOB=@J); S (SHOWME). There are also a number of "easier" to remember abbreviations, e.g. ABS (ABORTJOB #Snmbr); ABI (ABORTIO ldev); LF (LISTF @,1); SO (SHOWOUT SP). All these make it easier for the part-time System Operator to remember and easier for me when I have to walk them through a process. It is much faster to ask the part-time Operator to type SO rather than explain how to type SHOWOUT(space)SP.

Many of our hard to remember, but often needed, commands are also contained in the Operator UDC file. DSP defers a spoolfile (ALTSPOOLFILE #Onmbr;DEFER), PSP prints a deferred spoolfile (ALTSPOOLFILE #Onmbr;PRI=12) and DELSP deletes a spoolfile (DELETESPOOLFILE #Onmbr). DSON is easier to remember than DSCONTROL DSLINE;OPEN and FIXIT is a breeze compared to :RUN TERMDSM.PUB.SYS or PIT when compared to :RUN PSCREEN.PUB.TELESUP.

The Manager UDC file is set at the user level for MANAGER.SYS and for myself. This file contains many useful UDCs which I like, but which the part-time System Operators have little or no use for. The UDC command TOME fcopies a file to my screen for reading while TODEV sends a file to a designated printer. These are examples of things I do quite often but which the part-time Operators never have the need.

The Manager UDC file also contains a number of UDCs which access or run various CSL files/programs, e.g. DIRK runs DIRK of the TECH account, UDCUTIL runs UDCUTIL also of the TECH account and PLAY runs programs in the GAMES account (All work and no play makes Bob crazy!).

MECHANIZE, MECHANIZE, MECHANIZE

To make the system more manageable for the part-time System Operators and life more bearable for myself, I undertook to mechanize as many of our system utilities as possible. I figured the more I could accomplish via jobs, the less time I would have to spend walking the System Operators through the different processes.

Following our morning backups, a job streams which schedules the next backup. If today is Wednesday, it schedules a full backup for tomorrow. If today precedes a company holiday, it schedules the next backup for the next business day. This job then streams another which brings the system up for the day and automatically logs all users onto their appropriate terminal via the STARTSESS command.

The part-time System Operators also have specific tasks they are to perform periodically, such as advising the users to change their HPDESK password, sending me notification of their down time for the prior month, etc. In the past, I would call them or send a message reminding them it was time to perform the task. Tiring of this, I now have a job that, on the appropriate date, sends the Operators a reminder notice.

A whole host of other system tasks can be performed without having to get the part-time System Operators involved. Below is a partial list of regularly scheduled system utilities we perform via jobs which require NO System Operator involvement:

* Run three VINIT condenses at midnight every Saturday.
# Purge all LOG#####.PUB.SYS files that haven't been accessed in 7 days.
- Purge all READY spoolfiles that remain un-needed at the end of the week.
~ Run monthly capacity reports to check current system utilization and for predicting future usage.
^ Run CSL program REPORTER to obtain system usage information for forecasting purposes.
* Stream Predictive Support twice a week.
# Shut the system down each morning before the backup starts then restart it after the backup is complete.
- Start and stop various system-wide utilities, e.g. PRINT CENTRAL and BOUNCER.
~ Start and stop data communications links, e.g. SNA and DS, before and after each work day.
^ Check if the system clock needs to be changed for Daylight Savings Time, and if so, change it.

Since I manage one local and two remote sites, I generally have these utilities write their results to disc files, then have HPDESK mail the text of the disc file to me. Deferred spoolfiles are another good way to remotely view the results of these jobs. To me, these methods are better than letting the $STDLIST print to the line printer and having the System Operators mail the printout to me (You haven't seen our intra-company mail system!).

From the start, the part-time System Operators complained about the large volume of paper that did print on the line printer each day. Not only did they have to review all of it, hunting line by line for errors, they also had to decide what to trash (95%), what to keep (2%) and what to refer to me for further investigation (3%). To eliminate this headache, all our jobs have been modified to contain two items which eliminates much of this work.

First, so that we don't spend a lot of time reviewing a bunch of unnecessary spoolfiles (remember 95% above?), our jobs have been made somewhat self-error detecting. Following the job card, I reinsure the JCW CIERROR is initialized by :SETJCW CIERROR=0. Then just before the EOJ, I enter:

```
:IF CIERROR = 0 THEN
:   SET STDLIST = DELETE
:ENDIF
:EOJ
```

On jobs that perform multiple functions, I added an additional JCW called OUTPUT and initialize it by :SETJCW OUTPUT=0. At the beginning of each function I set the JCW CIERROR to 0 and at the end of each function I check it with:

```
:IF CIERROR <> 0 then
:   SETJCW OUTPUT = 1
:ENDIF
```

I close the job with:

```
:IF OUTPUT = 0 THEN
:   SET STDLIST = DELETE
:ENDIF
:EOJ
```

If the job completes without any CI errors being detected, the $STDLIST, to whit the spoolfile, is deleted. Now the only READY spoolfiles I see are those where a CIERROR was detected.

Secondly, each job card includes the entry OUTCLASS=LP,1. This defers the printing of the $STDLIST spoolfile so that it can be reviewed later using SPOOK5. If I decide that a printed copy is needed, I copy the deferred spoolfile to a disc file, use HPDESK to mail it to myself at my "home" location, and then print it locally.

Thanks to these two small but significant changes, I now spend about one hour per week looking at READY spoolfiles on three different systems rather than each of six Operators spending one hour per day.

DON'T LET SECURITY LOCK YOUR OUT

During this conference you have seen numerous classes which address system security. Almost every issue of every DP periodical will have at least one article regarding security. There are a number of vendors you will meet that will try to convince you that security must be your number one, top priority.

Is there such a thing as "too much security?" NO! and yes. As with all things in life, anything in excess can be detrimental under certain circumstances. This is what I discovered when we started off with a system security policy that was too strict for our circumstances.

Initially, MANAGER.SYS and myself were the only users with SM capability. OPERATOR.SYS was the only user with OP capability. Whenever a part-time System Operator needed to perform a system function, e.g. :SHOWOUT SP, they had to either take a hike from their desk to the computer room or they had to logoff as themselves and log back on as OPERATOR.SYS, then re-logon as themselves after they finished the task.

In talks with the part-time System Operators, I discovered this policy was costing us a minimum of 5 minutes in lost production every time one of the Operators had to perform some system function. Remembering that the time these people have allocated to the system is limited, forced me to re-evaluate our policy. Now each part-time System Operator has SM capability. SM was picked over OP because it totally minimizes the need for an Operator to leave their work station to perform system functions.

Also, through the added use of HPDESK Script Files which emulate the Operator UDCs, the part-time System Operators can now perform 98% of the needed system functions from their own logon without any lost "hiking" or "logging" time.

I also discovered during our talks that too much time was being spent by the Operators each morning helping users logon (It's amazing how many users can't remember HELLO or forget their MPE passwords!). I eliminated all such lost time by writing a job that logs every user onto the system each morning. We are able to do this with little or no loss of security because:

1. All but five of our terminals are located within physically secured Benefit Offices.
2. Each user has their own, unique terminal.
3. The User's Logon UDC file automatically loads all users into HPDESK which prompts for a separate, secured password.
4. The User's Logon UDC file does not allow the BREAK key to be used to stop a user's entry into HPDESK or to break out of it once signed on.
5. The User's Logon UDC file automatically logs each user off MPE if entry to HPDESK failed or after signing off HPDESK.

All this is great, but what about when a user is on vacation? Their terminal is now active and they aren't there to use it. Southwestern Bell's Corporate Computer Security policy states that any terminal that remains inactive after 15 minutes should be automatically logged off. In the past is was the System Operators' responsibility to deactivate (logoff) every active terminal for which there was no user that day. They were also responsible for insuring that every user logged off at the end of the work day. Again, you can imagine the amount of time this required.

To bring our systems into compliance with corporate policy and to eliminate the time spent by the part-time System Operators' on this task, we utilize another CSL program called BOUNCER. BOUNCER checks each session for a predetermined length of terminal inactivity. When that inactivity limit is reached, BOUNCER aborts the user's session.

Another feature of BOUNCER is that it keeps track of every session it aborts. By using BOUNCER's data file and a job that I wrote, the names of the last 10 users that were aborted are sent via HPDESK to the supervisors in each office. This now makes the bosses of each office responsible for monitoring system security thus eliminating the System Operators' involvement.

As mentioned previously, the CSL program PORTPASS is another way in which we make security work for us. Prior to PORTPASS, we unplugged the dial-up modems from the telephone network at the end of each work day. This of course prevented any possible dial-in security breaches. However, it also prevented me from dialing in from home to do any work at night. With the initiation of PORTPASS, the modems remain plugged in at night. PORTPASS runs after MPE passwords are completed but before any application can be run. Since PORTPASS disconnects the data link after one incorrect attempt the likelihood that a hacker will keep calling back trying different passwords is remote.

Another way we make security work for us is by customizing the file CATALOG.PUB.SYS. This is the place where all those nice, friendly logon errors messages are stored. Anyone with minimal computer knowledge can figure out the proper logon command sequence (:HELLO [username].[useracct]) just by watching the error messages MPE so promptly provides. I changed every error message within this file which pertained to an improper logon to read "INVALID LOG ON"; no explanations, no excuses. Since our users are logged on via a job each morning, the part-time System Operators know that if they see an invalid logon message on the console, someone is attempting to breach security.

NOT ALL OPERATORS ARE CREATED EQUAL

Quite often the decision regarding the appointment of the part-time System Operators is made by non-system supervisors and usually these part-time people continue to report to their non-system supervisor for job evaluation and salary treatment. They are just "loaned" to the system for a couple of hours each day.

As the System Manager, you may be lucky and get part-time System Operators who perform a task one time and remember it forever. More likely, these appointed Operators will be non-technical, non-computer minded in background and training and may or may not have the desire to learn. They also may or may not have the ability to apply the technical skills they do learn. Either way, you must learn to adapt your managerial skills to suit the needs of the part-time Operators and the situation at hand. Remember, the normal managerial "leverage" of job evaluation and salary treatment are not present.

OPERATORS ARE PEOPLE TOO

Due to limited computer responsibilities and/or limited budgets, the part-time System Operators may receive very little, if any, formal system training. The benefits of sending a part-time Operator to a $1,500, 5-day training class are generally not satisfactory to convince upper management of the expense. At times, this lack of knowledge and expertise may frustrate these part-time people. If allowed to continue unchecked, this frustration will eventually lead to feelings of system "inadequacy" or "incompetence."

You must constantly keep in mind that these part-time System Operators have been given the responsibility of supporting the system without any accompanying authority. You must be willing to transfer some of your authority to them. This can be accomplished in a number of small, but significant ways.

First, except for after-hours and weekends, insist that the end-users funnel all their questions and problems through the part-time System Operators; if the Operators cannot answer the question or resolve the problem, they will come to you. This imparts upon the end-users the idea that the System Operators, although only working part-time, are the system experts to be consulted.

Secondly, constantly remind yourself that these part-time System Operators are generally not technical, computer-minded people. Be sure that you do not talk down to them, nor that you talk over their heads. Remind them often that you are their support and their guide; you are not their dictator. They must feel free to ask questions or request explanations whenever they have the need.

Thirdly, allow the part-time System Operators to have whatever "crutches" they feel are necessary. If they feel they must have written notes or "crib" sheets for future reference, be patient. It may be a very minor, simple task to you, but to them it can be quite scary. This is not to say your should make the "crutches" for them; let those that need the crutches make the crutches. It's very possible they will learn a task just by making the crutch.

Fourth, consult with the part-time System Operators. I used to think that I knew what was best for the Operators and what I did would save them time. Once or twice I was wrong and ended up actually creating more work or requiring more time of them.

Last, but far from least, be willing to expend some of your time and effort in educating your part-time System Operators.

If you try to make the part-time System Operators conform to the system or to your way of doing things, they will avoid or delay the execution of each and every task they perceive as being "over their heads" or too "high-tech" for their abilities. Quite often this surfaces as the excuse, "I didn't have time.", and their non-system supervisors will be more than willing to tell you just how busy they are on their regular, full-time job.

I take a proactive approach to this problem. I make it a point to chat with each of the part-time System Operators at least once each week. This is my opportunity to obtain feedback from them on how they are reacting to the system, the users, the bosses and myself. It's amazing the number of problems that have been identified and resolved by this very simple procedure.

Whenever I introduce a new task, job or program, I always start off by explaining it to the Operators and I check with them again after it has been working for 2 or 3 weeks. To make this work, I must be willing to alter my plans if this new task is not compatible to the Operators' knowledge and abilities.

I also host an annual two day training/discussion session where all the part-time System Operators come together. Prior to the session, I solicit from them the topics they would like further explanation or training. The last four hours is reserved for discussion of topics raised from the floor. This session allows for the open exchange of ideas between myself and the Operators and amongst the Operators themselves. The Operators' supervisors are invited to attend the first day, however, the second day remains for the Operators' exclusively.

OPERATORS CAN BE TRAINED

One of my primary functions as a System Manager has been to develop and expand the technical knowledge and abilities of the part-time System Operators. A first step to this process was the development of a System Operator's Responsibility/Time Table. This document describes in some detail all the computer related tasks the Operators are expected to perform and when they are to be completed (daily, weekly, monthly, specific date or as needed).

This Responsibility/Time Table was a must not only for the part-time System Operators but also for their supervisors. If the supervisors who have job evaluation and salary treatment responsibilities over the Operators do not understand what tasks are to be performed, when they are to be completed, and how much time is involved in completing them, they will soon expect less time to be spent on computer functions and more on the primary job functions. This would leave the part-time System Operator, myself and the system between a rock and a hard place.

I also developed an easy to read, in-house training binder that explains in great detail how each item described in the System Operators' Responsibility/Time Table is to be performed. The Operators can then review this binder at their leisure and have it available whenever they have a need. This binder also covers some of the tasks performed by the System Manager. This provides the Operators with some insight into the "big picture" and helps them understand what is involved should they want to prepare themselves to replace me whenever I get my next promotion.

I am always exploring various means of training the part-time System Operators. This might be as simple as providing them data sheets which explain where in the existing documentation (MPE V COMMANDS MANUAL, SYSTEM OPERATORS MANUAL, etc.) answers can be found, up to and including detailed one-on-one training. Either way, or anywhere in between, I know that I have to expend a great deal of effort and patience to eventually obtain the desired results of a trained, self-sufficient part-time System Operator.

In our department, I am now utilizing an audio-digital training medium marketed by USER TRAINING SERVICES GROUP of Palo Alto, CA. This medium uses devices connected between the user's terminal and the HP3000 which records and plays audio and digital information on standard audio cassettes. If an Operator wanted information on how the LISTF command works, I could record a verbal explanation while simultaneously recording what appears on the my terminal's screen during a LISTF. This allows them to see what happens when a LISTF is performed and to hear me explain it as it appears. After receiving the tape, the Operator can then review, and re-review, the explanation whenever they have the time. The tape can then be sent to other Operators for their review or it can be stored for future reference.

There are any number of other training systems or packages on the market to fulfill your needs. Don't overlook these products as you browse through the vendor area or review product mail-outs.

KEEP YOURSELF TOUCHABLE

To further eliminate the part-time System Operators' anxiety, they must be able to "reach out and touch" the System Manager whenever THEY deem it necessary. If they become "stuck" performing a task, they will probably feel they must have the Manager, right then, to get them unstuck. Granted, they might be able to expend a little effort to locate the answer themselves, but all the while the users, bosses, and the task itself are all placing a strain on them to complete the task as rapidly as possible. If the Manager is not available, they quickly become candidates for the company's mental counselor.

The System Manager should do whatever is necessary to stay within quick communications of the part-time System Operators. Think back to the last time you tried to contact a medical professional after-hours. Between the time you hung up from the answering service and the time the professional called you back seemed like, and may well have been, hours. To the part-time System Operators, you are the professional that has the prescription to what ails them and they need you just as quickly as you needed your doctor.

Whenever a part-time System Operator is working after-hours or on a weekend (e.g. performing a full system reload) I always make sure they know where and how I can be reached. If they hit a "snag," many hours could be wasted trying to track me down to get an answer to their question. I also think it is a good idea to check in periodically just to encourage them in the task.

Maintaining this communications link can be accomplished through any number of today's telecommunications devices: Call Forwarding, Call Waiting, an answering machine, a personal pager or a mobile telephone are just a few. How you convince your company to pay for these items, however, is your problem.

Also, I do not encourage the part-time System Operators to be available to end-users after-hours or on weekends; I take this responsibility myself. By doing this I relieve them of the burden of keeping a second set of reference and personal notes at home, without which they probably couldn't answer the question or resolve the problem. I also am letting them know that I understand they only have part-time responsibility for the system and are not expected to treat it as a full-time job.

MAKE IT EASY ON YOURSELF

Every single thing you can learn about your job, the system, your applications or human resource issues, will enhance your ability to support the part-time System Operators. Every tool, every trick, every shortcut you learn can, at some time or another, help you or the Operators in keeping the HP3000 a soft bear skin on the floor rather than a raging wild animal crashing down the door.

For example, I learned about a program called CAP that allows a user with SM capability to switch logons without having to physically logoff as one user and log back on as another. This program also allows the user to keep the UDCs set on their original logon.

We utilize HPWORD template documents extensively and store these templates within the PUB group of each of the user accounts. It is necessary for me to frequently change from my normal logon to that of the Manager of each user account to install, modify and delete these templates. By running the CAP program rather than logging off, I save myself at least 3 hours per week.

This points to another fundamental I have placed into effect: Make the time available to attend every training class, seminar, Regional Users' Group meeting, vendor demonstration or any other learning opportunity that you or your company can or will afford. Granted, you get what you pay for, however, there are still a lot of very good learning opportunities available that are inexpensive or free for the asking.

You may walk away from a few of these sessions not having learned a thing. (Hopefully this will be among the free ones!) Others, like last year's INTEREX conference in Orlando, inundated me with so much good information that I am still trying to digest it all. From my experience, I have always gained a little knowledge at every meeting, session or seminar. At the very least, I have learned the name and telephone number of someone who has had experiences similar to my own, thus allowing me to network with other users for additional information.

Also take the time to read the trade journals. I never fail to review INTERACT, INTERRUPT, HP CHRONICLE and HP PROFESSIONAL cover to cover. Many of the articles are so far above my knowledge level that I don't always understand what I am reading, but when I get through, I always know a little more about the topic than before I started. And usually the articles which I feel are too basic always teach me some new trick or a better way to perform a particular function.

I do not overlook Human Resource or Management Development courses as I plan my personal training. Whether provided through the company, the local community college or on the open market, these courses help me to interact on a more professional level with the part-time System Operators and their supervisors. It is no small task to manage personnel for whom you have no managerial responsibilities for or to continually justify their need on the system to those who have this responsibility.

Last but not least, I don't let the problems I can't fix worry me. Case in point: I have been very concerned about system performance and how I might improve it. However, bottom line, with the "canned" products being 99% of what we run and me not being a programmer, there's not one single thing I can do to the programs to increase their effectiveness. If HPWORD or HPDESK aren't running at their peak, I can only grin and bear it. I haven't forgotten about system performance, I am just putting my resources elsewhere where I can have an affect.

LOOKING AT THE BOTTOM LINE

The bottom line of supporting part-time System Operators equates to a two-tiered solution:

A - Develop the technical expertise needed to streamline the System Operators' functions and to reduce the time required of them to support the system, and

B - Develop the human resource skills necessary to work with a variety of personnel who have a variety of interests. It is not enough to tell the part-time System Operators what to do. They and their management team must be understand and appreciate the importance of their role in supporting the system.

Whenever a System Manager combines technical expertise with good human resource skills, the result will be a well run computer operation where the part-time System Operators are constantly growing in their knowledge and skills. The ultimate payoff of these efforts will be demonstrated when one of these part-time people steps into your shoes as you move on to greater challenges.

Methods, Challenges and Madness - Small Shop Management

Joe Berry
Pekin Memorial Hospital
Court & 14th Streets
Pekin, IL 61554-5098

Introduction

The madness is the vision of having our system available to users twenty-four hours per day, an expert staff, and users who can trouble shoot potential problems before calling the MIS department.

The challenge is to manage a hospital information system with the equivalent of four full-time employees. This includes system management for a Series 70 CPU with over 60 users that must be available as many hours of the day as possible. The department is staffed on third shift every night for backup and daily batch processing, plus first shift through the week with non-staffed hours covered by beeper support.

We have developed a repertoire of methods to meet the challenge. These include a mix of in-house developed utilities, contributed library programs, and third party software programs. We firmly believe that quality will be the result when you train with good written procedures and extensive program documentation. Standardization of all programming conventions is a must along with maintaining current updated reference manuals for all products. A dedicated and loyal staff is a big plus.

The small shop is unique because everyone does a little bit of everything. In our shop all five of our staff must do 3000 system operations and management, help with applications software, provide personal computer hardware and software support, teach user training classes, learn how to use the tools that help them in their job such as Adager and MPEX, do programming using Powerhouse, and maintain in-house written programs. They must learn to wear many hats and keep many different projects going at the same time.

Small shop management is unique and challenging. It requires methods and madness to get the job done and remain almost sane.

Automation

It is our policy to automate everything that will save time and prevent manual effort. Memory, handwritten reminders on calendars, and manual tasks can no longer be relied upon to get the job done. There is too much going on at once. MIS departments need automation as much as or more than most of our users.

MPE job stream scheduling is one of the easiest and simplest ways to automate the operations environment of any shop. You can schedule those once a week or once a month jobs instead of trying to remember when to do them. Job scheduling requires a small investment of time to learn the syntax and can return many rewards for both the MIS department and the users. It is a simple way to automate and eliminate an error prone task.

There are programs in the CSL along with Streamx from Vesoft that help automate job streams by prompting for input from the user at the time the job is streamed. These programs eliminate the chore of repeatedly revising parms,such as dates, in job streams. They allow you to write more generic job streams; hence, reducing the amount of code you must maintain and eliminating the errors that can result from the failure to revise the parms. This is the type of help that is necessary to keep the staffs productivity at a peak and the error rate low.

There are several software packages on the market that will automatically examine the $Stdlist for job streams to find errors as defined in a master file. This is an area of great concern to many shops. The results of crucial jobs cannot be left to an operator who has several projects going at once; therefore, these packages represent a way to eliminate the probability for errors or omissions when reviewing $Stdlist. Some of these products are Job Rescue by NSD, Inc, Spoolmate by Unison, and Unispool by Holland House. All of these products do other tasks as well so they are very easy to cost justify.

Documentation/Manuals/Procedures

The key to bringing a new operator up to speed is providing a detailed set of operations procedures that eliminate the need for research steps and provide the reasons for the duties performed. These procedures must be kept current religiously and reviewed periodically by management to make sure they are still viable. The time it takes to write a new procedure will be returned many times over by reduced operator time and reduced errors. This is a very modest contribution; however, it ranks as the one with the most returns in my book.

The user must be provided with application manuals. These manuals should be kept current and made available to all who have the need for them. Users can be self sufficient; however, without proper documentation they are completely reliant upon the MIS department for assistance when a problem or perceived problem arises.

The manuals that came with the CPU and all other manuals related to the operating system, utilities, intrinsics, and third party software should be kept in an easily accessible location in the MIS department. Like the application manuals it is of paramount importance that they be kept current as every new update arrives. The manuals provide the reference material that everyone will need at some point. If they are not up to date they cannot provide the fullest support.

Standards/Policies

Standards for programming provide an accepted way to write programs. Standards are necessary for MIS because they provide consistency and eliminate surprises for both the users and the MIS department. Without standards or policies the MIS department would be like a plane without a flight plan.

Programming standards help the MIS department locate, maintain, and document programs. Some of the more important aspects of programming standards are naming conventions for both source code files and for object code files, documentation and modification requirements and format, and actual program structure. These standards require a little effort when the program is developed but avert research time later.

Even if you do not restrict disc space by user you must have a written policy regarding disc space. Without such a policy you fall prey to users who want to maintain many megabytes of data that could easily be archived. A charge back policy based on disc space is but one method to inhibit users from keeping a lot of archivable data on the system and it helps to prevent the users from viewing the MIS department as being uncooperative.

There should be standards that are shared with your users about MIS responsibilities and the hours of availability. A user should know what to expect of the MIS department and should know how to get in touch with the MIS department when ever it is necessary. Since we are not staffed 24 hours per day we have a policy that outlines when we are available, what type of services we offer and how to reach us if an emergency arises during the hours we are not staffed. This sets expectations for both the user and the MIS department. Without these many hours can be used debating who is going to solve a problem and when it will be solved.

Personal computers and the software and maintenance purchased for them require policies to block many undesirable effects that can happen if all users can purchase their own software and maintenance. A joint effort between users and the MIS department must be made in order to standardize the software purchased. This will help keep users' files compatible with each other and other software products. It will help the MIS department support the software if that is one of the MIS department's established responsibilities. Uniformity in hardware purchases help reduce costs because volume purchases for supplies can be made. Obtaining maintenance from one vendor has many advantages including reduced costs and potentially better vendor relationships can be built.

System Management

System management for the small shop becomes more of a science than an art. Tools have to be used to tune response time and monitor overall performance. Tools also have to be used to deter potential problems via preventative measures. Disc space must be monitored on a regular and conscientious schedule. Tools must be developed and used to prevent disc space from vanishing abruptly.

A small shop means that everyone has to have the knowledge to monitor system performance and tune accordingly; consequently, the monitors have to be simple and easy to use. The utilities that we have found that fit this category are SOO5E from the contributed library for on-demand monitoring or CPU usage, Auditor from the contributed library for historical and current CPU usage by user and/or by port, and OPT for more in-depth analysis of performance and system tables to fit the bill. None of these utilities are difficult to use or interpret but are powerful enough to serve the purpose well.

As a preventative measure we condense our disc drives regularly. This not only helps performance and but also helps us maintain larger chunks of contiguous freespace that helps prevent the need for reloads. We use a contributed library program that checks the discs and determines via user defined parms what drives are eligible to be condensed. I found that this program saves time and helps us make sure that the discs that need condensing get condensed promptly. Another preventative measure is the use of the contributed library program DISCIO. It determines which files on which devices have the most I/O's. With this program we can determine if any files need to be moved to a more or less active drive. Our policy is to put KSAM key files and KSAM data files on different drives to enhance performance.

Blocking factors are very key to performance and freespace. Datasets are reblocked according to Adager's formula. The contributed library program Block or MPEX is used to determine the optimum blocking factor for MPE files. We also use the optimum number of extents whenever feasible. For performance and space considerations we use an in-house developed program to monitor datasets to prevent too much wasted disc space and/or datasets that are too full.

Disc space is the entire MIS staff's responsibility. Written standards and regular (daily or weekly) monitoring of disc space is a must in any environment. In order to keep it simple and automated for our small shop we formulated an in-house program that not only prints daily reports but also monitors the growth patterns of datasets. This makes operations easier and helps us predict hardware purchases. The daily reports include a modified Query form sets listing and a more fully detailed FREE5 report. In addition to regular monitoring you need a written policy regarding the archiving of data. Without this policy no data is ever eligible for archival. To help improve freespace we automated the purging of system log files, Editor 'K' files, and Quad 'Q' files using MPEX to select via create date. There are contributed library programs that have this capability. Source code files and documentation files are Squished using Squisher from the Tech account of the contributed library. This cuts the disc usage by about 50%. We also require that a Recover Lost Disc Space be scheduled after every system failure or system hang.

Security is an important part of any shop. In a small shop it is important but not as important as in a larger shop. We maintain physical security by locking the doors and limiting access to the computer room. We use Security 3000 to secure our dial up ports and limit access to MPE via menus. Only the MIS department can get to a colon prompt.

Training

The single most important key for the success of a small shop is the training of both the operations staff and the users. Without proper training in the shortest time frame everyone suffers. Training is a big investment of time and money for any MIS department. The impact of the training is felt deeply in a small shop because of the limited number of people in the department. Training must be focused and intensive.

MIS staff training should consist of a formal document that outlines the minimum knowledge level acceptable after the training period. It also should estimate the number of hours each topic outlined in the training plan should take. This document will help both the trainer and the trainee. Both will know what is expected of them from the very beginning. In addition to the initial training plan there must be a formal plan to further cultivate the new employee and help him/her grow and be more valuable. Having this plan in writing helps to keep it a living document with goals and time frames. As we all know the world of computers is not a stagnant one and continuing education is a part of the territory. Take advantage of outside education and special interest groups including the local users' group. Confidence seems to be the one trait that you cannot teach an employee. Instill confidence and support them. Show them that mistakes can be corrected. After all, we all have made one every now and then. An employee that is too conservative and cautious to ever make a mistake lacks self-confidence and is not working at full capacity until they can dare to make mistakes.

Cross training in a small shop is a must. This again must be a part of a formal training plan that is followed from the date of hire. Without cross training vacations and terminations can have a prolonged effect on the MIS department and the entire user base.

Staff training and continuing education and development are musts for a small shop. The small shop cannot afford to settle for mediocre staff. You must get the best quality of staff possible and pay them accordingly. Not only is your staff a big investment of your time and money they are your department's image to the users.

The training of your users is just as important as training your staff. A small shop cannot afford to have a user base that is not somewhat self sufficient. Believe it or not users can be trained to be self sufficient. It doesn't happen over night and you cannot train all of them, but you can reduce the users' dependence on the MIS department.

Reduce the users' dependence on the MIS department by automating their environment in a friendly way, buy software packages that have been proven and are easy to learn, take advantage of outside sources for education, and make sure there is application documentation available. None of these are hard to accomplish and the rewards are superb.

Automate their environment to eliminating their need to call you for help. Allow reports to be printed outside the MIS department. Allow them to run their own reports from screens that maintain security but still give them independence. Create batch files for personal computers and purchase a good DOS shell to eliminate the users' need to learn DOS. Couple this with proven personal computer software packages and the user will be less dependent on you and feel like he has a little control over the automation in his life.

Take advantage of outside sources of education for your personal computer users as well as your 3000 users. Seminars offered by vendors, local colleges, and user groups are affordable and allow the user to interact with his peers. This will serve to reduce his dependence on the MIS department remarkably.

Provide good documentation that is easily accessible. Whenever possible lead the user through using the documentation. This will help instill in his mind that you use the manuals.

Conclusion

Small shop management is unique because the MIS staff needs to be a jack of all trade. Automation and planning are two of the most helpful assets to a small shop. In many ways the small shop fights the same battles as the larger shops, we just use fewer people.

None of the methods outlined for small shop management are difficult to accomplish. If only part of them were implemented at Pekin Memorial Hospital the results would not be as positive. It is all the methods combined with planning and formal documentation that meets the challenge head-on. The madness continues regardless of the methods and the outcome of the challenges but that's because we're in the field of computers where it requires madness to remain almost sane.

WIRING A MULTI-STORY OFFICE FOR VOICE AND DATA

PAUL EDWARDS
BRADMARK COMPUTER SYSTEMS, INC.
1506 ESTATES WAY
CARROLLTON TX 75006
(214) 242-6660

CASE STUDY: WIRING A MULTI-STORY OFFICE FOR VOICE AND DATA

ABSTRACT

Wiring a multi-story office presents many challenges. To allow the total flexibility of any voice or data device to be connected to any individual office outlet adds additional challenges.

This case study deals with an environment consisting of multiple HP3000 computers, a telephone switch with single and multi-line telephones, terminals, terminal printers, laser printers, PCs, and fax machines. The applications include word processing, Real Estate document preparation, remote site data communications, and program development. Physical wiring layout, voice/data flexibility, modular connections, RS232/RS422 considerations, and terminal/printer device connections will be discussed. Wiring diagrams for all connections will be presented.

The lessons learned in this case study will be applicable for small to large communications installations.

CASE STUDY: WIRING A MULTI-STORY OFFICE FOR VOICE AND DATA

This case study outlines the design and implementation of hardwired internal and external communications that were installed in a thirty lawyer law firm. Two prior installations for this law firm in previous locations provided the painful experience to complete this project with much greater success. The main areas this case study will deal with are Design Considerations, Environment, and Installation. Diagrams follow at the end.

DESIGN CONSIDERATIONS. Making the right connections from your computer system to your user's input and output devices are as important in the success of a computer installation as the time spent in the proper design of a software system. The design criteria should include flexibility, voice and data access, device independence, CPU and telephone switch selection, low cost, low maintenance, easy troubleshooting, and RS232 or RS422 connections.

Because no installation stays static for very long, the flexibility that is built into the system is the key for future growth and the inevitable changes. People will be moved from place to place in the office and new personnel will be added. The minimum of one dual modular plate should be at each current and proposed location. It is much cheaper to wire all possible locations initially than to add wiring later. To insure the various devices have the proper number of wires available, a three-pair connection to each modular jack should be used.

With the need for communications of all types, access to both voice and data channels is required. The devices used in most offices today are telephones, terminals, printers, Personal Computers, and FAX machines. Since most workstations need telephone access, this occupies one of the two available modular outlets in a minimum configuration. The choice of one or two duplex modular plates at each location is most critical because it determines future flexibility and cost. When in doubt, have more connections available than needed at the time of initial installation.

The use of each modular jack in a plate could be varied from time to time. The need to be independent of device type is key to flexibility. A telephone may be plugged in today and a printer tomorrow. The wiring therefore must be the same for all jacks in the installation. The choice of wiring each lead in the jack is made in the central termination location. An RJ-11 modular jack with 6 wires is used throughout the installation for both voice and data.

Selecting the CPU and telephone switch equipment is important for the future connection of devices. The newer computers have access for RS232 and RS422 data devices. Also connection of the computer ports through the telephone switch in a port contention switching configuration save the cost of ports on lightly used terminal devices.

Cost is very important to most companies today. By designing with the future in mind, costs are lower today and tomorrow. The cost of removing or moving existing wiring is mostly labor which is the most expensive part of wiring installations. Many of the new LAN systems will utilize the existing wiring discussed here, also. The only additional costs required to expand this installation in the future are port controllers for the CPU, trunk cables for the CPU-to-termination room connections, and wires and connectors for the additional devices.

This installation required the moving of punched-down jumper wires in the central termination room only. These changes can be made by system management personnel as needed. This keeps the maintenance to a minimum. Creation of complete documentation will assist with future maintenance. Strict adherence to standards set during the design phase is imperative.

Because the system uses modular connectors and punch-down blocks, troubleshooting can be done at various points in each circuit. Testing can be very difficult when several vendors are involved in the installation. By having several test points in each circuit, vendor finger pointing can be eliminated.

The choice of RS232 vs. RS422 is determined by the distance from the CPU to a device. The "supported" length of RS232 is fifty feet. Many people have made runs of almost one thousand feet. The practical length is determined by not only the distance of all cables and trunk lines but the number of connections in between and any electrical interference encountered. My rule of thumb is about four hundred feet for RS232. I use RS422 above that. Don't forget to measure all horizontal as well as vertical runs.

ENVIRONMENT. The environment of this case study consists of the building, all voice and data equipment, computer software systems, and user applications.

The building was multi-story and of recent construction. The office occupied several stories in the middle of the building. Since the floors were contiguous, wiring between floors was easy.

Voice equipment consisted of telephones, a Northern Telecom SL-1 switch, and Xerox fax machines. The computer systems were a Hewlett-Packard 3000 series 68 and a Hewlett-Packard 3000 series 70. Hewlett-Packard data entry terminals, Xerox dot-matrix printers, LaserJet printers, and Xerox spooled laser printers were operated by the user personnel. Data communications equipment consisted of Codex modems and multiplexors, HP support modems, programmer access modems, and Codex leased line modems for remote system access. The programmer access modems were also used with Telemon Network Engines for dial-out access to remote legal data bases for research. A Telemon PBX Engine was used to supply call recording information from the telephone switch to an accounting system on one of the HP 3000's.

The computer system software was standard HP MPE V-E fundamental operating system. Compilers were Basic and COBOL. Data base maintenance software was DBGENERAL. Special data base access was OMNIDEX. Data communications was originally MTS/3000 for the remote office. That was later changed to modems and multiplexors. Many programs were used from the INTEREX CSL for everyday operations. DS/3000 was used between systems for data transfer from a remote system and multiple application system access by users. POWERHOUSE was used for some of the application system development and production. HP LISTKEEPER was used for firm mailing lists.

The user application systems were HPWORD/3000 for word processing, legal time and billing from Computrac, and several in-house designed systems. The in-house systems consisted of Real Estate residential loan closing document preparation, Real Estate loan foreclosure document preparation, and word processing document file archival.

INSTALLATION. The installation was done in a modular fashion. It consisted of CPU, telephone switch, main termination room, floor termination rooms, wall plates with jacks, device connections, and documentation.

The wiring from the CPUs to the main termination room was done with trunk cables. At the CPU end, the 3-pin RS232 or 5-pin RS422 connectors were wired to the 25-pair trunk cables. This meant that 16 RS232 ports or 10 RS422 ports could be connected to each trunk cable. There were 12 RS232 or RS422 ports per ATP card in the CPUs. Because of this difference in cable connectors available and port connectors, careful planning was required to ensure the proper number of ports and cable connectors were matched to eliminate a gross mismatch in connectors. The mixture of RS232 and RS422 added an additional level of complexity to the already complex installation. The main termination room end of the trunk cables were punched down on blocks mounted at one end of the room.

The telephone switch was wired to the main termination room with 50-pair trunk cables. These trunk cables were punched down on blocks near the other end of the room. All port contention switch ports were included in this wiring.

In the main termination room, all connections were made to punch down blocks on one wall. The outside telephone trunk lines and modem phone lines were terminated on blocks at one end of the room. Next were the telephone switch blocks. The blocks for all the outlets on all floors were in the middle. The CPU connector blocks were on the other end of the room. This allowed the outlet connectors to be easily connected to CPU ports or telephone extensions. The outside phone lines were easily connected to the telephone switch blocks.

On each floor was a termination room. All individual jacks for each location on that floor were punched down on blocks in that room. One-hundred pair trunk cables ran from each floor to the main termination room blocks.

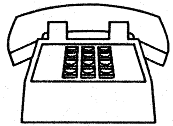
At each outlet, 3-pair wiring was made to each jack and a number was written in indelible ink on the plate for that jack. Each floor had unique numbers starting with one.

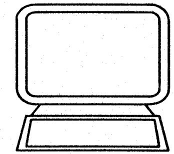
The connections from the individual jacks to the devices were made with flat silver-satin 6-lead telephone cord. The telephone cables were wired in reverse order from end to end while data cables were wired straight through. These connections were made easy with a special crimping tool and required no soldering. Since the wires were colored and the plugs were clear, the type of wiring (data or telephone) was seen quite easily by holding the two ends of the cable together with the locking tabs facing the same direction. The data cable had the wires in the same order while the telephone cable was reversed.

Standard 25-pin connectors are used for serial devices. A special hood was available with a modular jack on it. The internal wiring from the jack to the connector was made so that it conformed to the HP cabling manual. Since there were several types of terminal connectors, all connections were made to a 25-pin connector on the terminal or an HP cable. The 262X terminals had a 50-pin connector that required an HP cable.

Documentation of the wiring and device locations was most important for implementation and maintenance. Floor diagrams were used to locate each device. The number on each jack was written on the diagrams. A list was made by floor of each jack number, floor plan location number, device type, CPU connection , CPU port number, and the name of the device operator. This list was sorted in many ways and printed often.

Since the color code wiring scheme is not critical but the actual pin-to-pin wiring is, consistency is paramount. An industry standard telephone scheme was used to allow for future maintenance by installation personnel who were not involved in the initial installation. As with any properly implemented project, the more time spent in design, the less time spent in redoing the implementation.





CASE STUDY: WIRING A MULTI-STORY OFFICE FOR VOICE AND DATA

I. DESIGN CONSIDERATIONS

II. ENVIRONMENT

III. INSTALLATION

I. DESIGN CONSIDERATIONS

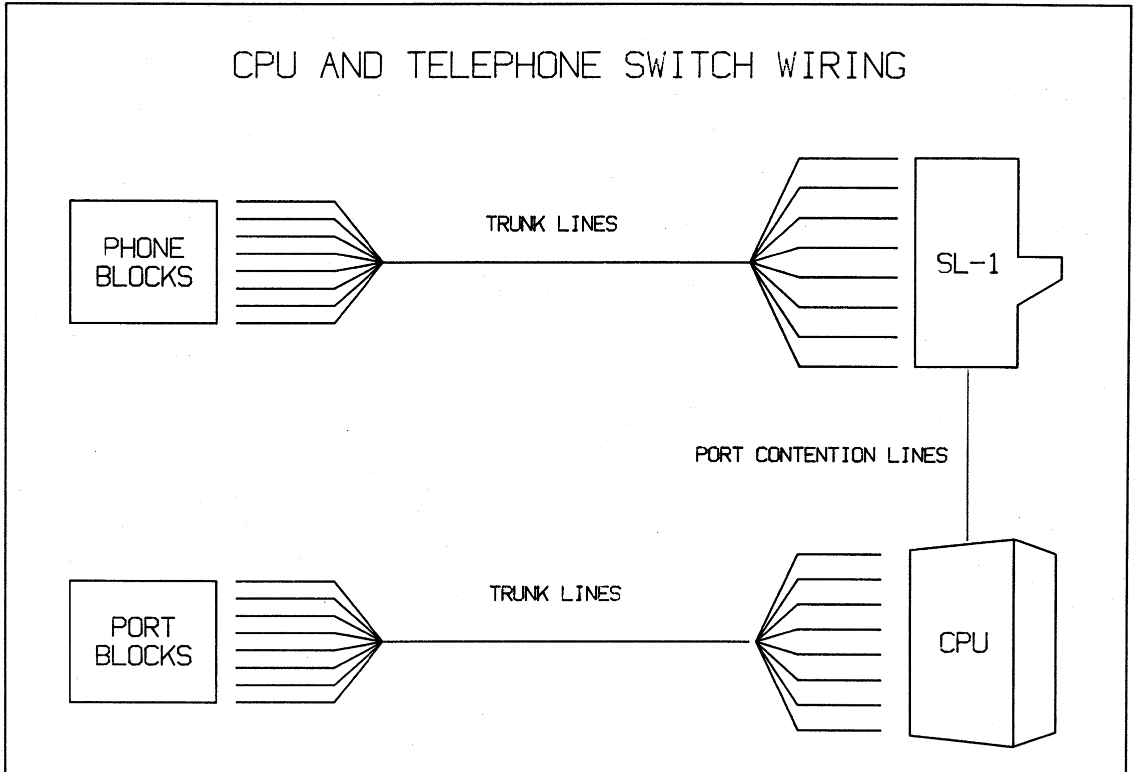
A. FLEXIBILITY
B. VOICE AND DATA
C. DEVICE INDEPENDENT
D. CPU/TELEPHONE SWITCH SELECTION
E. LOW COST
F. LOW MAINTENANCE
G. EASY TROUBLESHOOTING
H. RS232 AND RS422

II. ENVIRONMENT

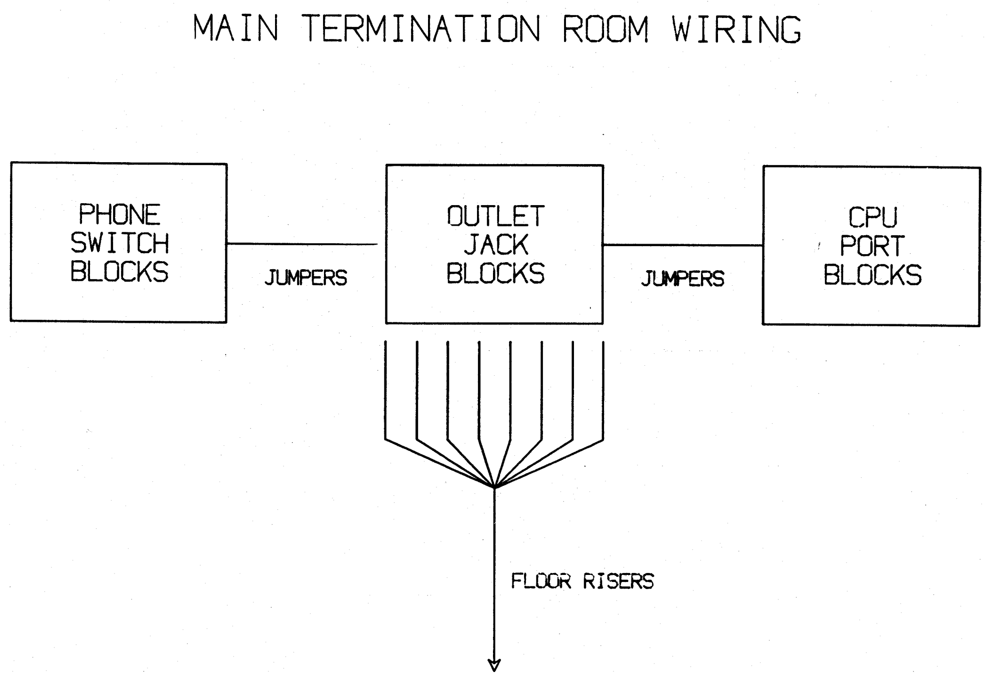
A. BUILDING
B. EQUIPMENT
C. SYSTEM SOFTWARE
D. USER APPLICATIONS

III. INSTALLATION

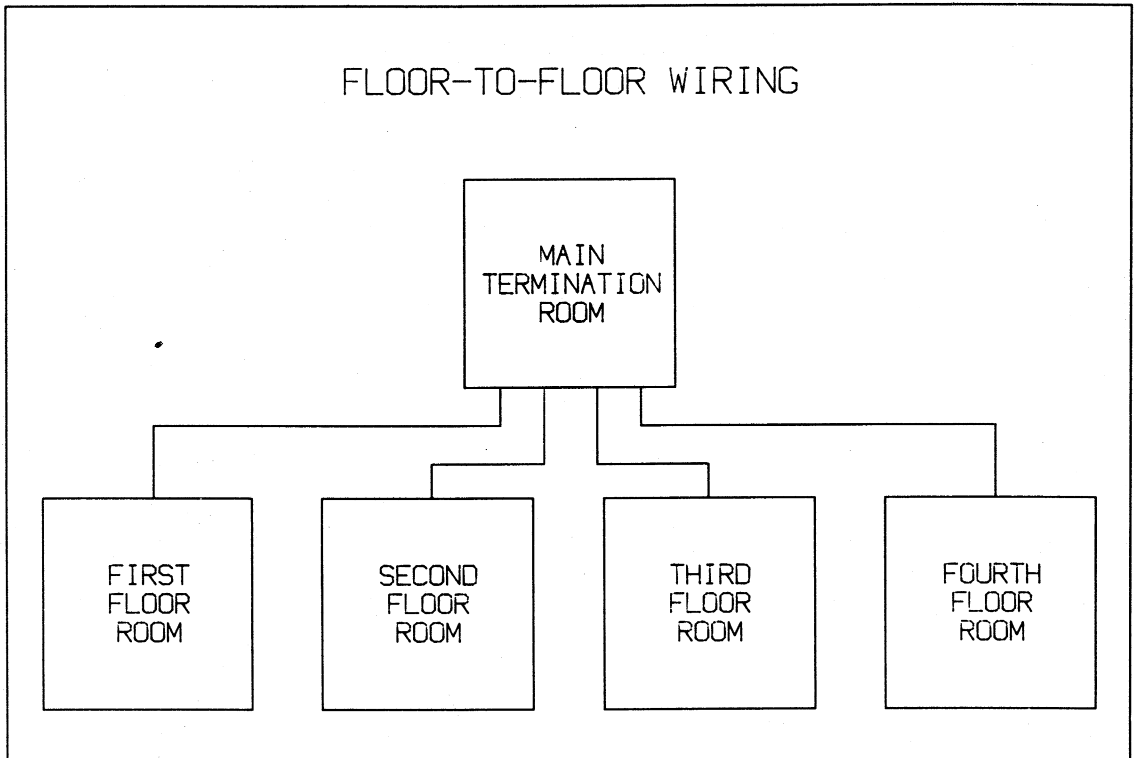
A. CPU
B. TELEPHONE SWITCH
C. WIRING ROOM–MAIN
D. WIRING ROOM–FLOOR
E. WALL OUTLETS
F. DEVICE CABLE
G. DEVICE CONNECTORS
H. DOCUMENTATION



WIRING A MULTI-STORY OFFICE FOR VOICE AND DATA
4508-12

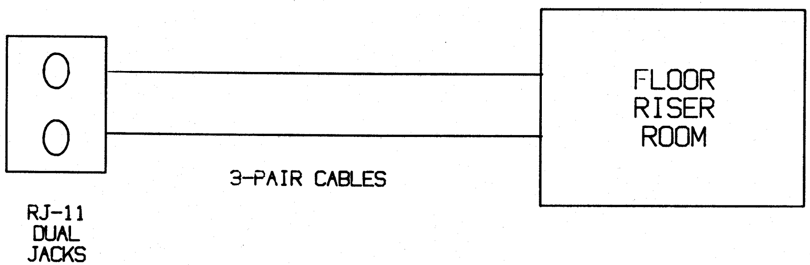


WIRING A MULTI-STORY OFFICE FOR VOICE AND DATA
4508-13



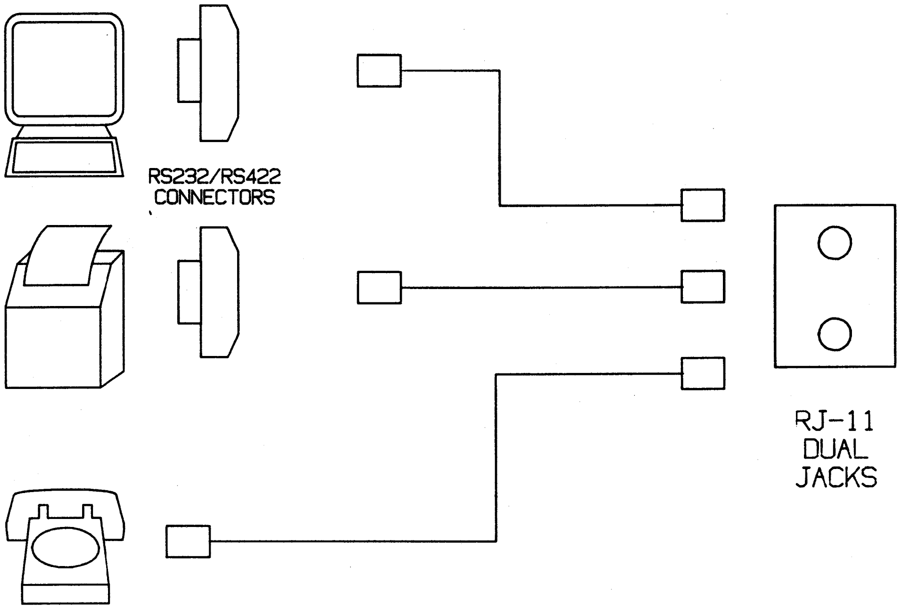
WIRING A MULTI-STORY OFFICE FOR VOICE AND DATA
4508-14

FLOOR WIRING



WIRING A MULTI-STORY OFFICE FOR VOICE AND DATA
4508-15

DEVICE WIRING



WIRING A MULTI-STORY OFFICE FOR VOICE AND DATA
4508-16

By Charles H. Wirl
President of Wilco Communications Inc.
Sunnyvale, Ca.

CABLING NOW AND INTO THE FUTURE

Most companies spend thousands of dollars for a computer system, have it all delivered in boxes, and suddenly during the unpacking begin planning their cabling methodology. Cabling is simple and straightforward, only three pins to connect, but can be an unsurpassed nightmare if not properly set up, documented and managed.

CABLE MANAGEMENT

A cable management system should be set up and constantly updated when adds, moves or changes are done to the system. Documentation is the key to keeping on top of the mounds of cable.

Every cable must be labeled with the same number at both ends. When installing wall plates, also label the cable inside the wall plate. This will assure that the identification number will not be lost, even if the wall plate is replaced.

When wiring the building and especially when adding new runs, always use the same color code for wires. Try to follow in someone's footsteps, even though red is not what you used at another company for pin #2.

PLANNING AND SAFETY

Make sure you are aware of the building codes for running low voltage communications cable. Some municipalities require you to enclose all cable in conduit. Plenum (TEFLON) cable allows you to run cable outside of conduits. Plenum cable is fire-proof and does not give off poisonous vapors when it is heated. It may be more cost effective to run TEFLON cable instead of enclosing all cable in conduit.

Plan the best size wiring closet to handle your needs now and for the next several years. Keep in mind that you may be installing a LAN or multiple circuits in the future.

DISTANCE

How far can you run cable at various baud rates? Many people ask this question every week. The answer is: it depends. What will work at one site will not work at another site. Electrical interference is not the same under every condition. A heavy duty motor, a fluorescent light, aircraft landing and other things may have an impact on how far the maximum distance is.

It is recommended to keep all runs at least two feet away from any fluorescent lights.

With thanks to Peter Hansen of Hughes Aircraft Company for the following chart:

TERMINAL SPEED	**CABLE DISTANCE**
300 baud	8000 feet
1200 "	4000 "
2400 "	2000 "
4800 "	1000 "
9600 "	500 "

Usually the first devices to suffer CPU distance affliction are printers. Characters begin to disappear (especially in long reports) and garbage characters become frequent. One approach is to buy a pair of line drivers to connect to each side of the cable. Several miles is a typical distance devices can now be run. The cost is under five hundred dollars for the equipment and can be used with local multiplexers to drive several devices with one cable.

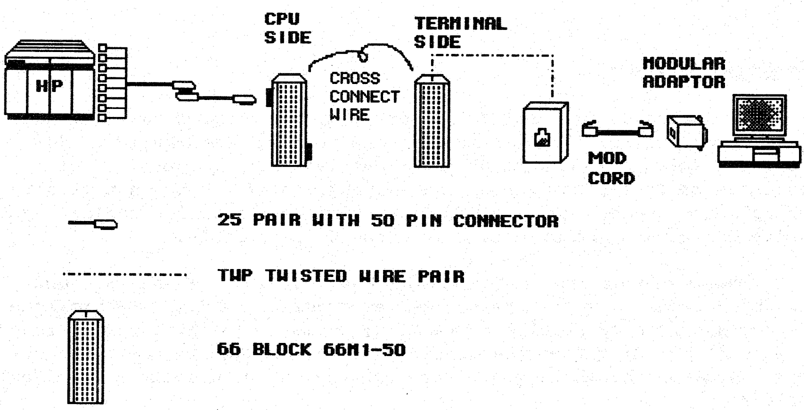
EQUIPMENT

A 66-block (aka RJ-21x, M1-50, punch down block) is a wonderful invention. It allows you to connect (splice) a twenty five pair cable to individual workstations. There are two hundred little silver pegs on each block. Each block handles two cables, one on the left and one on the right. One set of fifty pegs is used to punch down telephone cable #1. The next set receives the terminal cables, thereby splicing them to telephone cable #1. The process is repeated for the second telephone cable. The 66 blocks are also made with a connectorized socket so you plug in the telephone 25 pair cables directly and just punch down the terminal wires.

Another feature of a punch down block, is that it allows you to make a cheap patch bay. You would have two sets of blocks, the left set comprising all of the ports from the computer, and the right set

of blocks connecting to the terminals. Then by cross connecting the left and the right blocks, you can
activate selected terminals. Figure 1 shows the punch down blocks used to cross connect terminals.

FIGURE 1



MODULAR CABLING

The modular telephone adapter is now being used for many different computers. For the Hewlett Packard HP/3000 the RJ11 four wire jack is used. You run your standard two pair cable to an office and then install a modular RJ11 wall plate. A flat modular telephone cord runs from the wall plate to the terminal. Unlike that for the telephone, it is pinned straight through.

An RS-232 to RJ11 converter is attached to the back of the terminal, which converts to modular cabling.

Whenever you vacate an office or cubicle, you would remove the modular cord. This makes the cable less prone to getting run over by the cleaning lady's vacuum or the moving man's dolly.

The idea of modularity also allows you to easily change the existing configuration. At the terminal end dumb terminals and pc computers use different connectors, the modular concept allows the user to change the mod adaptor instead of changing the connector on the end of the cable. At the CPU end the user has the choice of using any port to any terminal by just pluming and replugging a telephone modular cord.

CAMPUS ENVIRONMENT

A campus environment takes more than one complex building and makes an overall cabling system very manageable. Each building is separated into a floor with at least one main wiring closet. All terminals on that floor run to that wiring closet. From that closet, the runs are punched down to 66 blocks (devices to splice cable). Twelve cable runs are channeled into one 25 pair cable.

These cables are called the backbone wiring. These runs and all other floors having terminal cables cross-connected to a backbone run meeting in a main building facility closet. If the computer room is in this building then the cables would be bridged to the computer room. A campus backbone would then run from this building to the next building.

Within each floor it is possible to sub-divide locations into other cabling closets. The standard 2 pair cable is run from the terminal to this closet. From the closet a twenty five pair cable would run to the main floor wiring closet or to the computer room wiring closet.

Each terminal cable should be labeled and identified. Also label the 25 pair cables so you know the floor and location just by reading the label. Telephone lines and data lines are usually found in the same wiring closet. Data communications lines such as leased lines for remote sites are easier to run to the computer room and diagnose problems when they are located near each other.

TESTING CABLES

You may want to test a cable to make sure it hasn't been broken by some sort of accident. You will need an ohmmeter and a piece of wire. Go to one end and short out pins 2 and 3. Then go to the other end and put the ohmmeter on pins 2 and 3. The ohmmeter should read zero, indicating a shorted position. Next check out pin 7 and 20 for continuity. This test will prove that the cable has not been broken. The ohmmeter should read open (or one for digital ohmmeters) when all jumpers are taken off. If not, the cable may be shorted somewhere along the run.

A toning device is used by the telephone company. They usually are yellow and make a siren like sound. Toning sets are used to trace cables that were not properly labeled or missing and lost cables. The wand is then waved over a mound of unlabeled cables and when the right one is found the wand makes the paramedic noise.

A more expensive device is a Time Domain Reflectometer. It sends a waveform of a certain frequency down the cable. By the amount of elapsed time it takes for the wave to bounce back, it calculates how many feet to a problem area. These may be shorts, opens and even splices. When troubleshooting coaxial Local Area Network problems, it comes in handy. It can also measure the distance of each cable run.

MPE XL VOLUME MANAGEMENT:
Private Volumes Made Easy

David Kohl
Hewlett Packard
2000 South Park Place
Atlanta, Georgia 30339

Few users of the HP3000 have ever taken full advantage of the benefits of Volume Management. This is partly due to the fact that very little information has ever been documented for someone who wants to use private volumes. In addition, the facilities for managing private volumes can be very cumbersome and often outweigh any benefits that might be gained.

This will hopefully all change with the introduction of new MPE XL Volume Management techniques. The availability of good documentation along with the straightforward management and maintenance strategies make the Volume Management facility easy to learn. In its simplicity, however, it also provides the flexibility for each user to gain the best benefits for himself.

In this paper, we will discuss the concepts and the components that make up the new Volume Management facility. Through a short case study we will examine some of the benefits that can be gained through the use of non-system volume sets. It is not the focus of this paper to explain detailed procedures for setting up volume sets, but we will take a brief look at the utilities used to do so. My hope is that you will come away from this discussion with the feeling that the potential benefits of Volume Management warrant further investigation on your part, (and with the confidence to take on the challenge).

VOLUMES, VOLUME SETS and VOLUME CLASSES

Let's begin with a brief discussion of MPE XL Volume Management. When we speak of Volume Management, we are simply referring to the management of disc storage. The Volume Management facilities of MPE XL provide the functions of volume initialization, maintenance and inquiry.

There are three basic components in the Volume Management subsystem: volumes, volume sets, and volume classes. Let's start with the smallest physical component, the volume.

It is common to think of each disc drive on the system as a volume but this is not the case. When we speak of a volume on MPE XL, we are referring to the disc media itself. This means that on a system with removable disc packs (HP7935 disc drives, for example), the number of volumes is not limited by the number of physical drives. We will see an example of how this can be beneficial later.

One or more volumes logically grouped together make up a volume set. MPE XL allows for one system volume set and multiple non-system sets, which are referred to as user volume sets. The system volume set is always present on the system and has the name MPEXL_SYSTEM_VOLUME_SET. The system volume set contains two types of storage space, permanent and transient. We will discuss both permanent and transient space later . Any other set that is mounted on the system will be a user volume set. These sets can have any name you wish to give them (for example, MY_USER_SET). These user volume sets can be mounted on the system as needed and contain only permanent space. If no user volume sets are defined on the system, all volumes will be part of the MPEXL_SYSTEM_VOLUME_SET.

Within each volume set, there are also two types of volumes, masters and members. The master volume is the controlling volume of each set and must be present in order to access the set. The master contains the Volume Set Information Table (VSIT), the free space map, the file label table, and the root node of the accounting directory for the set. The VSIT is a table that contains information about all of the volumes and classes in the set. A member volume contains no controlling information other than a volume label indicating to which set it belongs. It does, however, have its own free space map and file label table. Figure-1 is a comparison of the content of a master versus a member volume.

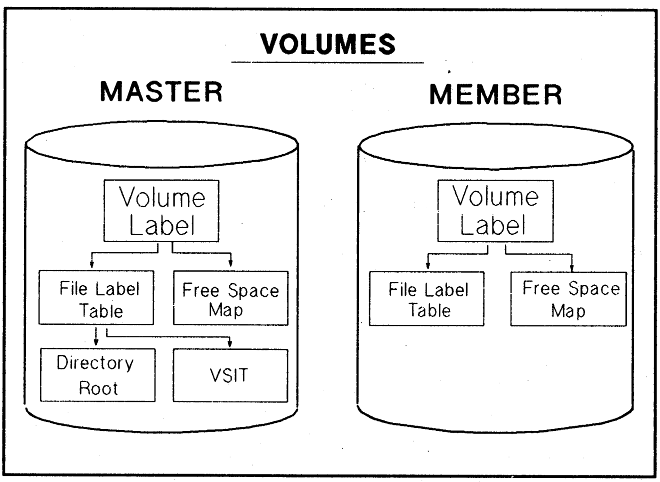


FIGURE-1

You can determine whether a mounted volume is a master or or member, and what set it belongs to, with the :DSTAT command.

Looking at the STATUS column in the following example, you can see that ldevs 1 and 3 are masters while 2 and 4 are members.

```
:DSTAT ALL
  LDEV-TYPE     STATUS     VOLUME (VOLUME SET -GEN)
  1-079370      MASTER     MEMBER1 (MPEXL_SYSTEM_VOLUME_SET)
  2-079370      MEMBER     MEMBER2 (MPEXL_SYSTEM_VOLUME_SET)
  3-079350      MASTER     MEMBER1 (USER_SET)
  4-079350      MEMBER     MEMBER2 (USER_SET)
```

Recall from earlier discussion that when we speak of a volume we are speaking of the disc pack itself. Therefore, all of the information we have just described is on the actual volume (disc) itself. This provides the ability for a volume to be recognized by the system when physically mounted on any drive. This is referred to as Automatic Volume Recognition (AVR).

We have just discussed two states that a mounted volume can be in: MASTER or MEMBER. When a volume is in one of these states, the data on the volume is accessible by the system. There are three other states that a volume can be in as well. These are LONER, SCRATCH, and UNKNOWN. A volume is in the LONER state when it associated master is not mounted or the set has been closed with the :VSCLOSE command. The SCRATCH state means that a volume is ready for initialization. UNKNOWN indicates that the volume does not have a valid MPE XL label on it. This could be the result of being brought over from another system or simply that it is a new pack that has never had a label written. Table-1 is a list of all five valid states for a volume.

VOLUME STATES

MASTER	Volume is mounted and accessible Contains volume set definition and data
MEMBER	Volume is mounted and accessible Contains data only
LONER	Volume is a member of a set that is closed or its master is not mounted
SCRATCH	Volume has been scratched with the SCRATCHVOL command in VOLUTIL
UNKNOWN	Uninitialized volume Does not have a valid MPE XL label

TABLE-1

The final component of Volume Management is the volume class. The volume class is similar to MPE V device classes in that it can be used to restrict a file to a certain device(s). There is another use of volume classes which assists one in best utilizing limited resources. It is possible to define more volumes in a set than there are physical drives to mount them in. In this case, the volumes can be grouped into classes according to the information that they contain. The volumes within a specific class can then be mounted as the data is needed.

VOLUME SET CREATION and MAINTENANCE

The utility used to define and initialize volume sets on the MPE XL system is called VOLUTIL. It is invoked by typing the command :VOLUTIL. Rather than take the time to go into detail on each command of VOLUTIL, Table-2 has been provided to list some of the basic setup commands. The functional definitions for each command are taken directly from the Volume Management Reference Manual (part number 32650-90045).

VOLUTIL Commands

ALTERVOL	Changes the amount of disc space that is allocated for permanent and transient storage
NEWVOL	Creates a new volume
SCRATCHVOL	Places a volume in the SCRATCH state
SHOWVOL	Displays information about a volume in a volume set
NEWSET	Creates a new volume set by defining and initializing the master volume
SHOWSET	Displays information about a volume set and its members and classes
NEWCLASS	Creates a new volume class
USE	Processes VOLUTIL commands in an ASCII file

TABLE-2

The NEWSET command is used to create the master volume for a volume set. The NEWVOL command create the individual members of the set.

As we mentioned earlier, there are two types of storage space on a system volume, permanent and transient. When you look at the commands in VOLUTIL, you will notice that several of them have parameters to set up or change the allocations for each type of space. Permanent space is disc space used for permanent structures such as files (permanent and temporary), the file label table, the free space map, and the directory. Transient space is for temporary structures such as stacks, heaps, and operating system data structures. Transient space is similar to the concept of virtual memory on an MPE V system.

Each volume in the system volume set is given a percentage for permanent and transient space when it is defined with the NEWVOL command in VOLUTIL. This value relates to the maximum amount of disc space of that type that can be allocated on the volume. For example, if you specify a permanent space value of 75% on an HP7937 disc drive (which has a total size of 2,232,192 sectors) only 1,674,144 sectors, or 75%, of the drive, will be available for permanent structures. The same rules hold true for transient space.

It is possible, and most likely the case, that the total percentage for permanent and trasient space will be greater than 100%. For example, the volume can be set up for 100% permanent and 100% transient. In this case the entire drive is available for use by either type of storage. Obviously, though, if 60% of the volume has been allocated as permanent space, only the remaining 40% will be available for use as transient space.

There is a danger in setting the allocations to 100% for both permanent and transient. If all of the available disc space is used up as permanent, you will be unable to allocate the transient structures needed to run programs. If all of the space is used by transient structures, you will be unable to build files. Caution should be taken to avoid this situation. There are special restrictions on ldev 1 which reserve a certain amount of transient space for rebooting. If ldev 1 is an HP7935 disc drive, the maximum permanent value allowed is 75%, if it is an HP7937, the maximum is 83%.

Caution must also be taken not to underconfigure the permanent space on the system. Simple mathematics tells us that if you have four disc drives and set each one up for only 75% permanent, you have removed the equivalent of an entire drive from your pool of available space. As a general rule, the system does not use a tremendous amount of transient space on any one drive other than ldev 1. With this in mind, you may not find it necessary to reserve a large amount on every volume. Also remember that user volume sets do not have transient space so all user volumes should be set to 100/100.

VOLUTIL has options on the show commands that will list the allocations on each volume. If you want to get a report on the usage, however, you should use the utility DISCFREE.

Appendix A of this paper has an example of the output from DISCFREE along with a detailed description of what each field represents.

ACCOUNTING STRUCTURES FOR USER VOLUME SETS

Whenever you logon to an MPE XL system, you will be logged into the system volume set unless special steps have been taken to direct you to a user volume set. This means that all of the files you build and access will reside on this set. In order for the files to reside on a user volume set, the system accounting structure must be duplicated on the set. To accomplish this duplication, there are two options available on the NEWACCT and NEWGROUP commands.

The first option available is the ONVS= parameter. This parameter tells the system on which volume set to create the directory entry. If this parameter is left off, the default is MPEXL_SYSTEM_VOLUME_SET. The second option is the HOMEVS=, and is valid only for groups. This parameter is specified when creating or altering a group on the system volume set to tell the file system which volume set to place the group's files on. The group must be created on a user volume set via the ONVS= parameter before it can successfully be "homed" there.

Let's look at a quick example. We want to create an account with the name ACCT1 on the user volume set called USER_SET, which has already been defined. We also want to create a group DATA whose files will reside on the user set. The commands to do this would be:

```
:NEWACCT ACCT1,MGR                          << 1 >>
:NEWACCT ACCT1,MGR;ONVS=USER_SET            << 2 >>
:NEWGROUP DATA.ACCT1;ONVS=USER_SET          << 3 >>
:NEWGROUP DATA.ACCT1;HOMEVS=USER_SET        << 4 >>
```

Statement number 1 simply creates the accounting structure on the system volume set and statement 2 creates it on the user volume set. Statement 3 creates the DATA group on the user volume set. Statement 4 finally creates the DATA group on the system volume set while at the same time setting its file pointer to the user volume set.

In this example, the PUB group is automatically created on both volume sets and remains "homed" to MPEXL_SYSTEM_VOLUME_SET since no HOMEVS= was specified for it. If a file is created in PUB.ACCT1, it will reside on the system volume set. If a file is created in DATA.ACCT1, it will be on the user volume set. Figure-2 shows the layout of the accounting structure on the two volume sets.

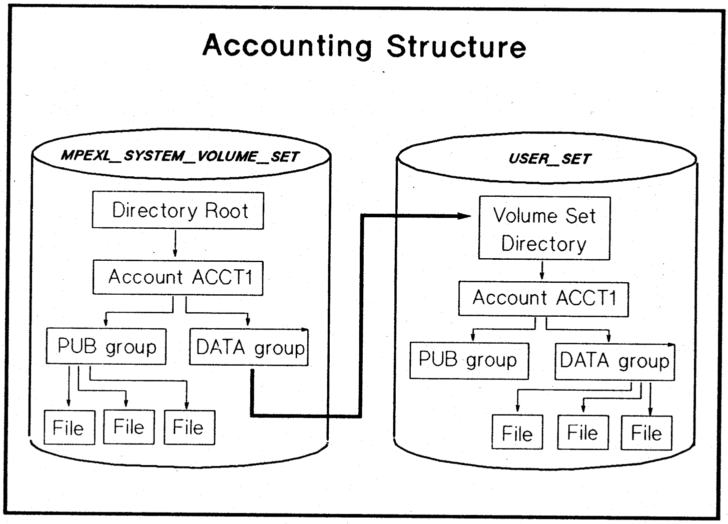


FIGURE-2

BENEFITS OF VOLUME MANAGEMENT

Now that we have discussed some of the "practices and pitfalls" behind Volume Management, let's begin looking at the benefits that can be gained by implementing good management strategies. To help us do this, we will be looking at the ABC Manufacturing Company.

The main application running on ABC's system is their manufacturing package. The manufacturing application handles all of ABC's day to day production and must be on-line twenty-four hours a day. If this application goes down, the whole system is basically down.

Three other applications running on the system are not as time critical. The internal accounting application, which handles inventory, receipts, and shipments, is run during the day but can be suspended for a short time if necessary. Payroll is also run during the day but only twice a month. The third application is customer billing. These billings are run nightly while the manufacturing application is the only other thing running on the system.

ABC also has a development group working on the system. This group is responsible for a large part of the system activity but it is not critical that they stay live at all times.

Effective Use of Limited Resources

The first benefit of a good volume management strategy is that it allows a user to make effective use of limited resources. ABC, for example, has only two HP7937 disc drives and eight HP7935 drives. They have decided to use the two HP7937s as the only members of the system volume set. This leaves them with eight HP7935s to be divided between the remaining applications. Analysis of their needs has determined that manufacturing and development applications will each need three of these drives. This leaves only two drives for the remaining three applications. Fortunately, none of these applications need to be running at the same time. Because of this fact, they can all share the same two drives and ABC will simply mount the proper packs for each application as needed. Figure-3 shows how the sets might be laid out.

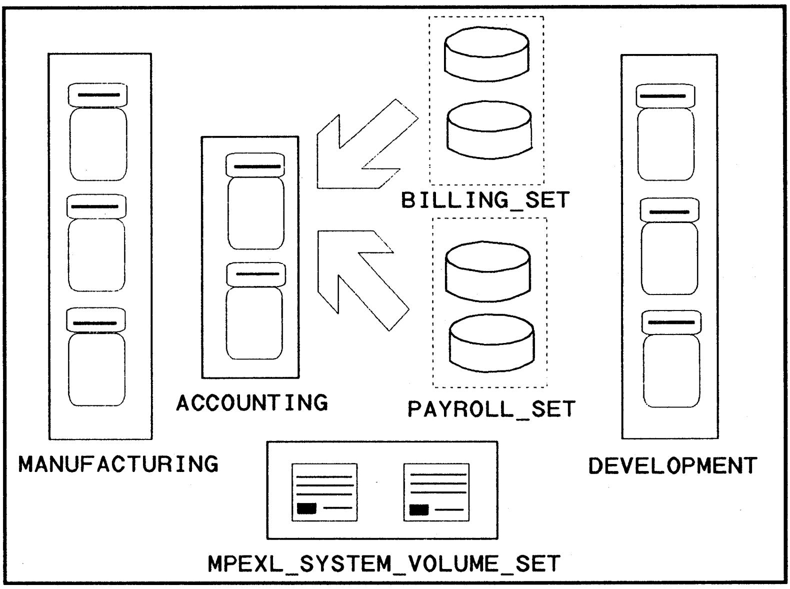


FIGURE-3

You can see that without the use of user volume sets, ABC would not have been able to load all of their applications unless they had purchased four more disc drives.

We can take this situation a bit further with the use of volume classes. The accounting and billing applications share some common data. If they are to reside on separate volume sets that are never mounted at the same time, the data will have to be duplicated on each set. This is an obvious waste of disc space. ABC has chosen instead to combine the two applications into one set. They will use volume classes to restrict the common data to the master volume. The program files and other data unique to each individual application will be divided into separate classes and separate volumes, so that only one class needs to be mounted at a time. Figure-4 shows the new layout with volume classes. You will notice that the payroll application remains a separate volume set.

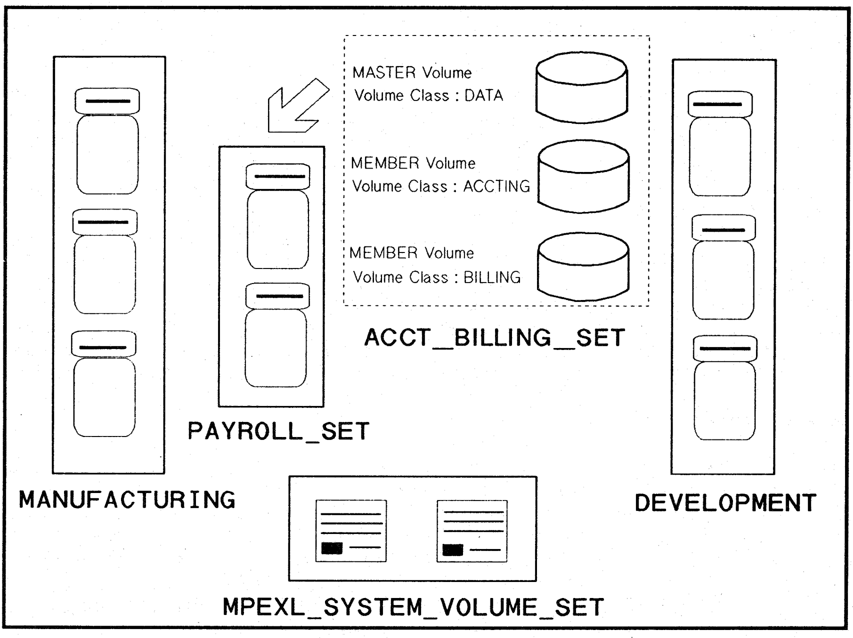


FIGURE-4

Reduced Probability of Failure

One of the greatest benefits of user volume sets is the reduced probability of system failures due to disc problems. When we speak of disc drive reliability, we measure the Mean Time Between Failure (MTBF). MTBF is a statistic that predicts how often failure is likely to occur. Assume for our discussion that the MTBF for each of ABC's disc drives is 50,000 hours and the probability of failure during this time is 5%. Since we have eight drives on the system, the probability of having a failure on any one of these drives is 40%.

The benefit of volume management is that a disc failure on a non-system volume set will not bring the entire system down.

The failure will only affect users accessing that set. Since ABC has only put two of their drives in the system volume set, they have reduced their probability of complete system failure from 40% to 10%.

Reduced Down Time

Through the use of user volumes sets, we have greatly reduced the probability of failure on the entire system. What do we do, though, if we have a failure on one of the drives in a user volume set. Assume that ABC has had a failure on one of the drives in their manufacturing volume set. We see in our layout (Figure-4), that both the manufacturing and development sets physically require three drives to be mounted. We have also identified that in the event of an emergency, the development applications can be shut down. So, after the failure in the manufacturing set, we can simply remove the three volumes of that set and remount them in the drives that currently hold the development set.

ABC has successfully brought their production back up without having to shut the system down. No other users were interrupted. Once the failing drive is repaired, the development set can be remounted in the production's old location to test the drive. This is all possible because the information about the sets themselves is on the volumes.

Faster Recovery

Perhaps the biggest gain you will notice by converting some of your system volumes to user volumes, is a reduction in the time to recover the system from hard failure. It is unfortunate, but likely that at some point you will be forced to do an INSTALL on your system. During an INSTALL, all volumes in the system volume set are initialized, wiping out any data stored on them. After the system is back up, all of these volumes must be redefined with VOLUTIL and all files restored. This can be a very long and tedious process.

In their implementation of user volume sets, ABC has greatly reduced the time to recover from such a situation. If all ten of their disc drives had been set up in the system volume set, they would all have to be redefined after an INSTALL. In addition, every file on the system would have to be restored. Since they have made their two HP7937s the only members of the system volume set, these drives are the only ones that will be initialized during the INSTALL. Once the system is up they will only need to define MEMBER2 (ldev 1 is always defined during the INSTALL) and restore only the files that resided on the system volume set. All of the structure and data on the user volume sets remains untouched and is current to the time of the failure.

Enhanced Security

Obviously, the last several points are the most significant gains of implementing a volume management strategy. Another benefit of partitioning your applications in such a way as ABC has, is enhanced security. Sensitive data, such as the payroll data, can be taken off-line when not in use to prevent unauthorized access. In addition, a user must possess UV (Use Volumes) capability in order to access any data on a user volume set. With this in mind, you can limit certain users to the system volume set to keep them out of the applications.

Assures Needed Resources

By partitioning their applications, ABC has also assured themselves of disc space being available for critical processing. Most of the wasted space on the system probably comes from the development accounts. By restricting them to their own set, other applications will not have to compete with them for disc space. In addition, space in the system volume set will remain available for such things as dump analysis.

Comparable Access Time

MPE XL is designed to make heavy use of volume sets. For this reason, user volume sets have been constructed to have access time comparable to the system volume set. There should be no performance loss in implementing them for your applications.

Learning the procedures for setting up user volume sets may seem a bit confusing to you at first. Once familiar with them, you will agree that they are easier understand than MPE V private volumes. Since this paper did not focus on the setup procedure, Appendix B of this paper has been provided as an example for the steps used to create all of ABC's volume sets. Hopefully, the benefits you are now aware of will make the effort to set up volume sets seem more worthwhile.

APPENDIX A

DISCFREE
OUTPUT DESCRIPTION

```
LDEV : 1
          DEVICE SIZE: 2232192
          TRANS SPACE:   55200             PERM SPACE:  106832
      MAX TRANS SPACE: 1674144         MAX PERM SPACE: 1674144

           FREE SPACE: 2070160
 AVAIL TO TRANS SPACE: 1618944   AVAIL TO PERM SPACE: 1567312
```

DEVICE SIZE - The total size in sectors of the disc drive. The drive in this example is an HP7937. An HP7935 would show 1579904.

TRANS SPACE - The amount of transient disc space currently being used on this device.

PERM SPACE - The amount of permanent disc space currently being used on this device.

MAX TRANS SPACE - The maximum amount of transient space configured for this device. Divide this value by the device size to get the percentage set up for transient space in VOLUTIL (1674144/2232192 = 75%).

MAX PERM SPACE - The maximum amount of permanent space configured for this device. Divide this value by the device size to get the percentage set up for permanent space in VOLUTIL.

FREE SPACE - Total amount of free space remaining on this device. Computed by subtracting TRANS SPACE and PERM SPACE from DEVICE SIZE.

AVAIL TO TRANS SPACE - Total amount of transient space still available for use on this device. Computed by subtracting TRANS SPACE from MAX TRANS SPACE. The result of this computation cannot exceed the value of FREE SPACE. If it does, the amount of available transient space will be equal to the amount of free space.

AVAIL TO PERM SPACE - Total amount of permanent space still available for use on this device. Computed by subtracting PERM SPACE from MAX PERM SPACE. The result of this computation cannot exceed the value of FREE SPACE. If it does, the amount of available permanent space will be equal to the amount of free space.

APPENDIX B

VOLUTIL Example

The following is an example of the command to set up part of ABC's system. The only volume currently defined is the master volume of the system volume set. The example shows the steps to complete setup of the system volume set and to set up ABC's manufacturing set. The other user volume sets are not included in this example.

```
MPEXL: VOLUTIL
volutil: :DSTAT ALL
  LDEV-TYPE     STATUS      VOLUME (VOLUME SET - GEN)
  1-079370      MASTER      MEMBER1 (MPEXL_SYSTEM_VOLUME_SET-0)
  2-079370      SCRATCH
  3-079350      SCRATCH
  4-079350      UNKNOWN
  5-079350      UNKNOWN

volutil: NEWVOL VNAME=MPEXL_SYSTEM_VOLUME_SET:MEMBER2 LDEV=2
*Verify: Initialize new member volume on ldev 2 [Y/N] ? YES
NEW and TEMP file deallocated for MEMBER2 (LDEV 2)
*Note: New member volume has been initialized on ldev 2.

volutil: NEWSET SNAME=MANUFACTURING MASTER=MEMBER1 LDEV=3
*Verify: Initialize new volume set MANUFACTURING:MEMBER1 on
         ldev 4 [Y/N] ? YES
beginning recovery
setup complete - beginning recovery of free space map and
                 label table
completed recovery of free space map and label table
completed recovery of files
begin posting of recovered files
recovery completed
NEW and TEMP files deallocated for MANUFACTURING:MEMBER1
*Note: New master volume has been initialized on ldev 3.

volutil: NEWVOL VNAME=MANUFACTURING:MEMBER2 LDEV=4
*Verify: Initialize new member volume MANUFACTURING:MEMBER2
         on ldev 4 [Y/N] ? YES
NEW and TEMP files deallocated for MEMBER2 (LDEV 4)
*Note: New member volume has been initialized on ldev 4.


volutil: NEWVOL VNAME=MANUFACTURING:MEMBER3 LDEV=5
*Verify: Initialize new member volume MANUFACTURING:MEMBER3
         on ldev 5 [Y/N] ? YES
NEW and TEMP files deallocated for MEMBER3 (LDEV 5)
*Note: New member volume has been initialized on ldev 5.
```

```
volutil: SHOWSET MANUFACTURING INFO=STRUCT

Volumes in set: MANUFACTURING

   MEMBER1
   MEMBER2
   MEMBER3

Classes in set: MANUFACTURING

   DISC

Volumes in class:   MANUFACTURING:DISC

   MEMBER1
   MEMBER2
   MEMBER3

volutil: :DSTAT ALL
  LDEV-TYPE     STATUS     VOLUME (VOLUME SET - GEN)
  1-079370      MASTER     MEMBER1 (MPEXL_SYSTEM_VOLUME_SET-0)
  2-079370      MEMBER     MEMBER2 (MPEXL_SYSTEM_VOLUME_SET-0)
  3-079350      MASTER     MEMBER1 (MANUFACTURING - 0)
  4-079350      MEMBER     MEMBER2 (MANUFACTURING - 0)
  5-079350      MEMBER     MEMBER3 (MANUFACTURING - 0)

volutil: EXIT

MPEXL:
```

Understanding DBRECOV

Author: *Andy Hering*
Hewlett-Packard
Data and Languages Division
Cupertino, California U.S.A.

INTEREX Conference;September 11-15, 1989;San Francisco, CA

Abstract

DBRECOV is the recovery mechanism for TurboIMAGE databases. It is a complex utility with a variety of available recovery options. As more customers implement user logging, the importance of understanding DBRECOV has increased. Unfortunately, many users are unfamiliar with DBRECOV and how it operates.

This paper will discuss how to use DBRECOV and will also explain the different options available that can affect the outcome of recovery. The different recovery algorithms will be discussed as well as some of the internal structures of DBRECOV. Performance of both DBRECOV/V and DBRECOV/XL will also be discussed.

The paper is designed for those users who wish to know more about DBRECOV and how to effectively use this utility to meet the needs of their organization. Readers should be familiar with TurboIMAGE and user logging.

Introduction

DBRECOV consists of two methods of database recovery: Roll-Back recovery and Roll-Forward recovery.

Roll-Back Recovery provides a means of rapid database recovery following a "soft" system crash (i.e. a system failure or a working loss of memory). Roll-Back uses the current database and log files. When invoked, Roll-Back will undo any incomplete database transactions. Intrinsic Level Recovery (ILR) on MPE/V must be enabled in order to ensure database integrity. Transaction Manager (XM) on MPE/XL automatically ensures database integrity.

Roll-Forward recovery can be used to provide database recovery in the case of a hard system failure (i.e. disk head crash). Roll-Forward recovery uses a stored back-up copy of a database and the current log file. The back-up copy of the database is updated with the completed transactions that were written to the log file. All incomplete transactions are suppressed.

DBRECOV also has many options that can affect the recovery operation. Several of these options will be discussed.

Using DBRECOV

Before any recovery can occur, some preparatory steps need to be taken. Figure 1 shows how to initiate database logging, and how to enable databases for the specific type of recovery desired. Multiple databases can share the same log identifier, which means transactions from all of the databases will be logged to the same log file. One thing to note is that enabling a database for Roll-Back will also automatically enable logging and ILR (on MPE/V). However, disabling a database for Roll-Back will not also disable logging and ILR (on MPE /V). This must be done manually. ILR will be discussed in more detail in the next section.

There are many options that can control the outcome of recovery. These options can be specified by using the CONTROL command. A list of all CONTROL commands can be seen in Figure 2. Since this list is fairly extensive, only the NOABORTS/ABORTS, NOUNEND/UNEND, STATS/NOSTATS commands will be discussed here.

The NOABORTS/ABORTS option controls DBRECOV's treatment of aborted transactions. An aborted transaction is any transaction that does not complete normally. An aborted transaction will have an ABEND record instead of a DBEND record in the log file. The CONTROL ABORTS (default) option will cause DBRECOV to treat an aborted transaction as though it had completed normally. If the NOABORTS option is used, DBRECOV will roll-out the aborted transaction. Any subsequent transaction that is dependent upon the aborted transaction will also be rolled out.

For example, if transaction 1 does a DBPUT to record number 1 and then aborts, it will be rolled out. If transaction 2 does a DBUPDATE to record number 1, it will fail since record number 1 no longer exists in the database. Transaction number 2 is considered a dependent transaction and will be rolled out.

The CONTROL NOUNEND/UNEND option affects the way DBRECOV handles transactions which do not complete due to a system failure. CONTROL NOUNEND (default) will cause DBRECOV to suppress transactions that do not complete due to a system failure. Transactions that are dependent upon these suppressed transactions will be rolled out. The UNEND option will cause DBRECOV to treat incomplete transactions as though they had completed.

The STATS option will print tabulated information from the log files similar to the tables printed after an actual database recovery; however, no databases are opened or recovered. This option is useful if the user wants to know what types of transactions occurred on the databases in the log file without having to perform any recovery. The STATS option is simple to use:

```
:FILE LOGFILE=ORDER001
:RUN DBRECOV.PUB.SYS
 > CONTROL STATS
 > RUN
```

The NOSTATS (default) option simply negates the STATS option.

In addition to specifying CONTROL options, the user may want to use the FILE command to route log records to individual user files. The FILE command will open a user file for a particular user. These files provide information about the outcome of recovery and can provide a useful tool for auditing purposes.

Figure 1: Preparatory Steps For DBRECOV

1. Acquire logging capability:
   :NEWACCT *acctname, mgrname;* CAP= *capability list* (include LG)

      OR

   :NEWACCT *acctname;* CAP= *capability list* (include LG)

   To aquire logging capability for a user:
   :NEWUSER *username;* CAP= *capability list* (include LG)

      OR

   :ALTUSER *username;* CAP= *capability list* (include LG)

2. Acquire a log identifier:
   :GETLOG *logid;* LOG= *logfile*,{DISC/TAPE/SDISC/CTAPE}
      [;PASS=*PASSWORD*] [; {AUTO/NOAUTO}]

   AUTO performs an automatic CHANGELOG when the disk log file becomes full. NOAUTO is the default.

3. Build a log file if logging to DISK:
   :BUILD *logfile;*CODE=LOG;DISC=[*numrecs*] [, [*numextents*]
      [,*initialalloc*]]]

4. Set the log identifier and flags:
   :RUN DBUTIL.PUB.SYS
   >>SET *data base name* LOGID=*logid*
     PASSWORD?*******

   If doing a roll-forward:
   >>ENABLE *data base name* FOR LOGGING
   >>ENABLE *data base name* FOR RECOVERY

Figure 1: Preparatory Steps For DBRECOV (Cont.)

If doing roll-back:
>>ENABLE *data base name* FOR ROLLBACK

Note: Enabling a database for ROLLBACK will also automatically enable the database for LOGGING and ILR (for MPE/V). However, disabling a database for ROLLBACK will not automatically disable a database for LOGGING and ILR (for MPE/V). This must be done manually.

5. Make a backup tape of the data base. (This is required for roll-forward and highly recommended for roll-back):
   :RUN DBSTORE.PUB.SYS

6. Start the logging process:
   :LOG *logid*, START

Figure 2: DBRECOV CONTROL OPTIONS

MODE 4	- RECOVER IN DBOPEN MODE 4;ALLOWS MODE 6 ACCESSORS.
MODEX	- RECOVER IN EXCLUSIVE MODE.
NOCHECKSUM/CHECKSUM	- DISALLOW/ALLOW RECORD CHECKSUM.
NOMATCH/MATCH	- DISALLOW/ALLOW CHECK FOR DATABASE/LOG FILE MATCH.
NOSTORE/STORE	- DISALLOW/ALLOW CHECK FOR THE DBSTORE FLAG.
NOSTAMP/STAMP	- DISALLOW/ALLOW CHECK FOR DATABASE AND LOG FILE TIME STAMPS.
NOLOGID/LOGID	- DISALLOW/ALLOW CHECK FOR DATABASE AND LOG FILE LOG ID's.
NOABORTS/ABORTS	- SUPPRESS/APPLY ABORTED TRANSACTIONS.
STATS/NOSTATS	- TURN ON/OFF LOG FILE REPORTING.
NOTSEQ	- DO NOT CHECK TO SEE IF LOG RECORD TIME STAMPS ARE IN SEQUENCE.
NORSEQ	- DO NOT CHECK TO SEE IF THE LOG RECORD NUMBER IS IN SEQUENCE WITH THE PREVIOUS LOG RECORD
NOUNEND/UNEND	- SUPPRESS/RECOVER TRANSACTIONS WHICH DO NOT COMPLETE DUE TO A SYSTEM FAILURE
EOF = NNN	- SPECIFY A LOG RECORD NUMBER IN ORDER TO CREATE AN ARTIFICIAL EOF
STOPTIME = mm/dd/yy hh:mm	- SPECIFY A TIME STAMP IN ORDER TO CREATE AN ARTIFICAL EOF
ERRORS = N	- SPECIFY THE MAXIMUM ERROR LIMIT BEFORE TERMINATION (DEFAULT = 3000)
WARNS = N	- SPECIFY THE MAXIMUM WARNING LIMIT BEFORE TERMINATION (DEFAULT = 3000)

The format of the FILE command is as follows:

> FILE *fileref,userref[,rmode,fmode]*

The *fileref* parameter is an MPE file reference in the form of *file name [/lockword][.group[.account]]* which specifies the destination file for each user's log records. The *userref* parameter, in the form of *username [/ident].account*, specifies which user's log records are to be copied to the destination file. The *rmode* parameter tells DBRECOV to return log records associated with transactions that are successfully recovered. *Rmode* can be used only for Roll-Forward recovery and can take on one of four values. *Fmode* directs DBRECOV to return log records associated with transactions which failed to be recovered. *Fmode* can be used with both Roll-Forward and Roll-Back recovery and can also take on one of four values. Please consult the *TurboIMAGE Reference Manual* for further details.

XM vs ILR

In order for Roll-Back recovery to work, a database must be in a physically consistent state. TurboIMAGE uses Transaction Manager (XM) on MPE/XL, and Intrinsic Level Recovery (ILR) on MPE/V to accomplish this task.

XM is an MPE/XL operating system service which provides TurboIMAGE/XL and other internal subsystems with a transaction based level of data and file integrity. From the TurboIMAGE/XL perspective, a transaction may be seen as a unique set of writes necessary to perform a single database intrinsic. For example, a DBPUT to a detail data set with one or more associated master data sets will result in multiple file system writes. A broken chain will result if the DBPUT is interrupted before all necessary file modifications are made. XM provides the same type of service that DBBEGIN and DBEND provides to database logging users. It is possible, however, to lose more than one database intrinsic if the XM journal in memory has not been posted to the XM log file on disk at the time of a system failure. XM will automatically perform recovery from the XM log file at boot up time. XM has enough information on disk to guarantee physical database integrity. XM will protect modifications to the root file as well as to data sets.

ILR is a feature of TurboIMAGE/V. ILR guarantees that at most, only one database intrinsic will be lost. ILR, however, provides a performance penalty; all modifications involving pointer manipulations must be logged to the ILR log file. The ILR log file has the name *<dbname>00* and is created when a database is enabled for ILR. TurboIMAGE/V will ensure that the "before" and "after" images of the modified data are written to the ILR log file before the data sets themselves are modified. ILR results in more file system writes being performed which decreases database performance.

ILR is also available on TurboIMAGE/XL, but the concept is entirely different. ILR can be used to ensure that only one intrinsic can be left incomplete. Once again, ILR implies a performance penalty. ILR will cause TurboIMAGE/XL to wait until the XM journal in memory is flushed out to the XM log file on disk before any modifications are made to the actual data sets.

DBRECOV is a two-step process. The first step is known as analysis time and the second step is known as recovery time.

Analysis time is when DBRECOV builds a recovery block in the Staging Area. A recovery block is a recoverable portion of a log file. The block contains at least 100 records with no open transactions (that is, every DBBEGIN will be matched with its corresponding DBEND). The only exception to this rule is in the last recovery block where there may be some open transactions due to a system failure. The period between recovery blocks is known as the quiet period.

On MPE/V, the Staging Area is a temporary MPE file to which DBRECOV writes recovery blocks. On MPE/XL, the Staging Area is a PASCAL HEAP structure in memory. Figure 3 shows the relationship between the log file and the Staging Area.

Recovery time is when DBRECOV performs the actual recovery of the recovery block. For Roll-Forward recovery, DBRECOV recovers every recovery block in the Staging Area. For Roll-Back recovery, only the last recovery block is recovered. This will be discussed in more detail later.

DBRECOV uses several internal tables and temporary files to accomplish recovery. Figure 4 illustrates the internal tables used by DBRECOV on MPE/XL, while Figure 5 illustrates the internal tables and temporary files used by DBRECOV on MPE/V.

The Database Table stores information about each database designated for recovery. An entry is created in the table when the database is opened. The table contains information about the number of transactions, DBPUTS, DBDELETES, and DBUPDATES that have occurred. In addition, the Database Table has pointers to the Process and Record Tables.

The Process Table stores information about each user process that accesses a database being recovered. The Process Table contains information such as the number of DBPUTS, DBDELETES, and DBUPDATES a process does, and also the number of the last incomplete transaction (if one exists).

On MPE/XL, the Process Table is built to accommodate a large number of users. However, due to architectural restraints on MPE/V, the Process Table cannot be built as large as it is on MPE/XL, and is divided into three parts. A temporary MPE file known as the Process File is used to store information for all processes accessing a database. The Process Table still exists, but only contains the 12 most active processes. Another table called the User Table is used to index the Process File entry and the Process Table entry (if one exists) for each process. Entries in the Process File are read into the Process Table when needed. Entries in the Process Table are linked together according to when the entry was allocated, with the latest one at the head of the chain. When the Process Table is full and more entries are needed, the entry at the tail of the chain will be moved back to the Process File, and the entry needed will be copied from the Process File into the Process Table.

Figure 3: Log File/Staging Area Relationship

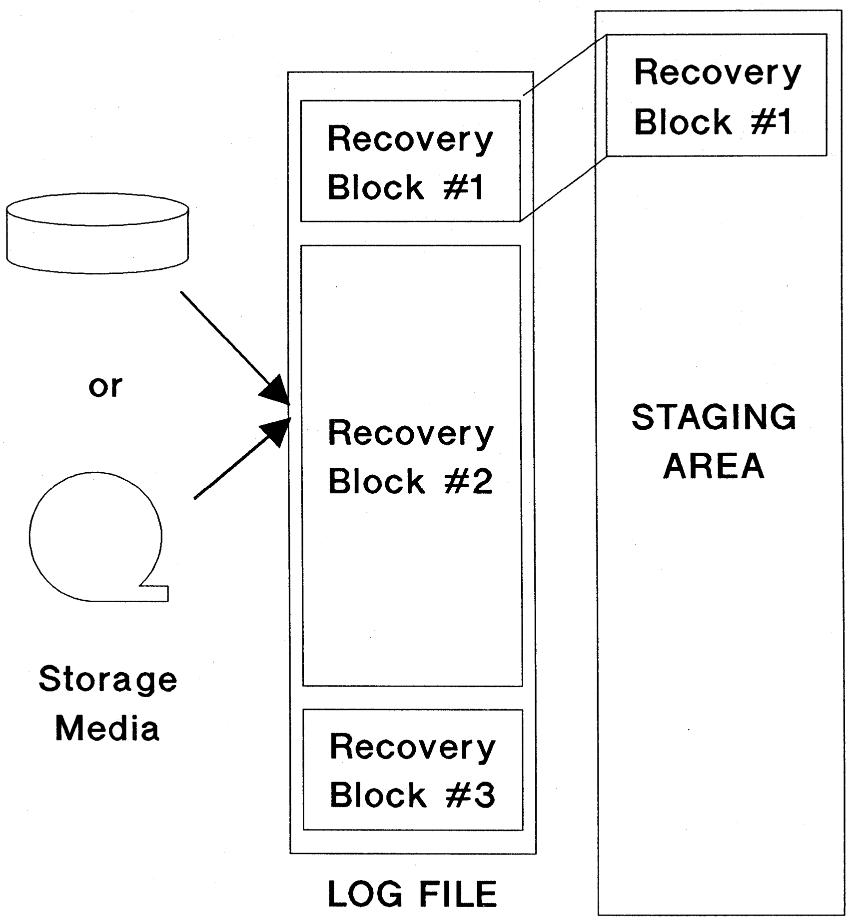


Figure 4: DBRECOV/XL Internal Tables

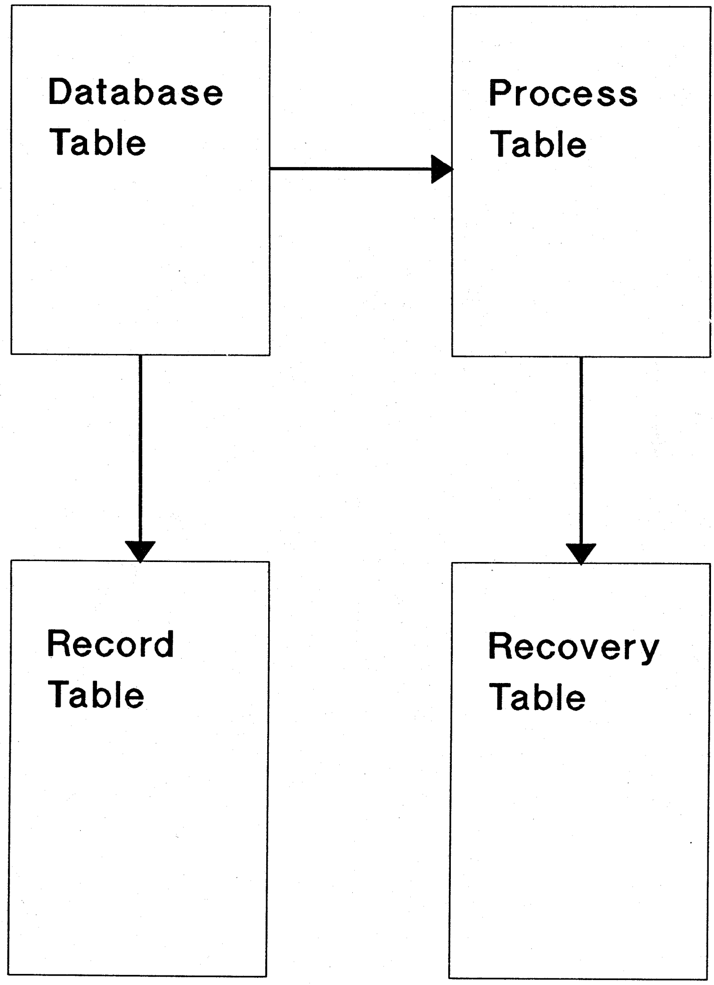
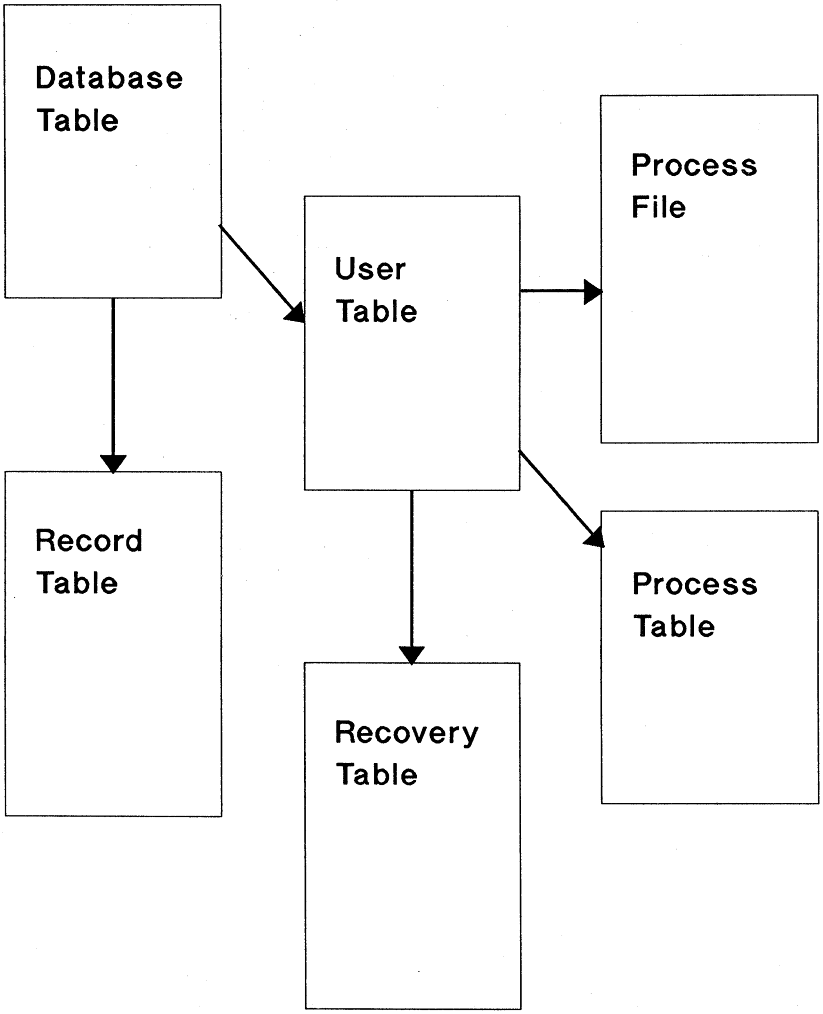


Figure 5: DBRECOV/V Internal Tables and Files



The Record Table stores the "before" and "after" record numbers of detail data set entries added during the recovery process in order to make subsequent references possible. During recovery, record numbers may change due to suppressed transactions. For example, transaction number 1 does a DBPUT to record number 1 of data set A and then aborts. Transaction number 2 does a DBPUT to record number 2 of data set A and then ends its transaction. If recovery is done later with the NOABORTS option, transaction number 1 will not be applied to the data set. However, transaction number 2 will be applied to the data set, but DBPUT will place the data in the next available record which is now record number 1 and not record number 2. Any subsequent DBUPDATE or DBDELETE that references record number 2 will now need to reference record number 1. The Record Table will map the old record number (from the log file) to the new record number (obtained from DBPUT) in order to make the correct retrieval.

The Recovery Table stores entries pertaining to the user recovery file declarations. Each entry contains the log number of the process, the recovery file name, and the type of information to be returned.

Roll-Forward Recovery

Roll-Forward recovery is the method of recovery needed when a system experiences problems due to faulty hardware or a severely corrupted database in which Roll-Back recovery is impossible.

The first step in Roll-Forward recovery is to purge all databases to be recovered and restore the back-up copies of the databases from tape. After the databases have been restored, DBRECOV can be run. The user can specify all desired options with the CONTROL and FILE commands. All databases that need to be recovered can be specified with the RECOVER command. DBRECOV will check the syntax, attempt to open the database(s), and create a Database Table entry for each database. Once the RUN command is accepted, the recovery process begins.

Roll-Forward will then build a recovery block. As mentioned before, a recovery block is a subset of the log file and contains 100 or more records with no open transactions. The only exception is in the very last recovery block, which may have open transactions because of a system failure. Once the recovery block is built, a pointer is positioned to the beginning of the Staging Area.

Roll-Forward recovery will attempt to apply as many transactions as it can to the database. All incomplete transactions will be suppressed. If a transaction cannot be applied because it is an incomplete or dependent transaction, Roll-Forward will invoke Roll-Back to roll-out the transaction and mark it dead. Once Roll-Back has completed, Roll-Forward can then resume again.

Once recovery has completed, a pointer is positioned to the beginning of Staging Area if recovery files have been declared via the FILE command. The Staging Area is processed again and all the appropriate log records are written to their corresponding files.

Roll-Forward will continue the above process for all recovery blocks. The process, database, logging, and recovery statistics are then printed out. All databases are now logically consistent.

Roll-Back Recovery

Roll-Back recovery will roll-out any incomplete transaction following a "soft" system crash. Roll-Back can be used in most situations to perform rapid recovery.

As with Roll-Forward recovery, the user can specify all desired recovery options with the CONTROL and FILE commands. Multiple databases can be recovered by specifying them with the ROLLBACK command. Roll-Back will check for correct syntax, attempt to open the database(s), and create a Database Table entry for each database.

After the RUN command is used to start recovery, Roll-Back will start to build recovery blocks. Roll-Back will discard every recovery block except the last one. The last recovery block is the one in which Roll-Back will perform its recovery.

Before recovery can begin, Roll-Back will determine which log record in the Staging Area it needs to roll-back to. This is known as the stopping point. It is defined as the DBBEGIN log record of the farthest incomplete transaction in the last recovery block. The stopping point is the point at which Roll-Back recovery stops and Roll-Forward recovery begins. Roll-Back will find the information it needs to determine the stopping point by analyzing the data in the Process Table.

Roll-Back will start at the last record in the Staging Area and roll-out ALL transactions up to the stopping point. Roll-Forward will then be invoked to reapply as many transactions as possible. If a transaction cannot be applied because it is dependent upon a suppressed transaction, Roll-Back will be invoked to roll-out the transaction and mark it dead. Once Roll-Back completes, Roll-Forward can then resume.

If recovery files have been declared with the file command, Roll-Back will position a pointer to the beginning of the Staging Area. A pass is made through the Staging Area, and the appropriate log records are written to their corresponding user files.

After Roll-Back has completed, the process, database, logging, and recovery statistics are printed out. All databases are now both logically and physically consistent.

Stop-Restart Recovery

Stop-Restart recovery is a feature that can be used by both Roll-Forward and Roll-Back recovery. Stop-Restart recovery can be used to stop a long recovery process and then restart it at a later time. It can only be used with multiple log files.

For example, if a user is recovering a database that has 50 log files, the user can stop recovery, say at log file number 27, by either renaming or removing the log file. When DBRECOV tries to open log file number 27 and can't find it, the program will prompt the user to either stop recovery or resume recovery. If the user responds RESUME, DBRECOV will ask that the log file be restored so that recovery can resume. If the user responds STOP, DBRECOV will stop the recovery process.

DBRECOV stops the recovery process by saving the recovery environment in a file called the Restart File. The Restart File is a privileged file that has the same name as the logid set in the database root file. The Restart File is created when the RUN command is issued. If a file already exists with the same name, DBRECOV will abort, indicating that there is a duplicate permanent file. The Restart File can be purged by running DBRECOV with the PURGE option (i.e. RUN DBRECOV.PUB.SYS,PURGE).

When recovery is stopped, DBRECOV saves all internal tables, files, and variables in the Restart File. The Staging Area, however, is not saved. Once everything is saved in the Restart File, DBRECOV will terminate.

Recovery can be restarted at a later time by running DBRECOV with the RESTART option (i.e. RUN DBRECOV.PUB.SYS,RESTART). DBRECOV will rebuild all the internal tables and files from the information stored in the Restart File. All internal variables will be set back to the values at the time recovery was stopped. The Staging Area will be rebuilt from the log files. After the environment has been restored, recovery can start from where it left off.

Stop-Restart recovery can be aborted if the user does not wish to restart recovery by running DBRECOV with the ABORT option (i.e. RUN DBRECOV.PUB.SYS, ABORT). This will also purge the Restart File. Note that using the ABORT option of DBRECOV may leave a database in a logically inconsistent state.

Performance

The performance of DBRECOV/V depends greatly on the number of users accessing the databases being recovered. Since the Process Table can only contain 12 entries, performance will start to decrease if more than 12 concurrent users access any combination of databases in a recovery block at one time. This is due to the fact that DBRECOV/V must swap entries back and forth between the Process Table and the Process File. Performance is significantly better if no swapping occurs and all process entries remain in the Process Table. In some instances, it is a better idea to recover only one or two databases at a time in order to avoid swapping Process Table and Process File entries.

DBRECOV/XL provides substantial performance improvements over DBRECOV/V. DBRECOV/XL's main performance gain is that it has a very large Process Table in memory and does not use the User Table or the Process File. This increases performance by not having to swap Process Table and Process File entries. In addition, DBRECOV/XL keeps the Staging Area in memory, which avoids the disk I/O present for the Staging Area in DBRECOV/V. It is in the best interest of the user to recover all databases from the same log file at one time. This eliminates the overhead of building the internal structures of DBRECOV/XL for every execution of the program.

The performance of DBRECOV on both MPE/V and MPE/XL also depends upon the complexity of the application programs which logged database intrinsics to the log file, and the structure of the databases being recovered.

In general, as the number of transactions dependent upon suppressed transactions increases, DBRECOV performance decreases. A transaction can be suppressed if it did not complete due to a system failure, or it aborted and the CONTROL NOABORTS option was specified. If there are other transactions dependent upon these suppressed transactions, then Roll-Back routines will be needed to roll-out these transactions, which will cause DBRECOV performance to decrease.

The structure of the database plays an important in DBRECOV performance. As the number of paths increase from a data set, the number of file system reads and writes increases which of course decreases performance.

Summary

DBRECOV is a very powerful and useful recovery tool. The many options available in DBRECOV can be used to tailor the recovery needs of a specific customer.

This paper has covered the internal design of DBRECOV, with a special emphasis placed on the internal tables and files used in the recovery process. The differences between DBRECOV/V and DBRECOV/XL were pointed out as well as some performance considerations. For those users who wish to know more about the externals of DBRECOV, the *TurboIMAGE Reference Manual* can provide detailed explanations.

SPL: Vision Of The Future Or Of The Past?

by

Stan Sieler

Allegro Consultants, Inc.

2055 Woodside Road, Suite 170
Redwood City, CA 94061
Voice: (415) 369-2303
Fax: (415) 369-2304

Abstract

Is SPL dead? If not, why not? How can SPL be used in a manner that allows easy porting from the Classic HP3000 to the HPPA HP3000? This paper addresses these questions, proposes some answers, and explores coding techniques that aid migrating from SPL to SPLash! and C.

Introduction

SPL is not dead. It's living in semi-retirement and has been spotted recently moving around Cupertino, California. Seriously, what is the status of SPL? This paper explores that question, as well as presenting techniques that can aid in either extending the life of SPL or in aiding the migration from SPL to C or SPLash!.

SPL was born with the first HP3000 in 1972, but we can trace its ancestry back to ALGOL (1960). SPL grew slowly, with the base features of the language being extended, until about 1978. A spurt of "late growth" came around 1982, but since then the language has been stable. The version currently available is called SPL/V. Given the unstated obsolescence of MPE V, future enhancements to SPL/V by Hewlett Packard seem quite unlikely.

For readers unfamiliar with SPL, let's briefly describe the language. SPL (Systems Programming Language) is a block structured language similar to ALGOL-60. SPL lacks some of the features of ALGOL, notably the ability to nest the declaration of procedures and a feature known as "blocks". SPL has some enhancements beyond ALGOL, including a primitive macro mechanism (DEFINEs), a source-file inclusion directive ($INCLUDE), direct access to the hardware stack (TOS), and the ability to intermix assembly language statements with high-level SPL statements (ASSEMBLE). Like ALGOL (and C), SPL has no built-in I/O capability. Like ALGOL (and unlike C or Pascal), SPL does not have the ability to declare "record" structures. (For a very detailed comparison of SPL to other languages, Eugene Volokh's multi-language comparison paper is highly recommended.)

When the RISC-based HP3000 line was first announced, the only "classic" HP3000 languages that were supported were Pascal, COBOL (1974), COBOL (1985), and FORTRAN 77. HP produced new versions of the compilers for these languages, versions that emit "Native Mode" (NM) RISC instructions. Since the introduction of MPE XL (the operating system for the RISC-based HP3000s), HP has added Native Mode compilers for C and RPG. Notably absent was SPL. A consortium of SPL enthusiasts spotted a market opportunity and, under the aegis of Software Research Northwest, developed and market SPLash!, a Native Mode version of SPL.

Since a Native Mode version of SPL exists, the vast majority of SPL/V programs can be migrated to Native Mode. This provides one answer to our question: Yes, SPL has a future, and it's called SPLash!

However, SPL (and SPLash!) are confined to HP computers running the MPE operating system (MPE V or MPE XL). With today's increasingly mixed computing environment, the the issue of portability should be considered when considering the death of SPL.

Since it is unlikely that other computer manufacturers will embrace MPE as their operating system, an alternative answer to our original question might be: No, SPL has no future because it is tied to one operating system.

So, we have two good answers: Yes and No. Because of this dichotomy, the rest of this paper is split into two sections. The first deals with using SPL in such a way as to prolong its usefulness. The second deals with using SPL with an eye towards migrating to C.

Staying Alive

Since this section is concerned with prolonging the lifetime of SPL programs, some of the reasons for choosing to write in SPL/V on MPE V should be mentioned. After all, if SPL/V was not a good choice for MPE V programming, then there would be no need to consider prolonging its life.

When performance is an important consideration, SPL/V is the language of choice on MPE V. The language and the CISC instruction set of the classic HP3000 were designed together, and no other language is as close a match for the instruction set. Many SPL statements are implemented as single instructions. (The language coming the closest is, perhaps, FORTRAN/V, an implementation of FORTRAN 66 which is not available in Native Mode on MPE XL.) The ability to drop into assembler provides absolute control over the speed of the of the final code.

Because SPL has no built in I/O facility, programmers must use the operating system intrinsics to access the file system. This results in "leaner" (and, unfortunately, "meaner") code. When compared with programs doing I/O in any other MPE V language (using the constructs provided by the language), SPL always wins. The reason is pretty simple: all other languages boil down to an interface to the same MPE intrinsics (e.g.: FREAD, FWRITE). Typically, these interfaces have general purpose code associated with them, which adds a layer of overhead not present in SPL programs. (True, it can often dramatically cut down on program development time, but that issue is not under consideration at the moment.)

SPL/V programs that are I/O intensive on MPE V will still be I/O intensive on MPE XL. Although it can usually be argued that the CPU efficiency of I/O intensive code matters very little (i.e.: it rarely matters if a program takes 2 milliseconds or 4 to process a record if it took the operating system 30 milliseconds to fetch the record), on MPE XL an extra layer of inefficiency is added when a program accesses the file system from Compatibility Mode (CM). Hence, even SPL programs can benefit from running in Native Mode. This can be accomplished by using the SPLash! compiler to compile an SPL program into NM code.

On MPE XL, an alternative method of accessing disc files exists, callable only from Native Mode. This feature is called "mapped files" (or, more properly, "user mapped files"). Mapped files have been explored in my paper "Mapped Files: What, How, & Why" [#1] and in the book "Beyond RISC!" [#2].

Prolonging the life of SPL/V programs means planning to use SPLash! to compile into Native Mode on MPE XL. This implies two important areas of concern: how to plan for imcompatibilities and how to plan to use new features (e.g.: mapped files).

Sometimes, Things Change

The RISC architecture deals with 32-bit and 64-bit addresses. 32-bit addresses are actually a short-form of 64-bit addresses, and are converted by the hardware to 64-bit form as they are being used. The Classic architecture deals with two kinds of 16-bit addresses: byte oriented and 16-bit oriented. The Classic HP3000 literature refers to these as byte-addresses and word-addresses, but the use of "word" is confusing with the advent of the 32-bit RISC word. Therefore, this paper will refer to Classic words as either "half words" or "16-bit words".

SPLash! can produce code for procedures in either of two different modes. "Splash-mode" procedures expect their parameters on the SPLash!-simulated stack (a 16-bit wide stack limited to 65536 bytes), and treat most addresses as 16-bits big. "Native-mode" procedures expect their parameters in a mix of registers and the native mode stack, and treat most addresses as 32-bits big. The programmer can choose which type of procedure he or she wants, with the limitation that splash-mode procedures can be called only from a SPLash! outer block OR other splash-mode procedures. Native-mode procedures can be called by splash-mode procedures, a SPLash! outer block, or from other native-mode procedures (produced by any language). If SPL code was being used as an SL (or RL) called by other languages, then the only choice is a native-mode procedure.

This produces two types of problems for SPL/V programs. Some programs have code which stores 16-bit addresses (byte OR half-word) into 16-bit variables:

```
integer i;              ! SPL integers are 16-bits
byte pointer p';
...
i := @p';               ! save addres of p' in i.
```

If this code is in a procedure marked as "option native", SPLash! will generate a syntax error on the attempt to store a 32-bit value (**@p'**) into a 16-bit variable (**i**).

The simplest solution to this problem is to change the type of the variable **i** to "double":

```
double i;
```

However, the program will then no longer compile under SPL/V. This is not acceptable to many programmers who want a single source to compile with both SPL/V and SPLash!. (Note: a similar problem faces Pascal/V and Pascal/XL programers. Only SPLash! has a totally edit-free solution.)

Two better solutions are available, both using compile-time logic. SPL allows source code to be optionally included (or omitted) based on the value of a "flag". SPL allows the use of 10 flags, named X0, X1, ..., X9. These flags can have either the value "ON" or "OFF". If X9 is chosen to have the value of "ON" for SPLash! compilations, and "OFF" for SPL/V compilations, then the above problem can be solved as either:

```
$if x9 = off                          ! SPL/V
   integer i;
$if x9 = on                           ! SPLash!
   double i;
$if
   byte pointer p';
   ...
   i := @p';
```

or:

```
$if x9 = off
   define addr = integer #;
$if x9 = on
   define addr = double #;
$if

addr i;
byte pointer p';
...
i := @i;
```

The second approach is recommended, as it involves using the least number of compile-time switches, and so produces the easiest to read SPL code.

A compiler flag (e.g.: X9) can be set to ON with the line:

```
$set x9 = on
```

SPLash! relieves the programmer from the need to constantly edit such a line (to "OFF" for SPL/V, and to "ON" for SPLash!) by extending the syntax of the $IF directive AND relying on a "feature" of SPL/V. The combination allows the following compiler directives to set X9 = ON automatically for SPLash! compiles, and to OFF for SPL/V compiles, with NO editing required!

```
$set x9 = off                   ! OFF = SPL/V, ON = SPLash!
$if x9 = on or xsplash          ! SPL/V ignores the "OR XSPLASH"
$   set x9 = on                 ! Yes, is SPLash!
$if
```

Some intrinsics return code addresses known as "plabel"s. On MPE V, a plabel was 16 bits. On MPE XL, a Native Mode plabel is 32 bits. Because of this change, the NM trap arming intrinsics (e.g.: XCONTRAP, XARITRAP), expect 32 bit parameters instead of 16 bits. This can be solved:

```
$if x9 = off
   define plabel = integer #;            ! 16 bit plabel
$if x9 = on
   define plabel = double #;             ! 32 bit plabel
$if
```

```
plabel
   old'plabel;
...
xcontrap (@cy'handler, old'plabel);
```

A few other intrinsics have had their parameters change from MPE V to NM MPE XL. These include: **createprocess, search** and **mycommand.** SPLash! comes with include files that, for splash-mode procedures, provide plug-compatible calling sequences. However, for programmers wishing to call the real **createprocess** intrinsic, a possible solution is:

```
integer
   parm;                                 ! desired new PARM value

$if x9 = off
   integer array
      itemvals  (0 : max'item);
   integer
      status;
   define
      dubl      = #;
$if x9 = on
   double array
      itemvals  (0 : max'item);
   double
      status;
   define
      dubl      = double #;
$if
...
itemvals (0) := dubl (parm);
...
itemvals (1) := dubl (logical ("CS"));          ! priority
...
createprocess (status, ... , itemvals);
```

Another example of an intrinsic with changed parameters is **genmessage**, whose **param1**, ..., **param5** parameters changed to 32-bit in Native Mode. A transparent method of calling this routine is:

```
$if x9 = off
   define virt = #;
$if x9 = on
   define virt = virtual #;
$if
...
intrinsic
   genmessage;
...
      ! pass 2 byte addresses as param1 and param2...
genmessage ( ..., virt (@param1), virt (@param2) );
```

Sometimes, They Stay The Same

Most SPL/V code compiles with no changes in SPLash!, even most **ASSEMBLE** statements. SPLash! simply emulates the Classic HP3000 instruction set when compiling an **ASSEMBLE**. Keep in mind that many uses of **ASSEMBLE** are probably not desirable, and that they give SPLash! little leeway for optimizing your code. Similarly, using TOS can easily lead to programming mistakes in SPL/V as well as in SPLash!.

Sometimes, Old Things Go

MPE XL is missing some procedures that were available on MPE V. Some of these procedures are completely gone (e.g.: the privileged **setbreakpnt**) and some exist only in Compatibility Mode (e.g.: **segmenter** and **sortinitial**).

Programs that used the **segmenter** procedure in SPL/V will have to change to something like:

```
   $if x9 = off
      ...
      segmenter (...);
      ...
   $if x9 = on
      ...
      move segment'command' := "PREP ....";
      command (segment'command', err, parm);
      ...
   $if
```

Programs using **sortinitial** should be changed to use the similar **sortinit**, which does exist in Native Mode.

Sometimes, New Things Come

MPE XL has added a number of new intrinsics, available only from Native Mode, including: **hpfopen** and **hpcicommand**. Some old intrinsics have an expanded range of parameters.

The **dascii** intrinsic (finally) allows a base of -10 (this is available in both CM and NM). A programmer wishing to obtain right-justified **dascii** output in a field of 12 bytes could do the following:

```
      integer len;
   $if x9 = off
      byte array scratch' (0 : 11);
   $if
      intrinsic dascii;
      ...

      move buf' := 12 (" ");      ! blank out the destination
   $if x9 = off
```

```
      len := dascii (val, 10, scratch');
      move buf' (11) := scratch' (len - 1), (- len);
   $if x9 = on
      dascii (val, -10, buf'(11));
   $if
```

Or, check the contributed library (or contact the author) for plug-compatible replacements for **ascii**, **dascii**, **binary**, a **dbinary** that support bases 8, 10, -10, and 16, and can run on MPE V or MPE XL.

For new intrinsics, they can be selected with compile-time logic. The following example shows how to call the command intrinsic and generate the appropriate error/warning message. Note that extra code is required on MPE V, as the **command** intrinsic does not print error messages, where the MPE XL **hpcicommand** intrinsic will (optionally) display error messages.

```
   $if x9 = off
      intrinsic command;
   $if x9 = on
      intrinsic hpcicommand;
   $if

   ...
   $if x9 = off
      command (buf', err, parm);
      if <> then
         genmsgu (2, err);
   $if x9 = on
      hpcicommand buf', err, parm, 0 <<print all errs & warnings>>);
   $if
```

Alas, Poor Yorick

This section deals with the "SPL is dying" side of the argument. If SPL/V is a dying language, what can we do while writing new SPL code for MPE V (or maintaining old SPL/V code) that can ease our eventual move to another language? What language should we aim towards?

The choice of target language will influence our actions, so that must be discussed first. Only two languages available on MPE today are feasible: Pascal and C.

Based on our earlier discussion, SPL can be considered "dying" only if portability to non-MPE based computers is important. In such a discussion, we clearly must consider HP's Pascal to also be "dying".

Why? Because HP's excellent implementation of Pascal/XL is excellenent precisely because it addresses a number of the limitations of standard Pascal. It provides extensions such as type coercion (known as "casting" to C programmers), pointer manipulation, crunched records, and structured constants. Programs written using these extensions, and efficient programs must use them, have portability problems similar to those of SPL/V. Additionally, the Pascal XL extensions are not available in Pascal/V, despite public announcements of their imminent release at the 1986 Detroit Conference, thus converting SPL/V to Pascal+/V is not feasible, even ignoring portability issues. The C compilers available tend to suffer a little less from portability issues.

Pascal lacks one important feature that makes conversion of SPL difficult: macros. The **define** statement of SPL allows extension of the language, and is impossible to simulate in Pascal (without the use of a pre-processor). C provides macros via #define.

Thus, C is our language of choice. Two different C compilers are available on the Classic HP3000 (from Tymlabs and CCSC), and one on the RISC HP3000 (from HP). I presented a paper on "How To Convert From SPL to C Without Making Waves" [#4] at the 1986 Detroit conference, and will not duplicate its contents here. Instead, the focus will be primarily on coding constructs for use in SPL/V programs that can be easily ported to C.

Encapsulate all I/O. When using file system intrinsics (e.g.: **fread** and **fwrite**), have only a single occurance of them, and put that in a procedure:

```
integer
   errno;                    ! global used for I/O errors
...
logical procedure read'file (fid, buffer, bytes);
      value   fid, bytes;
      logical fid, bytes;          ! unsigned, 16-bit
      array buffer;
```

```
        ! Returns number of bytes read.  -1 flags an error
  begin
  integer
     len;

  errno := 0;
  len := fread (fid, buffer, -bytes);
  if <> then
     begin
     fcheck (fid, errno);            ! fetch error code
     read'file := -1;                ! flag an error
     end
  else
     read'file := len;               ! return # bytes read

  end <<read'file proc>>;
```

Encapsulation (as shown above) makes porting a program to another machine much simpler. For example, an (approximate) translation of the above to C would be:

```
unsigned read_file (fid, buffer, bytes)
      unsigned fid, bytes;
      char *buffer;
        /* Returns number of bytes read.  -1 flags an error */
   {
   read_file = bytes;                /* probable result */
   while (bytes--) {                 /* while #bytes > 0 */
      readf (fid, buffer);           /* read one byte */
      if (errno) {
         read'file = -1;             /* flag an error */
         break;                      /* exit loop */
         }
      buffer++;                      /* increment pointer */
      };
   }
```

Note: the above was written by a non-C programmer, and it shows!

Once the I/O related routines are translated, the rest of the code translation should be straight forward.

Be aware that some data types of C have the same name, but a different meaning, as some data types in SPL. For example, **double** and **long** exist in both C and SPL, but C's usage is exactly backwards from SPL's. Problems such as this can be avoided by defining new data types. For example, the following SPL code fragment would be easy to translate into C:

```
define
   real64      = long #,
   int32       = double #,
   int16       = integer #,
   uint16      = logical #;

int32 ktr;
real64 sum, x, y, z;
```

```
...
ktr := 0d;
while (ktr := ktr + 1d) <= 100d do
   begin
   sum := sum + ...;
   end;
```

A conversion to C could be:

```
#define real64  double
#define int32   long
#define int16   short
#define uint16  unsigned

int32 ktr;
real64 sum, x, y, z;
...
ktr = 0;
while (++ktr <= 100) {
   sum += ...
   }
```

Bibliography

1. Language Comparison Paper by Eugene Volokh

2. "Mapped Files: What, How, & Why" by Stan Sieler
I.U.G. 89 Proceedings (Belgium Conference)
SCRUG 89 Proceedings

3. "Beyond RISC!" by Wayne Holt, Steven Cooper, Jason Goertz, Scott Levine, Joanna Mosher, Stanley R. Sieler, and Jacques Van Damme.
Software Research Northwest, Inc.

4. "How To Convert From SPL to C Without Making Waves" by Stan Sieler
Interex Detroit Conference, 1986
Paper 3126

Appendix 1

Sample INCLUDE file for SPL/V <--> SPLash! Declarations

```
$set x9 = off                    ! OFF = SPL/V, ON = SPLash!
$if x9 = on or xsplash           ! SPL/V ignores the "OR XSPLASH"
$   set x9 = on                  ! Yes, is SPLash!
$   symlen = 31                  ! allow up to 31 characters per ID
$if


define

$if x9 = off
   addr          = integer #,         ! 16 bits
   intdoub       = integer #,         ! integer (@... - @...)
   nil           = 0 #,               ! an unassigned pointer
   optnat        = #,
   plabel        = integer #,         ! 16 bits
   virt          = #,
$if x9 = on
   addr          = double #,          ! 32 bits
   intdoub       = double #,
   nil           = 0d #,              ! an unassigned pointer
   optnat        = option native; #,
   plabel        = double #,          ! 32 bits
   virt          = virtual #,
$if
```

LANs BEFORE TIME

by

John Stachowiak
Associate Director
Computer Services
N.W. Tri-County Intermediate Unit
252 Waterford Street
Edinboro, Penna. 16412

Editor's note...LANs BEFORE TIME is intended to be a beginners introduction to PC networking presented as an extension of a previous paper and an actual case history. Italicized terms are considered essential and are defined in the accompanying sequenced glossary.

IN THE BEGINNING.....

Ever since the beginning of time, it seems that man has been continually struggling with the limitations of technology and finding new ways to employ it in every day life. This is particularly true in the field of computer technology, an ever changing environment of mystery even to those such as ourselves who work in the field daily.

Where did it all begin?...I believe it was right in the beginning of time. Though history never mentions it, there was a computer like device hidden away in the cool deep darkness of the first cave. It is this influence that for centuries to come would find computer centers world wide hidden in isolated cool under ground facilities. One also can still observe some very cave man like tendencies within almost every end user community.

You say you don't believe in the theory of evolution, well how about another equally true coincidence. Again an unpublished theory is that the computer was created on the eighth day. This influence leads us to the source of our existence. After all who else works in the dead of night and weekends keeping systems operational when the rest of the normal workforce rests. Why else is there never enough time to complete projects to the end user's expectations. Yes, our environment expects an eight day week and our goals are set to the traditional seven day stretch.

TECHNOLOGY AND DATA COMMUNICATIONS

In all seriousness however, man has been confronted with the feat of making computers communicate with other computers and the end user since the invention of the first computing device. Compounding the problem is the rapid and ever changing nature of technology characterized by terms like *faster, smaller, cheaper, and more powerful*, and the list goes on and on. Our goals have also been ever changing and shifting from simple, *a single IBM Mainframe to another locally based IBM system*, to complex, *end user PC based communications to minis' and Mainframe scattered across the world.*

It is around these needs that the telecommunications industry developed and flourished throughout the years. In recent years we have all experienced the break-up of AT&T and the resulting problems of de-regulation. Our terminology is also dramatically shifting towards a new *lingo* of impressive acronyms all brought to you live by that newcomer in most shops, the PC.

WHERE ARE WE GOING?

I pose this question to data managers every where as the challenge of the 90's. With the popularity of the PC growing daily and becoming more affordable, all shops will soon have to consider the potential of this resource, the problems it will bring, and the obstacles of networking this hardware in an organizational-wide MIS plan.

OUR SOLUTION....

Unknowingly, we began this task some six years ago with the implementation of centralized word processing (HPWORD) based from our newly upgraded SERIES 70. Five stations became twenty-five virtually over night. With this invasion came the use of HP3000 based graphics (HPDRAW & EZCHART), a spreadsheet (VISICALC), user data base (HPLISTKEEPER), and electronic desk services (HPDESK).

Two years back we began seeking a solution to the CPU drain of this electronic office environment on our business computer. We migrated some thirty PC workstations to an in-house *LOCAL AREA NETWORK (LAN)*. We chose an HP solution, *OFFICE SHARE*, because it supported the bulk of our PC workstations at the time, HP150s, and provided a look alike work environment for our user base. They migrated to PC software solutions such as HPWORD, EXECUTIVE CARD MANAGER, and DRAWING GALLERY.

The HP3000 was also an integral part of our strategy and was linked to our PC LAN using *NETWORK SERVICES (NS3000)* and *RESOURCE SHARING*, both HP proprietary software products. Several varieties of terminal emulators were acquired to provide PCs with *BLOCK MODE* terminal capabilities. These included *ADVANCELINK*, *SESSIONS*, and *REFLECTIONS*. The choice here becomes more one of user preference, SESSIONS favored by those partial to the *WINDOWS* environment. As an observation, REFLECTIONS does seem to be the more popular choice among software integrators. *ADVANCEMAIL* was later adopted for users only requiring access to HPDESK mail services.

Sounds simple, right? At first glance yes but it was a totally new look at data communications. Our LAN *TOPOLOGY* was HP's Thin lan offering, a 10 Mbps. *ETHERNET* bus arrangement. At this time choices from HP also included the older 1Mbps. *STARLAN* and the evolving 10 Mbps. *STARLAN* both of which use standard *TWISTED PAIR WIRING* and *MODULAR CONNECTORS*, a very large plus. Again only the bus style topology would support the HP150s and we were locked into a *RG-58U COAXIAL CABLE* medium. Distance limitations and ease of installation were sacrificed because of this decision. Generally speaking, coax arrangements are only used for *BACK-BONE LINKS* and environments plagued with noise and other electrical interference.

The whole process was and continues to be a healthy career education. Not to get lost in detail, we will return to this experience at a later point in our discussion.

FROM AN END USERS POINT OF VIEW

It would be totally inappropriate at this point not to briefly summarize end user reactions to the changes we had implemented. The HP look-a-like software offerings were very well accepted. The use of well thought out configurations and automatic uses offered almost a transparent transition.

PC frustrations at first ran high....especially trying to remember to save frequently during long projects. Another missed and unanticipated problem was MSDOS. Going into the project we had assumed a certain degree basic knowledge about the end users PC workstation. After all, most had been using the same hardware for years. One can quickly take for granted the role which MPE plays for us in the HP3000 world, especially file management from a simple logon.

In total we spent much more time on issues of this nature. For many, it has taken some two years to realize how the LAN resources actually work in conjunction with DOS. In a subsequent installation we utilized a menu system to buffer the user from the sometimes harsh realities of DOS. Those who had the potential to interact directly with DOS were quickly identified by the level of frustration with the menu limitations and turned loose into real world of DOS, PAM in other cases.

STANDARDS COMPLIANCE

The saving grace in LAN technology in my opinion falls with a set of defacto standards known as the *OPEN SYSTEMS INTERFACE (OSI)* model and the *IEEE 802 STANDARDS*. Ethernet of almost any brand meets the 802.3 standard and thus becomes connectable to other standards with the proper interface and *PROTOCOL* conversion. HP uses *TCP/IP* protocol. Other common ones include *MAP, APPLE TALK*, and *TOPS* to mention a few of the more common ones.

Within the 802.3 environment HP routes data packets from station to station along the LAN by means of unique *IP ADDRESSES*. Novell on the other hand uses a unique *IPX ADDRESS* to route packets. Information moving between LANS of varying protocols must be translated/converted or *BRIDGED* to the dissimilar environment. The bridge, most times a PC SERVER of some type is charged with the task of examining each data packet to determine whether to re-address or forward the packet. Bridges are also used to isolate data traffic to a particular workgroup within a LAN.

What should you remember about standards? Know that they exist and stick to LAN solutions which comply with the standards. If not already, most can be bridged to one another in the very near future. Other than an inappropriate cabling scheme, all of your hardware can be used in an organization-wide information strategy.

THE LAN MARKET

Prior to getting too much ahead of ourselves in this discussion we should acknowledge that PC communications are becoming a very large market. However, it seems to focus around offerings from only a few vendors. At present, NOVELL boasts of a 60 some percent share of the market followed closely by 3COM who manufactures the HP 3C501 Thinlan PC card. These vendors market both the hardware which includes PC SERVERS, CONTROLLER CARDS, FILE SERVERS, TERMINAL SERVERS, COMMUNICATIONS SERVERS, and software, or the *LAN OPERATING SYSTEM.*

LAN operating systems vary significantly and offer a wide array of features. Ones which you should be aware of include inner-connectivity, bridging capabilities, number of users, topologies supported, SECURITY, print spooler controls, back-up protection, file access and locking conventions, and many more too numerous to mention. Remember, the LAN operating system is the equivalent of MPE or DOS to a multi-user LAN environment and as such should not be taken lightly.

Another feature most boast of is the provision of basic *ARPA SERVICES*. This is a grouping of some rather basic network services brought to us by our friends in the military and include TELENET, electronic mail exchange, simple file transfer, and virtual terminal capabilities. As a single vendor site, these mean nothing to me but users of multi-vendor equipment, DEC & IBM, might have a use for these services. Remember that *VIRTUAL TERMINAL (VT)* does not ensure full compatibility with a TYPE 10 HP terminal required for your view plus application screens.

The one topology we have yet to discuss is that of IBM's *TOKEN RING*. While there are other token/ring based topologies, IBM's is the most common. In the past it has been characterized as cumbersome and slow. In a token based topology, you can only communicate if you have the token and it is your turn to transmit. Whether or not a particular packet is for your station or not, a token based system will require you to open the packet, examine the address, and forward it. Only recently have new offerings in token based technology approached 16 Mbps. through-put.

As opposed to token based technology, HP uses a CARRIER SENSING MULTIPLE ACCESS (CSMA) system with COLLISION DETECTION (CD), or *CSMA\CD* for short, for regulating transmission. In this arrangement a carrier is generated along the transmission media much like a leased 4-wire circuit. Any one can transmit at will addressing specific destinations along the line. When packet collisions are detected, a time out occurs and the damaged goods are re-transmitted.

Comparing this data to HP's Office Share solution as I have throughout this paper, HP's LAN operating system is written by Microsoft and is commonly known as MSNET. It is not particularly known for its stand alone features. Security is in the hands of the person guarding the boot up disc, print spooling is not practical, and it's proprietary features can be frustrating on occasion. On the other hand it does offer true VT terminal connections and many of its other limitations are compensated by features delivered by HP3000 resident software.

Finally before leaving this discussion on the LAN market, a brief point of clarification. We have talked about standards and topologies and now operating systems. Most operating systems function over many different topologies simultaneously. For example if you acquire OFFICE SHARE and later want to change to NOVELL's NETWARE, your hardware will still function. Your decisions in PC networking involve strategies which address your topology (bus, token, or star), operating system (OFFICE SHARE, NETWARE, etc.) and transmission media (twisted pair, coax, fiber). Know your options and what will work best in your environment.

WHY THE HP SOLUTION WORKS FOR US?

Despite the revealing shortcomings of the HP solutions when compared to the rest of the market, the HP solution does offer several very significant benefits which must be weighed in the decision process. First is the very close relationship workstations have with the HP3000. In addition to virtual terminal, we enjoy the benefits of virtual disc, that is, the ability to address large areas of HP3000 disc through DOS in the PC world.

Secondly, each workstation on the LAN has a close working relationship to MPE controlled peripheral hardware. This includes specialized print resources such as lasers and high speed dot matrix printers. In this environment even the tape drive becomes a sharable resource for PC hard drive back-ups. The key to network resource deployment becomes a series of unique *SHORT NAMES* on each server which represent each user and peripheral resource.

This solution also is extremely transparent to the end user. In fact, the proper deployment of automatic uses can automate and mask the identity of which network resources are being used to perform routine tasks. The traditional MPE spooler process also does prove to be far superior to the MSNET based spooler in some instances, however, spool file naming conventions are generic and derived from the sending short name id. This can result in a spool queue of numerous identically named print files.

Finally, HP offers a single source for both hardware and software support. Knock on wood! We have yet to be confronted with a problem that HP has not been willing to sort out, despite what ever third party software is being run on the PC.

MAKING YOUR CONNECTION

Once you start installing your LAN based solutions you quickly forget how all of your workstations and resources are configured and connected to your system. Most will choose some type of singular cabling strategy, that is one cable for each device. Some shops still use the serial port on the PC for asynchronous connections to a host and a secondary LAN connection for PC networking. In fact many find serial LANS, something we will discuss later, a very cost effective alternative.

A rather new piece of hardware some encounter in the world of LAN technology is the *TERMINAL SERVER*. The terminal server equates to the asynchronous concept of multiplexing groups of peripherals across a common data link. For sites that elect a singular cabling scheme, the terminal sever can be adapted via MAU etc. to the LAN based cable. In pairs these devices packetize and translate data from the connected devices to a transmittable 802.3 format.

Virtual Terminal connections always log on to the HP3000 through a series of devices configured as *VTERM*s. They acquire session numbers just the same as other users but the device-id is a cluster of ports which physically don't exist. They are nodes as addressed by NS 3000 cabled to the *LANIC CARD* in the HP3000 card cage and attached to the LAN cable via *MEDIUM ATTACHMENT UNIT (MAU)*.

As PCs boot to the LAN, they also acquire sessions for every shared resource allocated to the user. These as well as the PC to HP3000 spooler communications are handled by several background jobs on the host computer. The entire sequence of jobs must be present and functional in order for any of the network resources to be accessed.

COMPOUND LAN COMMUNICATIONS

About a year back we acquired a small HP3000 GX as a development machine and demo unit for distributed processing concepts. This was a very basic HP3000 equipped with a LANIC, NS3000, and RESOURCE SHARING. With minimal effort this second processor was attached to a 1Mbps Starlan hub with a modular connector, the Starlan hub to a 10 Mbps. bridge, and the bridge to our Thinlan cable with a MAU. Again with minimum configuration, this second HP3000 was communicating with our other HP3000, PC LAN, and two PC SERVERS.

From our discussions thus far, this exercise emphasizes the flexibility of a single vendor solution, standards compliance, and bridging of varied protocols and speeds.

OTHER LAN BASED OPTIONS

On occasion we have referred to the term SERVER in many different contexts. One limitation of most modern day PCs is their inherent ability to complete only a single task at a time. This is where the multi-tasking abilities of OS-2 come into play making use of far greater memory and processor resources. HP's new *OPEN VIEW* products and *LAN MANAGER* are examples of OS-2 technology that will most certainly add a new chapter to PC networking techniques in the very near future.

Placing OS-2 aside, it is not uncommon to come across clusters of PCs in a PC LAN which stand idle performing only a single specialized server function when called upon. Some of the more common ones include communications (a bank of out-going dial modems), LAN bridging, file server duties, and possibly redundant back-up.

While currently undocumented, the HP PC SERVER is supposedly capable of controlling an internal dial modem cluster with the use of HAYES SMARTCOM II software and a PC FAX BOARD for shared facsimile services. Both of these points are currently being validated in actual practice.

EXTENDED PC LANS

Geographically, our user base is scattered across a three County rural area in Northwestern Pennsylvania. As the trend to strategically network continues to evolve, many of these remote sites will opt for inclusion in some type of *WIDE AREA NETWORK (WAN)* structure. The challenge of this feat becomes distance, cost, and through-put.

Once outside of a fixed location, it becomes extremely difficult and expensive to move true LAN traffic. A data rate of 10 Mbps. can be moved by one of three methods. These include *FIBER OPTIC CABLE*, *MICRO-WAVE TRANSMISSION*,or a complete *T-SPAN*. For most, all of these option are either unobtainable or cost prohibitive which moves us to several other scalable alternatives.

Probably the most common means is that of Bridging LAN segments over a 56Kbps. digital leased circuit. Also affordable is this same arrangement but over parallel links. For most applications, this alternative will provide very satisfactory results.

Another alternative gaining momentum is that of BROADBAND or more commonly known as CABLE TV (CATV). In this method, a device called a *BUFFERED REPEATER* is attached by MAU to the in-house LAN segment and by *RF MODEM* to the CATV link. Again 802.3 standard compliant transmission is directed to the *CATV HEADEND* on a *GUARD CHANNEL* then reversed as necessary for the receiving station. This can be a very cost effective alternative for some but on the other hand you have very little control over carrier regulation.

HP as well as other vendors also support what is known as a *SERIAL LAN CONNECTION*. In the HP world, this connection can be established with the limitations of dialed modems or other direct connect asynchronous cabling to an HP3000 server. Only the resources of the host HP3000 will be accessible in this method. Through-put and speed will be critical factors in endorsing this solution. It is, however, very useful for the laptop user.

The market does also offer these same dialed connections to PC based systems via communications server. Novell is again particularly well known for these solutions. The day is rapidly approaching when the NETWARE environment can be directly bridged to OFFICE SHARE. This will be a very long awaited day for users like us who have a strong MacIntosh user group amidst a DOS based OFFICE SHARE environment.

OTHER EFFECTIVE NETWORKING STRATEGIES

Why network?...What are your objectives?...Both good questions to answer even before deciding your PC networking direction. For us, we needed the benefits of centralized office support in a manageable environment. We wanted to control software distribution, maximize resource deployment and utilization, and most importantly safeguard our information resources. A full blown PC LAN was our only viable approach with all the givens.

Others however simply answer this question as a need to share our laser printer. Cases of this nature can easily be corrected with any one of several *GADGETS* such as the common *PRINT SPOOLER*. The print spooler is a large stand alone buffer of varied size which is asynchronously connected between a single or group of PCs and allows the PC print output to exit the PCs I/O card thus immediately freeing the device for other use. A page of straight printed text consumes approximately 2 bytes of buffer space.

Gadgets also include crude *AB SWITCHES*, *CONTENTION DEVICES*, and specifically for the HP SERIES II LASERJET, a buffered *OCTOPUS INTERFACE* for up to four users. The key here is read and ask questions prior to purchase. Stick with a reputable vendor and most will permit money back guarantees if you are not totally happy with the results.

Another source of inexpensive PC network equipment are any number of quasi asynchronous LANS. These usually cable the serial ports of several PCs together and also provide a software disc to configure and control the coordinated resources. Again......buyer beware!

X.25 PRIVATE PACKET SWITCHED NETWORKS

Although not new to most seasoned data communications experts, *PRIVATE SWITCHED NETWORKS or X.25* are becoming quite popular. We acquired ours nearly six years ago and are extremely impressed with the increasing utility a box of it's age continues to produce. At the point we purchased the *SWITCH* and accompanying *PADs* there were not many vendors in the market. We continue to support the TELLABS 33x and 44x lines because of their track record, diagnostic features and expandable modular design. HP also is becoming very aggressive in this market.

While most will purchase these boxes for widely distributed switched communications, it can be used quite flexibly as an automated local resource switch for a small PC office. An 8 or 16 channel unit can easily support several printers, modems, switched communications links, and of course a variety of different workstations which include terminals. An inexpensive print buffer will also extend the utility of printer resources.

SMALL MINI CLUSTERS

As we began this piece with a blatant acknowledgement of the change technology has brought about in the field of data communications, the entry of the small mini-computer to the market offers some new rather exciting options. In particular the new HP3000 GX & LX models can almost be cost justified replacements for yesterday's 8 channel statistical multiplexer. IBM, as well as DEC, has similar sized processing resources.

A typical scenario might equip a remote GX to function as mini server to a small PC office LAN. With the addition of additional hardware, any locally connected PC could also DS a connection to a larger corporate computer for updated business information. The possibilities are virtually endless.

HOW TO PURCHASE PC NETWORKING HARDWARE

The most common source of information and guidance on this topic is the vendor community at large. For the most part they are a pretty reliable source of factual information. Your HP sales rep would also be another source of information on HP specific issues.

My original goal with this paper was to share a very basic two year experience with PC LANS and OFFICE SHARE with others contemplating the big leap forward into time. There aren't as many scars to hide as one might think. Probably the worst is the hundred and one different times I have listened to an HP SE preach the defacto standard and that famous programmed explanation of the *seven* layer OSI model.

You know what?.....I bet it also is an eight layer model but no one quite realizes it!

Transferring Printfiles From a VAX Cluster to an HP3000 over Ethernet

By

Louis P. Tremante
(formerly) Assistant Director, Computer Services
Union College
Schenectady, NY 12308

Transferring Printfiles From a VAX Cluster to an HP3000 Over Ethernet (abstract)

Union College uses a cluster of six VAX's for academic computing and two HP3000's for administrative computing. Two 45ppm laser printers are attached to an HP3000/48 and this paper describes the method used to transfer student output from the VAX cluster to the laser printers. All systems are connected to an ethernet network. The network protocol used on the VAX cluster is Decnet, which is not IEEE 802.3 compatible, thus making it impossible to talk directly to NS/3000 on the HPs. Special hardware and software was installed on one VAX to permit file transfers to the HP. Decnet task to task communication is used to transfer printfile information to the central VAX where carriage control problems are resolved, and a banner is created and appended to the print file. The files are uniquely named and transferred to a special account on the /48, which is periodically polled. When printfiles are found, spoolfiles are created using the appropriate environment to produce 1-up, 2-up, or 4-up output, as specified by the VAX user. Programs used are written in DCL and MPE command language, FORTRAN, and C.

INTRODUCTION

The academic computing facility at Union College consists of a cluster of five DEC VAX computers. Two HP3000 computers provide support for administrative computing. In addition, one of the HP's acts as a print server for student output from the VAX cluster. Student output from the VAX cluster is at time very high in volume, especially near the end of each term. HP 2680 laser printers were selected to provide this print output due to low cost, good quality laser output, and high speed (45ppm). There was no direct interface which would connect the 2680 to a VAX, so a scheme had to be devised to get the print files from the VAX to an HP print server.

BACKGROUND

Prior to the installation of VMS 4.0 in 1985, no cluster-wide print queues were available. This meant that each of the five processors on the cluster had their own queues and could only direct output to printers directly connected to them. If the HP printer was to be used, two major problems had to be solved. First, some method of simulating a cluster-wide queue would have to be devised. This would permit users on any machine to direct output to a single system printer. A program was written which used the VMS "mailbox" feature, to pass printfile names to a central process running on a VAX 11/785 named TED. This local program runs on each processor in the cluster. The VMS "PRINT" command is trapped and control passed to the local process. The printfile names are extracted from the print command and sent via a mailbox to the central process running on TED. This central process queues these filenames and prepares the files for transmission to the HP. This process generates the appropriate banner sheet, copies the file from its home processor, and initiates transmission of the file to the HP.

The second major problem to be solved was the actual transmission of the print files to the HP3000 to which the laser printer was connected. A program named HASP, originally written to provide remote job entry(RJE) capability between VAX and IBM computers had been modified to operate between a VAX and an HP3000. This program uses an asynchronous 9600 baud line to transmit data from the VAX to the HP. On the HP, a system program (MRJE/3000) establishes a link to the HASP program on the VAX and accepts the printfiles as they are sent across. Once on the HP, MRJE then delivers the print files to the HP spooling facility which queues them for printing on the laser printer.

NEW PROBLEMS

While this system has been functioning quite well for several years, there are some good reasons why we had to change it. We have received several updates to the HASP software but because of the way in which the special process on the VAX interfaces with the HASP, we have been unable to apply the updates. Our fear was that the update would not function in the same way with our "in house" process. When it became necessary to update to VMS 5.0 it was also necessary to install a new version of HASP so we no longer had the option of running with the older version.

The 9600 baud line did not present a problem with small files, however the transmission time for large print files was unacceptable. Cluster-wide queueing is now available, so the simulation using "mailboxes" is no longer necessary.

Only two-up (2 pages on one sheet) output was available for the print files. The HP laser printers have the capability of producing output using a variety of environments and it would be useful to take advantage of these features however using the MRJE scheme, to do so would be extremely difficult, if at all possible.

The VAX Cluster and the two HP3000's are connected to the same ethernet segment. The VAX's communicate with one another using Digital's proprietary network protocol, DECnet. The HP's communicate with each other using a protocol which conforms to the IEEE/802.3 standard for ethernet. Unfortunately, DECnet and IEEE/802.3 are incompatible, otherwise the network transfer of printfiles from the VAX to the HP would be a relatively simple matter. The main advantage of the ethernet connection is its high transfer rate of 10 megabits per second.

NEW SOLUTIONS

By adding a special hardware board manufactured by Interlan, to the VAX, the protocol differences can be overcome. In fact the Interlan card transmits packets which are 802.3 compatible at the data link and physical layers of the ISO model, thus permitting the two nodes to communicate. Special HP software (NS for the VAX) must be run on the VAX to permit communication with the TCP/IP based NS/3000 software running on the HP. This now permits file transfers between the VAX and the HP. The file transfer command DSCOPY, allows the user to specify the file to be transferred and indirectly the print environment to be used.

The potential now exists to take advantage of the functionality of the 2680 laser printer.

The problem we now face is to take a print file, add appropriate banner pages to it, and issue a DSCOPY command to transfer it over the ethernet to the HP. The current version of VMS (5.0) provides a cluster-wide queue from which print files can be accessed. There were three techniques considered for implementing this. First considered was modifying the VMS print symbiont; a second possibility was to write a device driver which would gather records into a file and present to the DSCOPY command.

The third technique, and the one successfully implemented, was to use DECnet task to task communication to process the print files. DECnet tasks are programs running on different network nodes (each processor on the cluster is a node) , which can communicate with one another. One task resides on each node and traps "PRINT" commands. This task then requests a logical link to a target task on the VAX(named GAR) with the Interlan card, the node which can communicate with the HP. These two tasks exchange information (filenames, data, etc) and then the task on GAR invokes a command procedure which creates the banner and DSCOPY the new file to the HP.

We choose to provide users with the option to select 1-up, 2-up, or 4-up output. They would do so by issuing the appropriate PRINT1, PRINT2, or PRINT4 command on the VAX. The command procedure invoked by these commands would pass the appropriate environment parameter along so that ultimately the correct environment file is used by the HP spooler. Unfortunately, it is not possible to issue a DSCOPY command to a file equated to a printer. If it were, each file equate could contain the appropriate environment file reference and a spoolfile would be created directly as a result of the DSCOPY command. Since we couldn't do it the easy way, we created three groups on the HP; LP1, LP2, and LP4. The routine which creates a file name for the files to be transferred, would append the appropriate group designation so that the files to be printed 1-up would end up in the LP1 group, 2-up in the LP2 group etc. A process running on the HP would then wake up each 15 minutes, scan the groups for new files, and FCOPY them to the laser printer through the appropriate environment.

DETAIL OF VAX PROCEDURES

HPPRINT1 (see appendix A)

(For the sake of this description, the command PRINT1 will be used however the reader should bear in mind that it could be PRINT2 or PRINT4 as well.) A user on any processor issues the command PRINT1 <filename> [d]. This command would be trapped and the DCL command procedure HPPRINT1 would be invoked. HPPRINT1, on the local node, performs the following functions:

1) Checks for the existence of the filename parameter.
2) Checks for wild cards (not currently permitted).
3) Gets username and node name.
4) Builds full file specification evaluating any logical definitions.
5) Defines remote task on the node GAR.
6) Checks for delete after print option (see [d] in PRINT1 command).
7) Executes the program VAXTOVAX establishing link with GAR

VAXTOVAX (see appendix B)

This routine is necessary because the program which invokes the common VAXTOHP task on GAR must have system privileges. HPPRINT1, executing in a user account would have the privileges of that user which in almost all cases would be insufficient. By making VAXTOVAX an "installed" program on the VAX, the user can invoke a privileged routine without having the privileges himself. DCL programs can not be "installed" therefore the routine was written in "C". An additional advantage of "C" is that it accommodates command line arguments, which facilitated invoking VAXTOVAX from HPPRINT. VAXTOVAX performs the following functions:

1) Open "task" which is the FORTRAN program VAXTOHP on the processor GAR, which communicates with the HP.
2) Transmit the parameters it received from HPPRINT1 to VAXTOHP:
   a) username
   b) file specification
   c) file name
   d) node name
   e) environment code
   f) delete code
   g) record attribute (FORTRAN carriage control or not)

VAXTOHP (see appendix C)

When the program VAXTOVAX invokes "task" which has been defined as VAXTOHP, a command file VAXTOHP.COM is run which contains the single command Run VAXTOHP.EXE. It is a requirement of DECnet task to task communication that the remote process be started in this way.

The program VAXTOHP was written in FORTRAN for at least two reasons: there was a relatively straightforward example in the DECnet documentation of task to task linkages using FORTRAN, and the application was not a difficult one for this language. It could have been written in any language supported by VMS. VAXTOHP performs the following functions:

1) Input parameters from VAXTOVAX running on a local node.
2) Using equivalence statements, some of these parameters are used to construct a constant "startjob" which contains the DCL command to start the command file "template". Startjob is spawned by the lib$spawn command, when VAXTOHP has completed building the banner etc. "template" actually contains the DSCOPY command and will be described below.
3) Build a banner file naming it Xhmmsshh.lpn where
   - h = rightmost digit of hour
   - mm= minutes
   - ss= seconds
   - hh= hundredths of seconds
   - n = environment (1,2,or 4)

   (the time values come from the system clock and provide the banner with a unique name which is compatible with the HP.)
4) Scan the print file for carriage control characters and tab characters, converting and shifting as necessary.
5) Spawn startjob. Since any of five processors could request print services simultaneously, each request results in an independently spawned process with a uniquely named printfile.

TEMPLATE (see appendix D)

Template is a DCL command procedure spawned by VAXTOHP. If the process completes successfully, it does not return to VAXTOHP. If it does not complete successfully an error condition is returned to VAXTOHP which makes note of the error. Template performs the following functions:

1) Builds HP compatible filename of the form filename.lpn
2) Sends message back to user indicating new name of file to be transferred.
3) Issue DSCOPY and transfer file to HP.
4) Delete print file if requested.

note: I wish to acknowledge the considerable assistance provided by Sara Dearing, Sr. VAX System Manager, and Bruce Senn, Sr. HP System Manager, in the successful implementation of this project.

```
$! ************************  PRINTHP1 ***************************

$! This command file establishes link with common print routine
$! on GAR.  Parameters are passed to GAR to enable the common
$!  routine  to build  banner  and  printfile.   This  routine is
invoked
$! by the command print1.
$! ************************************************************

$ filename = p1

$ask:
$ if filename .eqs. "" then inquire filename "_File"
$ if filename .eqs. "" then goto ask
$ star = f$locate("*",filename)
$ length = f$length(filename)
$ if star .eqs. length then goto continue
$ write sys$output "***  Wild cards not permitted at  the present
time ***"
$ exit
$continue:
$gosub getfilespec
$ filespec = f
$ user = f$getjpi("","username")
$ node = f$getsyi("nodename")
$ rat =  f$file_attributes(filespec,"rat")
$ if rat .eqs. "" then rat = "none"
$ assign NL: sys$output
$ define/nolog task "gar""hpprint password""::""0=vaxtohp"""
$ deassign sys$output
$ if p2 .eqs. "D" then goto callvax
$ if p2 .eqs. "DELETE" then goto callvax
$ p2 = "N"
$callvax:
$ vaxtovax :== $$1$dua2:[hpprint]vaxtovax
$ vaxtovax  "''user'" "''filespec'"  "''p1'" "''node'"  1 "''p2'"
"''rat'"
$ exit
$getfilespec:
$ f = f$search(p1)                    !store file spec in f
$ if f .nes. "" then goto next1       ! null if file not found
$ write sys$output "File ",p1,"   not found"
$exit
$next1:
$ c = f$locate(":",f)                  !save location of colon.
$ lb = f$locate("[",f)                     !  save location of  left
bracket [.
$ drive =  f$extract(0,c,f)                !  split apart drive  and
```

```
directory.
$ newdir = f$extract(c+1,50,f)
$start:
$ trns  = f$trnlnm(drive)                       !  translate drive  if
predefined,
$ if trns .eqs. "" then goto exit1  ! if not f is filespec.
$ len = f$length(trns)                  ! get  length of translated
drive etc
$ rb = f$locate("]",trns)           ! find ].
$  if  rb .eqs.  len then  drive  = trns  ! only  drive component
returned if
$ if rb  .eqs. len  then goto start      ! no ]  found, return  to
translate again.
$ rd = rb-1
$ rdot = f$extract(rd,1,trns)       ! check for dot to left  of ]

$ if rdot .eqs. "." then goto next4
$ write sys$output "error, no .] in dir spec ",trns
$ exit
$next4:                                         ! build  new directory
component.
$ colon = f$locate(":",trns)
$ drive = f$extract(0,colon,trns)   ! split drive and directory
$ dir = f$extract(colon+1,len-colon-2,trns)
$ d1 = f$extract(0,f$length(dir),dir)
$ d2 = f$extract(1,f$length(newdir),newdir)
$ f = drive + ":" + d1 + d2
$ goto start
$exit1:
$! write sys$output "filespec= ",f
$return
$ exit
```

```
/*  Program run from  PRINTHP1.COM, PRINTHP2.COM and PRINTHP4.COM
which accepts the  username, full file specification,  file name,
node,  number  corresponding  to the  appropriate  hp environment
(1-up, 2-up  or  4-up),  and a  D  or an  N  to indicate  if  the
printfile is to be  deleted as command line arguments  and writes
them out to sys$net using decnet task to task communication. */

#include <stdio.h>
main(int argc, char *argv[])
{
     int i;
     FILE *fd;

/* open task that is defined in PRINTHPx.COM */

     fd = fopen ("task","w");
     if (fd == NULL) { printf("file open error"); exit(-1); }

/* print username,  full file specification, file  name, node, hp
environment  number  and  D  for  delete  or N  for  nodelete  of
printfile to the opened task via sys$net */

     for (i=1; i<argc; i++)
          fprintf(fd,"%s%s", argv[i], (i<argc-1) ? "\n" : "");

/* close the task (sys$net) */
     if (fclose(fd) ==  EOF) { printf("error closing file");
                           exit(-1); }
}
```

```
C
C     THIS FORTRAN PROGRAM GETS PRINTFILE INFORMATION FROM
C     HPPRINTx THROUGH VAXTOVAX.C
C     A BANNER FILE IS BUILT AND THEN THE PRINTFILE IS READ
C     RECORD BY RECORD TO CONVERT ff CHARACTERS TO FORTRAN
C     CARRIAGE CONTROL. THEN tabs ARE INTERPRETED AND THE
C     NECESSARY SHIFTING DONE.  THE CONVERTED PRINTFILE
C     RECORDS ARE THEN ADDED TO THE BANNER.  THE COMMAND
C     PROCEDURE template IS THEN SPAWNED AND THE BANNER
C     AND PRINTFILE IS DSCOPY'D TO THE HP.
C

      character  user*12,filespec*100,f1*15,dat*9,tim*8,stx,so,si
      character node*4,SK*3,env,datetime(23),bann*12,banner(12)
      character fs*100,startjob*137,startj(137),ff,del,rat*3
      character input*133,output*133,buffer(134),tab
      equivalence(bann,banner,startj(112)),(startjob,startj)
      equivalence(filespec,startj(11)),(user,startj(125))
      equivalence(del,startj(137)),(fs,f1)
      equivalence(input,buffer(2)),(output,buffer(1))

      INTEGER*4 istat,flag

      DATA  tab/9/
      DATA  ff,stx,so,si,node,env,flag/12,2,14,15,'init',0,17/
      DATA startjob/'@template ffffffffffffffffffffffffffff
     1ffffffffffffffffffffffffffffffffffffffffffffffffffff
     1ffffffffffffff bbbbbbbbbbbb uuuuuuuuuuuud'/
      DATA bann/'Xhmmsshh.lpn'/
c
      OPEN (UNIT=1,NAME='SYS$NET', ACCESS='SEQUENTIAL',
     1     FORM='FORMATTED',  CARRIAGECONTROL='NONE', TYPE='OLD')

c
      call date(dat)
      call time(tim)
 10   read(1,100,END=30,err=900)user
      read(1,100,end=30,err=900)filespec
      read(1,100,end=30,err=900)fs
      read(1,100,end=30,err=900)node
      read(1,100,end=30,err=900)env
      read(1,100,end=30,err=900)del
      read(1,100,end=30,err=900)rat
c
c     open new banner file, build file name
c
      a=lib$date_time(datetime)
```

```
      banner(2) = datetime(14)
      banner(3) = datetime(16)
      banner(4) = datetime(17)
      banner(5) = datetime(19)
      banner(6) = datetime(20)
      banner(7)= datetime(22)
      banner(8)= datetime(23)
      banner(12)= env
      write(6,*)banner,datetime
      open(unit=2,file=bann,status='new',recl=133,err=901,
     1access= 'sequential',blocksize=133)
      go to 11
 901  write(6,*)'open failed on device 2, banner.dat'
      go to 35
c
c     send hex 2 to prespace hp
c
 11   continue
      write(2,105)stx
 105  format (a1)
c
c     skip 3 lines
c
      do 16 i=1,3
 16   write(2,*)' '
c
c     build banner
c
      write(2,*)'      ',so,user,si
      do 18 i=1,15
 18   write(2,*)' '
c
      write(2,*)sk,'V     V    AAA    X     X          ',
     1' OOO    U     U  TTTTT   PPPP    U     U   TTTTT'
      write(2,*)sk,'V     V   A     A   X     X          ',
     1'O     O  U     U     T     P     P  U     U     T   '
      write(2,*)sk,'V     V   A     A    X X             ',
     1'O     O  U     U     T     P     P  U     U     T   '
      WRITE(2,*)SK,'V     V   A     A     X              ',
     1'O     O  U     U     T     PPPP    U     U     T   '
      WRITE(2,*)SK,'V     V   AAAAA    X X               ',
     1'O     O  U     U     T     P       U     U     T   '
      WRITE(2,*)SK,' VVV    A     A   X     X          ',
     1'O     O  U     U     T     P       U     U     T   '
      WRITE(2,*)SK,'  V     A     A   X     X          ',
     1' OOO     UUU      T     P         UUU      T   '
c
c     skip 5 lines
c
      do 21 i=1,5
 21   write(2,*)' '
c
```

```
      do 22 i = 1,3
 22   write(2,102)user,f1,dat,tim
C
C     skip 15 lines
c
      do 25 i=1,5
 25   write (2,*)' '
c
      write (2,104)user,node,filespec


c
c     skip to next page
c
      do 26 i=1,18
 26   write(2,*)' '
c
c     skip env-1 pages
      if (env.eq.'1')j=0
      if (env.eq.'2')j=1
      if (env.eq.'4')j=3
      do 28 i=1,j*61
 28   write(2,*)' '
c
c
      open(unit=3,file=filespec,status='old',err=515)

 50   read(3,106,end=52,err=51)input
c
c  scan for tabs and shift right
c
      do 54 i = 2,125
      if(buffer(i) .eq. tab) then
        do 53 j = 133,i+1,-1
        k=mod(i-1,8)
        if (k .eq. 0) k = 8
        idelta = 8-k
        buffer(j) = buffer(j-idelta)
 53     continue
c
c       blank fill behind shift
c
        do 531 m=1,idelta+1
 531    buffer(i-1+m) = ' '
      endif
 54   continue
c
c
c
      if(rat .ne. 'FTN')then
        do 55 i=2,133
        if (buffer(i) .eq. ff)then
           write(2,108)'1'
```

```
            goto 50
          endif
 55       continue
c
          write(2,107)output
        else
          write(2,107)input
        endif
        go to 50
C
C
 51     write(6,*)'read error on ',filespec
        go to 30
 515    write(6,*)'error opening ',filespec
        go to 30
 52     continue
        write(6,*)'end file on ',filespec
        close(unit=2)
        close(unit=3)
c
c
c       setup parms and spawn command file to
c             issue dscopy to hp
c
        istat = lib$spawn (startjob,,,)
        if(.not. istat)go to 903

 30     WRITE(6,*)'END OF vaxtohp'
        go to 35
c
 35     CLOSE(UNIT=1)
        close(unit=2)
        stop
c
 100    format(A)
 101    format(3a)
 102    format('   VAX/VMS  >>>',A12,' <<<       ',A15,
       1' ',A9,' ',A8)
 104    format('   USER: ',a12/'   FILE: ',A4,'::',A50)
 106    format(a132)
 107    format(a133)
 108    format(a1)
c

 900    write(6,*)'error reading file data on sys$net'
        go to 35
 903    write(6,*)'error in spawned process'
        go to 35
c
        END
```

```
$set verify
$ on error then goto next
$ cctl = "c"
$ group = "1"+cctl+f$extract(11,1,p2)
$ hpnam = f$extract(0,8,p2)
$ newp2 = hpnam+"."+group
$ rename 'p2 'newp2
$ reply/bell/user = 'p3 "PRINTFILE ''p1' QUEUED AS ''newp2'"
$ on error then goto copyerr
$ dscopy 'newp2 "gz#manager/???.vax,''group'#''hpnam'"
$continue:
$ delete 'newp2;*
$write sys$output "p4= ",p4
$ if p4 .eqs. "D" then delete 'p1
$ exit
$copyerr:
$ on error then goto next
$ dscopy 'newp2 "hp#manager/???.vax,''group'#''hpnam'"
$ goto continue
$next:
$ reply/bell/user = 'p3 "error in print file transmission"
$ exit
```

Design, Implementation and Support of Applications Using Remote File Access: A First-Hand Experience

By Kevin Darling
The Gap, Inc.
Eastern Distribution Center
3434 Mineola Pike
Erlanger, Kentucky 41018
(606) 283-1100

What is RFA?

Remote File Access or RFA is a service that allows you to create, open, read, write, and close files or devices on remote HP 3000's. Implied is the ability to access physical devices like tape drives and printers as well as disc files. The RFA functions use the same HP file system intrinsics that are used to control files locally. These intrinsics are literally sent to the remote machine for processing there. Programs can either use the system intrinsics available for file access or they can use the file access options available for the particular language the development is being done in.

One enhancement to the NS/3000 RFA over DS/3000 RFA is the introduction of nowait I/O. As implied, the program requesting the nowait I/O does not have to "block" itself pending the completion of the I/O request. Execution continues and verifies the results of the access at a time that is appropriate. This form of access requires the application to be in Privileged Mode. As with any use of Privileged Mode, developers need to heed the warnings printed by HP in the documentation.

Accesses Using RFA

Remote File Access can be done interactively or programmatically. In both cases a valid :FILE equation or FOPEN call needs to be issued to gain access to the remote file. Examples for the interactive use of RFA can include the following.

Example 1 - Copying a file from a remote to local machine.

First an environment needs to be defined for accessing the remote node and then the file can be copied:

```
:DSLINE RMT3K
:DSCOPY DOCUMENT:RMT3K[MGR.PROD]
```

This will allow DSCOPY to programmatically logon to the remote machine and then copy the file to the local machine.

Example 2 - Using a file on a remote machine.

Using the same structure to setup the environment, a file equation for the file to be accessed needs to defined before it can be used:

```
:DSLINE RMT3K
:REMOTE HELLO MGR.PROD
:FILE OUTPUT=RMTMSGS:RMT3K
:FCOPY FROM=LOCALMSG;TO=*OUTPUT
```

Or assuming that the "file" is a device:

```
:DSLINE RMT3K
:REMOTE HELLO MGR.PROD
:FILE PRINTER;DEV=RMT3K#LP
:LISTF @.@,2;*PRINTER
```

In the case of the programmatic access, the environment on the remote node needs to be established. Once completed, the local application can access the remote "file(s)" using either the standard HP intrinsic calls or the file access routines of the language. The reference to the node on which the "file" resides can be specified on the :FILE statement with the formal designator being used in the actual open call. For example:

```
:DSLINE RMT3K
:REMOTE HELLO MGR.PROD
:FILE X=X:RMT3K
...
FOPEN (X,...)
```

In Pascal applications, the file name used in a non-intrinsic open can actually contain the node name. In the use of the intrinsic FOPEN, the node location can either be referenced in the file name or device parameter of the call. Examples include:

```
:FILE X=X:RMT3K
```

Pascal -

```
        ...
OPEN (X, 'X:RMT3K');
```

FOPEN'S -

```
FOPEN (X:RMT3K,...)
or
FOPEN (X,...,RMT3K#,...)
```

The file equation can also be used to override the program definition for a file name even if it includes a node location. For example, the following alters the predefined node location:

```
:FILE X:RMT3K=X:NEWNODE
...
FOPEN (X:RMT3K,...)
```

As seen above, system intrinsics can be used in the accessing of the remote files. Use of an intrinsic on the remote file gives the same results to the program as if the file existed on the local machine. File system intrinsic condition codes retain their current definition and the occurrence of network connection errors return CCL condition codes. These conditions can then be processed by an FCHECK to determine the exact source of the error. As added insurance, the file's formal designator can be reviewed by the FPARSE intrinsic to insure it is syntactically correct.

One of the key elements of the Remote File Access facility is the :FILE command. The :FILE command as seen above is used to specify the formal designator for the remote file. As a brief review, following is the syntax of the :FILE command with specific reference to the remote access parameters:

```
                          ┌                                        ┐
                          │=*filedesignator                        │
                          │=$NEWPASS                               │
                          │=$OLDPASS                               │
:FILE filedesignator      │=$STDIN                                 │
                          │=$STDINX                                │
                          │=$STDLIST                               │
                          │=filename[:nodespec][,filedomain]       │
                          └                                        ┘

       [;DEV=[[envname]#][device][,outpri][,copies]]
       [;VTERM]
       [;ENV=envfile[:nodespec]]
       [;valid equation options]
```

References to specific documentation of the parameters can be found in the MPE Commands Reference Manual.

Message Files and RFA

Message files are the method used for communication between processors on the HP 3000. Processes are able to use RFA facilities to access message files to help distribute processing across CPU's. By means of normal MPE file system intrinsics, a process can write a transaction to a remote message file which are then read by the local program.

Message files also have the ability to help link multiple processes on multiple nodes. Such an implementation would include special applications to oversee the message traffic and direct the message to the correct processor. This program could be categorized as a "message switcher". Such a program running on each node would do the following:

* Open a message file as the reader.
* Open a message file for each local program as a writer.
* Establish any remote contacts necessary for operation.
* Open any remote message files.

Any of the local programs could communicate with programs running on other nodes by passing transactions through the local message switcher. Once received, the message switcher then routes the transaction to the correct processor on the correct node (see figure 1). Using the message file facility, interprocess communications can be simplified.

An RFA Implementation

To understand the implementation, we need to describe the functionality of the system. The environment began with one distribution center. The distribution center was expanded to support another division. Each division's data was to be maintained separately and only the shipping portion was to be merged. By merging the shipping functions, they were able to take advantages of cost savings. But even with the merging of the shipping data, each division required easy access to their data.

To support this from an applications stand point, a redesign of the shipping data flow was needed. Data was going to be centralized and yet it was necessary to provide access by both machines. To do this, we defined one machine as the master system and the second a slave. The master system would contain all the data for shipping processing while the slave system would only maintain data that was required for the users to access. This design required some data duplication; but, it also allowed for some balancing of processing. We took advantage of a smaller processing volume to locate the master system thus allowed us to improve the response time in most cases even with the data duplication.

As seen in figure 2, we designed a system that had two main message switching programs. One for processing picking information and the second to handle the processing internal to the shipping system. In both cases the programs had multiple input message files they had to deal with. When a transaction required transmission to the other system for processing, it was transmitted across the LAN to the remote input file.

Message Switching Example

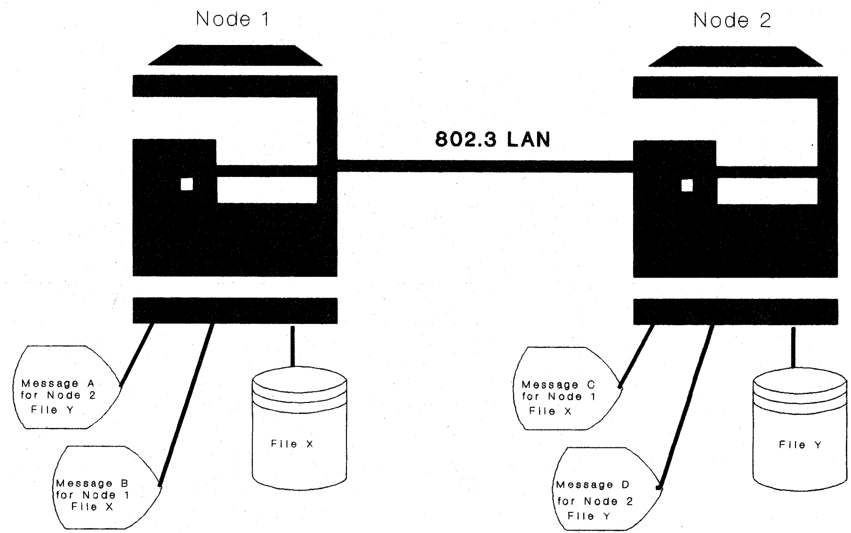


Figure 1

System Using Remote File Access

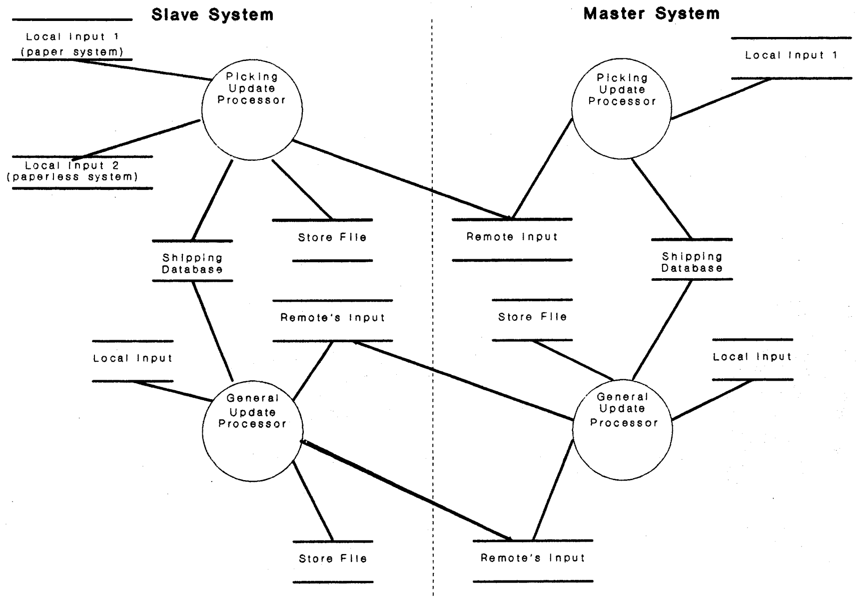


Figure 2

Coding Issues

Some problems encountered in the development included:

* A need for multiple inputs
* A need for a methodology LAN failure processing
* A need for online processing status requests
* A need for transaction integrity

The use of multiple inputs had two purposes. In the case of the picking system, the slave system had two different systems that input came from. On the master system, the inputs were for the local picking system and the remote update requests from the slave system. The general update processor used the two inputs to determine if the transactions needed to be transmitted to the other machine for update. If a transaction was processed from the local input file and the data was required to be updated on the other machine, it would be sent on to the other processor upon completion of the local update. If the transaction was processed from the file designated as the input from the remote system, the updates would occur but the transaction would not be routed back. This prevents the possibility of run-away transactions and the possibility of the queues getting full from the run-away scenario.

A serious issue to contend with is the issue of handling LAN failures. Some sort of processing scenario had to be built into the code to handle such failures and not loose the transactions. The solution was to use the file intrinsics to handle the I/O to the remote files. Whenever a write was completed, the status was checked to determine if the write was successful. If it was not, a local store file would be opened and all transactions from that point on would be routed to this local file. The program then notifies the Operator of the failure. When the situation is corrected, the Operator can reply to an outstanding reply the program checks ever so often and when it detects the Operator has said the problem is fixed, the program opens the local store file and the remote file again. It then reads the stored records and transmits the entries across. If there is a failure during the recovery, it continues normal processing by opening the store file to append transactions.

Since these programs run all the time as batch jobs, reading the input queues and processing when needed, we provided code to allow operators to reply to the outstanding reply of the program with a command to request the processing status. This function is also used to change the processing requirements of the program, such as input ratios for the input files.

The transaction integrity issue had been initially addressed with the existing system with the processing of most critical updates by a single processor. This was carried forward in the design of the remote input file processing. All transactions are still routed through discrete processors to insure that the data is

Processor Considerations

During our implementation of the system initially, we had a Series 68 defined as a slave system providing updates to a Series 70. This was all well and good for the first few months. But, during the fall we upgraded the slave system to a Series 950 to handle the other processing requirements on that box. This made the master system inundated with updates since it was not able to keep up with the processing of the 950. In some cases, we found that some updates were being impacted since our input ratios for the input files were biased incorrectly. After some fine-tuning of the system, we were able to limp well enough along until the second Series 950 was installed for the other division.

Other Performance Issues

As the system has been enhanced over the last year, we have found some issue with the Remote File Access facilities. One critical problem discovered during some prototyping showed that the time to access a standard MPE file remotely made the program take a great deal of time longer than if the file resided on the local system. If remote file access like this is required, attempts should be made to either use Remote File Transfer to get the file to the local machine before accessing or possibly some type of message switching system get more than one record at a time. Also, performance can be improved for processes that frequently open, close, and access files by keeping the remote file open to that remote environment. As with most processing, the first open of a file takes a significant amount of resources. However any other opens to that same remote environment are more efficient since the resources are shared. Also HP recommends that applications using many files open the files with the largest blocksize first.

Conclusions

The overall implementation of Remote File Access has been very successful. The use of the design for error processing has been critical to the success of the processing and has been effective in preventing major system problems. Much like RFA, Remote Database Access also is an effective tool for the distribution of processing. In the case where LAN's can be used for inter-CPU connection, high volumes of transactions can be handled easily.

WHO OWNS MY NETWORK?

Presented by
Eric S. Fisher
Fisher Systems Consulting
320 Old Ayer Road
Groton, MA 01450

+1-508-448-6556

ABSTRACT

The instinctive answer to the question of network ownership is, "Of course I DO!" However, the real answer is more complex than it seems. The person paying the bills for the network certainly owns the physical plant: the cable, the network hardware, the communications lines, and so forth. But who owns the software environment? How can we avoid becoming locked into one vendor's proprietary architecture? Is it worth the effort? What are management trade-offs for "open" versus "proprietary" approaches? What is really meant by "open" architectures and "standard" protocols? What are the cost trade-offs? How do we plan for the future in an open versus proprietary environment?

In this paper, I will analyze some hidden costs of network ownership and try to provide some criteria for answering the above questions.

INTRODUCTION

There are many components to a workable network design. They divide into hardware and software, workstations and servers, applications and systems. Each must be chosen carefully in order to operate easily with the others. At every point in the design process, several questions arise:

- o Is this component compatible with the other components selected so far?
- o Does this choice fit in with long-term plans? Does it matter?
- o By making this choice, what options become available? Which are ruled out?
- o Does this choice enhance or reduce my overall control and thus ownership of the network?

To illustrate these points in a practical context, I will discuss design issues involved in selecting three network environments: Novell, 3Com, and HP OfficeShare. I have chosen these three because each typifies a different answer to the above questions. In addition, I will discuss various cable plant designs that work well in these environments.

HARDWARE INTERFACE COMPATIBILITY

There are three major points at which hardware interface compatibility has been defined: bus, AUI and cabling medium. Each point can be thought of as a "boundary" where different suppliers or technology can meet. Figures 1, 2, 3 and 3a show the position of the boundary for each compatibility level. Design issues apply at each point.

BUS COMPATIBILITY

Input/output buses become an important issue when enough machines with the same type of bus come into the market to make it profitable for third party manufacturers to compete. With the advent of the Apple II and the IBM PC, the "add-in" market for microcomputers grew to a potential of several million units. Today, "industry standard" buses fall into the following major categories:

"AT" BUS

The "AT" bus, so called because IBM labelled its Intel 80286-based PC the "PC/AT", is a 16-bit extension of the original IBM PC bus. It is the most common bus in existence today. Interface cards abound for this bus architecture, enabling application designers to create many custom environments at very low cost. Although this bus architecture is generally associated with the Intel 80x86 microprocessor, other CPU's, especially the Motorola 680x0 series, have been successfully interfaced. Because of the size of the AT bus market, adapter cards designed for this bus are substantially less expensive than any other type.

NuBus

The NuBus is Apple's Macintosh bus. The Macintosh computers are based on the Motorola 680x0 microprocessors and the NuBus is optimized for this CPU. While not as common as the AT bus, the NuBus/Macintosh market is still large enough to allow lively competition and much innovation. However, the smaller size of the market is reflected in the generally higher price and more limited variety of interface cards. Intel 80x86 applications are limited to DOS coprocessor cards in this environment.

Micro Channel

Perhaps the most controversial of all the standard buses, IBM's Micro Channel is also the most recent. Introduced with the PS/2 family, this bus is purported to solve all of the problems inherent in the architecture of the AT bus. It currently suffers from the same problem as the NuBus: the population of machines is not large enough to bring costs of adapter cards in line with those for the AT bus.

Card compatibility requirements, however, go much farther than simple bus compatibility. Just because a particular circuit card will not crash the computer does not mean that it will operate with the desired software. Different cards use different interrupt vectors, different memory addresses, different registers to perform the same functions. In order for a particular network interface card to work with 3Com 3Plus software, for example, a "driver" must be written to operate that card in that environment. Independent vendors of these cards, like Western Digital, will supply drivers for the major PC network environments, including 3Plus and NetWare. OfficeShare is not considered a major environment, and choosing OfficeShare means choosing cards that H-P supports.

Some hardware vendors recognize the problem of supplying drivers for every software environment, and attempt to solve it by designing their cards to act the same as some more common device. VGA cards are a common example, where "register compatibility" has become a buzzword for total conformity to behavior characteristics defined by IBM for the VGA standard. Networking cards are a much wilder environment, where each manufacturer has attempted to gain some market advantage by locking customers in to specific hardware. Some progress is being made, however. Recently H-P introduced their "StarLAN 10" cards and claimed that they were compatible with the much more common 3Com 3C503 "Etherlink II" card. Whether they will in fact perform with any driver written for the 3C503 is yet to be seen, but the effort made is certainly commendable.

AUI COMPATIBILITY

The Attachment Unit Interface (AUI) is the point at which a network interface card connects to the media adapter, often called a transceiver. The transceiver is responsible for placing the digital signals on the cabling and receiving the signals from other devices. In contrast to the bus interfaces, which are all "de facto" industry standards, the AUI interface has been strictly defined by an international standards body, the IEEE. Two types are common in office environments:

IEEE 802.3 CSMA/CD ("Ethernet")

The IEEE 802.3 standard is an evolution of the original Xerox Ethernet environment. It is sometimes known as "Ethernet 3" to distinguish it from "Ethernet 2", a mostly compatible predecessor. Most 802.3 installations use coaxial cable, but there is nothing in the standard that requires it. One of the advantages of the AUI interface is that it defines the network in terms of signal levels and pin locations, not in terms of the actual cabling medium. Several vendors have taken advantage of this fact to implement 802.3 network technology on twisted pair copper and fiber optic media. Given a transceiver that responds properly to the AUI signalling, the network interface card does not need to know anything about the physical environment that is supporting it.

IEEE 802.5 Token Ring

The IEEE 802.5 standard is an evolution of the IBM Token Ring research and development. Developed more recently than the Ethernet environment, its evolution has been closely coordinated with the IEEE standards body. Thus, there are no predecessor technologies for 802.5 environments. The predominant cable medium for 802.5 networks is shielded twisted pair copper. However, as in the 802.3 case, an AUI connection can be serviced by any transceiver following the 802.5 specifications.

One of the most important decisions to be made in the initial stages of network design is the selection of 802.3 or 802.5 for the fundamental topology. Commonly referred to as "Ethernet" or "Token Ring", the two are totally incompatible at the hardware level. Most major local area networking software, including 3Com and Novell, will run on either. Most issues of performance and reliability are matters of personal preference. The major concerns are for the other equipment to be used on the network. Most minicomputer and mainframe manufacturers have a preference for one topology or the other. Hewlett-Packard, for example, is exclusively in the Ethernet camp. If the network design is to include interoperability with Hewlett-Packard computers, then, the networking hardware must be of the 802.3 type.

CABLING MEDIUM COMPATIBILITY

In network design, far too little attention is usually paid to the transmission medium itself. It is easy to assume that coax is required for Ethernet and twisted pair is required for token ring, but as long as the transceiver follows the AUI interface rules it does not matter what the transmission medium is. Therefore, decisions about cabling medium can be made independently of decisions about network topology. The following types of cable are the most commonly used:

50-OHM COAXIAL CABLE

The original medium for Ethernet, this cable comes in two sizes, "thick" and "thin". Thick cable is now rarely used except for backbone networks because of its high cost and difficulty to manage. Thin cable, however, is still one of the most common Ethernet wiring schemes.

ADVANTAGES

- o Interface cards are relatively inexpensive and available for every type of network software
- o Adding a device to the network is very easy: simply add one more loop to the cable

DISADVANTAGES

- o The cable itself can be as much as ten times more expensive as twisted pair
- o A fault in the cable disables every device in the same segment of the network; network segmentation devices are expensive and thus minimally used
- o Location and diagnosis of cable faults are time-consuming because of the electrical nature of coax

UNSHIELDED TWISTED PAIR

Commonly thought of as "telephone wire", unshielded twisted pair is an option for both Ethernet and token ring networks. IBM token ring has been available on this medium since its introduction, but only within the past two years has Ethernet technology evolved to the point that it can run reliably at full rated speed (10MB) on this medium. However, a number of vendors, including Synoptics, Cabletron and Hewlett-Packard, offer 10MB 802.3 equipment for unshielded twisted pair cable.

ADVANTAGES

- o This is the least expensive cable now available

- o The "hub and spoke" ("star") topology of twisted pair equipment limits failure from cable faults to the device being connected via the faulty cable

- o Fault isolation is automatic; repair is very quick

- o In many cases, especially in more modern buildings, it is possible to use existing telephone cable for the network

DISADVANTAGES

- o There are hidden costs: every eight to twelve network devices require a concentrator; to calculate an accurate cost of the network per device, a portion of the concentrator (hub) must be included

- o Only time will tell whether the manufacturers of this equipment have solved the RFI/EMI problems of carrying such powerful signals of such high frequency over unshielded cable

- o Adding a device to the network can be difficult if a cable run from the device location to a concentrator is not already in place

- o There are as yet no standards for the cabling medium side of the network; the IEEE 802.3 committee is working on it, but at present there is no guarantee of interoperability between hubs of one vendor and network interfaces of another

- o Each vendor of twisted pair systems sells a transceiver (MAU) that meets the 802.3 AUI specifications on the device side and connects via twisted pair to the hub. These MAU's can be expensive. Network cards with internal MAU's, quite common for thin coax, are currently rare for 802.3 twisted pair networks and tend to be restricted in the software that they support.

- A twisted pair hub counts as a repeater in a network. 802.3 specifications limit the repeater count to four (i.e. there can be no more than four hubs between devices on a network) before some sort of bridge must be introduced. This limit increases the care required in cable plant design.

SHIELDED TWISTED PAIR

Shielded twisted pair cable has been the preferred medium for IBM token ring since its inception. Some Ethernet implementations are also designed for it. Shielded twisted pair is available in a number of configurations, inluding one that matches the electrical characteristics of unshielded cable and can thus be substituted for use by equipment designed for unshielded cable.

ADVANTAGES

- EMI/RFI concerns are greatly reduced
- This is the only cable type guaranteed to carry IBM's 16MB token ring
- Can be used for any present or anticipated twisted pair scheme
- Even though more expensive than unshielded, still considerably less than coax

DISADVANTAGES

- More expensive (by about 50%) than unshielded
- If substituted for an unshielded specification, may require heavier gauge wire to carry the same distance
- This wire must always be installed new -- telephone wire is never shielded
- Shares network equipment disadvantage with unshielded twisted pair

FIBER OPTIC CABLE

Only recently has the associated equipment cost for fiber optic networks become low enough to allow consideration for any but the most demanding applications. Today, however, the equipment is sufficiently reasonable in price to make it possible to design backbone networks using fiber optic technology.

ADVANTAGES

- Totally immune to electromagnetic interference; also does not generate any
- Has the highest speed capability of any medium

DISADVANTAGES

- Cable cost is highest of all
- Standards are still very immature

SOFTWARE COMPATIBILITY ISSUES

If the hardware issues can be somewhat confusing, they are nevertheless far more straightforward than the software issues. In the software area, there are two "faces" of interest to network designers: the portion that looks at the network, called transport services, and the portion that works with the application, called presentation services.

NETWORK TRANSPORT SERVICES

Corresponding to layers one through four of the OSI model, network transport services are responsible for moving data from one device in the network to another. In order to do this, network communications protocols are used. The first step toward interoperability is to have all transport services use the same protocols, and they don't. Two major variants in use today are:

UDP/XNS

Xerox Network Services, the first protocol developed for Ethernet, is still widely used as a basic component by Novell, 3Com and many others. UDP, Universal Datagram Protocol, is the layer four service most commonly used with XNS.

TCP/IP

Transmission Control Protocol/Internet Protocol was developed by the Department of Defense Advanced Research Projects Agency (ARPA) for global networking systems. Adapted by some local area network vendors for network transport, this protocol is still the most successful in providing interoperability among disparate systems. Hewlett-Packard networking systems are based on this protocol. However, as shown in Figure 4, TCP/IP is not the whole story. The 802.3 standard, most often considered a hardware standard, also has software standard elements. 802.3 packet framing is not the same as Ethernet framing. H-P networking, having come later than ARPA, was designed to the 802.3 standard and is thus incompatible at the data link layer (layer two) with ARPA services.

While all three variants shown in figure 4 can coexist on the same physical network, they cannot talk to one another without translation services. To use the telephone system analogy, the situation can be compared to international telephone services where all people use the same telephone equipment and all use voice -- but some speak English, others French and still others German.

NETWORK PRESENTATION SERVICES

Where transport services manage movement of data on the network, presentation services provide a consistent application interface, so that developers of workstation and PC software do not need to be concerned about the particular network environment in which their applications will operate. Over the years, several de facto standard presentation services have been developed. Especially in the PC area, these have allowed application software developers to adapt their packages to network environments without extensive customization. Some of the most common examples are:

MS-NET

Microsoft Networking Services, commonly called the "redirector"

NETBIOS

An alternative workstation program interface to MS-NET, with similar but not identical services. Working at a slightly lower level than MS-NET, it appeals to network application designers because it more easily overcomes performance problems. Some networking software supports both interfaces.

ARPA SERVICES

A series of application interfaces designed to move information across disparate networks. The most common of these are ftp, a file transfer protocol; telnet, a virtual terminal interface; and smtp, an electronic mail system.

NFS

Network File System, a file sharing environment common in UNIX based systems.

Note that presentation services are only loosely connected to the transport services. MS-NET, for example, is supported on virtually all transport services. While this makes application design easier, it does not really solve interoperability problems, in the sense that a PC running MultiMate with NetWare will not be able to access a MultiMate document residing on a 3Com 3Plus server. Any stack in Figure 5 could be placed on top of any stack on Figure 4 and the application software would work the same.

OTHER PROBLEMS

CO-PROCESSOR CARDS

One thing common to all network service software is the amount of memory it takes and the CPU resources it consumes. A number of manufacturers have solved some of both problems by introducing network interface cards that have their own CPU and memory. These coprocessor cards run the network transport protocols without requiring the host processor to intervene. Only a small amount of host memory is required to contain the interface programs. Unfortunately, a number of problems arise:

- One manufacturer's coprocessor card cannot run another manufacturer's network software.
- XNS cards cannot run TCP/IP, and vice versa
- As if coprocessor cards were not expensive enough, they are only supplied with an AUI interface or a thin coax MAU. Twisted pair installations are out of luck.

NETWORK OPERATING SYSTEMS

Workstations run PC DOS, OS/2 or UNIX. Servers are not required to run any of the above. Although most do, Novell in particular uses a proprietary operating system for its servers.

CONCLUSIONS

Design of a network that serves present needs without hindering future growth is a very difficult task. Many seemingly irrelevant details must be carefully considered. True interoperability is a dream of the future; today a compromise is always required. A good network design will always include compromises that allow future growth and encourage evolution of standards.

HARDWARE INTERFACE COMPATIBILITY

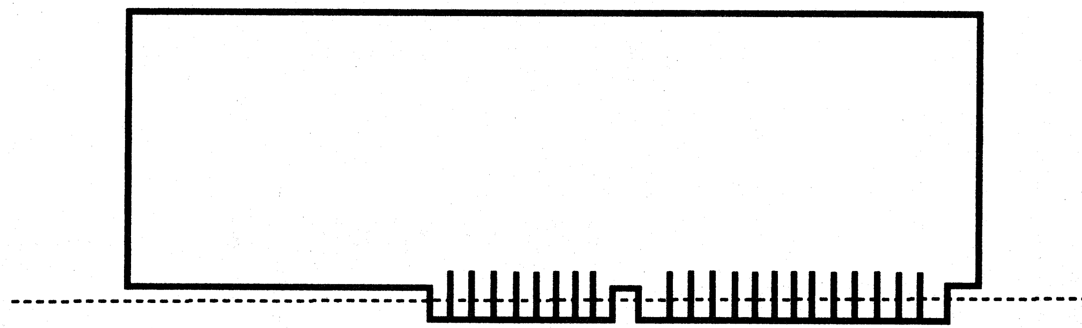


FIGURE 1: BUS COMPATIBILITY

HARDWARE INTERFACE COMPATIBILITY

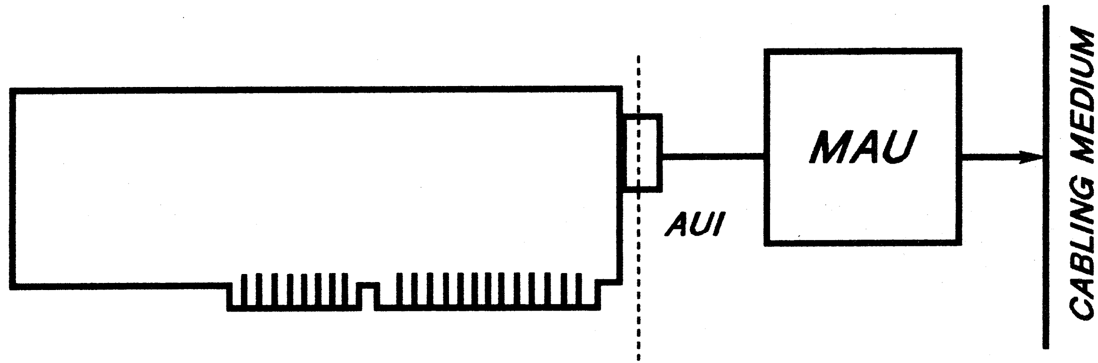
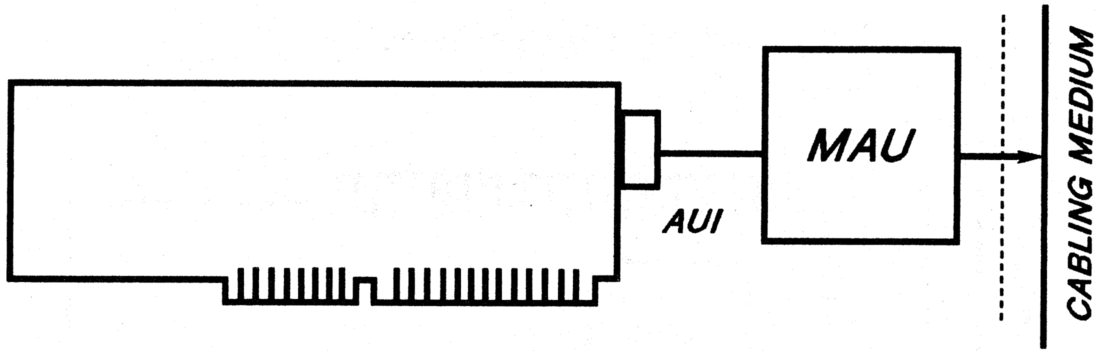


FIGURE 2: AUI COMPATIBILITY

4603 - 16

HARDWARE INTERFACE COMPATIBILITY



4603 - 17

FIGURE 3: MEDIA COMPATIBILITY

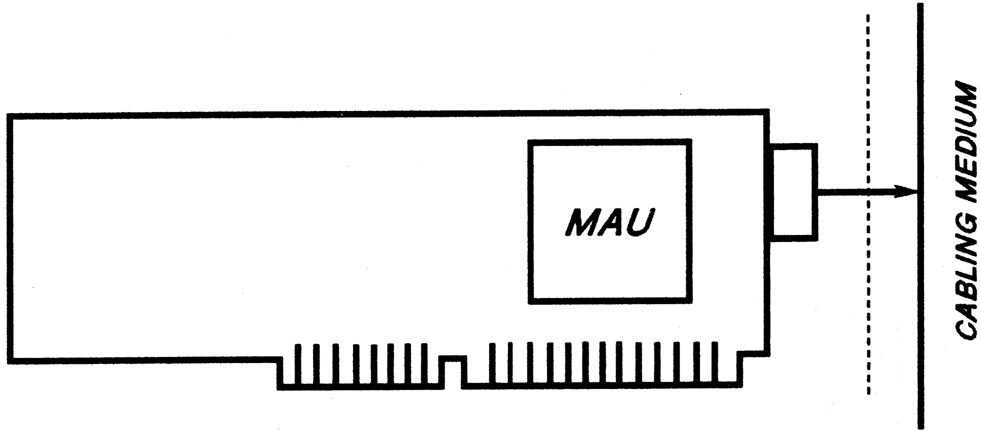


FIGURE 3A: INTERNAL MAU

SOFTWARE INTERFACE COMPATIBILITY

4603 - 19

TRANSPORT	TCP	TCP	UDP
NETWORK	IP	IP	XNS
DATA LINK	802.3	E-NET	E-NET
PHYSICAL	802.3	802.3	802.3
	NS OFFICESHARE	ARPA	3COM NOVELL APOLLO

FIGURE 4: TRANSPORT AND BELOW

SOFTWARE INTERFACE COMPATIBILITY

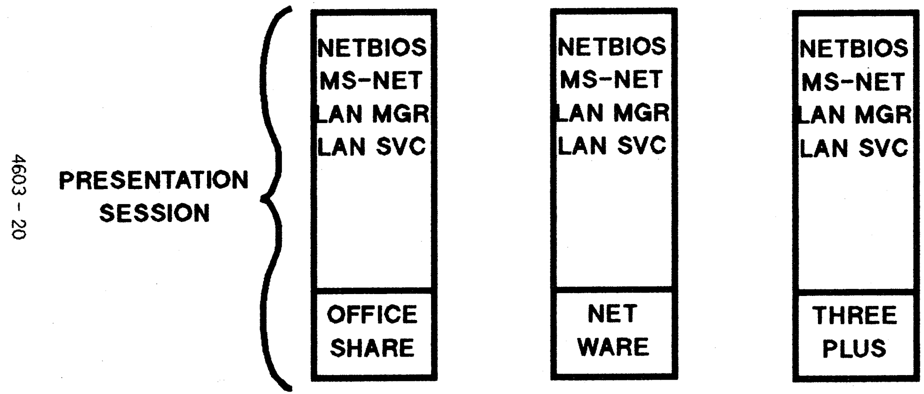


FIGURE 5: LAN SERVICES FOR PC SOFTWARE

THE HP3000 CONNECTION

"RS232 TO LAN"

Dave Heck/Lewis Dalrymple
The City of Tempe
120 East 5th Street
Tempe, AZ 85281
(602) 731-8777

*** INTRODUCTION**

You're right, the title does sound a little like an international spy network, which is smuggling 'who knows what' from code name 'RS232' to a mysterious place called 'LAN'.

Well, you are partially right, there is a little bit of intrigue in the "HP3000 CONNECTION", however, we hope to solve that mystery by showing you some simple methods for connecting peripherals to your HP3000 Computer.

We at the City of Tempe, with our (9) HP3000 (and growing) computer networks, have experienced about every situation one might imagine and have therefore developed many great techniques for streamlining peripheral connections to HP3000's from the Micro3000 through the series 95X.

*** RS232?????????**

We will begin by describing the primary method used in connecting HP3000 Computers to other data communications devices.

The code name is 'RS232C', which is a specification published by the Electronics Industries Association (EIA), governing the interface between data communications equipment (DCE), such as modems, and data terminal equipment (DTE), such as computers.

There are two basic parts to the 'RS232C' standard:

1) Electrical characteristics of the signals crossing the interface...

2) Connector/Pin numbers and the functions of the signals of the interface....

Table 1 describes both pin numbers and signals.

PIN #	NAME OF SIGNAL	ABBR.
1	GROUND	
2	SEND DATA	SD
3	RECEIVE DATA	RD
4	REQUEST TO SEND	RTS
5	CLEAR TO SEND	CTS
6	DATA SET READY	DSR
7	SIGNAL COMMON	
8	DATA CARRIER DETECT	DCD
15	TRANSMIT CLOCK	DB
17	RECEIVE CLOCK	DD
20	DATA TERMINAL	DTR
24	EXTERNAL MODEM	
25	OUT OF SERVICE	OOS

*** THE CONNECTIONS BEGIN**

Hewlett-Packard elected to use the EIA RS232 specification (standard) and the 25 pin D-Type connector, commonly used by the telephone and computer industry, as the standard for the first HP3000 computers. Yes, we can all (well, nearly all except you youngsters) recall the ADCC provided by HP as the standard interface to all peripheral devices. This required cables with 25 pin connectors on both ends to interface the computer and most other peripheral devices (terminals, printers, etc.).

Figure 1 shows a typical 25 pin connector/cable.

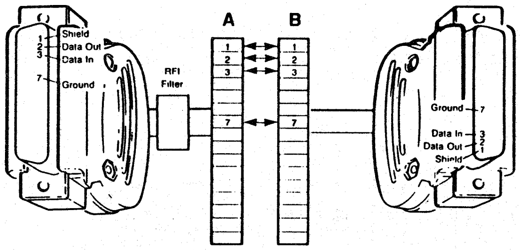


Figure 1

It was soon determined that most connections (terminals, printers, etc.) could be made with only three of the 25 pins available in the D-type connector:

Pin - 2 Send Data
Pin - 3 Receive Data
Pin - 7 Signal Common

This greatly reduced both connector and wiring costs for most interface requirements.

As new HP3000 computers evolved, requirements for more users (ports/connectors) indicated that the large 25-pin D-type connector used up far too much panel space and thus must be replaced.

This change produced a replacement for both the D-type connector and the ADCC port.

The ATP/3-pin port/connector became the new standard on all HP3000 Computers.

This opened the way for a new series of cables/connectors for the serial interface.

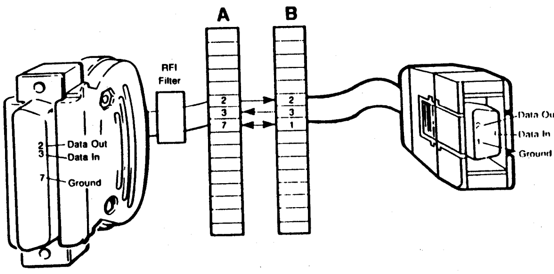


Figure 2

At this point we had eliminated the 25-pin D-type connector on the computer end of the cable and the troublesome 'hold down' screws which were either missing or too short.........or what ever........

Interconnection cabling was still a problem, since special cables still had to be assembled and then 'pulled' throughout the data processing area.
Add to this the RS232 standard cable length limit (50 feet) which was also adopted by HP (a most conservative standard).

Well, like most other things we are told not to do, we started stretching 50 feet to 100 feet and then to 1000 feet plus with few or no operating problems.

We had now achieved reasonable distances (the user didn't have to live in the computer room), however, we still had the problem of special cables and the runs to each peripheral device.

*** ENTER MODULAR WIRING**

The next advancement just seemed to answer all of our prayers, and it had been right at our finger tips all the time--modular connectors/wiring.....

The idea of using the modular plug/jack wiring scheme, developed by the telephone industry, opened up all the doors to the next generation of data communication wiring/connections.

Just what does this mean in the simplest terms??????

First of all, it allows interface wiring to be accomplished by using existing multiconductor, twisted pair cables used by the telephone industry, something that is available in almost every building location. Secondly, all available RJ11 (4/6 conductor) and RJ45 (8 conductor) modular connectors and connector/jack assemblies could be used to interconnect computers and peripheral devices.

As part of this windfall came the DB-25 modular adapter, a 25-pin D-type connector with an RJ11 or RJ45 interface connection.

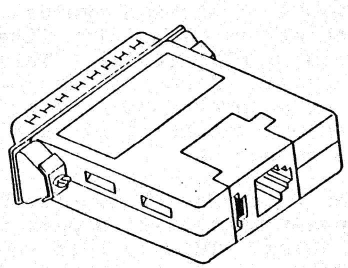


Figure 3

This meant that we could now interconnect our computer and terminals/printers without making and pulling special cables and without any solder connections.

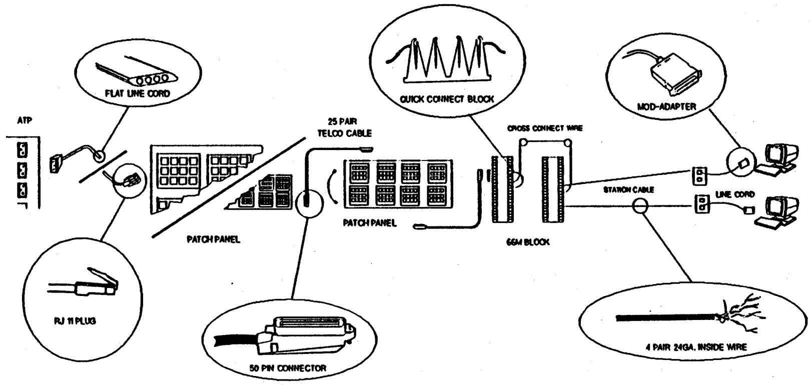


Figure 4

The modular wiring technology provided a wide range of connecting devices, including plugs/jacks (RJ11/RJ45), patch panels, quick-connect blocks (M66), and of course the modular adapter.

Our illustration (Figure 4) shows a typical connection path from the ATP connector, using flat line cord (4 conductor) with a modular plug (RJ11). This assembly is plugged into a patch panel which consists of 12 jack groups that are either connected to a second panel group or a preconnectorized quick-connect block via a 25 pair telephone type cable/connector.

Once we are connected to the quick-connect blocks, we can cross connect to conventional twisted pair conductors in existing telephone cables. This in turn routes to the user destination. We can then simply extend our connections out to wall jacks (RJ11) and provide the final connection through a RJ11 line cord and modular adapter. As simple as that........

Simple, efficient, and most of all cost effective. Oh yes, and let's not forget flexible.

The patch panel allows us the ultimate flexibility in being able to change ports/peripheral devices by changing the position of a patch cord on the patch panel(s).

* RS232 INTERCONNECTING METHODS

Using the modular wiring system provides us with a wide range of connection options which we will illustrate in the following examples:

+ Small Business System

For all you users with a Series 37, 42, or Micro 3000 computer, we recommend the arrangement shown in Figure 5.

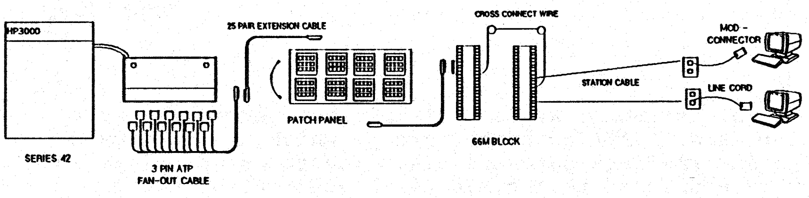


Figure 5

Here we have used a 12 connector (ATP) octopus cable wired to a 25 pair (50 pin) telco connector. This cable is extended (if required) with another telco cable to an intermediate patch panel.

We use patch panels for the greatest connection flexibility, however, this could be eliminated by connecting directly to a quick-connect wiring block. From the blocks you can route through existing telephone station cable to the appropriate user location(s). The rest is simply using the RJ11 line cord and the modular adapter to your terminal or printer.

+ The Computer Room

Computer room wiring, for you larger system users (Series 58, 70, 900), is handled in a similar manner as described for the smaller systems. Figure 6 provides a typical example:

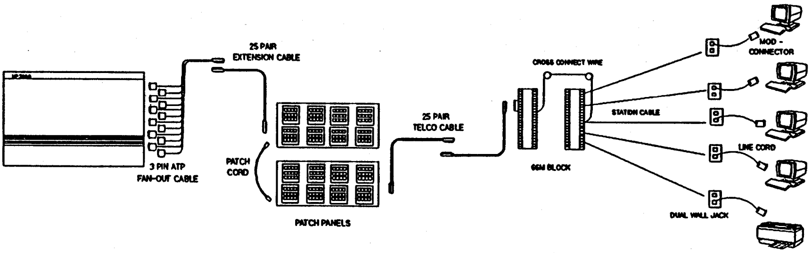


Figure 6

In this example, we use multiple octopus cables (one for each 12 ATP connector group) which are connected to intermediate patch panels. The output panel is in turn connected to the M66 block and routed to the user location via multiconductor telephone cables.

Final connections are again made via the RJ11 plug/jack assemblies and modular adapters.

If you have multiple computers in one computer room, and are prone to rearranging/upgrading your systems, we would further suggest patch panels mounted below your raised floor, near each computer. This way you can make your initial connections to RJ11 jack patch panels using individual ATP/RJ11 cables. This allows complete flexibility of all connections within the computer room without altering any 'down stream' wiring or patches when you move or upgrade to a new/different computer. This is great when you upgrade to a different machine and ports/LDEV's are changed.

+ Multicomputer/Multilocations

Until now, we have discussed only direct connect computer requirements.

Well, what if we have remote (outside the RS232 direct limit) computers/peripheral devices? Say a thousand feet or even 5 miles away.

The modular wiring scheme works just the same way, but with some different connection devices as shown in Figures 7 and 8.

In Figure 7 we show you an example of a LUXCOM fiber optic multiplexer link that we used to communicate to 200+ users in a building approximately 2000 feet from our central computer room.

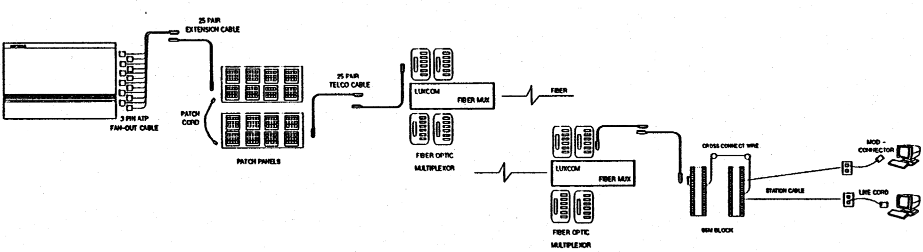


Figure 7

Again, we use the typical octopus fan-out-cable with 12 3-pin ATP connectors attached to intermediate panels using telco type connector/extender cables.

The output patch panel is also connected via 50 pin telco connector/cables to the LUXCOM multiplexers. It is, however, worth noting that the output patch panels were custom wired using a 3-wire connection scheme to provide 16 channels for each 50 pin connector, which matched the number of channels on each LUXCOM I/O module. This greatly simplified our overall wiring scheme.

The multiplexed signals are transmitted over fiber optic cable to the remote site and the connection process reversed.

Outputs were then routed through an existing telephone cable to the 200 user locations throughout our remote building complex.

Figure 8 illustrates our use of modular wiring for networks requiring modem/multiplexer connections for remote sites being serviced over leased telephone lines.

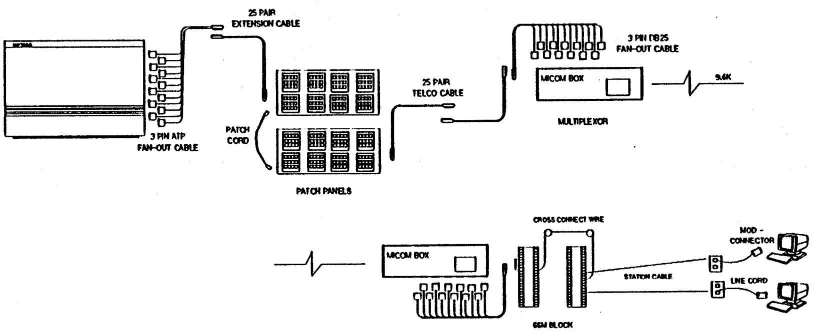


Figure 8

As our example shows, we have used the basic 'extremity' connection using octopus cables, patch panels, line cord/RJ11 connector assemblies, and modular adapters for the final peripheral device connections.

The new piece is how we connected the multiplexers.

Again, we use custom fan-out-cables with 12 DB-25 connectors (standard EIA interface for multiplexers) to allow us to attach directly to the patch panels through 50 conductor telco cables/connectors. All other connections are the same as shown in earlier examples.

*** SERIES 950 RS232/LAN CONNECTIONS**

A new 'twist' was introduced with the precision architecture machines, in that all direct connections are through Distributed Terminal Controllers (DTC) which are LAN (local area network) connected to the computer.

Well, not to worry, there is a simple solution, especially if you have adopted the modular wiring concept.

Although the LAN wiring/connection concept is a little bit 'foreign', we soon learn about BNC connectors, medium access units (MAU), and LANICS (the equivalent of the INP). These are all the items needed to make the coaxial cable connections which are the primary interface between the 900 series computer and the DTC.

Before we get too deep into this connection method let's start with some basics about the LAN.

+ THE LAN

A Local Area Network(LAN) is a communication system that provides the capability of economically distributing information between intelligent entities such as persons, workstations, and computational facilities in a local environment. Rather than being an end in itself, a LAN is the means of putting distributed systems together--ie. for integrating the components of automated systems.

o What Are The Benefits????

1) Efficient communication between devices.
2) Economical communication between devices.
3) Low interconnection costs.
4) Vendor independence.
5) Versatility of modification and expansion.
6) Support of multiple applications.
7) Application integration.
8) Protocol standardization.

9) High reliability of network components.
10) Common resource sharing.

o **Where is it all going?????**

Obviously in the HP world, with the 'Precision Architecture' machines, we have already arrived in the 'LAN' era for interconnecting both computers and/or workstations.

We at the City have already expanded our PC base to approximately 250 and now have over 20 StarLAN subnets throughout our facilities.

Continued reduction in hardware costs and an ever widening product/vendor base certainly indicates an ever increasing use of local and wide area network systems.

+ LAN BASICS

o **Cables** - In the HP LAN environment, there are three types of acceptable transmission media: ThickLAN, ThinLAN, and UTP (unshielded, twisted pair).

ThickLAN cable is the classic ethernet cable. It consists of a .0855 inch solid copper center conductor surrounded by a dielectric, two foil shields and two braided shields.

ThinLAN cable, on the other hand, is the less traditional "cheaperNET" cable. It consists of RG58 A/U coax with only one outer braided shield and a stranded center conductor. ThinLAN, however, does have several attributes which make it more desirable than ThickLAN. It's flexibility and ease of attachment make it far more practical in most installation environments and provides the lowest overall cost per connection/maintenance.

UTP (unshielded, twisted pair)

UTP, by far, is the most desirable transmission medium in the office environment today. It is the 'new kid' in the LAN cables, but it already is widely proven/accepted. UTP is not only flexible and easy to install,but it is inexpensive and in most cases already exists (your telco wiring).

HP has two LAN products based on UTP; StarLAN (1 Mbps) and StarLAN 10 (10 mbps).

FIBER OPTIC

Fiber optic cable is commonly used as a 'backbone' medium connecting LANs in different buildings together. All of its properties make it ideal for heavy traffic, rough environment, and long distances.

Fiber optic cable is simple in design, using a hollow core fiber (measured in microns) surrounded by a solid glass 'cladding' and finally a protective outer covering.

Light emitting devices are used to send signals over the cable while a detector receives the signals and converts them back to electrical inpulses.

Fiber represents the 'high speed' communication method of the future.

- **CONNECTING TO THE LAN**

ATTACHMENTS

The attachments to the LAN medium is accomplished in (3) three different ways depending upon the actual medium used (ThickLAN, ThinLAN, or UTP).

ThickLAN cable connections are made through a 'vampire tap'. This tap is designed so that you can attach it to an active network without disturbing the operations of the network.

The signal probe of the tap pierces the cable (through a hole that you drill with a coring tool) to make contact with the center conductor.

The tap is then connected to a signal converter (transceiver) called the Medium Attachment Unit (MAU).

THINLAN

Connections to the ThinLAN cable are made through BNC 'tee' connectors. Two sides of the BNC tee connector connect to the LAN cable; the third side connects to the ThinMAU. A 1 - meter cable connects the ThinMAU (normally a permanently attached cable) to the applicable LAN component (repeater, bridge, PC interface board, etc.). This cable is called the AUI or Attachment Unit Interface.

UTP

Attachments to unshielded, twisted pair (UTP) are as numerous as the mechanical connections available for wire. However, we tend to only think of those which we have previously described for 'Modular Wiring'.

The source of the UTP-LAN output originates with the RJ45 output of a StarLAN Hub (1 Mbps) or the StarLAN 10 Bridge (10 Mbps).

These signals are then routed/connected using a wide variety of connection devices, including RJ11 plug/jacks, patch panels, M66 quick-connect blocks, only to mention a few.

Computer connections, via UTP, are accomplished with a 'Twisted Pair MAU', which has an RJ45 jack for twisted pair connection and an AUI cable/connector for attachment to the HP3000 LANIC board.

o **LIMITS (DOs and DON'Ts)**

There are some basic rules that define both the medium and how it must be used. We have developed the following table to show you some of these specifications/comparisons in Figure 9.

	ThickLAN (802.3)	ThinLAN (802.3)	StarLAN 10	HP StarLAN
Topology	BUS		STAR	
Data Rate	10 Mbps			1 Mbps
Cable Type	Belden 9880 (50 Ohm)	RG 58 A/U	Unshielded Twisted Pair	
Segment Length	500 Meters	185 Meters	100 Meters	250 Meters
Network Span	5 X 500 Meters	5 X 185 Meters	3 Cascades @ 100M	3 Cascades @ 250M
Stations Per Segment	100	30	12 Per Hub	11 Per Hub
Transceiver Spacing	2.5 Meters	.5 Meters	N/A	
Media Access	Transceiver / AUI cable	BNC"T" / Thin MAU	RJ 45 Type Modular Connector	
Stations Per Network	1024			

Figure 9

Now that we have covered the basics, we can get back to connecting our 950 computer.

As we indicated early, the interconnection to the 950 is via the DTC which is actually connected to the LAN. The output of the DTC is the familiar ATP.

What a relief, at least some things have not changed.

Figure 10 gives you a typical look at the connections needed for the 900 series machines.

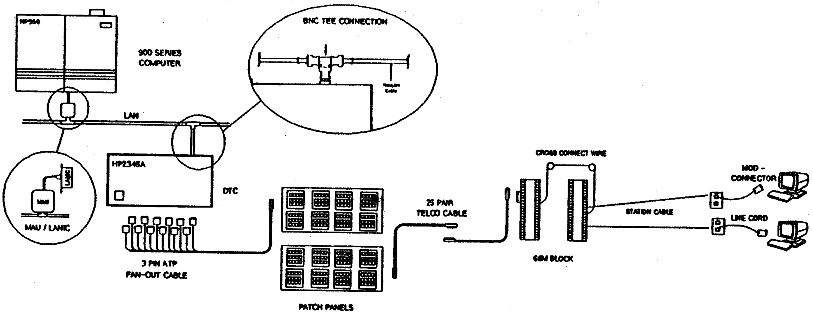


Figure 10

Pretty simple, as you can see, just a few new parts and then everything else is the same as we have described so many times before.

We again use the familiar octopus cables, patch panels, and of course at the end of all this the RJ11/modular adapter.

*** LAN/WAN CONNECTIONS**

It is now time to demonstrate some of the LAN/WAN connections and networks.

+ SYSTEM LAN = THINLAN

We refer to the LAN which is the basic 'backbone' connecting our computers, DTCs, and bridges as the System LAN.

In our network we have elected to use ThinLAN cable which offers us the greatest flexibility and the lowest overall connection cost.

Figure 11 depicts a typical network using the ThinLAN cable/connection techniques:

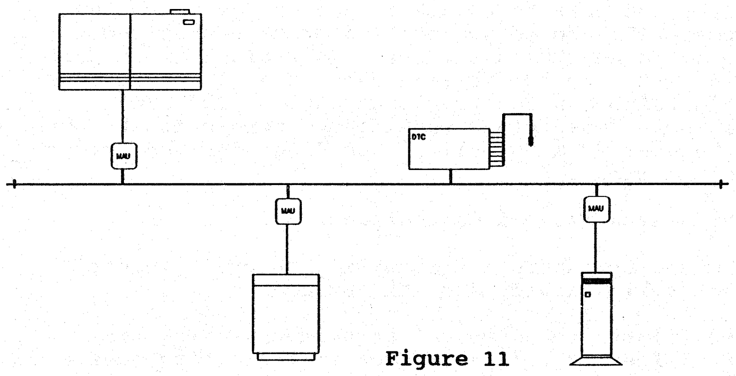


Figure 11

+ Starlan = PC LAN = Office LAN

The StarLan (subnetwork) consists of personal computer (PC) users in a common location, sharing resources (programs, files, printers, etc.). These subnetworks also have access to other resources/computers attached to the overall System LAN/WAN.
Figure 12 illustrates our typical StarLAN subnetwork configuration:

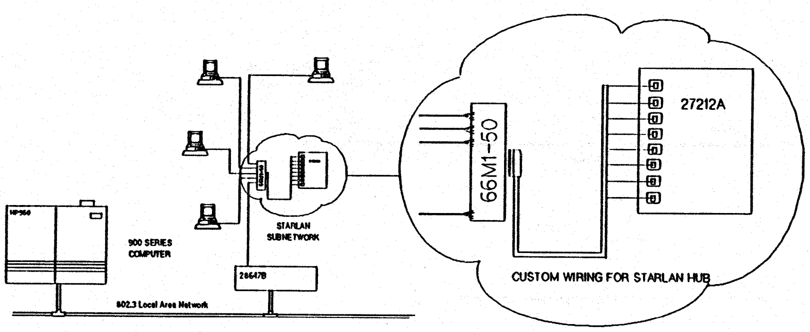


Figure 12

We use a 10:1 (10Mbps/1Mbps) bridge to provide isolation between the LAN and the subnetwork traffic as well as provide medium conversion (twisted pair to thinLAN). The bridge output and user PC's are connected to a preconnectorized, quick-connect block (M66). This block is in turn connected to the Starlan Hub via a custom RJ45/50 pin telco connector/octopus cable. The RJ45 octopus cable plugs provide the connection to both the user PC's as well as one 'leg' for the bridge connection.

+ **Wide Area Network Connection**

Remote computers are network connected (in our network) via wide area MAC bridges.

We illustrate a typical arrangement using two Wellfleet LN series bridges communicating over a telco 56Kbps DDS line. (Figure 13)

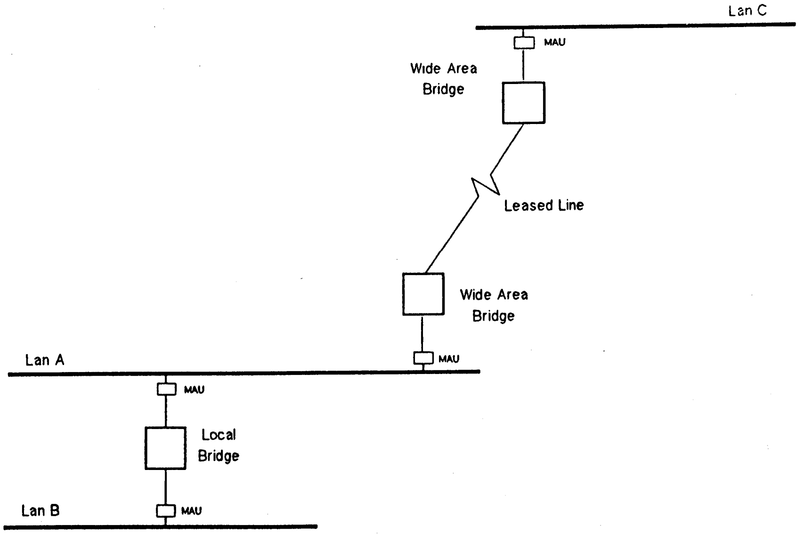


Figure 13

The bridges are connected to the respective LAN's using BNC connectors/Medium Access Units (MAU).

Once in place, the bridges are transparent to the local area networks.

+ THE COMPOSITE NETWORK

We can now take all of the pieces and interconnect them into a "Composite Network' or CATENET.

In Figure 14 we take the System LAN(s), StarLANs, and connect them via a Wide Area Network (WAN) bridges.

This network now uses all of the many connections from RS232 to LAN..........

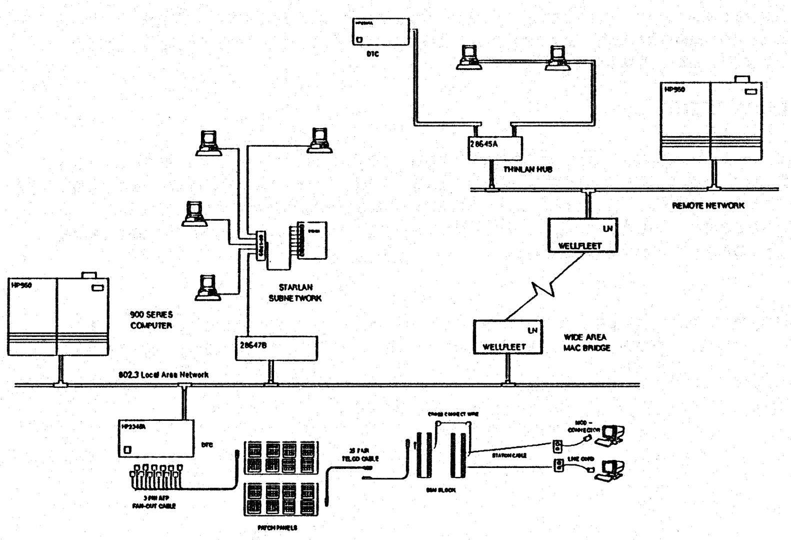


Figure 14

*** PUTTING IT TOGETHER AND KEEPING IT RUNNING**

Well, we have told you a lot about connecting things to the HP3000 computer. In concluding, we want to leave you a list of items that we found to be very helpful in 'putting it all together and keeping it running'.

Tools

Punch down tool for M66 quick-connect blocks, RJ11/RJ45 crimping tool, wire strippers

Test Equipment

LAN Scanner(3COM), voltohm meter, audio tone generator, telephone buttset

Trouble Shooting

LAN analyzer(Sniffer), loopback connectors, LAN (AUI) inline monitor, breakout box, protocol analyzer, bit error rate tester

*** CONCLUSION**

As you can see, the connection techniques have made many changes over the past 10 years and are now entering the LAN era. But, no matter where the new connection technology leads us, we always seem to need some of the 'old' basic connections.

We hope this brief presentation of our techniques will help you in your current and future HP3000 CONNECTIONS.

HP StarLAN/IBM Token Ring Connectivity

Charly Said-Nejad
Indata, Inc.
6059 Bristol Parkway
Culver City, California 90230
(213) 417-8289

1. Introduction

The development and implementation of OSI standards promises to make new and expanding networks easier and less expensive to operate in multi-vendor environments. Today, many major vendors have shown their support for OSI by implementing their standards and providing gateway connections. This paper focuses on the interconnectivity between IEEE 802.3 and IEEE 802.5 LANs. Since the HP Starlan and the IBM Token Ring Networks are two major implementations of the above, all the discussion in this paper will be directly applicable to both. Except for these network products from HP and IBM, and the corresponding network operating systems, no other network products will be discussed. The aim of this paper is to explore the interconnectivity issues in a general rather than a vendor specific way where various products could be applied. The discussion will be limited to the physical, data link, and network layers of the OSI reference model which is sufficient for network interconnection.

Furthermore, Wide Area Networks (WANS) and X.25 connections will not be discussed. The main focus will be on interconnectivity of networks in an office building. Before addressing interconnectivity issues, the lower three levels of the OSI model will be discussed. Next, the IEEE 802.3 and IEEE 802.5 protocols will be touched upon. Finally, the various methods of connecting networks will be discussed.

2. An overview of the physical, data link and network layers

The physical layer deals with the nature of the transmission medium, electrical signaling, and device attachment. The data link layer provides a well defined service to the network layer, determining how the bits of the physical layer are grouped into frames, dealing with transmission errors, regulating the flow of frames so that slow receivers are not swamped by fast senders, and general link management. The data link layer consists of Media Access Control (MAC) and the Logical Link Control (LLC).

The MAC sublayer is the lower part of the data link layer and deals with methods for allowing a particular node to transmit on the data transmission channel available to it. OSI has left open the implementation of the MAC sublayer and therefore, three standards have emerged. This paper deals with the IEEE 802.3 and IEEE 802.5 standards. Above the MAC sublayer is the Logical Link Control (LLC), which is architecture independent and can be used with each of the physical LAN implementations. LLC receives services from the MAC sublayer. These services allow the local LLC sublayer entity to exchange LLC data with peer sublayer entities. This provides a means to successfully transfer data and/or control information from one machine to another.

Finally, the network layer is concerned with getting packets from the source all the way to the destination. This function may require making many hops at intermediate nodes along the way. It is at this level that two networks first establish a connection.

3. The IEEE 802.3 MAC Sublayer Protocol

The 802.3 frames structure as shown below is made up of series of bits. Each frame starts with a preamble of 7 bytes

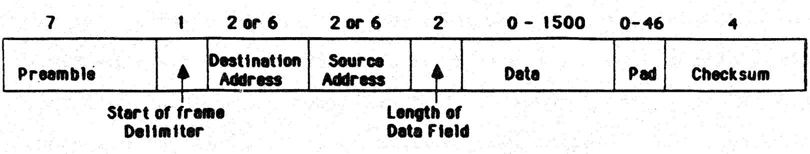


Figure 1: The 802.3 frame format

followed by the start of frame byte to denote the start of the frame itself. Then comes the destination and source addresses. The length of the data and the data itself follows next. The maximum size of data is 1500 bytes. After a series of null bytes there is a checksum field. It is here that a checksum algorithm is applied to detect errors. The framing, addressing and error detection is part of the services performed by the data link layer. Another service is media access management which involves collision avoidance and collision handling. The following is a brief summary of this process.

The IEEE 802.3 standard utilizes baseband technology, where data can only be transmitted to one node at a time. The 802.3 protocol is known as Carrier Sense, Multiple Access System with Collision Detection (CSMA/CD). This simply means that if a node wishes to transmit a data packet, it must first listen (Carrier Sense) to insure that no data is currently being passed over the network. If the network is available, the data will be broadcast. If the transmission was clean, that is, no collision, the node will again listen to the network for traffic. If two nodes attempt to transmit data across the LAN at the same time, then a collision occurs. When collision is detected, the node detecting the collision will issue a "jam" broadcast to all nodes on the LAN and a collision recovery routine is executed and the node will wait a random amount of time before re-transmitting its data.

4. The IEEE 802.5 MAC Sublayer Protocol

The basic operation of the MAC protocol is straightforward. When there is no traffic on the ring, a 3 byte token circulates endlessly, waiting for a station to seize it. When the token is captured, the station modifies it to a start of frame sequence and appends additional fields making it a complete frame.

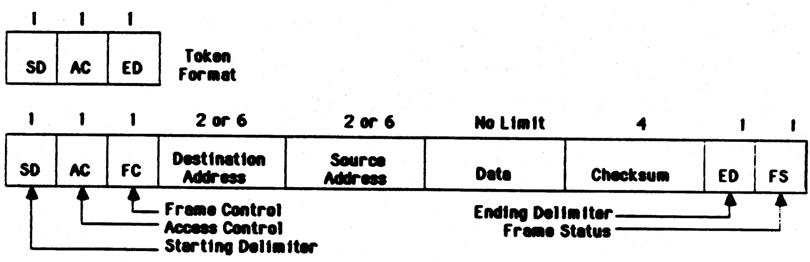


Figure 2: The token frame and the 802.5 frame format

When the information transfer is complete, the sending station generates a new token. The frame format is more complex than the IEEE 802.3 frame. The SD field indicates the start of frame. Following SD is the Access Control (AC) field which provides information necessary to allow access to the medium. The Frame Control (FC) byte follows which distinguishes data frames from various control frames. Next comes the Destination and Source Addresses (DA, SA), followed by data which can be as many bytes as a particular implementation allows. The Checksum field (CS) and Ending Delimiter (ED) fields follow. Finally, there is a Frame Status (FS) field. The basic function of this field is to provide automatic acknowledgment for each frame and increase reliability as much as possible.

5. Internetworking Issues

In this discussion we shall assume that a collection of IEEE 802.3 networks (e.g., HP StarLAN, star topology) need to communicate among themselves and with IEEE 802.5 networks (e.g., IBM Token Ring), and vice versa. A coaxial backbone is assumed connecting the floors of a building and various host computers. Also we assume that the backbone and the StarLAN networks run under NS3000/V link software which is the network operating system. The token ring drive and the IBM PC LAN programs are running on the Token Ring network. Interconnecting the various departmental networks to the backbone is the objective which we shall address.

In the above environment there are four possible internetwork communication methods: StarLAN to StarLAN, StarLAN to Token Ring, Token Ring to Token Ring, and Token Ring to StarLAN.

Several problems become apparent. The first problem is that the IEEE 802.3 networks and the IEEE 802.5 networks use different frame formats at the MAC sublayer. As a result, any copying between different LANs require reformatting, which takes CPU time, requires new checksum calculation and introduces the possibility of undetected errors due to bad bits in the bridges memory. A second problem is that interconnected LANs do not necessarily run at the same data rate. Too many packets may be sent to a single address. This situation is called flow control. The HP StarLAN has 1 Mbps and 10 Mbps implementations. The

IBM Token Ring network runs at 4 Mbps. Therefore, when forwarding back to back frames from the 10 Mbps LAN to the 4 Mbps LAN, the connectivity black box will not be able to get rid of the frames as fast as they come in. Frames will have to be buffered and stored in memory. The amount of memory then becomes an issue. A third problem in LAN connectivity is that the maximum frame length is different between the two network types. The general boundaries are 1500 bytes for IEEE 802.3 and 5000 bytes for the Token Ring networks. The following section will explore ways to connect the networks, given these constraints.

6. Homogeneous Internetworking Solutions

First, we shall consider IEEE 802.3 to IEEE 802.3 connectivity. An HP StarLAN 10 Mbps transparent bridge can easily accomplish this. 1 Mbps and 10 Mbps traffic can be handled using such a bridge assuming we have both implementations of StarLAN in place. The various StarLAN networks may either be connected to the backbone, or using a tree structure, each network has a parent-child relationship to networks below. The master network can be connected to the backbone using a bridge and the subnetworks can also connect to the master network using a bridge. When a workstation in network needs to communicate with another workstation in another network, the bridge will determine which subnetwork the data belongs to. Therefore, it provides address filtering, which causes traffic which is not intended for a subnetwork to by-pass the subnetwork. Also, as traffic increases, it is advantageous to break the network into two or more segments linked by bridges, thereby, reducing the number of devices on each segment. This will in effect reduce collisions which in turn will increase the throughput on the system. Another benefit of breaking up the network is to exceed network limitations such as maximum distance and number of devices supported. The frames will broadcast to all the stations in the intended subnetwork. Various algorithms have been designed (spanning tree) to determine and store the unique paths between LANs.

From IEEE 802.5 to IEEE 802.5 no special problems exist. IBM bridges can interconnect these IBM Token Ring LANs together. The major difference between these bridges is that they are source routing bridges. The algorithm used to determine the path between the 2 networks is significantly different and is outside the scope of this paper. One advantage that Token Ring to Token Ring connections enjoy is that they can support routing bridges. Therefore, 2 parallel Token Ring networks can use two bridges to communicate. The load will thus be split among the two bridges resulting in increased throughput. IEEE 802.3 connections can not make use of multiple bridges. It is also important for the bridges in the above two communications mode to manage flow control, as explained in the previous section.

The situation becomes more complex when trying to interconnect IEEE 802.3 to IEEE 802.5 and vice-versa. What is needed is a connection black box able to handle the 2 different frame formats and sizes, different communication protocols, and different data rate. To resolve this issue, we need to understand what are routers and how they differ from bridges.

7. Routers, Bridges, and Non Homogeneous Connections

While routers and bridges can connect distinct LANs, increase LAN capacity and LAN distance, and provide network traffic isolation (Packets not belonging to a subnetwork are not passed to that sub-network), they have a fundamental difference. Bridges copy packets between adjacent networks using same network protocol and operate at the MAC level. They do not interfere with network layer protocols and are therefore, transparent to the distinct workstations. In addition, bridges interconnect LANs with a uniform address domain, and therefore, no address conversion is required and in the majority of cases, they connect homogeneous LANs (i.e., same MAC protocol).

In contrast, routers are not transparent to the user protocols. They implement the network protocol (e.g., IP, DECNET, XNS, NS3000, etc.) and, thus, have peer counterparts in other routers as well as in user workstations and hosts. They terminate the MAC and LLC layers of each connected LAN and permit translation between different address domains. Because of the higher protocol different processing overhead, the router generally delivers lower throughput than the bridge. It provides, however, more efficient routing and flow control than a bridge, since it operates at the network level and can exploit the traffic management procedures that are part of that layer. In addition, the source and destination machine may not be adjacent and belong to two distinct networks. Furthermore, since routers operate at the network layer, they have the capability to fragment data packets received from the source machine and reassemble at destination.

In terms of IEEE 802.3 to IEEE 802.5 and IEEE 802.5 to IEEE 802.3 connectivity, bridges can not be used since the StarLAN implementation, as all IEEE 802.3 implementations do not handle source routing bridges, which are used by IEEE 802.5 In addition, IEEE 802.5 networks do not use transparent bridges used by IEEE 802.3 Therefore, what is needed is a router which can understand the specific network protocols on either side. Given that capability, we have solved the problem of IEEE 802.3 to IEEE 802.5 connectivity. But how about IEEE 802.5 to IEEE 802.3? In this case we are dealing with IEEE 802.5 frame size of about 5000 bytes needing to transfer to a network allowing 1500 maximum frame size. The router can take care of fragmentation and assembly. The data packet will have to be split into several packets and reassembled at the other end. The problem with bridges is that they only have access to the source and destination of the packets and can not look inside the data to determine its size. If they can not handle the extra bytes they will simply drop them from their transmission.

In practice, the topology of the interconnected networks will play an important role in determining when to use routers or bridges. In many cases, both options are feasible and performance and cost considerations determine what equipment to choose. In other cases, routers are required and the question may become how to minimize their use and still accomplish the interconnectivity. The inter-network configuration and topology will provide some answers. But in general, the guidelines in this paper can help in deciding what equipment is needed for a given environment.

8. Summary

This paper attempted to provide some general guidelines for IEEE 802.3 and IEEE 802.5 connectivity by explaining is some detail the physical, data link, and network layers of the OSI reference model, and the MAC sublayer protocols used by IEEE 802.3 and IEEE 802.5 networks. Next it applied these concepts to network interconnectivity and a discussion was held regarding the merits of bridges and routers and how they can accomplish the connectivity.

PCs, HP 3000s, & LANs: The Virtual Terminal Connection

Randy Robinson

Walker Richer & Quinn, Inc.
P.O. Box 31876, Seattle, WA 98103-1876 U.S.A.
Telephone: (206) 324-0350 TELEX #: 311 743 WRQ UR
FAX: (206) 329-7565

The first step in integrating PCs and minicomputers is making the data communications connection. Local area networks are the medium of choice, offering the potential of high speed communications and connectivity to multiple hosts. There are a variety of popular LAN products available. Unfortunately, not all these products are compatible with one another. Integrating an HP 3000 with a PC LAN may not be straightforward.

The purpose of this paper is to illustrate the basic requirements for making a terminal connection from a PC LAN to an HP3000. To understand the virtual terminal connection requires knowledge of basic networking principles, as well as the use of gateways and communications servers.

The requirements for making a terminal connection with HP's OfficeShare and Novell NetWare will be discussed in some detail.

I. Basic Concepts

The commonly understood definition of a computer network is a collection of connected computers. Connected means capable of exchanging data.

Two types of networks are wide area networks and local area networks. Wide area networks connect computers which are physically distant from each other. An example of a wide area network is the ARPANET, which is the Department of Defense network. It links computers at government and university locations around the world. X.25 public data networks, such as Telenet and Tymnet, are wide area networks that provide links for commercial computers.

Local area networks (LANs) link computers that are physically close to each other, usually in the same building, and typically on the same floor. Examples of local area networks are OfficeShare from Hewlett-Packard and NetWare from Novell.

Networks are also distinguished by the communication media, or channels, which physically link the computers. There are two broad categories of links: point-to-point and broadcast. Point-to-point is a link consisting of nodes which are connected via numerous cables or leased telephone lines.

Any node on the network can only directly communicate with nodes which are physically connected to it. To communicate with a node not physically connected to it, it must pass through an intermediate node which is physically connected to it. For example, in figure 1, for node A to communicate with node C, the message must pass through node D. (The message can also pass through B, then F, to get to C.)

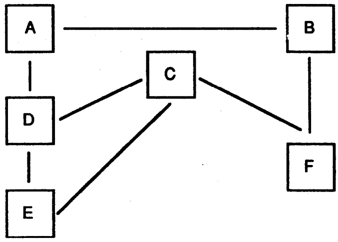


Figure 1 A Simple Point-to-Point Network.

Broadcast is a link in which all the nodes share a single communication channel. In a broadcast network, every node can communicate directly with every other node. LANs are typically broadcast networks.

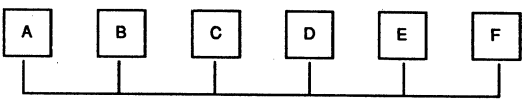


Figure 2 A Simple Broadcast Network

II. The ISO Model

Computer networks are designed in a very structured way. The International Standards Organization (ISO) has a model for this structure. This model, the Reference Model of Open Systems Interconnection (OSI), has seven layers. Each layer is a separate entity. For example, layer 2 on one computer carries on a conversation with layer 2 on another computer. The rules and conventions used in this conversation are collectively known as the layer 2 protocol. The layers on each computer must also communicate with the layers above and below it on the same computer. The mechanism to do this is called an interface.

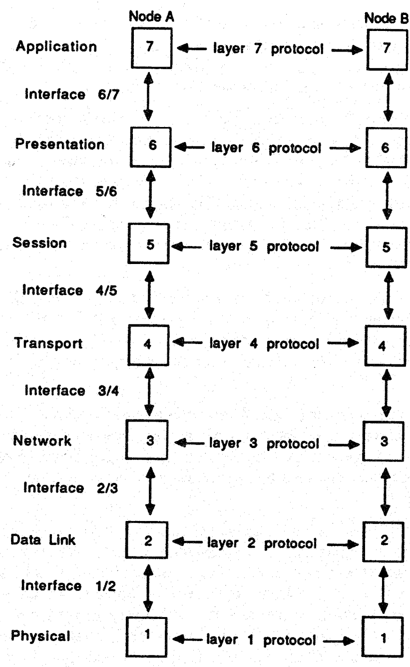


Figure 3 The ISO Model

Figure 3 illustrates the ISO model. Nodes A and B are the source and destination nodes exchanging a message. No data is actually transferred from layer n on one machine to layer n on the other machine. On the machine which is sending the message, each layer passes data and control information to the layer immediately below it until the lowest layer is reached. This layer transfers the data to the receiving machine. On the receiving machine, each layer passes data to the layer immediately above it until the highest layer is reached. For the conversation to be successful, each layer must use the protocol for its layer and must adhere to the interfaces used between layers.

Each layer provides services to the layer above it. A brief description of the services provided by each layer follows:

Physical This layer transmits the raw bits over the communication channel.

Data Link The data link layer takes a raw transmission facility (the physical layer), and turns it into a line that appears error-free to the network layer. This means that the data link layer must detect any transmission errors and either correct or re-transmit the data. This typically involves packet sequencing and CRC calculations.

Network This layer routes the messages from the source node to the destination node. As an illustration, consider figure 1. If node D wishes to send a message to node B, it must know what route to take. One possible route is to send the message to node C with an instruction to forward the message to B. An alternative would be to send the message to node A with a similar instruction. One function of the network layer is to know about the speed of the different routes. In this example, sending the message to node B requires only 1 intermediate node, whereas sending it to node C requires 2 (nodes C and F).

Transport The transport layer maintains connections between source and destination nodes. A connection is the data path between processes. A node may support multiple processes, each of which may have a connection to processes on different nodes. It is up to the transport layer to route the data from each connection to the correct process.

Session The session layer is the connection manager. A presentation layer process negotiates with the session layer to establish a session with a remote presentation layer process. An example is providing a logon, including password, so that files may be transferred from an account (directory) on one node to an account on another node.

Presentation The presentation layer performs various data manipulation functions, such as data compression and encryption. File transfer and virtual terminal protocols are usually considered presentation services. One example of a virtual terminal protocol function is echo. When a character is typed at the keyboard, that character is usually 'echoed' on the screen. In the RS-232 environment, the host computer usually generates the echo. In a LAN environment, the virtual terminal layers on the local and host nodes typically negotiate whether the echo is generated locally or by the host.

Application The content of the application layer is up to the individual user processes. An example of application processes is terminal emulation on a PC and an interactive (terminal oriented) application running on a mini computer.

The ISO model is just that, a model. None of the commonly used networking protocols exactly follows the model. (If you believe the advertising literature, they all do.) The distinctions between layers 5, 6, and 7 are especially difficult to determine in most networks.

In the best of all worlds, protocols at the same layer would all be interchangeable. In other words, a network could be built by deciding which protocol to use at each layer independently of the choice made at any other layer. In practice, this is not true.

Typically, the physical and data link layers are coupled. One exception to this is Ethernet and 802.3. At one time, Ethernet and 802.3 had different specifications for the physical layer. Currently, they are the same. At the data link level, 802.3 and Ethernet are different protocols. An 802.3 node can not converse with an Ethernet node. However, because 802.3 and Ethernet share the same physical layer, 802.3 and Ethernet nodes can be on the same network. A node that can understand both 802.3 and Ethernet can converse with any node on the network.

Network and transport layer protocols are also usually coupled. A good example of this is TCP/IP. These are protocols defined by the Defense Advanced Research Projects Agency (ARPA), and are always used together on the same network.

Presentation layer protocols are often coupled with network/transport protocols, as in ARPA, but this is not necessarily the case.

In any network, there may be several presentation layer protocols. A good example is ARPA, which has a virtual terminal protocol, Telnet, and a file transfer protocol, FTP, at the presentation level (as well as others).

While there are agreed upon definitions of the various network layers, the interfaces between layers are not as well defined. Specific interfaces will be discussed in section IV.

III. Specific Protocols

As was discussed earlier, none of the common networking protocols exactly adheres to the ISO model. Therefore the representation of the of the layers of the following networks should be considered approximations.

The following charts show the protocols used by ARPA, NS/3000 (the networking software for the HP 3000), OfficeShare and NetWare. For the presentation layer, only virtual terminal protocols are listed.

Table 1.1
ARPA Protocols

Layer	Protocol	Standard
Presentation	Telnet	ARPA
Session	None	
Transport	TCP	ARPA
Network	IP	ARPA
Data Link	Various	
Physical	Various	

By and large, network vendors who have implemented ARPA protocols for LANs have used Ethernet for the physical and data link layers. Many vendors offer alternative physical/data link protocols, but almost all provide Ethernet.

Table 1.2
NS/3000 Protocols

Layer	Protocol	Standard
Presentation	NS/VT	HP proprietary
	Telnet*!	ARPA
Session	None	
Transport	TCP	ARPA
Network	IP	ARPA
Data Link	802.3	IEEE
	Ethernet*	DEC/INTEL/XEROX
Physical	802.3/Ethernet	IEEE

*Not available on MPE/XL (Spectrum) machines
!Available from Wollongong

The datalink layer is 802.3. On classic HP 3000s running MPE V-Delta-5, Ethernet is also available. Ethernet is not available on MPE/XL systems.

The network layer is IP, which is an ARPA standard.

The transport layer is TCP, also an ARPA standard.

There is no session layer that is distinct from the transport. Again, this conforms to the ARPA standard.

At the presentation level, HP uses a proprietary protocol, called Network Services Virtual Terminal (NS/VT). A third party vendor, Wollongong, offers Telnet, the ARPA standard, for classic HP 3000s.

Table 1.3
OfficeShare Protocols

Layer	Protocol	Standard
Presentation	NS/VT	HP proprietary
	Telnet	ARPA
Session	None	
Transport	TCP	ARPA
Network	IP	ARPA
Data Link	802.3	IEEE
Physical	802.3/Ethernet	IEEE

OfficeShare for the PC uses the same protocols as the HP 3000, except that Ethernet is not available as a data link protocol.

Table 1.4
Novell NetWare Protocols

Layer	Protocol	Standard
Presentation	Novell Proprietary	Novell Proprietary
Session	None	
Transport	IPX	Novell proprietary
Network	IPX	Novell proprietary
Data Link	Various	
Physical	Various	

Because of the commercial success of Novell NetWare, most PC LAN card vendors provide Novell network software with their cards. This means that NetWare will run with all of the common physical and data link protocols that are available for the PC.

IPX is the protocol that provides network layer and transport layer services. This is a proprietary protocol.

Novell NetWare uses a proprietary presentation layer virtual terminal protocol. This protocol is limited in that it is only used with an asynchronous communications server (see section VI).

IV. Specific Interfaces

There are 3 interfaces which are of special interest. Referring to figure 3, these are interfaces 2/3, 5/6, and 6/7. For purposes of this discussion, the 2/3 interface will be called the data link interface. Because the session layer is generally not implemented independently of the transport layer, the 5/6 interface will be referred to as the transport interface. The 6/7 interface will be referred to as the virtual terminal interface.

On the HP 3000, the interface to the data link layer is not published.

On the PC, there are currently no common interfaces to the data link layer, although two have recently been proposed. Microsoft and 3Com have developed Network Device Interface Specifications (NDIS). Novell has announced the Open Data-Link Interface (OLI). The intent of both these interfaces is to allow multiple sets of PC networking software to co-exist and use the same LAN card. Because there is no standard way in DOS to share hardware resources, multiple data link drivers cannot share the same LAN card. However, a single data link layer could service more than one network layer. There are some networks vendors who provide this capability using their own interfaces. Examples are Ungermann-Bass and Excelan. Both of these vendors provide the ability to simultaneously run multiple network layers on the PC.

A standard interface to the ARPA transport layer (interface 5/6) is known as Berkley sockets. This standard was developed in the Unix environment, specifically Berkley Unix. The HP 3000 has a set of intrinsics which roughly correspond to Berkley sockets, but not exactly. OfficeShare has NetIPC, which roughly corresponds to Berkley sockets, but not exactly.

Most PC networks that use TCP/IP provide a socket interface. Novell NetWare does not have a socket interface to its transport layer.

On the PC, the most common interface to the transport layer is NetBios. Most PC networks have NetBios support, including OfficeShare and NetWare.

Another PC interface to the transport layer is the MS-Net Transport Layer Interface. This is very similar to NetBios. OfficeShare has an MS-Net interface.

Novell has a proprietary interface to the transport layer. For purposes of this paper, this interface will be termed Novell's Application Program Interface (API). Novell also has a NetBios interface.

On the HP 3000 the interface to the virtual terminal layer, whether it is NS/VT or Telnet, is through the terminal driver. An application (layer 7) program can read and write data to the virtual terminal protocol by doing FREADs and FWRITEs to the terminal.

On the PC, there are two common interfaces to virtual terminal protocols. These are through software interrupt 14 (hex) and interrupt 6B (hex). The interrupt 14 interface is the standard BIOS serial communications interface. Interrupt 6B is an interface developed by Ungermann-Bass which has been adopted by several network vendors.

OfficeShare has a proprietary interface to its virtual terminal protocol.

A terminal emulation program running on the PC can read and write data to the virtual terminal protocol by adhering to the interface that the particular protocol supports.

V. The Terminal Connection

Local area networks are becoming a popular means of connecting computers together, and as a means of connecting terminals (and PCs with terminal emulation software) to host computers. This paper is limited to a discussion of the problems involved with connecting PCs with terminal emulation software to HP 3000s.

The HP3000 provides three types of terminal connections. These are RS-232 (using ATPs on 'classic' 3000s and DTCs on XL machines), X.25 virtual terminal (classic 3000s only), and LAN virtual terminal. Any of these connections may be used to connect a PC on a LAN to an HP 3000.

VI. The Asynchronous Connection

An asynchronous communications server is used to connect a PC to an HP3000 via an RS-232 connection. (An asynchronous communications server is sometimes called a terminal server.) An asynchronous server is a device with a LAN connection on one side, and serial ports on the other side. It can be a PC or a specialized 'box'. The serial ports can connect to any asynchronous device such as a modem, a terminal, or a host computer.

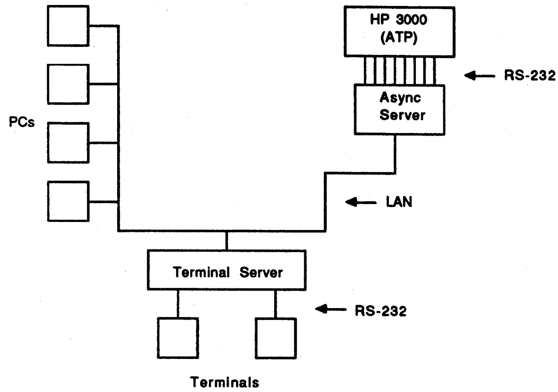


Figure 4 LAN Configuration Using Asynchronous Communications Servers

In this configuration, the HP 3000 performs standard RS-232 terminal I/O. There is no multiplexing of the RS-232 line which takes place. This means that for every PC which is to have concurrent access to the HP 3000, there must be a separate RS-232 connection between the asynchronous server and the HP 3000.

The asynchronous server must perform the same layer 1-4 protocols as the rest of the nodes on the network. It further has to perform a presentation layer virtual terminal protocol. The PC must also perform this same virtual terminal protocol. There is a client-server relationship between the PC and the server. The PC typically requests a connection to a specific RS-232 port on the server, and then data flows between the PC and the HP 3000.

The terminal emulator must use the presentation interface that the PC virtual terminal protocol is using.

A specific example of how this works is the Novell Asynchronous Communications Server (NACS). The server consists of a PC with special serial hardware which allows up to 16 RS-232 ports to be installed in the PC. The server software then runs on the PC. The server software performs a proprietary virtual terminal protocol. It uses Novell's API to communicate with the IPX protocol.

On the workstation PC, the virtual terminal protocol is performed by the Novell Asynchronous Server Interface (NASI) program. This program uses Novell's API to communicate with the IPX protocol. A terminal emulator uses the Interrupt 6BH interface to communicate with NASI.

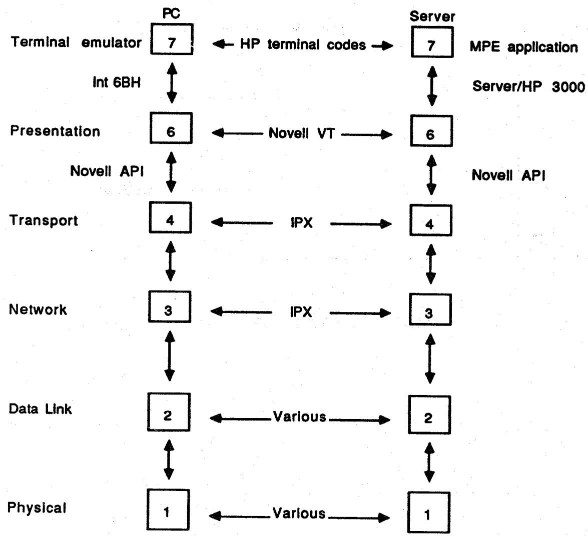


Figure 5 Protocol Stacks for Novell Asynchronous Communications Server

Figure 5 is a chart of the protocols used with the NACS. In this situation there is no session layer. On the PC, there are typically 3 programs which must run. These are IPX.COM, which performs the data link, network, and transport protocols (the physical protocol is performed by the network card), NASI.EXE, which performs the virtual terminal protocol, and the terminal emulator. On the server there are two programs, IPX.COM and NACS.EXE.

VII. The X.25 Connection

The X.25 server works much like the asynchronous server, except that the connection between the server and the HP 3000 is an X.25 link. In this case, the HP 3000 uses an INP.

VII. The LAN Virtual Terminal Connection with OfficeShare

Making a LAN virtual terminal connection with OfficeShare is fairly straightforward. Figure 6 is an illustration of the LAN configuration. The HP 3000 must be equipped with a LANIC, and must be running NS/3000.

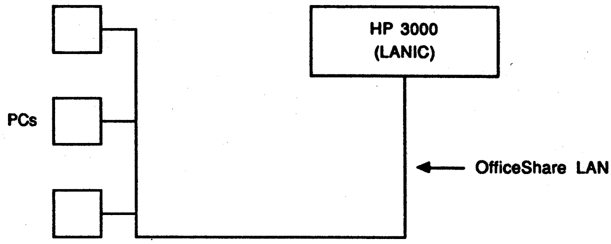


Figure 6 LAN Configuration Using OfficeShare

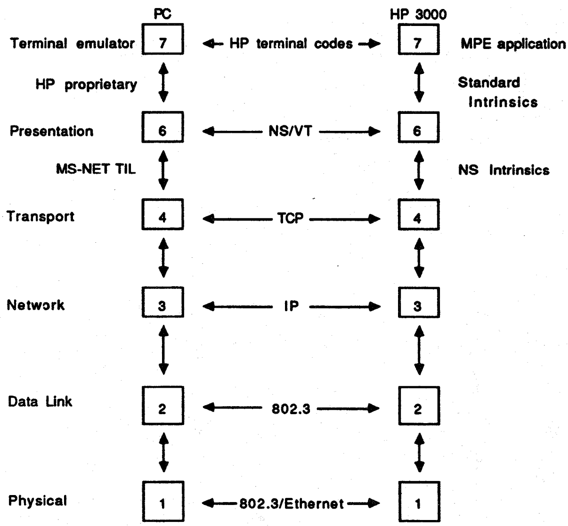


Figure 7 Protocol Stacks for OfficeShare

Figure 7 illustrates the protocols used in this configuration.

IX. The LAN Virtual Terminal Connection with Novell NetWare

Making a LAN virtual terminal connection with NetWare is not straightforward. The problem is that the protocols used by NetWare are not the same as those used by the HP 3000. A gateway is necessary to make the connection. A gateway is a network node which can understand different protocols, and convert from one to the other. HP has a gateway product called NetWare Link. Figure 8 illustrates the network configuration using a gateway.

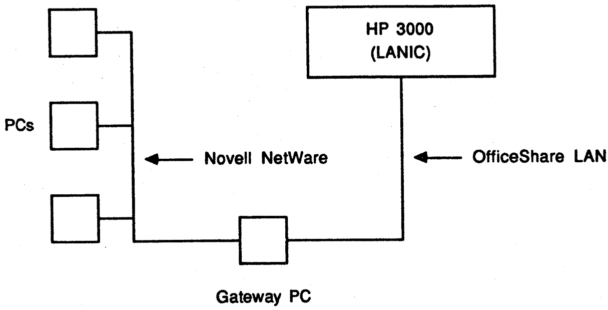


Figure 8 LAN Configuration Using NetWare Link

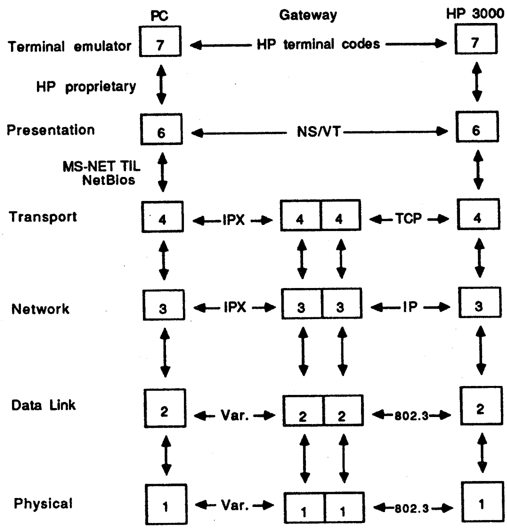


Figure 9 Protocol Stacks for NetWare Link

The gateway runs OfficeShare and NetWare simultaneously, and has a program which bridges the two. Figure 9 illustrates the protocols used in this configuration. The gateway is performing a conversion at four layers, including the physical. This means that the gateway must have two LAN cards, one for NetWare and one for OfficeShare. Even if NetWare is running on a 3Com 3C501 card, which is the same card used for OfficeShare, two cards are necessary. This is because there is no defined data link interface which would allow separate network/transport protocols to share the same card (see section IV).

The transport interface on the PC is a little unusual. The program which performs the virtual terminal protocol is a DOS device driver (VT.SYS). This program uses the MS-Net TIL interface to communicate with the transport layer. HP has written a program which traps these calls, and converts them to NetBios calls. The gateway does the reverse, converting NetBios calls into MS-Net TIL calls. This means that VT.SYS did not have to be modified to work in this environment.

X. The Telnet Option

Telnet is the ARPA standard for virtual terminal. Many network vendors, (including HP in the case of the HP 9000) offer Telnet/TCP/IP products. In general, the data link/physical protocol used by these vendors is Ethernet. This set of protocols is available on the 'classic' HP 3000s, but not on MPE/XL machines. Ethernet support is included in MPE V-Delta-5. Telnet support is available through a third party, Wollongong.

Wollongong has not previously offered any HP 3000 products. The Telnet product required some joint development with HP. In the early versions of this product, there have been bugs, some of which have been traced to Wollongong, and some of which have been traced to HP. The performance of this product has not been as good as the performance of NS/VT with OfficeShare.

The major advantage of using Telnet is the connectivity options it provides. Telnet/TCP/IP/Ethernet has emerged, over the last two or three years, as a standard for much of the LAN industry. Most Unix systems that have LAN support use these protocols. Many companies offer PC products that use these protocols. There are also several asynchronous (terminal) servers on the market. These protocols offer the most connectivity options across the broadest range of computers.

Unfortunately, connectivity between Novell NetWare and Telnet hosts requires either gateways or data link layers which support multiple network protocols. Interlan (formerly Micom-Interlan), has a NetWare - Telnet/TCP/IP gateway which operates much like HP's NetWare Link. There are several PC LAN vendors who provide the ability to run NetWare and Telnet/TCP/IP protocols simultaneously on a PC. These vendors include Ungermann-Bass, Excelan, and Wollongong. In all these cases, specific PC LAN hardware is required. This means that existing Novell Networks that do not use any of these LAN cards must use a gateway to connect to a Telnet host.

XI. Summary

When integrating an HP 3000 into a single LAN, a network manager or system manager must consider the type of terminal connection between the PCs and the HP 3000, the terminal connection to other hosts that might be on the network, and the type of PC LAN software.

If HP 3000s are the only host computers on the network, and OfficeShare is the PC LAN software, then the integration is simple and straightforward.

If the network environment includes hosts from non-HP vendors, and those hosts have Ethernet rather than 802.3 data link protocols, or if the LAN software is from Novell or another non-HP vendor, the integration becomes complex. In this situation, compromises must be made concerning cost and the complexity of the terminal connection.

The networking trend is toward the ARPA protocols. This trend is bound to continue, and plans to integrate PCs and HP 3000s should take this into account. Unfortunately, the most popular PC LAN software, Novell's Netware, does not support the ARPA protocols.

To help simplify the integration, HP should offer Ethernet and Telnet options for MPE/XL (Spectrum) systems, and work with Wollongong to improve the Telnet option for MPE systems.

New Communication Strategies for Integrating PCs with Mainframes

Steve Brockway

Walker Richer & Quinn, Inc.
P.O. Box 31876, Seattle, WA 98103-1876 U.S.A.
Telephone: (206) 324-0350 TELEX #: 311 743 WRQ UR
FAX: (206) 329-7565

PCs have gained overwhelming acceptance in the workplace. They provide excellent response time and generally have a high quality user interface in both graphics and text modes.

The host (an HP 3000 or VAX), on the other hand, is a multi-user system that has developed over time into a high-performance database engine. The number of applications and tools available on the host make it the natural machine for management of valuable corporate information.

If you are an application or tool developer, how can you harness the power and flexibility of the PC interface while maintaining the efficiency and integrity of host-based data?

The purpose of this paper is to identify and discuss the issues facing application and software tool developers as they begin to use the PC as a presentation tool for their host applications.

Concepts

This discussion of PC-to-Host integration will cover a number of topics you are familiar with as developers.

Many of the topics - such as user interface and security - are ones you have dealt with before. However, the pervasive use of PCs is forcing most of you to change your views of even the most well understood concepts.

What are the PC-to-Host integration concepts and how can you best incorporate them into your new products?

1. The Host User Interface: Is the PC Better?
2. The Host as a Database Engine
3. The Communication Link
4. Determining the PC/Host Work Breakdown
5. Security

1. The Host User Interface: Is the PC Better?

An experienced PC user who is asked to move to a minicomputer-based package, such as Visicalc/3000 or HP-Slate, is inevitably going to be disappointed - not necessarily because of the capabilities of the software, but because the user interface is noticeably weaker.

Why is this the case? PC users are accustomed to pop-up windows, inverse-video selection bars and pull-down menus. In graphics mode, modern user interface concepts such as true WYSIWYG (What You See Is What You Get) word processing is possible.

PC User Interface Advantages

Most of you are familiar with VPLUS/3000, HP's block mode forms package. VPLUS is good at what it does: it provides a way to enter and edit data using the terminal exclusively, and sends data to the host only when a screen is complete. This technique transfers some of the processing from the host to the remote device, a desirable goal. There are a number of advantages to this approach that can be further enhanced by using the PC.

- The ability to respond to data items as soon as they are entered allows for better editing of data. The PC application programmer can include routines for data validation and reformatting that give the user immediate feedback. Thus bad data items are reported at once, not after a full screen of information has been entered.
- The PC application can make prudent use of help windows. When a field is incorrectly filled out or the user asks for help, help windows can be displayed.
- As screens are designed, movement between fields can be determined by the specific data item entered. This eases the job of order entry by letting the computer make decisions for the user.

2. The Host as a Database Engine

Until recently, only mainframe-based operating and database systems have been realistic choices for DP departments. DBMSs (Database Management Systems) such as IMAGE have provided rollforward recovery, item-level locking, shadowing, and other advanced capabilities, while operating systems such as MPE have supplied a reliable platform for applications development. The idea of running a serious, "bread-and-butter" application using a LAN and Dbase III, for example, has not been greeted enthusiastically.

Today, however, there's a new class of high-powered, PC-based database machines. Examples include Microsoft's SQL Server, IBM's OS/2 Extended Edition, and Gupta Technologies' SQL Server.

These database engines execute on a dedicated PC (the database server) that is connected to a LAN. Programs running on the users' PCs then make database requests to the server. The database files are stored on the server's disk, and are shareable among all users on the LAN. The difference between these products and the previous generation of PC databases is that these systems, in combination with improvements in LAN operating systems, have some of the advanced features described above. For the first time, an organization can seriously consider moving its MIS applications to a totally PC-based environment.

While the PC database server/LAN approach is completely viable, there may be some reasons not to move in that direction.

- First, this technology is quite new, even by computer industry standards. Some of the products mentioned above are still, at this writing, in beta testing.
- Second, you probably have several MPE programmers on staff who may not have a great deal of experience with PC programming. (Remember, PC programming is not the same as knowing DOS commands and being able to run Lotus 1-2-3). Moving to a new DBMS such as SQL would entail retraining your MIS technical staff or hiring additional people.
- Third, a LAN does not have a batch system. For certain types of processing, such as report generation, payroll calculation and other non-interactive activities, it would be better to submit a batch job and forget about it. This cannot currently be done with the PC LAN server approach.
- Finally, you probably have a large base of existing MPE applications, using IMAGE or possibly KSAM, that would be incompatible with a new database standard. It would certainly be more convenient if your new PC application could share the same databases used by your current systems. For example, say that your accounting department is using MCBA software, and they want to add some new interactive functions that read data from MCBA's IMAGE database. If your new DBMS standard is Paradox, you will either have to cook up some data transfer scheme, or stay with the usual MPE program and its terminal interface.

As you can see, all this is leading to a proposal. You want to get the best of both worlds, the fast and productive PC user interface, plus the multi-user database and shared resources of your HP 3000. What you need is an arbitrator, a software layer that will take requests for data from the PC, give them to the minicomputer, and return the results to the PC. In the remainder of this paper we will discuss how this might be done using existing products.

3. The Communication Link

If PCs are to be the presentation tool and the centrally located mainframe is to be the database engine, the missing element that ties these two pieces together is the communication link. Typically, communication link discussions turn to topics such as Client/Server systems, APPC, LU 6.2, and peer-to-peer protocols. All are important topics, but they tend to complicate basic applied technology with too much theory.

The basic requirement for such a communication link is that it provide a *transparent, error-free exchange* of data between a program on the PC and a program on the host.

What does the transparent, error-free exchange of data mean?

- Transparency: The exchange of data between the PC and host should not be dependent on the physical connection. Considering the diverse PC-to-host connections, X.25, direct connections, modems, and LANs, this is not a trivial task. Since many companies rely on these connections, the communication tool should also support them. Otherwise your application will work in one situation but not in another.
- Error-free exchange: Data exchanged between a PC program and a host program must arrive at its destination without errors. Error detection and retransmission issues are low level, bit-oriented tasks and are therefore the responsibility of the communication tool. Application developers should not have to get involved in low level data communication functions.

From your perspective as application or tool developers, the PC-to-host communication tool you choose should be intuitive, easy to use, and mask the complex issues of peer-to-peer communications.

At this point, you should feel comfortable with the three concepts presented: using the PC for user presentation; the host as a centralized database engine; and choosing a communication tool that fits the needs of your application.

The next section discusses the criteria for choosing the correct communication tool. This decision depends on your application and the skills of the people implementing the application.

4. Determining the PC/Host Work Breakdown

In the application scenario discussed thus far, part of the processing is done by the PC, and part by the host computer. Exactly which functions each will perform determines the PC-to-host communication tool best suited to your needs.

There are two techniques that work well for PC and host communications in the HP 3000 marketplace:

- Message Exchange - A low level communication tool that provides the PC and host with basic SEND/RECEIVE data functionality. The PC can SEND/RECEIVE a message to/from the HP 3000. The host can also SEND/RECEIVE messages.

  If your application requires extensive database inquiries or heavy use of the communication link, a message exchange architecture is best suited for you. A message exchange architecture requires you to develop both a PC application and a host application.

  Two products in this category are:

NETIPC	Interprocess communication from HP Officeshare. IPC provides error-free message exchange between a PC and HP 3000 but only over Officeshare.
PPL	Process-to-Process Link from Walker Richer & Quinn. PPL provides error-free message exchange over direct connections, modems, 15 different LAN connections, X.25, and multiplexers.

- Intrinsic Server - Supports direct calls to MPE and all IMAGE intrinsics from the PC application. Using an intrinsic server, a software developer can implement applications that reside totally on the PC.

  This approach is well suited for applications that are inquiry only. Two products in this category are as follows:

Cooperative Services	HP's intrinsic server product.
PPL Toolkit	Walker Richer & Quinn's intrinsic server.

Let's look at some examples of applications where these communication techniques can be applied.

A Distributed Mail System: Why Message Exchange Works

Consider the case of a PC-based electronic mail system with the host as the mail server. Probably the best approach here is to make the host server the centralized mailbox, and have each PC user maintain his own local mail messages, both inbound and outbound. On demand, the PC connects to the host, delivers outbound messages, and retrieves inbound messages. Once disconnected, the user can browse through his mail and compose replies. This application would require a dual database capability, an intelligent host component, and a fast file transfer facility.

Message exchange is best for this application for the following reasons:

- It allows the development of an intelligent host application that integrates PC mail messages with the host mail system and accepts mail messages bundled as one large file.
- PC memory is saved because much of the mail processing is done on the host.
- The communication is mostly file transfer oriented, something that intrinsic servers don't handle well.

Order Entry: A Case for the Intrinsic Server

As another example, consider an order entry system for a distributor of office supplies. In this system, clerks take orders over the phone, giving information to customers about stock availability, prices, etc. You want the clerk to reach a given inventory item quickly, without having to enter a code. To do this, you will use the graphics capability of the PC to draw pictures of inventory classes (such as paper, writing instruments, etc). Using a mouse or touchscreen, the clerk will select a class, which will lead to further picture menus or to part description codes. Once a specific code is selected, a database lookup on the host is performed to bring down current information about the item. If the item is ordered, a transaction will be completed with the host database.

Since this application is inquiry oriented, the intrinsic server approach works well:

- Only the PC application must be developed.
- All necessary information is stored in IMAGE and therefore can be accessed through intrinsics.
- Data communication activity is limited. Data is exchanged between the PC and host only when a code is selected or a product is ordered.

Writing PC Applications

Those of you who do not have a great deal of PC programming experience may be wondering exactly how much effort it will be to get up to speed in this area. The answer will vary, depending on which PC programming language you use and how sophisticated you want the user interface to be. If you do your current development in COBOL or FORTRAN, you will be happy to hear that there are a number of excellent PC compliers available for those languages. While there is no PC implementation of SPL, programmers experienced in SPL should have no trouble learning C, which, along with assembler, is widely used by professional developers.

If you are worried about writing code to do fancy pop-up windows and such, take heart. There are several library packages available that will do all that for you. For instance, if you want to generate a pull-down menu, the program simply specifies a list of menu options, makes a call to a library routine, and gets a selection number in return. Many of these packages are written expressly for C programs, so it may be worth your while to make C the new standard in your shop. Examples of such packages are C-scape, Vermont Views, CXL, Windows, and Presentation Manager.

As far as database access routines are concerned, the complexity of your program depends on what you are doing. If you choose the message exchange approach, the actual file I/O or database calls are done in your host program or through a remote procedure. The PC simply sends and receives data buffers, the structure of which is defined by you.

If you choose the intrinsic server model, your PC program can make IMAGE or other MPE calls directly. This will make life a little easier for your experienced HP programmers. However, they must still learn a PC programming language.

5. Security

A critical issue for every MIS shop is security, i.e., making sure that data cannot be manipulated by unauthorized users. How does a PC based distributed system address security concerns? As with terminal based systems, the first level of security is at logon time, using user-ids and passwords. It is important that you do NOT allow the PC to automate the logon task, using command files with hardcoded passwords. Otherwise, anyone with physical access to the PC can make an "end run" around your security system. Instead, the PC should prompt the user for passwords, inserting them into command files. In this way, complicated logon sequences that must negotiate through modems, switches, etc., can be automated without compromising security.

Another pitfall to be aware of, particularly with the intrinsic server approach, is allowing the PC program to use Privileged Mode. If the host-based intrinsic server is a PM program, or the PC can command it to go into privileged mode, then in effect you have disabled the HP's security system (for a knowledgeable programmer). There is no easy workaround to this problem, other than designing your applications so that no PM work is required.

Finally, there is the slightly different issue of database validity. We have mentioned that reliability is extremely important to organizations: a corrupt database can be quite expensive. With the intrinsic server approach, a PC or data communication line going down in the middle of a multi-update transaction could lead to an inconsistent database. While there is the risk of this situation even in a terminal based system, it is important that system designers be aware of this issue when deciding on the proper communication tool for their situation.

Conclusion

Using the PC as a front end to your host-based database applications is a powerful technique for improving the usability of your MIS applications. In order to use the idea effectively, the developer must have a clear understanding of the nature of his application and how processing should be distributed between host and PC. With use of proper design techniques, a distributed application will perform effectively in today's world of multi-CPU environments.

HPDECIBMUNIXDGORANYTHING
(Multivendor Networking with the HP 3000)

by

Donn Lewis
The Apex Group, Inc.
7151 Columbia Gateway Drive
Building F
Columbia, MD 21046
(301)-290-1606

Steve Zalewski
The Wollongong Group, Inc.
1129 San Antonio Road
Palo Alto, CA 94303
(415) 962-7200

Many companies, educational institutions, and government agencies have acquired, over time, computers from several different manufacturers. The reasons for this vary. Independent divisions make separate purchase decisions; businesses merge or are acquired; or there is a need for a specific software applications which only runs on a particular computer system. In the university environment, various computers from different vendors are used to illustrate various operating systems and hardware technologies for students.

To provide the ability to share information between compatible computers, many organizations have created "networking islands" composed of computers all from the same vendor. Within one organization/company, there may be, for example, an AdvanceNet from Hewlett-Packard, a DECNET from Digital, and several PC LANs running NetWare. The islands may even be sharing the same transmission medium such as an Ethernet LAN or private X.25 wide area network. Though they are co-existing on the same medium, the networking islands, with their vendor-specific networks, are unable to communicate.

Beginning in the public sector, and gaining increased private-sector momentum in the last two years, there has been significant movement toward using industry-standard networking protocols. Two reasons for this growth are:

1. The ever increasing need to share more and more information instantaneously, regardless of what computer and in what location the information resides, and

2. The growth of public networks interconnected through the Defense Data Network and the worldwide internet. It is estimated that there are between 100,000 and 500,000 computers which are accessible through the internet. The internet demonstrates a cost-effective way for users to share valuable, non-proprietary information with colleagues around the world.

Today, the protocol of choice for multivendor networks in most companies, universities, and government agencies is Transmission Control Protocol/Internet Protocol (TCP/IP). Available since the late-1970s, TCP/IP is well-defined and is supported on over 150 different computers. It is the protocol used to communicate on the worldwide Internet.

This paper will discuss TCP/IP; its implementation on the Hewlett-Packard HP 3000; case studies of multivendor networks using HP 3000's; managing TCP/IP networks; and a possible future transition strategy to ISO's OSI (The International Organization for Standardization's Open Systems Interconnection) protocol suite.

What TCP/IP is

TCP/IP is the name given to the protocols adopted by the U.S. Department of Defense for the communication and interconnection of systems. Formally referred to as the Internet Protocol Suite, it provides several important user benefits:

TCP/IP allow users to interconnect computers running different operating systems,

TCP/IP provides an easy method of developing new communications protocols as part of the network, because the TCP/IP implementation is mature, well defined, and layered,

TCP/IP allows concurrent communication among several systems using one physical device, since it automatically multiplexes user data over the single device,

TCP/IP provides highly reliable transmission over an extremely wide range of media, and

Through the application services, TCP/IP provides the ability to do file transfers, exchange electronic mail, and execute remote log-ins.

The more common name, TCP/IP, comes from the two core protocols in the suite: Transmission Control Protocol and Internet Protocol, Figure 1 shows how parts of TCP/IP map to the OSI Reference Model. TCP and IP provide functionally equivalent to that defined by layers 3 and 4 of the OSI Model. TCP also contains some layer 5 functionality. TCP/IP considers the implementation of the data link (level 2) and physical layer (level 1) as independent. The network implementor decides which physical medium is used to interconnect the host systems.

IP routes data among hosts on the network and permits store-and-forward of data on the network. TCP contributes much to the popularity and power of TCP/IP. TCP's main function is to provide a reliable, byte-stream-oriented virtual circuit for application processes. As IP multiplexes protocols, TCP multiplexes connections. TCP provides the ability for multiple connections through a single network attachment. TCP allows referencing of individual processes (applications/users) within a host computer, while IP is limited to referencing only the entire host computer.

Above the TCP layer, direct parallels to the OSI model do not apply very well. The protocols that operate above TCP are not distinctly defined as residing in either the application, presentation or session layer. However, most of them could be placed at the application level. The most popular are: Telnet (Network Virtual Terminal), FTP (File Transfer Protocol), and SMTP (Simple Mail Transfer Protocol).

TCP/IP and OSI REFERENCE MODEL

Layer

7 APPLICATION

6 PRESENTATION

5 SESSION

4 TRANSPORT

3 NETWORK

2 DATA LINK

1 PHYSICAL

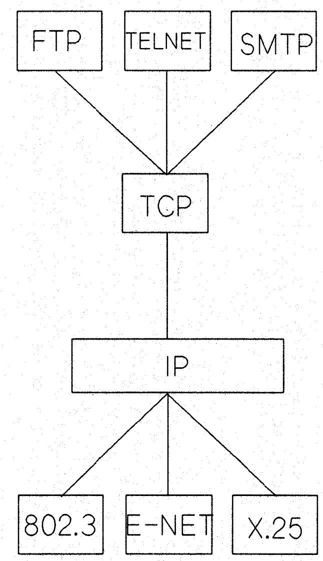


FIGURE 1

The Apex Group, Inc. 7151 Columbia Gateway Dr Columbia, MD

WIN/TCP for MPE/V

WIN/TCP for MPE/V provides the TCP/IP Application-level ARPA Services for Hewlett-Packard HP 3000 computers running MPE/V. HP 3000 Series 3x, 4x, 5x, 6x, 70, Micro 3000 and Micro 3000 XE/GX/LX are supported over X.25 and 802.3/Ethernet connections. The specific ARPA Services are: TELNET, FTP, and SMTP. Figure 2 shows the relationship of WIN/TCP for MPE/V to other HP 3000 software.

The TELNET service allows users to log onto an HP 3000 from a remote computer as though the terminal were directly connected to the HP 3000. The remote terminal may be connected to any of the 150 different types of computers which support TELNET or to a terminal server which supports TELNET. The connection can be over 802.3/Ethernet or X.25. A PC running a terminal emulator such as HP's AdvanceLink or Reflection from Walker, Richer, and Quinn, is also supported. The PC could be directly connected to the Ethernet cable and running WIN/TCP for DOS. Figure 3 shows some examples of how terminals and PCs could be connected to an HP 3000 as a virtual terminal using the TELNET protocol.

Wollongong's TELNET capability for MPE/V is transparent to block-mode applications. HP block mode terminals, or PCs with block mode terminal emulators, can access VPLUS or user block mode applications on the HP 3000 as though they were directly connected.

WIN/TCP for MPE/V also provides a TELNET client capability which allows an HP 3000 user to "TELNET" from an HP 3000 to another system on a TCP/IP network. For example, assume that on the same Ethernet cable there is an HP 3000, a DEC VAX, and a third computer running the UNIX operating system. An HP 3000 terminal could become a virtual terminal to the DEC VAX or the UNIX system.

FTP provides for the transfer of different types of files between the networked systems. FTP follows MIL-STD 1780. Binary as well as ASCII files can be transferred. The file and system security of both the host and client systems are maintained.

SMTP provides a means of transferring mail messages between users of different mail systems. For example, you could send a mail message using HPDESK to a VAX user if both users are on the same network. The sender does not have to know the route or how the receiver is connected to the network. This capability is one of the most widely used features of the Defense Data Network (DDN) and the worldwide internet. If connected to the Internet, by supplying the receiver's address, a user could reach the millions of mail users on the 100,000 to 500,000 Internet-connected computers.

To send a message, the HP 3000 user utilizes HPDESKManager's Foreign Service Connection. The user composes the message as if the receiver were on their HP 3000. To specify the receiver's address, the sender can give the complete address or use HPDESK's distribution list capability. For example, the list "zalewski" could be used instead of the address: sjz%subloc%loo@twg.com When distribution lists are used, the sender is completely unaware that the receiver is on a non-HPDESK mail system or even where the sender is located. No routing or addressing information needs to be known by the sender.

HP 3000 TCP/IP IMPLEMENTATION

HP SOFTWARE	HPDESK
TELNET FTP	SMTP
MPE V - V-Delta-2 or LATER	
ANY CLASSIC HP 3000 (GX to Series 70)	
LANIC INP	

FIGURE 2 The Apex Group, Inc. 7151 Columbia Gateway Dr Columbia, MD

TCP/IP PHYSICAL CONNECTIONS

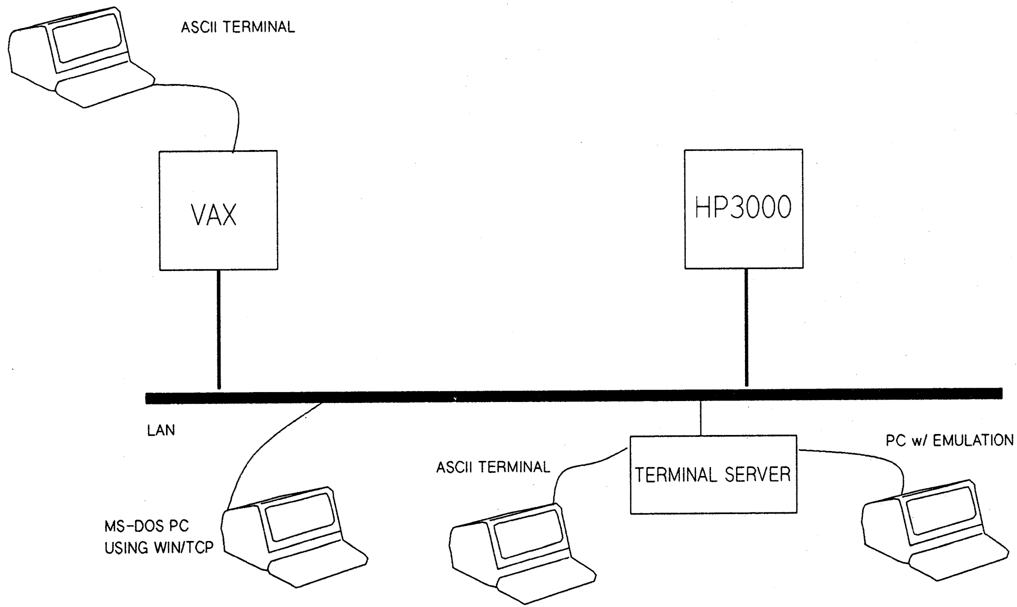


FIGURE 3

The Apex Group, Inc. 7151 Columbia Gateway Dr Columbia, MD

WIN/TCP for MPE/V is available for all MPE/V systems from the Micro 3000 LX to the Series 70. MPE release V-Delta-3 or later is required. V-Delta-5 or later will allow HP 3000's to interoperate on the same physical cable with ethernet hosts (DEC VAX etc.) without the use of a gateway. SMTP requires HPDESK.

Some Case Studies:

The real power of TCP/IP on the HP 3000 can be seen in some of the case studies presented here. In figure 4, An HP 3000 Series 58 is used to provide interoperability with a series of DEC VAX systems several thousand feet away on a fiber optic link, as well as with the Defense Data Network (DDN) via X.25. In order to provide filtering and DDN connections, a gateway from cisco Systems provides routing of TCP/IP packets between the HP and DEC LANs and interfaces to the core gateways of the DDN.

In figure 5, the user required a common solution to interfacing an HP 3000 Series 70, several DEC VAX systems and a remote IBM 3090 running MVS. In this case the VAX and HP 3000 systems are using WIN/TCP products from The Wollongong Group, a hardware and software solution for the IBM system from Advanced Computer Communications (ACS 9315 and ACCES/MVS). The IBM system is located remotely from the HP 3000 and the VAX systems, so a pair of cisco gateways provide a link between the two LANs at T1 speeds (1.544 Megabits/sec). The system is used to provide both terminal emulation (TELNET) and file transfer (FTP) services to all the hosts in the network. The network management software on the cisco gateways provide information to the network manager about traffic loads, status of the physical links and provides both local and virtual console capability.

In figure 6, a user running V-Delta-3 is connecting an HP 3000 Series 70 through a cisco gateway to an IBM RT running AIX (IBM's unix). The system provides for workstations connected to the RT to access the HP 3000 and other NS services resident on the HP 3000 for connect to IBM MVS services. Because they are using V-Delta-3 release of MPE, the use of a gateway is required.

Some Pitfalls:

One of the problems in dealing with industry standards like TCP/IP is that each vendor implements them in a slightly different manner. This can create solutions which theoretically work, but in practice do not. Because of these systematic differences, the user should plan appropriately to obtain design, engineering and installation support from a qualified network intergrator when planning and executing a multivendor network.

U.S. NAVY HP3000 TCP/IP Network

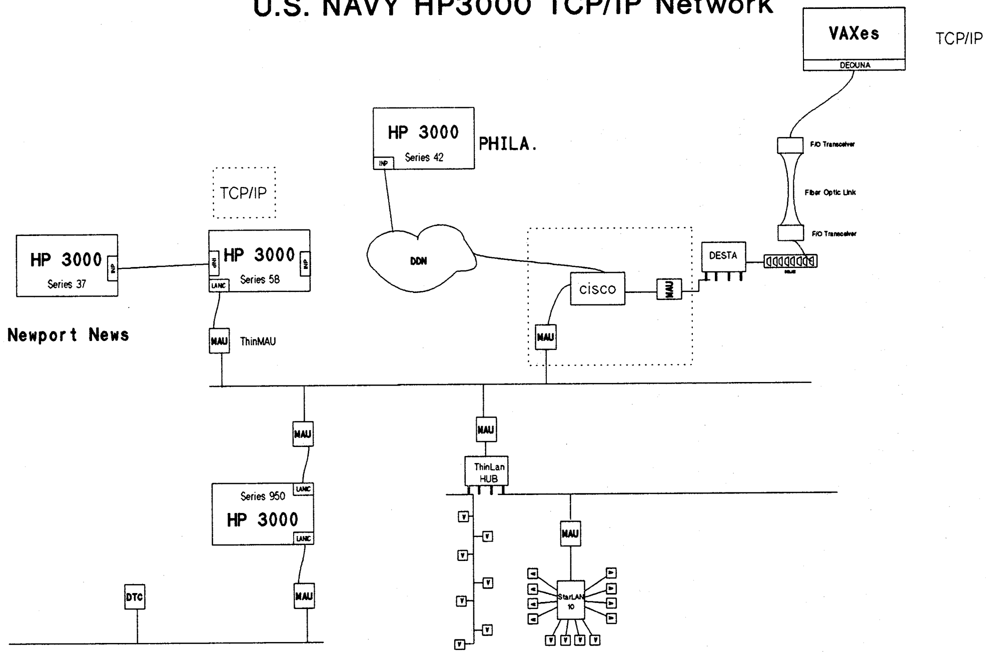


FIGURE 4 The Apex Group, Inc. 7151 Columbia Gateway Dr Columbia, MD

IBM, HP3000 and DEC VAX - TCP/IP Network

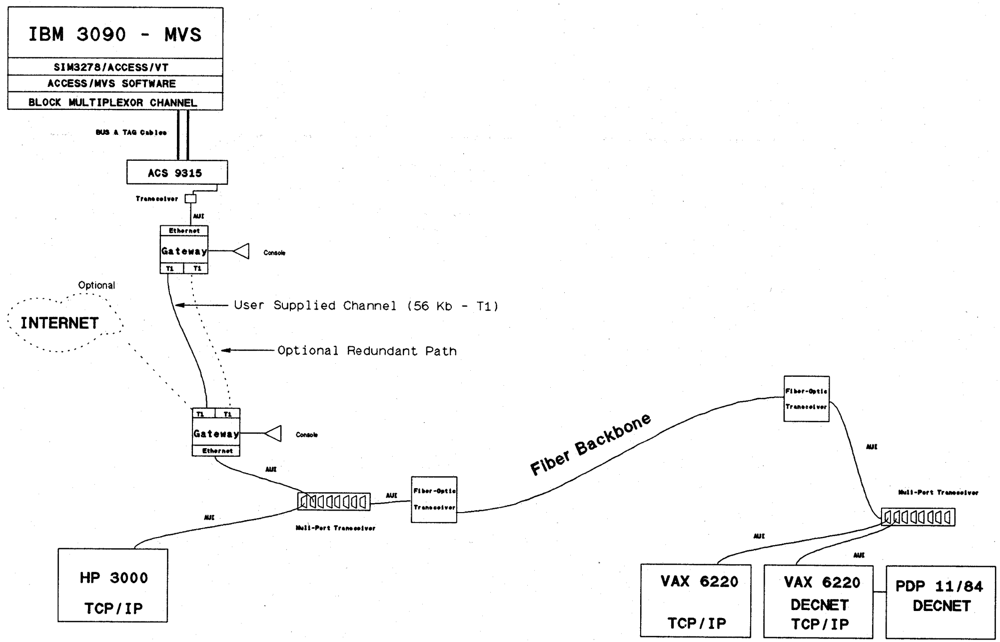


FIGURE 5

The Apex Group, Inc. 7151 Columbia Gateway Dr Columbia, MD

HP3000, IBM UNIX and SUN

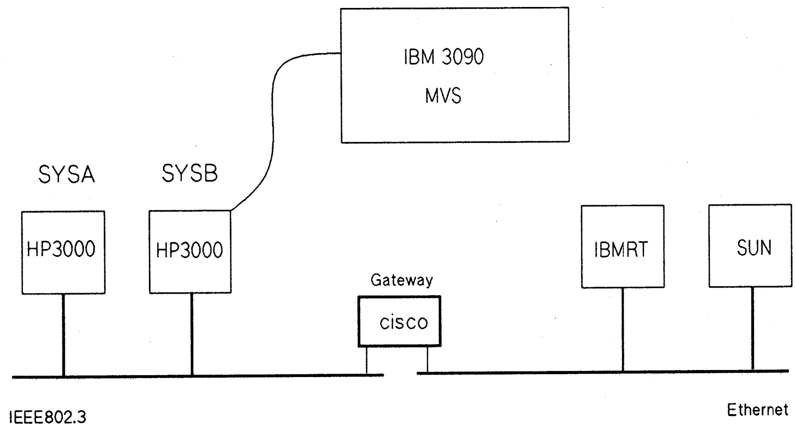


FIGURE 6

The Apex Group, Inc. 7151 Columbia Gateway Dr Columbia, MD

Transition From TCP/IP to OSI.

In the future, as the higher OSI layers (layers 3 through 7) are implemented by more computer vendors, there will be a transition to OSI from TCP/IP. We would like to end this paper with some ideas on a reasonable way to make the transition from TCP/IP to OSI. The first question which comes to mind when someone speaks about OSI is "When will there be products generally available?". Estimates vary widely from immediately to the next "few" years. Regardless of when a particular product is available, some things are very definite:

> Not all data communications and computer vendors will have products available at the same time,
>
> Some of the computers on the existing TCP/IP network will never have OSI provided on them because they are an older or low volume product, and
>
> Not every division/organization will be eager to upgrade the network and, in fact, for the previous two reasons, may be unable/unwilling to ever upgrade from their present TCP/IP network.

For many customers, transition will involve the coexistence of TCP/IP and OSI with a gradual phase-in. For large networks, this phase-in transition may take as long as ten years.

When users are choosing products to build networks today, they should insure that the vendors have a commitment to provide migration to the OSI model in the near future.

The time to consider the transition to OSI is when you begin implementing or expanding your TCP/IP network.

Summary

Today, TCP/IP is the protocol of choice for implementing multivendor computer networks. With WIN/TCP for MPE V, HP 3000 computers have the ability to fully participate in an organization's network.

How a Logical Information Network Can Help Systems Managers Maximize Network Performance, Minimize Cost

by

Howard J. Gunn
Vice President, Sales and Services
Gandalf Data, Inc.

How a Logical Information Network Can Help Systems Managers Maximize Network Performance, Minimize Cost

ABSTRACT

The continued evolution of a digital wide area transport network by the telephone industry presents end user organizations with a myriad of new and complex issues involving tariffs and technologies. This paper will describe an OSI compliant, customer premise based Logical Information Network (LIN) structure, that allows organizations to capitalize on the technical evolution and tariff modifications that follow the technology. The LIN provides the structure for combining time-sensitive performance control (circuit switching) with store and forward (packet switching) capabilities over the telco transport network, while developing OSI compatibility with traditional dial, X.25 and future ISDN services.

The Networking Dichotomy Perspective

The concept of networking as a means to share expensive, complex or scarce resources has been an accepted axiom at least since the time of Moses -- one boat and no one could tread water for forty days. Over the years, the concept of sharing valuable resources has been honed to an advanced art, but not a science. Networking has failed to evolve into a science because, there are no immutable laws of expensive, complex or scarce, that apply as clearly as they did to Moses. Our capitalistic society places a premium on progress, performance and competitive advantage and that causes the relationship of value and cost to shift with respect to volume, time and distance. The underlying factor that drives this shift in networking values is often perceived as technology and the deployment of technology. But, other factors constrain and even control technology.

Think for a moment about the value of the only printing press. Its value was constrained, at the time, by the ability to read. In other words, it isn't the technology itself that creates value, but it is how the technology fits or can function within the constraints of the environment. Printing Bibles instead of Keno cards made sense at the time, but Keno cards certainly seem to generate more value today, in certain places around the world.

If networking is only an art form and technology is constrained by the environment, it only stands to reason that some networking artists are better than others. In the same vein, businesses must realize that as the environment changes, so will the art form. Today, in the U.S., we are living through a period of significant changes in the information processing and the transport environment. At the same time, new technologies, such as imaging systems are finding their way into generalized use, as a means to create a competitive advantage. How these factors are integrated by today's firms will determine their competitive fate in the early 1990's.

Today's Environment

The North American experience with open competition has expanded from a simple government controlled experiment in market allocation, to a full fledged test of technology and complexity. In a traditional sense, there has only been one Wide Area Network (WAN), since the early 1900's. The value of this network transport system to the end user was based on the dictation of a simple pricing model and the government's desire to exclude all other alternatives. With the creation of MCI, the breakup of AT&T and the eventual evolution of multiple WAN carriers, the technology controlling tariff structure of monopoly has disappeared. In its place, we have multiple carriers and multiple technologies. Figure 1 shows the technology based

"Zones of Advantage" in terms of bits (volume) and sites. In addition, there are at least six vendors in each zone and some vendors in multiple zones of technology.

The rapid introduction of both WAN technology and suppliers has created new opportunities for buyers, in comparison to the traditional controlled technology introductions of the monopoly era. Price and technology are no longer constrained by tariff. The only constraint in the environment is now competition and fit within the environment. Each potential source of transport can now push its technology on the user, via prices and services, that typically can offset the cost of the technology introduction.

Figure 2 shows a set of generalized cost curves, with respect to distance and volume, independent of the technology curves and their zones of advantage. As with the variability of technology, distance, volume and performance needs all are all factors in making an economic decision. It is clear that each organization can create a competitive advantage for itself, in terms of responding to these cost curves. The problem is the complexity of multiple carriers and multiple technologies makes it is almost impossible today to select a single technology for all applications, distances or volumes.

The importance of the competitive advantages that can be gained is shown in Figure 3. Industries are spending between .5 and 2% of their total revenues, on WAN transport services. These costs are in addition to what is spent on WAN facilities that are privately owned. Companies within these industries that can improve their WAN performance or reduce their cost for WAN transport, with respect to their competition, obviously flow the advantage directly to Net Profit -- and that is what eventually drives the value of a company's stock.

But, since each WAN transport vendor has the liberty to provide transport services, at a price that best suits their own network architecture and technology, how is an end user organization to cope. Each organization already has a set of computers, a changing computing environment and a the variables of time, distance and volume. Selecting an appropriate computing configuration, that can exploit the various cost advantages offered by the carrier's technologies seems like an insurmountable challenge. This paper suggests that the end user organization has to evolve a Logical Information Network (LIN) architecture, to be in position to gain from deregulation, gain from the explosion of technology choices and gain from the myriad of changes sure to come from the expanding role of computers and communications.

What is a LIN?

A LIN, or Logical Information Network, is a design architecture, that allows the end user organization to control cost and performance through the introduction of software processing within the seven layers of the Open System Interconnection (OSI) model. The intent of the LIN is to allow the organization to develop OSI compliance, while taking advantage of the myriad of WAN technologies and cost alternatives available for the movement of information over WAN transport facilities. As shown in Figure 4, a LIN would allow communications between or across Enterprise Networks independent of architectures and protocols.

The OSI model itself is a fairly stagnate view of how various processes can be arranged to produce applications that are computing hardware independent. Although OSI is an obviously important conceptual step toward applications evolution, as shown in Figure 5, it is evident that the telephone costs don't seem to enter into the picture. It seems obvious that there is more then the technical variables of the OSI model to consider, in the real world of networking applications.

Figure 6 displays the same OSI model and how it can be related to business functions, at the application layer. This chart also shows the number of specifications to be considered in the physical, logical and network environment that support the data processing, office and telecom applications. And, if there are already some 2000 unique ways through the OSI stacks and at least six different suppliers through the low level transport, the end user organization has a maze that requires a more practical architectural framework for making WAN transport decisions.

Enter The Art Form

To achieve resolution of the maze, above the simple establishment of connections the network level in OSI model, implies the insertion of programming and logic at, or above Level 4 (quality, cost and performance level) and at or below Level 5 (session level), to ensure security within the computer session. In Gandalf, we call this point between Levels 4 and 5, the Program Interface level or PI for short. Figure 7 reflects this PI and how it can be used to manage host, voice and lower level functions.

What Does a LIN Provide?

The idea of a LIN is to produce the same effect on WANs as IBM's LU 6.2 concept brings to host computing in the SAA environment. Programmers at the higher levels can assume that they are writing to a standard interface, that is device and transport independent. Just as importantly, network operators can manage

WAN physical, logical and network layers, independent of target resources constraints, if the PI exists and can manipulate the arriving data correctly. For example, the same physical, logical and network WAN media can be used to carry voice, data and video, in whatever volume desired by the performance and cost criteria of the leasing organization up to the "capacity" of the media. This is in direct contrast to today's environment that dedicates electronics and the media to extend a single function over the network media. Even the future ISDN environment has the same level of restriction between sender and receiver, as evidenced by the channelization of the 144 kilobit pipe. By following the architectural design parameter of a LIN, each leasor would be able to use the entire 144 kilobit channel to carry whatever information needed to be transmitted and create whatever channelization is appropriate for the transaction. The output from the LIN would route the appropriate bits to the target resource via a much lower costing premise media. This combining of the heterogeneous traffic onto a single WAN facility would allow the electronics and the media to be run at 60 to 70 percent occupancy, as opposed to the current 15 to 20 percent occupancy load of most WAN linkages in use today.

A LIN would also provide the natural evolution stage to the future OSI environment, without the cost or performance robbing degradation sure to accompany computer processing of multiple stacks of protocols. As shown in Figure 8, the LIN draws a network software manageable boundary between the logical and physical Network Addressable functions and the logical and physical Path Control functions. By imposing the software control in the network, managers nor users will not have to concern themselves with either lower level or higher level compatibility. The network connection itself can be programmed to transform the arriving traffic to the native mode of the target resource, without concerning itself with the compatibility of the native mode, at the source. And, as shown in Figure 8, compatibility with both pre-OSI resources (SNA & DNA Models) and post-OSI models (SAA & OSI) will simplify and may actually speed up the transition and acceptance of OSI and SAA.

Why Network Processing?

Implementation of the transport layer control function in the network implies preprocessing of the bit streams to minimize transport costs, by maximizing the occupancy of the linkages. Given the WAN costs of today and the expected pricing over the next five years, our models indicate that WAN savings alone will pay for the relatively inexpensive cost of "preprocessing" the data streams and filling WAN channels to maximum capacity. For example, a single B channel or switched 56 circuit in today's environment can support 64 sessions of 19,200 bit per second asynchronous traffic (about 30 packets per second). Obviously, such traffic is already preprocessed in terms of stat muxing or

PAD functionality today. The importance of the LIN is that the original asynchronous traffic wouldn't be reprocessed back to async at the receiving end. Some of the packets could be preprocessed directly to TCP/IP and fed directly to hosts. Other packets could be pre-processed into SDLC and fed into a Front End Processor, as if they came from a communications controller. Other packets could be routed direct to voice PBX trunks.

Why is it Important to Consider a LIN?

Simple competition and the global nature of future competition are somehow related to the timeliness and availability of information. In this environment, up-to-date information is no longer enough. Companies will and already need to have access to information residing in multiple storage devices, in different locations, on different transport networks.

For example, the time related facets of the On-Line-Transaction-Processing (OLTP) industry are already beginning to obsoleting certain concepts of the time shared host architecture envisioned by OSI (remember OSI preceded the PC, MAC, SUN, VECTRA, Appolo and ATM). Although OSI proponents will obviously try to patch-on just a few more specifications, to expand the 2000 or so ways to be OSI compliant, sooner or later the two dimensional OSI model will be torn. If OLTP doesn't do it, the technological imperatives of the next advances in memory management and computing will surely be even more robust than the OSI model. For example, imaging systems, such as Filenet's and diskless workstations, such as Gandalf's 8980, can already support multiple sessions on different media to independent hosts, while they are concurrently running DOS under NOVELL. Although OSI helps formulate the local screen management for these devices, the same access medium to the user is carrying ASCII, EBCIDIC, graphics and voice -- TODAY.

Maybe the most important reasons to consider the LIN architecture and products that support network preprocessing, are the concepts of flexibility and evolveability. In this context, we can only rely on simple analogies to history. The computer, as we know it in its electronic form is about forty years old. If it is in the same state of development as the telephone after forty years, I can assure you the rapid growth period in computing is still in front of the industry. And, unlike the telephone industry, who managed its technology through monopoly, you, the networking artist for your company, do not have this luxury.

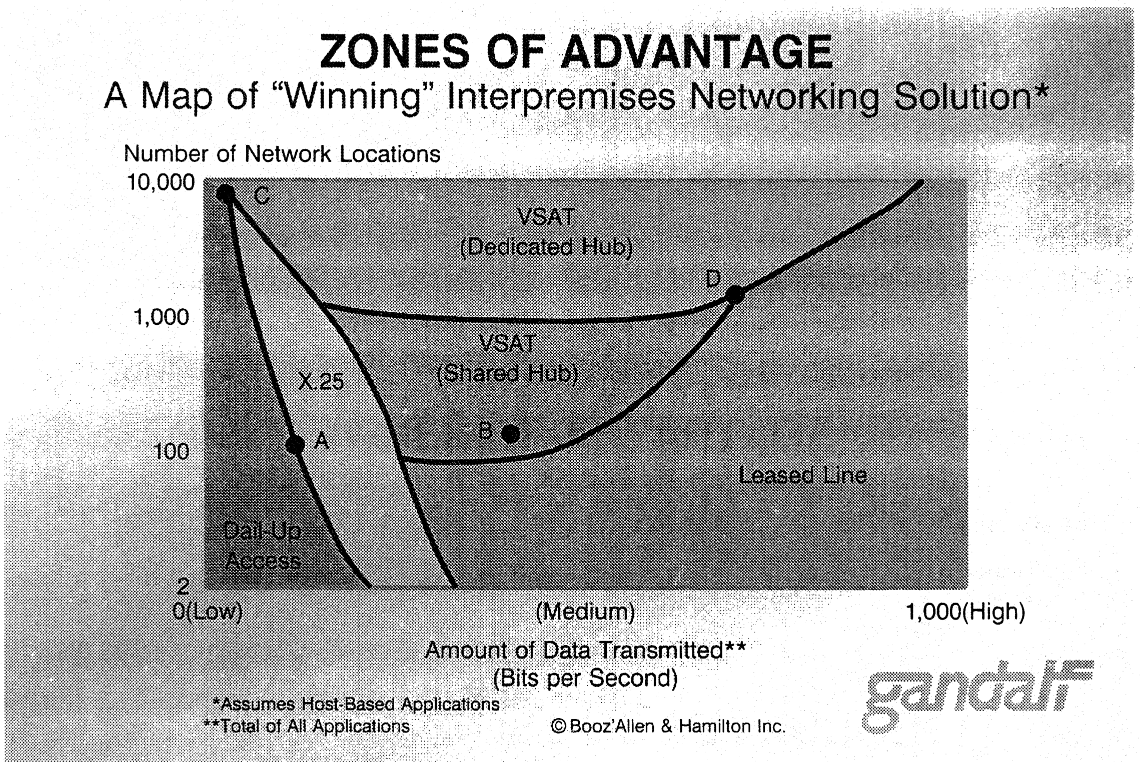
LIN Based Products

Various products, such as X.25 switching, LAN gateways and networking T1 devices have been designed to support the first three levels of the OSI model. Each product provides a high degree of homogeneous networking, within the scope of the first three levels. Most of them have separate network management managers with Level 4 functionality over the devices.

Newer products, such as the Gandalf StarMaster Network processor, depicted in Figure 9, and the AS 400 from IBM have been expressly designed to support the OSI/SAA communications architecture. Other vendors, including HP AdvancedNet and Data General are evolving similar architectures, that manage source and target resource independently.

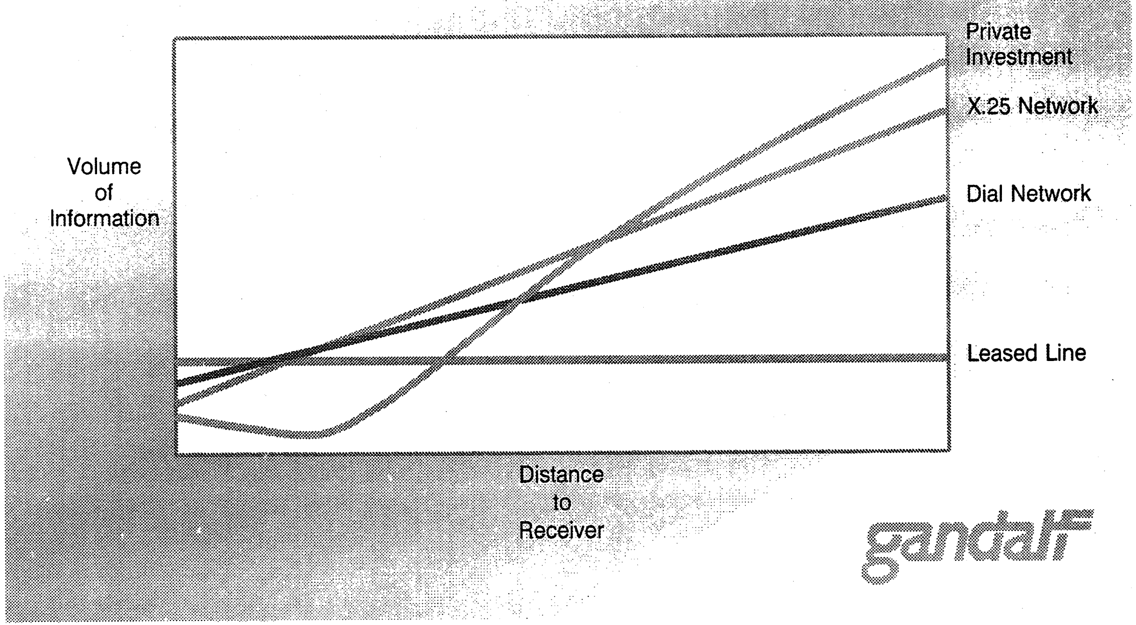
Although the StarMaster and AS 400 conceptually handle the transformation the same way, there is still the age old issue of store and forward versus intelligent peripherals. Which approach fits is primarily a function of volume and performance. As shown in Figure 10, StarMaster deploys both technologies within the same architecture. Figure 10, reflects Gandalf's use of the dedicated peripherals that perform the pre-processing. The intent of the distributed processing design is to dedicate sufficient computer power to simplify performance management and potential generalized degradation of a single host shared computer system, such as the AS 400.

Obviously, such products exist to provide the tools for the networking artist to get through the LAN/WAN maze of cost and performance management while providing applications. It is just as obvious, that volume of traffic, existing operating environments, artistic expression and the needs of the company weigh heavily on the expression of technology. In fact, the important issue is not how to erect a LIN, but, how rapidly businesses actually get network artists commissioned.



4634 - Figure 1

GENERAL COST OF TRANSPORT



4634 - Figure 2

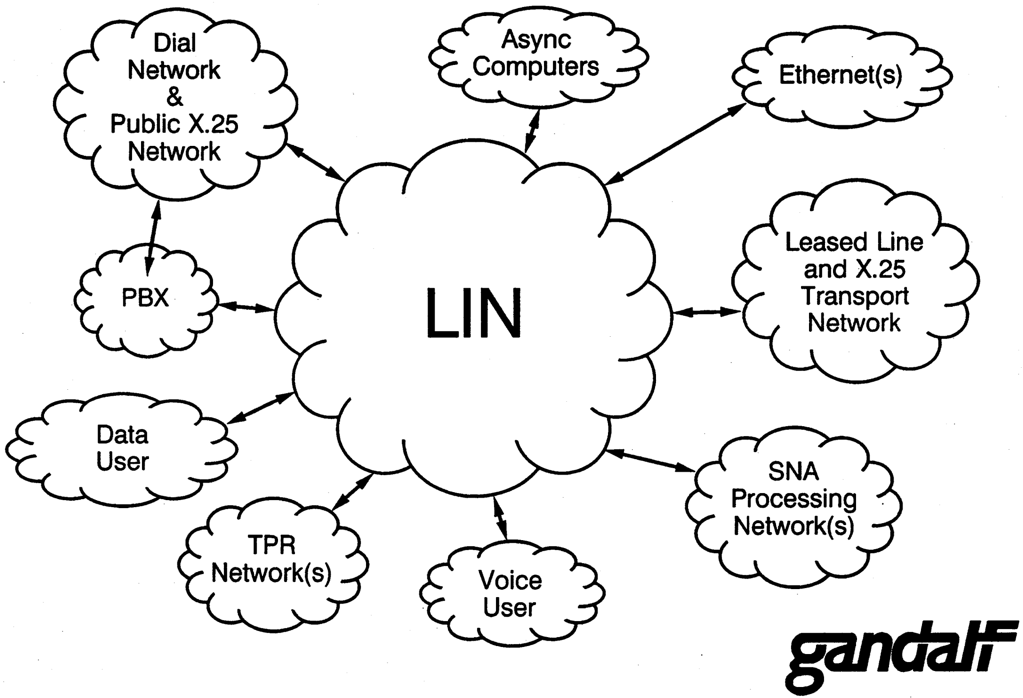
TELECOM EXPENSE AS A PERCENTAGE OF TOTAL REVENUE

Rank 1987	Rank 1986	Industry	Percentage of Revenue 1987	Percentage of Revenue 1986
1	2	Airlines	2.05%	1.60%
2	16	Banks and Bank Holding Companies	1.33%	0.58%
3	1	Office Equipment and Computers	1.32%	1.70%
4	4	Transportation and Trucking	1.31%	1.25%
5	7	Universities and Nonprofit Organizations	1.09%	1.10%
6	18	Service	0.94%	0.52%
7	*	Electrical and Electronics	0.90%	*
8	10	Aerospace	0.75%	0.66%
9	24	Textile and Apparel	0.72%	0.42%
10	9	Manufacturing	0.67%	0.68%

**Data for Previous Year Is Not Available*
Source: International Communications Association, Dallas

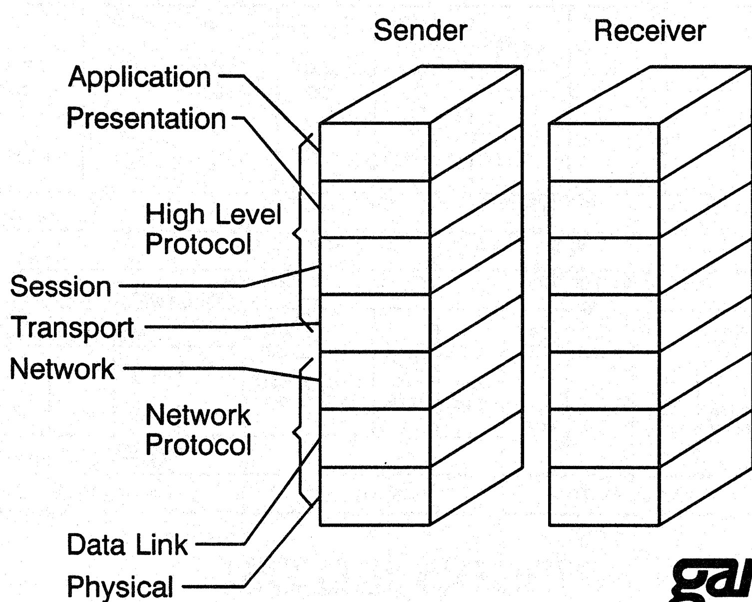
gandalf

4634 - Figure 3



4634 - Figure 4

OSI MODEL



gandalf

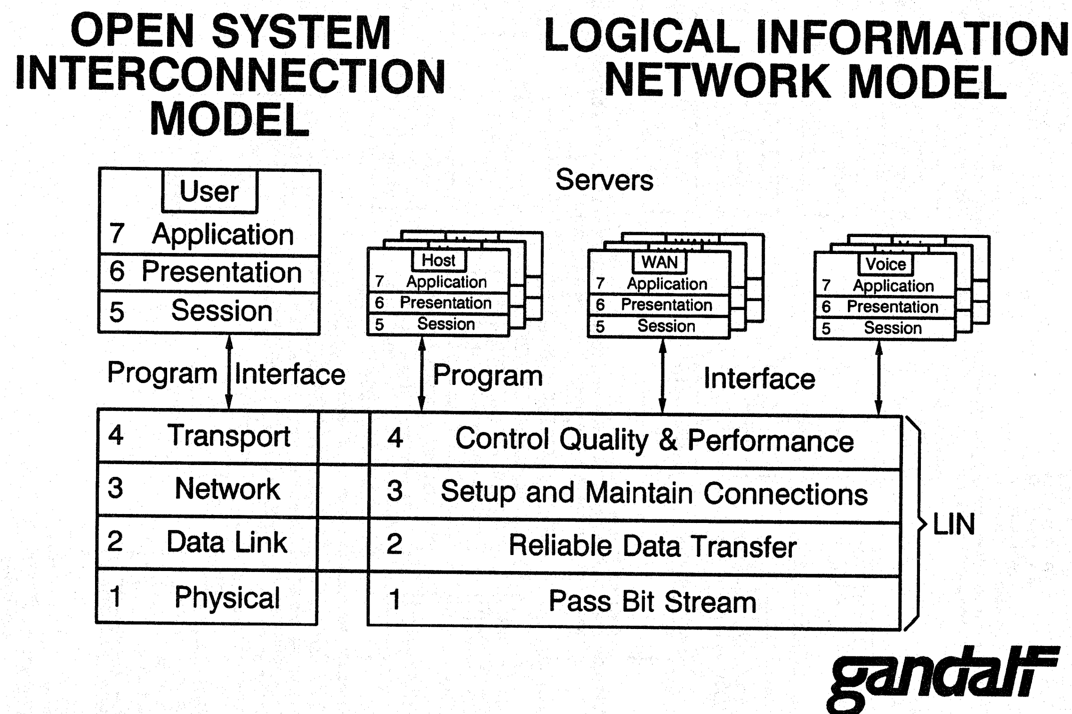
4634 - Figure 5

ISO/OSI MODEL

Layer	Office Function				DP Function				Telcom Function				
7	Document Structures T73, ECMA101								Tele Text F200 T60	Fax FIXY	Video Text	Mixed Mode T73 T200	Application Service
7	Dir	File	Mail X400	Print	File Trans ISO 8571	Job Trans	Prog-to Prog-Comm	Virtual Term					Application Service
7	Remote Operation and Services X410												Application Service
6	Presentation (Transfer Syntax and Codes) ISO 8822, 8823, 8824, 8825, X409, T50, T6, T61, T100												Application Service
5	Session ISO 8326, 8327 X215, X225, T62												Application Service
4	Transport ISO 8072, 8073 X214, X224, T70												Transport Service

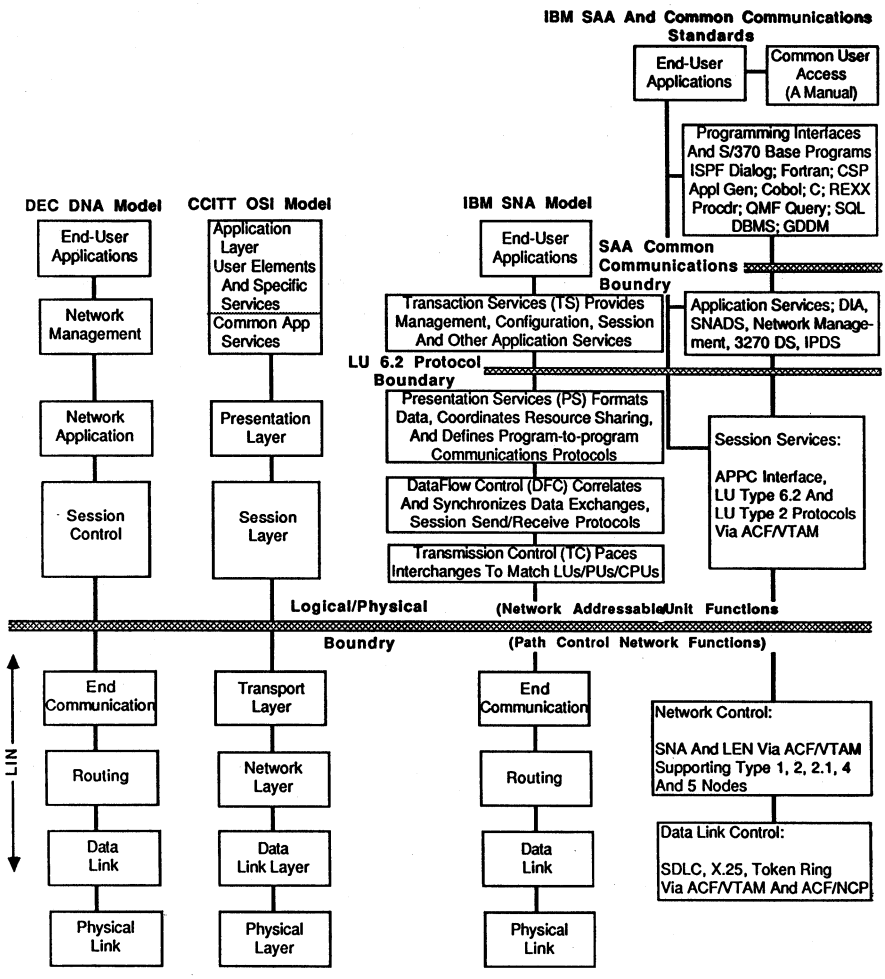
Layer									
3	PSTN	CSDN	PSDN	ISDN	CSMA/CD	TPR			Transport Service
2	ISO 8208	ISO 7776	ISO 8878	ISO 8877	ISO 8802.3 8802.2	ISO 8802.5 8802.2			Transport Service
1	V....	X21	X25	I....	8473	8473			Transport Service

gandalf



4634 - Figure 7

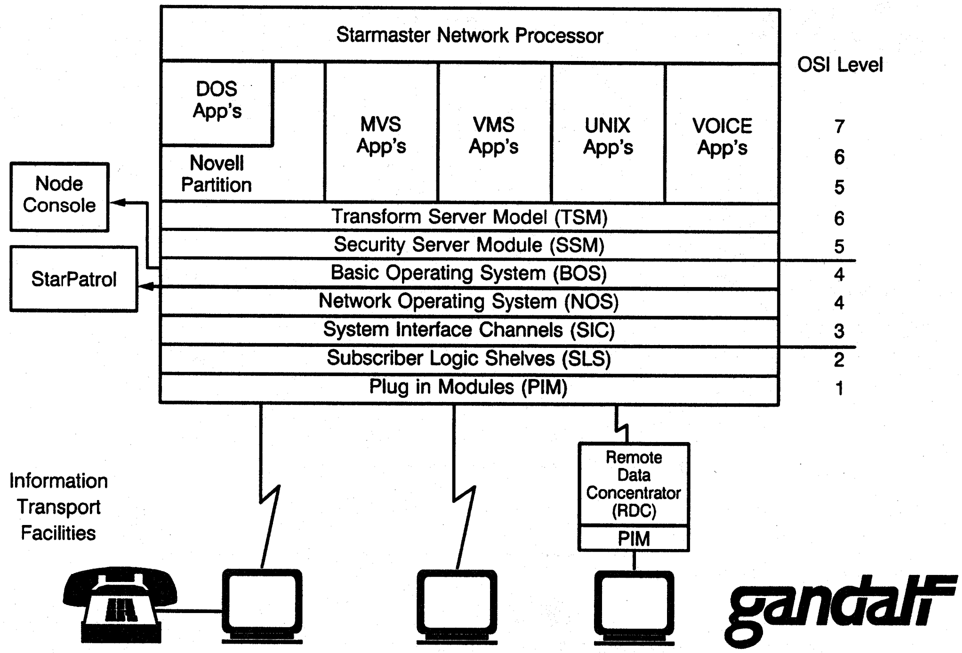
LIN Fits With SAA, DNA, OSI And SNA



ACF/NCP = Advanced Communications Function/Network Control Program
ACF/VTAM = Advanced Communications Function/Virtual Telecommunications Access Method
APPC = Advanced protram-To-Program Communications
CSP = Cross System Product
DCA = Document Content Architecture
DIA = Document Itnerchange Architecture
GDDM = Graphical Data Display Manager
IPDS = Intelligent Printer Data Stream
ISPF = Interactive System Productivity Facility
LEN = Low Entry Networking
LU = Logical Unit
PU = Physical Unit
QMF = Query Management Facility
SDLC = Synchronous Data Link Controls

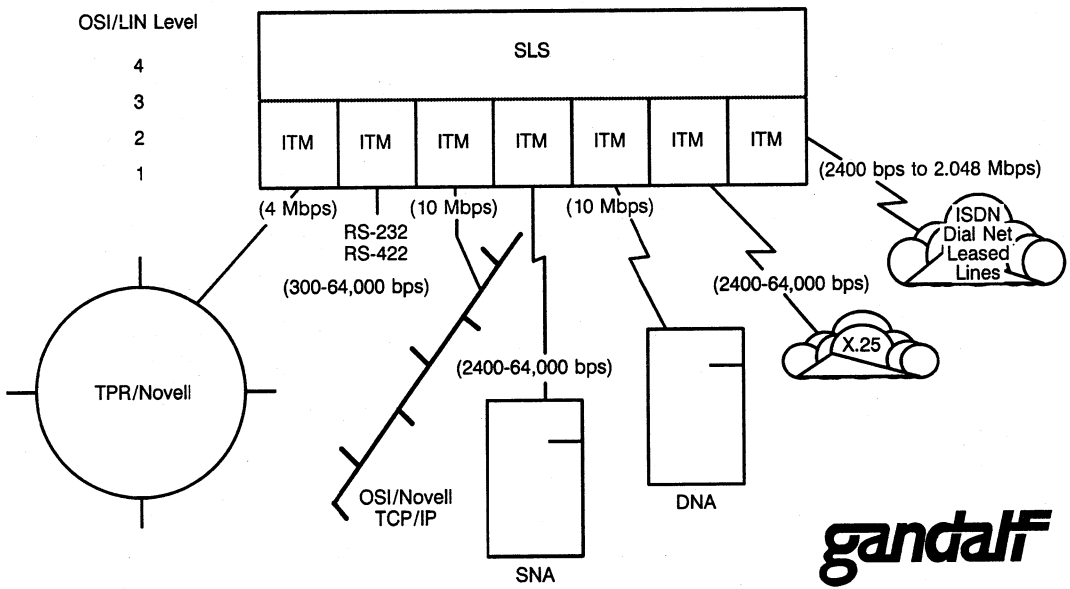
4634 Figure 8

STARMASTER NETWORK PROCESSING SYSTEM



4634 - Figure 9

STARMASTER INTER-NET TRANSFORM MODULES (ITM)



4634 Figure 10

PREMISE WIRING SYSTEMS;
GUIDELINES FOR THE DECISION MAKER

BY

MARK INDERMILL
PRECISION INTERLINK CORPORATION
11241 COLOMA ROAD, SUITE D
RANCHO CORDOVA, CA 95670
(916) 631-0323

Every business that has a computer and a telephone system has a need to connect that computer and telephone system with terminals and/or telephone instruments out on desks or in special workstations. Generally, there is one person who takes ownership of the small problems of connecting a terminal here or a telephone there and keeping the office systems up and running for the staff. When the problem gets larger, that is, when there are more than just a few terminals or telephones to deal with, when the office is moving to a new location, or when there is a major reshuffle within the office, it is time to contract for the cabling work to be done by a professional low voltage cabler.

The purpose of this paper is to give the reader some insight and guidelines to use in evaluating office wiring for business. We will begin by describing the problem that businesses face in taking full advantage of today's communication technology. Then we will look at some of the complexities and issues surrounding premise wiring systems, and finally, we will provide some methodology for evaluating your needs, evaluating the network design for your office environment, and evaluating the vendor.

HOW IS TODAY'S COMMUNICATION TECHNOLOGY BOTH A BOON AND A BURDEN FOR TODAY'S BUSINESSES?

THE BOON: Hardware sophistication translates to increased user convenience, flexibility, increased functionality, and lasting value.

Since the AT&T divestiture, competition has resulted in vastly improved voice and data communication technology available to business. Now a business can install its own cabling systems, thus taking full advantage of the latest in technology, as well as achieving economies of scale by addressing voice and data communications together rather than as two separate problems.

THE BURDEN: Hardware sophistication also translates into increased user responsibility.

Before divestiture the telephone company was the sole owner of the telephone system and was solely responsible for all service and maintenance. After divestiture, voice communication hardware was no longer under the sole purview of the telephone company -- installation can legally be performed by anyone. Contracting for this service with someone who has a valid contractor's license and a communication equipment specialty will ensure a certain level of quality, but the responsibility for an office wiring system that works is yours.

Now, you own the wire and the cable plant; you are the installer; you are the sole source of maintenance and support; you are the designer of office wiring systems.

SO WHAT?

The installation and full utilization of sophisticated equipment generally requires technologically sophisticated installation techniques. Not only is it important to understand the technology and what it can provide for your business, it is also important to understand the complexities of the problem.

WHAT BASIC FACTORS MAKE INSTALLATION COMPLEX AND DIFFICULT?

There are a host of factors that contribute to the complexity of voice and data communications wiring today. With the number of vendors, wiring types, equipment types, and an assortment of other things to be wary of it is not difficult to see that answers to a few well posed questions should be the first step in the planning process.

Multiple equipment vendors. (Sure, your shop is all HP, but what about the phone system?) Many data processing environments include equipment from many different vendors. You may have an IBM mainframe and HP satellites. Or, an HP system and a ROLM PBX. You may have a DEC machine acting as the front end for your nationwide network. You may have IBM PC's out on the office floor. Seldom does one find a completely homogenous shop.

Equipment types. Almost every desk has a telephone. Many have terminals. Some of those terminals are personal computers. A lot of those PC's are connected by LAN's. Many of these devices are connected to printers. Some are connected to modems and multiplexors. Some businesses have computers on site, some are served by remote CPU's.

Wiring types. If you are an IBM user you are familiar with coaxial cable. IBM also uses TYPE 1 and TYPE 2 connection technology, a combination of coax and twisted pair in the same insulating sleeve. A Wang user works with dual coax. Telephones use twisted pair. There are applications for twinax. Do you really require shielding on all your cables? Does the building code require teflon coated cable in your environment?

Other variables. There are other variables that play a role in defining your wiring system requirements. What communications protocols are you using? Are your telephones digital or pulse? What electrical specification are you using, RS232, RS422, 802.3? What media -- IBM Type 1, 2, 3...99? How large is your building and is the wiring plant designed for maximum occupancy? (It should be.) What is the maximum run length for cables? Is radio frequency interference (RFI) going to be a problem? Is there electromagnetic interference (EMI) present at your site?

There is a perceived lack of connectivity between different types of communications devices.

There are unstructured wiring configurations.

There is an almost total lack of documentation. No records of existing cables.

And, it really doesn't surprise anyone that most businesses suffer from inadequate office planning.

HOW TO PLAN

Plan for the future. Most companies will pay high installation costs for data and voice communications equipment. The incremental cost of planning and installing for the maximum occupancy rate of the office space is small compared to the cost of adding to and changing an inadequate wiring system.

If your business is like many others, you will have to put up with many intra-office moves, in addition to any growth in personnel. It is not unusual for 50% of the installed telephones and data terminals to be moved each year. For data terminals, this problem is compounded because of the variety of cable types usually associated with a variety of equipment. These cable types will likely have to coexist with future products.

Also, multi-business office buildings have medium to heavy communication requirements and characteristically have a high rate of occupant turnover and office rearrangement. Because the demand for communications is diverse, these buildings should have a cable distribution system that provides maximum flexibility.

WITH BURDEN ALSO COMES OPPORTUNITY -- WHAT IS THE OPPORTUNITY?

Increased flexibility. Today's technology permits installation of a cable plant flexible enough to provide performance compatible with the old cable types and still support new wiring designs. In some cases, even existing cable plant can be converted to a new, more flexible installation.

Increased support capability. Today's technology allows almost total flexibility in supporting equipment from various vendor types. The same cable plant can support IBM, Hewlett Packard, & DEC equipment as well as a majority of telephone system types concurrently with only slight modification.

Economies of scale. Although telephone systems are frequently considered separately from data communications systems, this is not necessary. Many installations can use the same premise wiring system for both voice and data. The initial cost of this kind of piggyback installation is cheaper than installing both systems separately and frequently easier to maintain.

User maintainability. The user may now manage his own inside wire and save thousands of dollars in maintenance costs. Users can place and replace terminals and telephones in any workstation at will and without fear of bringing down their entire communications network.

HOW CAN YOU MAKE THE MOST OF THE OPPORTUNITY AVAILABLE?

By knowing your specific needs, what questions to ask, and by qualifying the installation vendors you can increase your network's flexibility, add lasting value, increase your ability to support your users, and reduce maintenance costs.

Below are some needs analysis questions specific to individual users or groups of users in a building. (Or, if your business is large, individual departments within your business.) The answers to these questions can significantly shape the design of your office network.

1. Are the users known?
2. What are the users' communication requirements, both near term and long term?
3. Will the building be leased or owner occupied?
4. What are the system requirements?
5. What environment (indoor, outdoor, aerial, underground) is required?

6. What are the electrical code requirements?

7. Do you have a choice on cable raceways?

   In a new building almost any raceway system can be installed. In existing buildings, the choice is much more limited.

   If underfloor raceway is used, then access is the primary concern. Not only for primary installation, but more importantly for follow on installation or modification.

8. What is the number, size, and location of the wiring closets?

   These are the primary cable distribution centers for office telephones and data communications equipment.

9. What are the local building codes?

   Building codes are commonly different for old and new construction. Will the media you are considering comply with these codes?

Cost Factor Notes:

- The cost of adding additional or unplanned cabling is not limited to the cost of materials and labor.

- The cost of adding cables to existing buildings, compared to installing cables during initial construction of the building, increases after the ceiling is installed and increases more after the building is occupied.

- Additional time is lost due to disruption in the work place, dirt and noise associated with drilling holes in concrete and dry wall, and a general loss of productivity among office workers.

- Visual appeal of "after construction" installation may not be as great in pre-planned installations.

It is possible that when all elements of each office system are evaluated, there is opportunity for economies of scale in choosing a network backbone that will accommodate some or all of the system requirements. Large savings can result both in the initial installation and the subsequent maintenance and support.

Below are needs analysis questions specific to types of systems in a building (such as telephones, security, environmental controls, life safety systems, data terminals, PC LAN's).

1. What is the type and the number of cables each system uses and how are they configured?

2. What building and electrical codes apply?

3. What security measures are required for cable, data, and equipment?

4. What is the building environment required for each system?

5. Will the interior layout be stable?

6. How many devices does each workstation require?

7. Is it reasonable to expect that the business (and hence the workstation requirement) will grow?

8. What is the incremental cost of adding "pre-wires" to the office network design?

9. Is employee training adequately provided for? Can some workstations or conference rooms be reconfigured for training new employees (or old employees on new systems)?

HOW DO YOU KNOW THAT YOU HAVE THE RIGHT VENDOR?

There is no secret to installing a good premise wiring system. It is done all the time. A good installation is merely one aspect to a complete and successful premise cable plant. More important than installation is a thoroughly researched network design. More important than installation is documentation of the job when it is complete. More important than installation is how you the customer feel about the completed job. Office wiring is the life blood of your business. It is important to do it right. Below are some questions to ask yourself and your vendor.

1. Does the design cover all aspects of my requirements?

2. Is the design flexible enough to accommodate future requirements?

3. Is the designed network large enough to accommodate growth?

4. Do I understand my own requirements enough to evaluate the design? If not, a qualified professional should be retained to provide this service.

5. Is the vendor willing to educate and train me on the features of the cable network to reduce the "surprise" factor?

6. Is the installation work professional?

   a. Do the installers appear to know what they are doing?

   b. Does the installation have a professional appearance? Are all cables out of site where possible and dressed down appropriately where it is not possible to hide them?

   c. Are installers polite to office workers who are present and do they exercise care in keeping the disruption to a minimum?

7. Is the network thoroughly tested before being turned over to the customer?

8. Does the installation meet the design specification?

9. If I want to maintain this network myself, do I have sufficient documentation to do so?

10. Is documentation complete enough to allow efficient maintenance of the network or must I pay to re-train the vendor every time the network requires service?

Once the vendor has done his job, here are some questions to ask yourself about the installation.

1. Does the vendor's design match the specification?

2. Did the vendor perform the work in a timely fashion and meet the production schedules?

3. Does the vendor document his work?

4. Was the network 100% when it was turned over to the me?

5. Is the vendor easy to work with?

6. Can this vendor be recommended to colleagues?

7. Does the vendor make me feel comfortable with my installation?

WHAT IS THE LIFE OF YOUR WIRING SYSTEM?

Wiring is not a dynamic part of your office environment. Once it is in place it does not change, nor is it subject to stresses and strains as it lays within the walls or furniture panels. Wire ages well. The only points of wear are at the user interfaces; the patch panels or the blocks where the wire makes its appearance at the workstation. These connectors have a limited designed life of about 150 matings per connector. You will get fewer matings if line cords and patch cords are roughly handled, or are used on a fully loaded patch panel. The strain relief mechanism for these connectors is designed for the weight of the cable itself and modest mating efforts. If the patch cord you are moving is under 30 other patch cords the impact on the life of the cords and patch panel appearances could be significant. With these factors in mind, a little quick math should reveal how soon you can expect some deterioration in your wiring system based on the level of activity in your office environment.

Prewired locations experience no wear until they are utilized. The materials used in connectorizing wire do not oxidize quickly, nor is there much deterioration in the wire itself. If you decide to utilize a prewired location it should work the first time, even after a year, or two. If it doesn't, chances are that it never was suitable for communications. 100% testing can ensure that your prewires are an asset, not a worry. Insist on it. If the vendor warrants the work a prewire problem will be fixed free of charge, but that doesn't help you in an emergency.

Documentation of your cable plant is the most important factor in ensuring its long and full utilization. It is highly probable that the wiring system will outlast several managers. A good set of documentation will reduce the learning curve for each successor, reduce the number of "shoot-from-the-hip" quick fixes, and reduce the number of wiring man hours involved in any office move. Documentation and labeling of all workstation appearances can eliminate countless hours of hunting for communications problems. Documentation of all computer ports and patch panel appearances can help make large systems manageable.

THE BOTTOM LINE

Our recommendations for anyone responsible for a major internal office reshuffle, a location change, or modernizing the current facility are these:

Know your office system requirements.

Consider all those requirements in the premise wiring design.

Make sure your vendor builds a system that can be maintained with the least amount of cost and effort.

Recognize office system maintenance costs in your budgeting process.

Insist that your office wiring system is thoroughly tested and documented. A "pre-wired" location has value only if you know where it is going and from where it has come. Don't pay to re-train your service technician on your network every time you have a maintenance problem.

Design for the maximum occupancy of the office space, not what you currently plan to use. Good businesses have a way of growing. Wouldn't it be nice if your premise wiring network didn't get in the way of that growth?

Procedure for Gaining Control of an X.25 Network

#4660

Kimberly D. Weinmann
Hewlett-Packard Company
Colorado Telecommunications Division
5070 Centennial Blvd.
Colorado Springs, Colorado 80919

Taking control of an X.25 network through maintenance and troubleshooting

Network Management

Management of an X.25 packet switching network is a critical job. Much reliance is being placed on X.25 networks by a growing number of companies. Network control is the goal. Or putting it in more realistic terms, minimizing downtime and maximizing customer satisfaction is what you strive for. Problems must be avoided and, when they occur, must be identified and quickly resolved. Organized, efficient network management is the best way to achieve your goal of an available, efficient X.25 packet switching network.

Network Management consists of five major areas of responsibility

- network maintenance
- performance optimization
- network planning
- accounting
- network security.

This paper will focus on the area of network maintenance in general and troubleshooting in particular. For information on the other four areas of responsibility, refer to the HP product note titled "X.25 network performance analysis", publication number 5952-5120.

Proactive and reactive methods of network maintenance will be discussed. Proactive maintenance is the ability to detect and foresee problems before they occur, preventing downtime and user difficulty. Reactive maintenance involves troubleshooting problems as they occur, minimizing downtime and customer dissatisfaction for each occurrence.

Optimally, your network should appear transparent to your users. In reality, many things can suddenly make your X.25 network the center of attention. Many things can disturb the data flow across an X.25 line. Quality of the lines, noise causing transmission errors, the quality and reliability of your equipment itself, are all possible sources of problems.

A network manager should have the ability to foresee problems and resolve them in a timely manner. Effective maintenance techniques, both proactive and reactive, are an integral part of network management.

Network Control

In network maintenance, there are several questions you should ask yourself to determine the importance of being in control of your network.

- What is the impact to your business when your X.25 network is down?
- Does your company/supervisor expect you to insure 99.9% network uptime?
- Are your users nonproductive when the X.25 network is down?

If you're not in control of your network, you could be getting complaints from both management and users.

Let's define major components of control for an X.25 network.
- high customer satisfaction
- low network downtime

In an effort to pin down the degree of control you currently have over your network, ask yourself the following questions.
- Are you able to tell if the line is degrading and will soon cause users to complain?
- When a user complains, are you able to locate the problem in a minimum amount of time?

If the answer to either or both these questions is no, chances are good that you could be in better control of the network. As shown in Figure 1, control is directly related to your ability to maintain and troubleshoot your network. Network control results in customer satisfaction and minimal network downtime.

Figure 1

Increasing network maintenance results in high network control which minimizes network downtime and customer dissatisfaction.

Taking Control of Your Network

The road to controlling your network involves an active role in maintaining the network. This entails locating problems before they are noticed by network users, and troubleshooting those that do get noticed so downtime is minimum. Procedures for network maintenance are discussed below.

1. When setting up your network, use high quality equipment, especially at the physical level where problems occur most often. The better the quality of your equipment, the less likely that it will be the cause of problems.

2. Know the characteristics of your network when it is operating correctly. This will enable you to easily recognize problems when they occur. Fingerprint your network. A fingerprint is the result of a set of tests run against your network's lines during proper operation. Examples of fingerprint tests are percentage line utilization during peak and nonpeak hours, and percent information frames with bad frame check sequences. The tests can be easily repeated on a line when a problem occurs. The results can then be compared with the fingerprint for that line. This will verify that the problem is with the network. If the results match the fingerprint, the problem is probably not with the network. The HP 4954A protocol analyzer and the HP 18370A network performance analyzer can be used for such fingerprinting. User misunderstanding or improper use of the network could be possible culprits to the problems you experience.

3. Use proactive maintenance methods to spot problems before they cause performance degradation for the customer. This is called performance analysis and allows resolution of problems with no downtime. There are two basic types of performance analysis.

   a. Monitor line utilization regularly and determine a utilization percentage limit for your lines for a set period of time. For example, set a utilization limit on a line at 60% regular traffic averaged over a one hour period. Monitor the frequency with which this limit is exceeded. If necessary, consider adding lines to your network.

   b. Monitor data transmission quality of the physical level of the network. For example, if the frame error rate (bad FCSs/#I frames) is above .1%, the situation should be investigated because it could point to a degradation in the data transmission facilities.

If it is not possible to spot a problem before your customer does, the best thing that can be done is to minimize downtime and customer dissatisfaction. This is done through troubleshooting. Troubleshooting is reactive testing, but does allow control over the amount of time the link is down by helping pinpoint the problem. This puts control of the network back in your hands. You no longer must rely on your equipment vendors to find the problem. After all, you have more incentive than your vendors to get the network up and running quickly. Common components of troubleshooting are as follows.

1. The first step in troubleshooting is to verify the problem. In some instances there may not be a network problem. It may be a user misunderstanding or an improper use of the network. It is desirable to do this verification from the data center. For example, if a call is not getting through, use a PAD and terminal in the data center to place a call to the same address as the user. If the results match the user complaint, then the problem has been verified.

2. Utilize any network equipment diagnostic information available. Many pieces of network equipment have built-in diagnostics which can help you determine where to begin in the troubleshooting process. The PAD, for example, provides PAD service signals to the user's terminal which gives an indication of why calls are cleared. For example, a call might be cleared due to network congestion and the PAD will notify the user through the PAD service signals. Also, some modems have diagnostics built in which give an indication of problems on the phone line. Make sure you utilize the diagnostic information already available to you.

3. Now it is time to pinpoint the problem using troubleshooting techniques. The faster troubleshooting can be done, the faster the problem can be fixed. This results in better control of the network. The steps of network troubleshooting will be discussed in the next section. Systematic troubleshooting will greatly increase the control you have over your network.

4. Once you have pinpointed the problem by troubleshooting, it must be fixed quickly. If you rely on vendors to fix problems, you can maximize your credibility with these vendors with the ability to give them as much information as possible about the problem. You should know what information each vendor needs when you place the service call. For example, if you have isolated the problem to the phone line, you should have your circuit number and analog test measurements ready when you call your carrier company. This gives you credibility in the carrier's eyes and will expedite problem resolutions.

Control Through Troubleshooting

As mentioned in the last section, an important step in network maintenance is to spot problems before they cause a problem for the user. Not all problems can be caught in this way and others may slip by. In these situations, the next step in network maintenance is to troubleshoot and pinpoint the problem so it can be corrected quickly.

Trouble reports from network users take a variety of forms. Some of these are
- no response to an input
- garbage on the terminal screen
- inability to place a call which presented no problem in the past
- jobs aborted for no apparent reason.

Once a problem has been reported by a user, you must verify and then isolate the source of the problem so it can be fixed. The steps to problem isolation are
a. level 1 (physical interface) troubleshooting
b. level 2 (link or frame level) troubleshooting
c. level 3 (packet level) troubleshooting.

Troubleshooting in the order shown above allows you to eliminate problems in an organized and consistent way. In some situations, it may be best to skip one or two steps. For example, a user may have a call cleared and get a PAD service signal displayed on the terminal indicating that facilities requested were not available from the network. This points directly to a level 3 problem. There is no reason in this situation to start at level 1.

In most situations, though, when the problem is not as obvious, you should start at level 1 and proceed logically through the outlined steps until the problem is found. Because many problems are caused by the physical interface, this method usually finds the problem quickly and saves time.

The type of troubleshooting performed at each level will be nonintrusive. Nonintrusive testing does not interfere with data traffic on the line. This is important because even though one user may be having a problem communicating, others on the same line may have no problems. You don't want to prevent everyone on the line from communicating by downing the line to do intrusive testing. You may encounter a few difficult problems which will require intrusive testing when nonintrusive fails. Troubleshooting involving intrusive testing uses tools such as bit error rate testers and X.25 emulators.

In order to facilitate nonintrusive testing, a patch panel is suggested. This device provides a port for each line in your network. Plugging into a port via the patch panel allows nonintrusive monitoring, without breaking the connection between DTE and DCE. When connecting a piece of test equipment to a line via a patch panel, the term "patched in" is used.

Troubleshooting steps are discussed in some detail in the following sections. For an outline of this process, refer to the flowchart in Figure 2.

Figure 2

Troubleshooting in an orderly manner allows problems to be located quickly.

Troubleshooting Steps

Level 1 (physical interface) troubleshooting

X.25 at level 1 defines the mechanical, electrical, functional and procedural characteristics to activate and deactivate the physical connection between the DTE and the DCE. The item transferred by this level is a BIT. RS-232C, V.35 and X.21 are the level 1 specifications used to define the interchange circuits for signal ground, data send, data receive, control and timing circuits.

Because RS-232C is the most commonly used interface, RS-232C lead names will be used in this discussion. V.35 and X.21 have similar leads with similar names.

The reason to start the troubleshooting process at level 1 is that the physical interface is a common source of network problems. After installation, the higher levels of the network structure malfunction less often. Also, it is fairly easy to determine if the basic functions of the physical interface are working or not. One of the most common problems at level 1 is a break in the physical connection. This could be due to many factors, such as a down phone line or improper cabling.

If the appropriate interface leads are up, the physical connection is probably good. By the leads being 'up', we mean their status when the network is operating properly. The way to determine if these leads are up is to monitor them. You need to look at both sides of the connection, DTE and DCE. The leads to look for from the DTE are:

Data Terminal Ready	- DTR	normally on when DTE has power
Ready to Send	- RTS	normally on if DTE can send
Transmit Data	- TD	toggles if DTE is sending
External Transmit Clock	- ETC	clock not always used

The leads to look for from the DCE are:

Data Set Ready	- DSR	normally on when DCE has power
Clear to Send	- CTS	normally on if DTE can send
Carrier Detect	- CD	on indicates DCE can send
Receive Data	- RD	toggles if DCE is sending
Receive and Transmit Clocks	- RC, TC	toggles if the clock is working

The two tools most commonly used for lead monitoring are breakout boxes and protocol analyzers. The HP 4951C and HP 4952A protocol analyzers can provide both with the full breakout box contained in their RS-232C interface pod.

Breakout box

A breakout box's primary function is to monitor leads. It uses LEDs to do this. Each interface lead is represented by a red and green LED. When the breakout box is patched into the line under test, the lights will indicate on (with a green LED) or off (with a red LED) conditions for each lead. If no LED is lit, it could indicate a cable problem. If an LED is red, it could indicate a protocol problem. This is a very easy way to quickly see the status of the physical interface. For example, if there is a break in the physical interface on either side of the line, no lights will be on for that side. Or, when a phone line in the network is down, the LED for CD will be red.

Protocol analyzer

Typical protocol analyzers today,such as the HP 4951C and HP 4952A, are multifunctional tools that include a breakout box and many other tools for network maintenance. They have the ability to view all data and lead activity on the link. They can nonintrusively monitor a line and look at the data passing between the DTE and DCE. When patched into a line, the HP 4951C and HP 4952A allow data at all three levels to be monitored and decoded. At level 1, you can see the important leads in relation to the DTE and DCE data. An example of this is the data and state display format on the HP 4951C or the HP 4952A shown in Figure 3 where RTS, CTS, DSR and CD are graphically displayed along with the data from the DTE and DCE. If problems exist on an X.25 line, such as inactive leads, they can easily be seen in this display format.

If the problem is not isolated at the level 1 physical interface, move on to level 2 troubleshooting.

Figure 3
This data and lead status display allows problems with control leads to be spotted quickly.

Level 2 (link or frame level) troubleshooting

X.25 at level 2 is the link-level interface (sometimes referred to as the frame level). At this level, the procedure to access the DTE/DCE link to allow data exchanges is defined. The item transferred at this level is a FRAME. This level of your packet switching network uses a protocol called LAP-B which is a synchronous data link control protocol. This link level insures error free data transmission between the different nodes of the network.

The first type of level 2 tests to perform are those that locate physical interface problems that were not evident in level 1 troubleshooting. These are more common than level 2 problems. Nonintrusive level 2 troubleshooting can provide indications of poor line quality. These indications are

- bad frame check sequences (FCS) - indicate bit errors during transmission
- rejects (REJ) - indicate frames received incorrectly.

There are two procedures that can be used to locate bad FCSs or REJs. Read on to find out how the HP 4951C and HP 4952A can help here.

The first procedure is to monitor the line using a protocol analyzer to decode the frames and watch for occurrences of FCSs or REJs. The frame decode display format of the HP 4951C and HP 4952A are shown in Figure 4.

Bad FCSs on any of the frames flash "B" in the FCS column for these frames. REJ frames are displayed in the Type column. The HP 4951C and HP 4952A can be programmed to trigger on events (such as bad FCSs and REJs), count them, sound an alarm when they occur and highlight them in the data buffer for later viewing. This means you don't need to sort manually through pages of printouts to find problems. The analyzer can do it for you.

Figure 4
A frame level decode shows each frame on the X.25 line decoded for monitoring problems can be seen easily.

The second, and simplest procedure is to utilize the X.25 and SNA link level performance analysis package available on the HP 4951C and HP 4952A to count bad FCSs and REJs on your X.25 line. Many other level 2 events are counted and can be logged to disc for later review. Figure 5 shows the link level performance analysis in action. In this situation, bad FCSs occured on the line. REJs also occured. The X.25 line tested in this example has a problem with the physical interface. At this point, the problem must be isolated further.

Figure 5
Statistics of link-level information is useful in locating X.25 problems. Link-level events are counted by the analyzer so the user does not need to manually count bad FCSs or REJs.

The second type of level 2 tests to perform are those that locate problems caused by configuration errors and bad clocks. The most effective procedure for this is to monitor the line with a protocol analyzer, utilizing a frame decode like the one seen in Figure 4. This display decodes the address, type, sequence numbers, poll/final bit, first 9 bytes of data and frame check status for each frame on the X.25 line. Decode the frames on the problem X.25 line. If you see only RRs (receiver ready) and/or information frames, the level 2 is functioning properly. If you have already tested level 1 and found no problems, the problem is probably at level 3. If you see SABMs (set asynchronous balanced mode) and/or DISCs (disconnect) on the line, there are two situations to check for to pinpoint the problem.

1. If there are SABMs from each side of the line (DTE and DCE where DCE is defined to be the network and DTE is defined to be the subscriber) using the same address, then one of the sides has been improperly configured because each side thinks it is the DTE or DCE. For example, in Figure 6, the SABMs from both DTE and DCE are using address 01Hex. The DCE is improperly configured, because commands from the DCE should use address 03Hex.

   Figure 6

   One possible level 2 problem is incorrect configuration. The DCE here is improperly configured as a DTE, and so is using address 01Hex rather than 03Hex.

2. If the SABM/UA (UA is an unnumbered acknowledgment) link initialization process repeats over and over, the problem is that the DTE and DCE are using different clocks which are not in phase. This situation is diagrammed in Figure 7. To fix this problem, make sure both DTE and DCE are using the same clock.

Figure 7
If DTE and DCE do not use the same clock, they will not be able to communicate.

If neither of these situations is occurring , the problem is probably not at level 1 or level 2. Move on to level 3 troubleshooting.

Level 3 (packet level) troubleshooting

X.25 at level 3 is the packet or network level. The item exchanged by this level is a PACKET. The protocol at this level defines the procedures for the exchange of packets containing control information and user data between the network and subscriber.

If level 1 and 2 nonintrusive troubleshooting does not locate the problem, it will probably be found at level

3. There are two types of problems found at level 3. Those caused by system software and those which are user related. Problems caused by software are very rare. User problems such as improper addressing or network congestion are much more common. The best tool for level 3 troubleshooting is a packet decode on a protocol analyzer. An example of this is shown in Figure 8. This decode makes it very easy to see problems at level 3 because each packet of an X.25 line is decoded by packet type, quality and delivery bits, Modulo, logical channel number, sequence numbers and the more bit.

Figure 8
A level 3 packet decode displays all packets on an X.25 network broken down into all components. Level 3 problems can often be spotted using such a decode.

The first step in level 3 troubleshooting is to determine if activity exists at level 3. Using the packet decode, look to see if any packets are crossing the network.

- If there are no packets for more than 200 seconds, then, as strange as it seems, level 3 is up, there is no problem with the level 3 software, and there is no activity at level 3. This assumes levels 1 and 2 are working properly. Also, if you see only RRs (receiver readys) and/or data packets, there is no problem with the level 3 software.
- If there is activity in under 200 seconds, and it does not consist only of RRs and data packets, then there could be a problem with the level 3 software. Continue to locate the possible problem.
- If one side of the network is sending restart packets every 200 seconds or less, which are not responded to, then the side of the network not responding is down at level 3. The level 3 software possibly needs to be rebooted.
- If there are no restarts, the last situation to look for is one side of the network sending a call request which is not answered with a call confirm, but with a clear request. A diagram of this situation is shown in Figure 9. This means the call is not going through. To find the reason for the clear, look at the cause codes and diagnostic codes contained in the clear packet. This can be done with the packet decode on some protocol analyzers. There are many reasons why a call may not go through such as network congestion, invalid address or a busy number.

Figure 9

<u>If a call is not getting through the network, look at the cause and diagnostic codes for a reason.</u>

If none of the level 3 problems discussed are found, either the problem was missed during troubleshooting or the problem is not at levels 1, 2 or 3. If you are convinced there is a problem with the network, the line should be taken down for intrusive testing at the different levels. This could include bit error rate testing at level 1, DTE and DCE simulation at level 2, or network and subscriber emulation at level 3.

Troubleshooting tools available

Troubleshooting can be achieved with a wide range of test equipment. Prices range with the capability provided, from about $200 to $20,000.

The most commonly used equipment for troubleshooting is the breakout box and the protocol analyzer. Table 1 provides a summary of the capabilities of both as troubleshooting tools for level 1, 2 and 3 problems. The easier problems can usually be located with a breakout box monitoring leads. Some more difficult problems require more complex tools, such as level 2 and 3 decodes, triggering and statistics packages. It is ideal to have test equipment containing all the tools for troubleshooting. This allows the difficult and the easy problems to be found quickly every time, maximizing network control.

Table 1
Breakout box and protocol analyzer capabilities as X.25 network troubleshooting tools for levels 1, 2 and 3.

4660-21

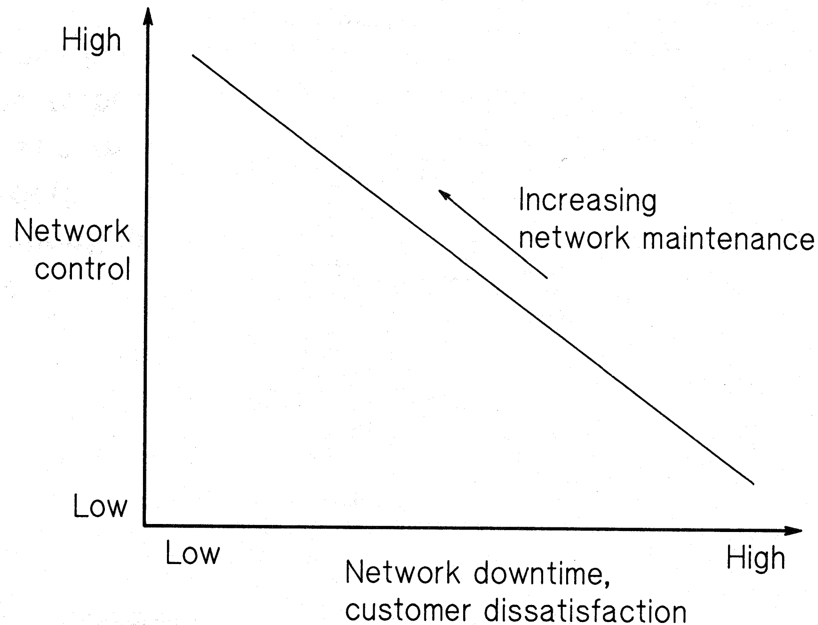


Figure 1 – Increasing network maintenance results in high network control which minimizes network downtime and customer dissatisfaction.

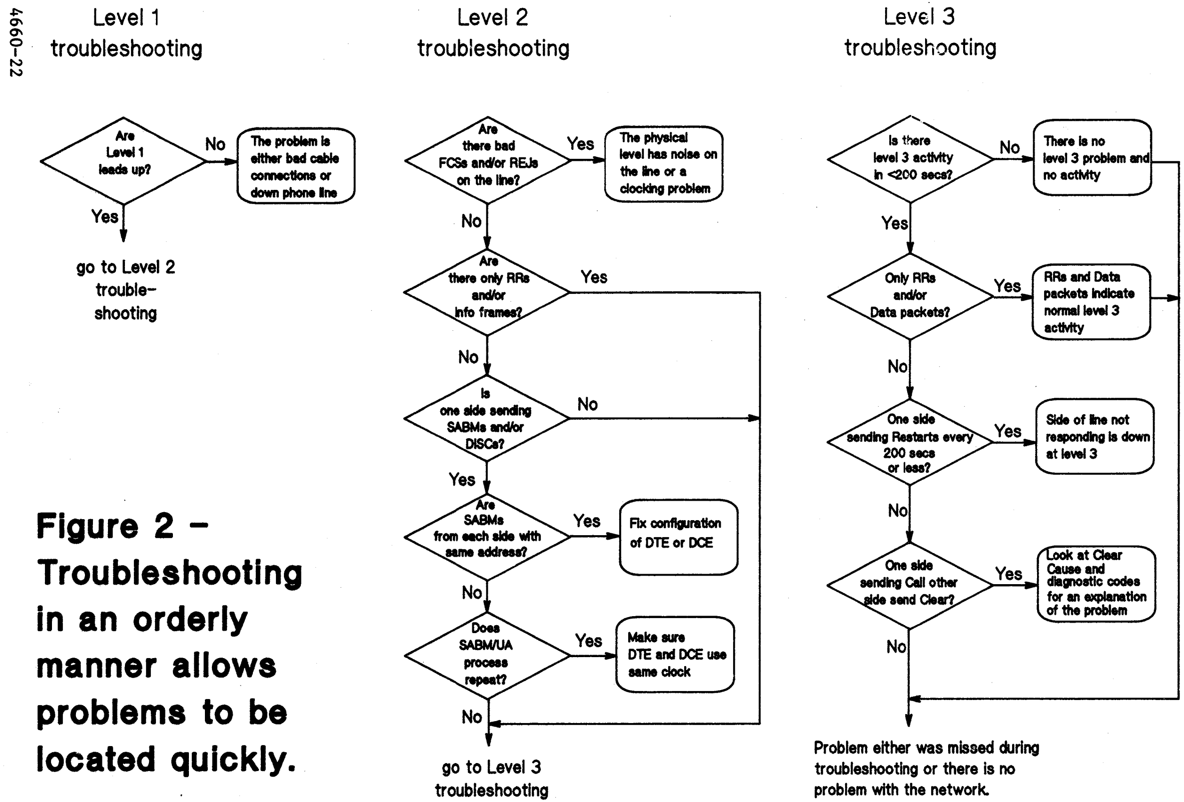


Figure 2 – Troubleshooting in an orderly manner allows problems to be located quickly.

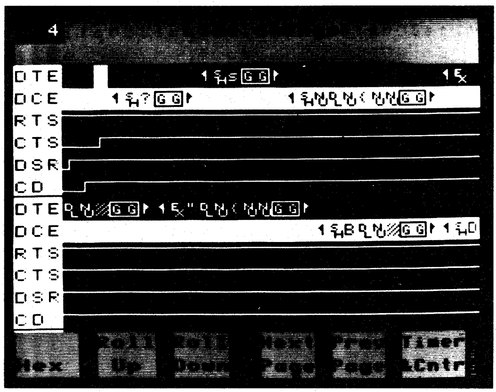


Figure 3 - This data and lead status display allows problems with control leads to be spotted quickly.

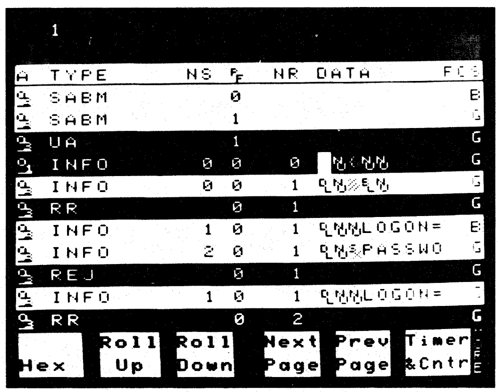


Figure 4 - A frame level decode shows each frame on the X.25 line decoded for monitoring, problems can be seen easily.

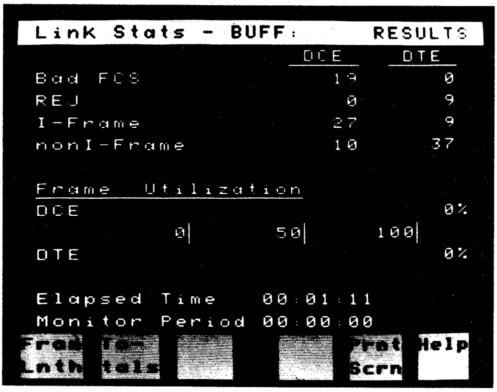


Figure 5 - Statistics of link-level information is useful in locating X.25 problems. Link-level events are counted by the analyzer so the user does not need to manually count bad FCSs or REJs.

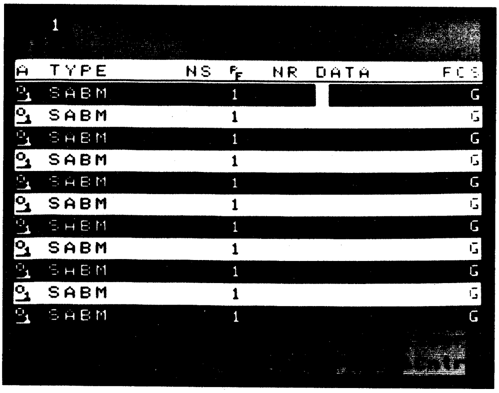


Figure 6 - One possible level 2 problem is incorrect configuration. The DCE here is improperly configured as a DTE, and so is using address 01Hex rather than 03Hex.

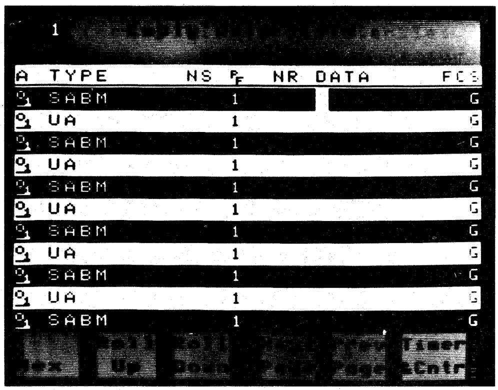


Figure 7 - If DTE and DCE do not use the smae clock, they will not be able to communicate.

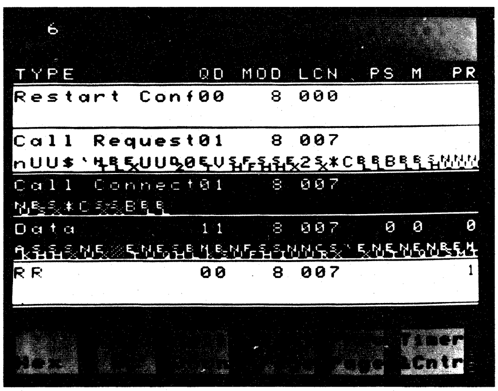


Figure 8 - A level 3 packet decode displays all packets on an X.25 network broken down into all components. Level 3 problems can often be spotted using such a decode.

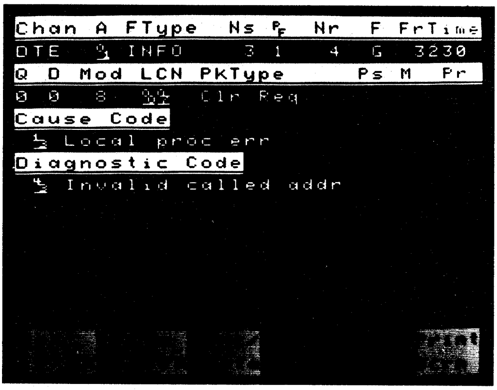


Figure 9 - If a call is not getting through the network, look at the cause and diagnostic codes for a reason.

4660-27

Troubleshooting Tools

	Level 1	Level 2	Level 3
Breakout box	Monitor lead activity with LEDs.	XX	XX
Protocol analyzer	Monitor lead activity with LEDs or on a data and lead status display.	Monitor, trigger decode level 2 frames. Perform statistics.	Decode level 3 packets.

Table 1 – Breakout box and protocol analyzer capabilities as X.25 network troubleshooting tools for levels 1, 2 and 3.

IFACE110

HP PRIVATE PACKET NETWORK
CASE STUDIES

#4661

Joelle GAUTHIER
HEWLETT-PACKARD
GRENOBLE NETWORKS DIVISION
5 Av Raymond Chanas
38320 Eybens - FRANCE

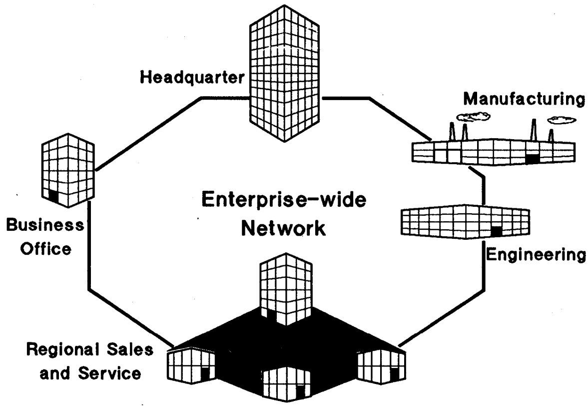
COMPETITIVENESS...

Maintaining the competitive edge in todays' changing market place is becoming more and more dependent on efficient management of information flow. Getting the right information to the right people at the right time means cutting operation costs, boosting productivity and increasing overall customer satisfaction.

Essential to optimizing competitiveness and effectiveness is the enterprise information system, which requires the integration of systems, applications, and networks in a fashion which serves the entire enterprise.

HEWLETT-PACKARD UNDERSTANDS YOUR ENTERPRISE NETWORK NEEDS

A typical enterprise wide environment reflects geographical dispersion, equipment from multivendors, rising datacommunication costs, integrated applications and information flows between Headquarters operations, Sales and Service offices, Development facilities and Manufacturing sites, each having a specialized information environment.

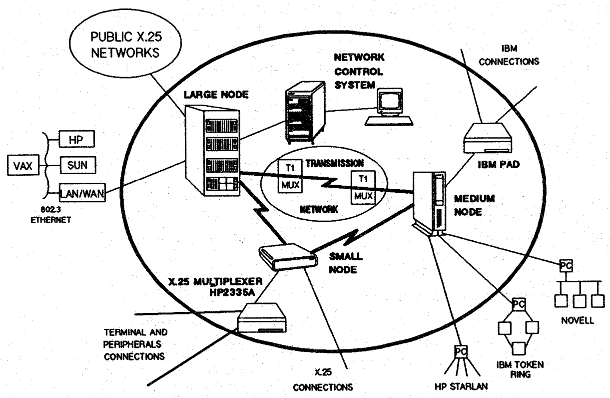


Organizations like yours need a global network that will provide complete control over these data and information exchanges. The enterprise information system needs to be cost-effective, reliable, invisible to the user and have the ability to operate in conjunction with a wide variety of computer systems, and assure a smooth growth path. Also to be fully efficient, the enterprise network has to be backed by the best possible support and service.

Hewlett-Packard recognizes the importance of an enterprise-wide network and provides such an infrastructure with Hewlett-Packard Private Packet Network (HP PPN). The HP Private Packet Network is based on the international networking standard CCITT X.25, as well as fully compatible with IBM System Network Architecture (SNA) and DECNet. It provides the opportunity to construct a secure data network, with extensive network management facilities, and a reliable, modular design which optimizes availability and cost of ownership. With HP's T-1 multiplexer support program, HP Private Packet Network users can now take advantage of the increased reliability, performance and cost savings of T-1 equipment.

The HP Private Packet Network consists of a family of switching nodes, asynchronous Packet-Assembly/Disassembly devices, protocols converters and an advanced network management system based on an HP9000 Unix mini-computer.
In addition to assuring you of vendor independence HP PPN provides you with system connectivity to T-1 multiplexers and Local Area Network (LAN) devices. HP PPN also supports through servers or gateway systems the CCITT X.400 protocols for electronic mail.

HP PRIVATE PACKET NETWORK



MAJOR COMPANIES USE HP PRIVATE PACKET NETWORK

Hewlett-Packard has installed several HP Private Packet Networks around the world and as evidence by the major users hereafter mentioned. Companies have relied on HP to provide reliable and secure enterprise network capabilities to manage all of their multivendor communications.

SES, Hertz-Europe, SGS-Thomson Microlectronics, Longs Drug Stores and Hewlett-Packard Corporations are some of the companies which implement HP Private Packet Networks to manage their multivendor, enterprise-wide communications.

STOCK EXCHANGE OF SINGAPORE

At SES, an HP Private Packet Network links 30 member brokerage firms throughout Singapore to manage trading volume up to 118 million shares per day.

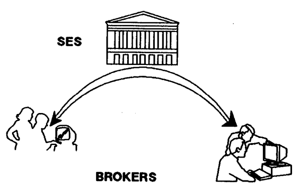
"We selected the HP X.25 Private Packet Network because there can't be any errors or downtime in our business," said Paul Phillips, MIS director of SES. "If a line fails, HP's network finds another path without interruption and report what's happening."

"Our business mission is to increase transaction volume while lowering the cost of doing business so that SES can become an industry leader in the financial business. HP's networking capabilities are helping us to achieve our goal," said Phillips.

STOCK EXCHANGE OF SINGAPORE

BUSINESS NEED : ENHANCE ATTRACTIVENESS OF SES AS A PLACE TO INVEST

- HP PPN INSTALLED END 87
- 30 SITES DEPLOYED



Business

- Stock Exchange

Business Issues

- No Downtime Tolerated
- Ineffective Access of Brokers to SES
  - Unreliable Lines
  - No Rerouting
- Lack of Ressources to Develop a Network

HP PPN Benefits

- Increase Transactions
  - Total Reliability
  - Fast Response Time
- Lower Costs

HERTZ-EUROPE

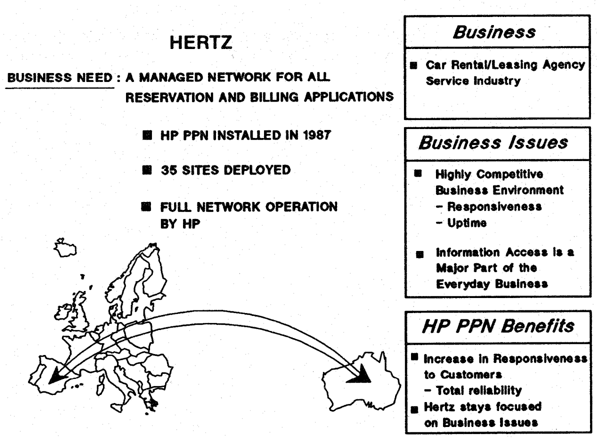
"Hertz has a car rental outlets in the airports, major cities and key towns of over 130 countries. Pulling these outlets together into cohesive unit is a mammoth task.

"With Hewlett-Packard's help we have started to network all our European rental outlets together. Our aim is not only to be identified as one company - HERTZ - but to act as like one company as well, providing a consistently excellent level of service across the board.

"We are already seeing the benefits of networking in the UK, France, Germany, Italy and Switzerland. It is enabling us to manage our fleet more efficiently. Our marketing department can see the immediate impact of promotions and price changes, outlet by outlet, and make adjustments as necessary. And our customers are seeing the benefits of improved reservation services and standardization of documentation and forms.

"HP supplies the hardware and telecommunications systems, negotiates with the local country PTTs on our behalf, and also monitors and supports our network from its Bristol offices.

"Working with HP is enabling us to grow our network a lot faster than we would otherwise be capable of doing. So our customers are benefiting that much quicker", comments Joe Bournat, Director of Management Information Systems at Hertz Europe Limited.



"HP's extensive Private Packet Network support is one of the several reasons why Hertz-Europe chose the company to manage our enterprise-wide network.

"Hertz is in the business of care-hire, not in setting up and managing European information systems. When management decided to extend the network to Europe, HP readily came to mind. The fact that HP has recognized network experience and could serve as a single vendor in managing the entire Hertz project clinched the deal. Because our entire network is managed by HP, Hertz has eliminated operating costs for administration, training and personnel," said Bournat.

HP SUPPORT AND OPERATION SERVICES

Continued value of your information system investment is guaranteed by Hewlett-Packard's comprehensive support envelope. Recognizing the specific requirements Enterprise-wide Networking creates, Hewlett-Packard has supplemented its worldwide outstanding support organization to include centers of expertise in Enterprise-wide networking.

HP provides private network support to customers including Hertz on a worldwide basis through HP's Customer Network Centers (CNCs) in Atlanta, Georgia; Bristol, England; and Singapore. Staffed by a team of networking experts, each center offers network-management support services on an hourly or full-time basis. By relaying on HP to support a company's private network, users can benefits from cost savings in personnel and training.

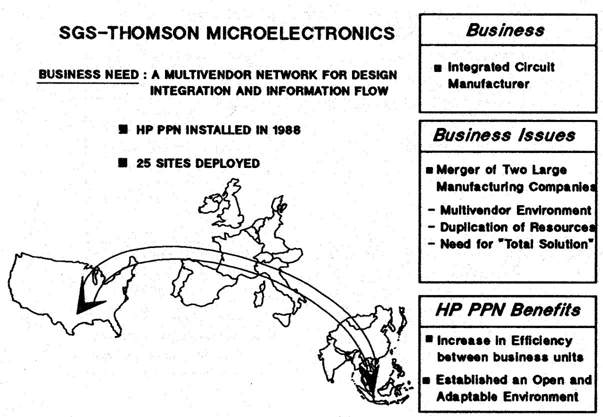
HP's support services range from consulting, network design, project management, training and actual 24 hour network operation.

SGS-THOMSON MICROELECTRONICS

For chipmaker SGS-Thomson microelectronics, deciding which vendor to select for the company's X.25, private packet-network solution boiled down to five basic requirements -- multivendor connectivity, network control, security, reliability and cost effectiveness.

SGS-Thomson Microelectronics selected HP among five major networking vendors to design and install its pilot network to connect multiple-computer systems from HP, DEC and IBM; and provide complete network management from the network.

The SGS-Thomson Microelectronics pilot network was operational within two weeks. Connectivity improved immediately because terminals previously connected to a single computer now could switch among HP and DEC computers for specific applications. With fewer communications lines needed to connect multivendor-computer systems, company wide communications became more reliable and available on the network. SGS-Thomson Microelectronics was ready for full deployment at the conclusion of the pilot network.



LONGS DRUG STORES

Longs owns and operates 240 retail drug stores through six western states. A key element in Longs business strategy is to set up stores as independent operations that can be networked so that each store can purchase and price its own inventory, yet share data communications among headquarters and sister stores.

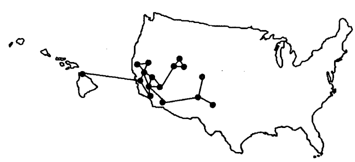
"Because we run a decentralized operation, a network that provides peer-to-peer capabilities and high-performance communications is extremely important," said Bill Gates, director of information services for Longs.

"Since a hierarchical implementation did not make sense to us, we looked at other networking alternatives that could provide a reliable and efficient means of distributing our communications. For Longs Drug Stores, HP's network provides a total solution by allowing more timely management information to be transferred among our stores and home office."

LONGS DRUG STORES

BUSINESS NEED : A NETWORK PROVIDING PEER TO PEER CAPABILITIES AND HIGH-PERFORMANCE COMMUNICATIONS

- HP PPN INSTALLED IN MID' 88
- 15 SITES DEPLOYED



Business

- Retail Chain of Drug Stores

Business Issues

- Stores working as Independant Operations
- Information Shared among Headquarters and Sister Stores
  - Rapid Access to Information

HP PPN Benefits

- Timely Management of Information
  - High reliability
- Improve Customer Service
  - Credit Card Verification

HEWLETT-PACKARD

Hewlett-Packard few years ago, begun long range planning of it's network infrastructure. A needs assessment determined that the next generation internal network needed to satisfy the following needs :

* **Supportability** - that it should be based on standards;

* **Reliability** - that it should be able to accommodate equipment component and link outages and provide alternate paths between corespondents;

* **Expandability** - that it should have no constraints to expansion for additional sites or growth of traffic;

* **Adaptability** - that it should be based on a technology which accommodates growth and change, and can serve both interactive as well as batch users;

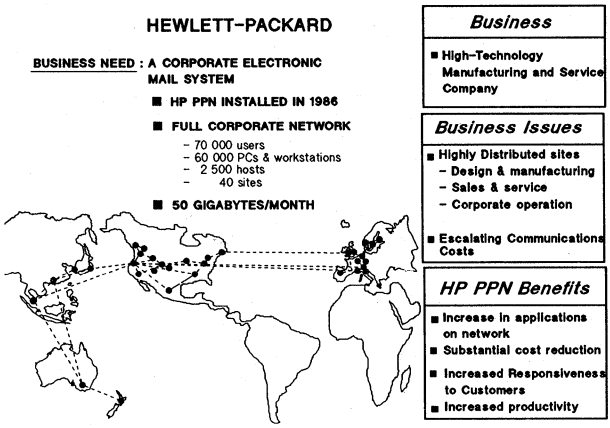
* **Efficiency** - that it allows for the efficient use of circuit bandwith given the performance requirements of both batch and interactive users.

It was concluded that X.25 packet network technology best satisfied these needs.

Deployment of X.25 networks began in the sales regions, where standalone Dynapac switches and HP2334 asynchronous PADs were installed. In view of the limited functionalities of such switches and the tremendous traffic increase, HP started in second half of 1986 the installation of HP NET a backbone class X.25 network based on HP PPN.

As this backbone grew, the smaller networks migrated from public network interconnection to become HP NET tributary networks. HP NET has grown to more than 40 large nodes around the world. The private packet network today connects over 70,000 users through the connections of more than 60,000 personal computers and workstations and more than 2,500 hosts. HP NET is carrying over 50 gigabits of data per month.

Electronic mail, sales order administration, engineering as well as purchasing and inventory management are the applications running over the network.



In the U.S., and increasingly in other countries where high capacity private line circuits are becoming available, the HP Private Packet Network is set up on top of a transmission network, made of T-1 multiplexers which carries voice traffic. The T-1 circuits having enough idle capacity to carry data lines at significant lower cost than single leased private line circuits, the HP NET topology took advantage of this inexpensive bandwith to link major nodes.

Incorporation of non-native X.25 traffic into HP NET X.25 network is also influence by the marginal cost of bandwith, that is , the availability of T-1 or other high capacity circuits can influence the decision to se protocol converters to convert traffic to the X.25 protocol to be carried on a packet network. Within HP, where the cost of private line circuits is low, most commonly in the U.S. because of T-1, SNA or TCP/IP lan internet traffic is often carried over separate T-1 channels rather than being converted to X.25. This is strictly a price/performance issue, not one of functionality. Where the cost of circuits is high, SNA and TCP/IP internet traffic is converted to X.25 and carried over HP NET packet network.

HP NET is managed cooperatively from several locations within HP. There are network administration groups in Palo Alto, in Geneva and Honk Kong. Each of these administrative groups has a Network Operator Console (NOC) from which they can configure and monitor the entire network, and has primary responsibility for supporting the sites in their region of the world. Network configuration changes, user support, billing policy, and operations monitoring is performed directly by these group for their users. Additionally, operations support for event management is provided by the Customer Network Center in Atlanta for two shifts per day. This relieves the small teams in Geneva and Hong Kong from having to monitor individual network events during their daytime shift.

The implementation of HP Private Packet Network has contributed to improvements in the assembly of and access to financial and analytical data, reductions in production cycle time, shortened product development cycles and enables the company to reduce overall inventories.

In retrospect, after the break-even point, annual network savings started at about $7 million, and are increasing annually. In fact, despite more than a three fold traffic volume increase over the last two years, leased line expenses actually declined in FY'88.

SUMMARY

Reading and hearing about such success stories from your industry peers, suppliers or even competitors, are powerful incentives to get start on a similar project. More and more companies are discovering the benefits of wide area networking. A private packet networks means better control, security, and the X.25 standard ensures multivendor connectivity.

HP PPN can help you to be more competitive. And HP will work with you through all phases of your HP PPN implementation -- from Planning, design, and training to operation and maintenance. Our reputation for customer satisfaction is unmatched because we back our solutions with top-rated support and services -- worldwide.

Contact us today to find out exactly how Hewlett-Packard Private Packet Network, the heart of the enterprise-wide networking solution can make your organization more competitive.

The Transition from TCP/IP to OSI Networks

April 21, 1989

Dave Morse
 # 4663

Colorado Networks Division
3404 East Harmony Road
Fort Collins, CO 80525-9599

The Transition from TCP/IP to OSI Networks

The industry-wide movement to OSI protocols is underway. HP has been a long term proponent of the OSI architecture and OSI protocols. This paper will discuss motivating factors for the OSI movement and HP's strategy for providing OSI products.

HP began its movement to OSI in 1983 with endorsement of an OSI architecture for all future HP networking products. This represented the first step in the movement from a proprietary architecture (DS or Distributed Systems) focused on connecting HP systems to each other to an industry standard architecture implementing standard protocols focused on multivendor connectivity. Today, HP's dominant networking products (NS and ARPA services) are based on the OSI architecture and defacto standard protocols. The result is excellent multivendor interconnectivity based on the protocols most commonly implemented today. In the future, HP's dominant networking products will be based on OSI protocols. During the transition period, HP will maintain the strong multivendor connectivity available today.

The movement to OSI protocols will not take place overnight. There are numerous practical limitations slowing the movement. Applications drive the need for networking services. Today's applications use TCP/IP based services like NS and ARPA. New applications will be developed based on OSI services, but they will grow gradually. Support for existing applications will be an important part of the migration process. Many existing systems will not be upgraded to OSI protocols. Modification of existing applications to use OSI services may be difficult or impossible.

HP believes that the movement to OSI protocols will be characterized by a long period of coexistence of TCP/IP based protocols and OSI protocols. This period could stretch to ten years or longer. Accommodating this long coexistence phase is a key aspect of the HP OSI strategy.

The first phase of the OSI migration will be characterized by OSI pilot programs. This phase is occurring today. The motivation to experiment with OSI is stronger in some industries and geographical areas than in others. In this phase there is typically not a strong need for the pilot networks to communicate with the existing networks. Since the new networks are experimental, there is often a desire to keep them separate. HP has been aggressive in offering OSI products such as MAP. We plan to continue with additional products to comply with new OSI profiles such as GOSIP.

The second phase of OSI migration will be characterized by OSI subnets. OSI applications will emerge that require OSI services. Compatibility with existing applications will be important since few systems will run only OSI based applications. The HP strategy for this phase is to offer dual protocol stacks which support both TCP/IP and OSI based services. Systems configured with dual stacks can communicate with existing TCP/IP only based systems and with systems running OSI protocols. HP's current ARPA/NS products for the HP 9000 computers are examples of a dual stack implementation. These products do not provide a complete dual stack, but support both ARPA and NS services and IEEE 802.3 and Ethernet links. These links and services are supported transparently to the user or programmer. A similar approach will be used to support complete NS/ARPA/TCP/IP and OSI stacks in the same system.

The HP strategy for the later phases of OSI migration is still evolving. The goals of the strategy are clear. HP will provide compatibility with the installed base of TCP/IP based systems as these systems change over to OSI. HP will also continue to support multivendor connectivity throughout this transition period. In order to provide this multivendor connectivity, HP's transitional products must be consistent with those provided by other vendors. HP is working aggressively in various industry networking forums to ensure that we do provide this compatibility. It is clear that dual protocol stacks will play an important part in this transitional period.

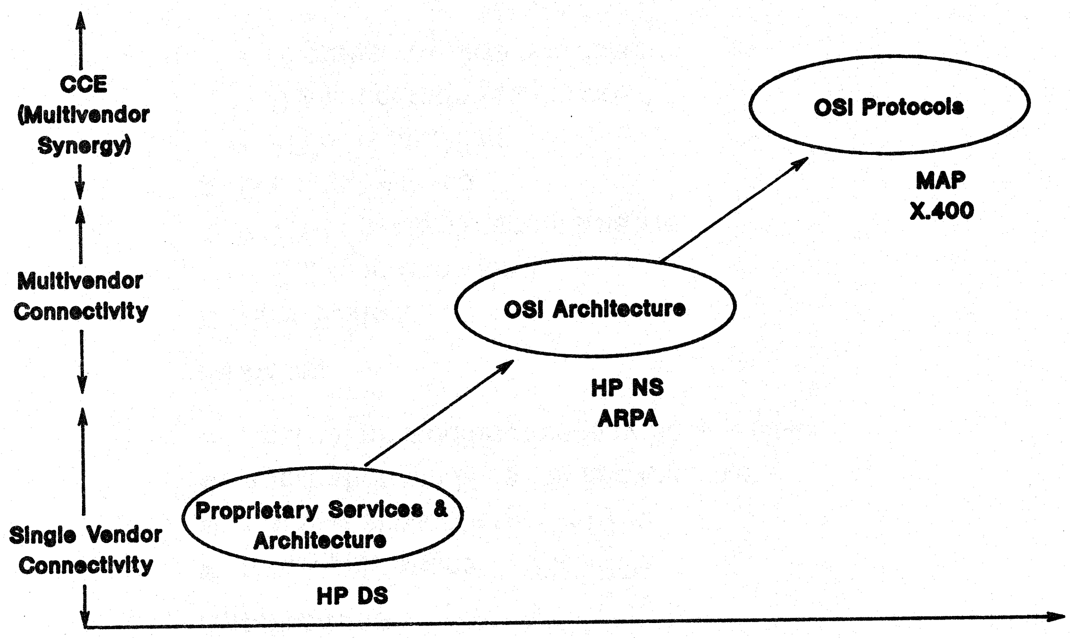
Other product requirements are less clear at this time. Both gateways and mixed protocol stacks could play a role. Gateways can be of two types, application layer gateways and lower level gateways. Application layer gateways completely covert an OSI service over an OSI protocol stack to a TCP/IP based service over a TCP/IP protocol stack. Such a gateway can connect a "pure" OSI network to a "pure" TCP/IP network. Lower level gateways convert protocols at lower levels, for example, TCP to the OSI transport protocol. Mixed protocol stacks would provide TCP/IP services (such as ARPA) over OSI transport protocols or OSI services (such as FTAM) over TCP/IP transport protocols. A combination of mixed protocol stacks and transport layer gateways could provide compatibility between networks running a single transport protocol (either TCP/IP or OSI) with both OSI and ARPA services.

At present, there are advocates in the industry for both the pure stack approach to providing compatibility and the mixed stack approach. The current debate about the evolution of the TCP "internet" used to connect numerous research institutions exemplifies this. The U.S. Department of Defense (DOD) supports the pure stack/application gateway approach. The Internet Engineering Task Force (IETF) supports mixed stacks. HP is actively involved in these debates and will offer suitable products to maintain the multivendor connectivity required by our customers.

It is likely that non-OSI services will be required to supplement the OSI services. These could be required for a number of reasons such as application compatibility, performance, or additional functionality. HP will evaluate the need for supplemental services and provide those required to maintain useful multivendor interoperability.

HP's approach to the evolution of OSI is to build on the existing OSI architecture of HP AdvanceNet and provide early availability of OSI products. HP will provide tools to ensure multivendor connectivity during the long transition phase between today's TCP/IP networks and the future's pure OSI networks. HP will offer dual protocol stacks in the early transition phase. Later in the transition, HP will offer gateways and mixed stacks to maintain multivendor connectivity. Throughout the transition, HP will work to drive the definition of standards and will implement them aggressively.

The Deployment of OSI Networks



Practical Limitations on Evolution

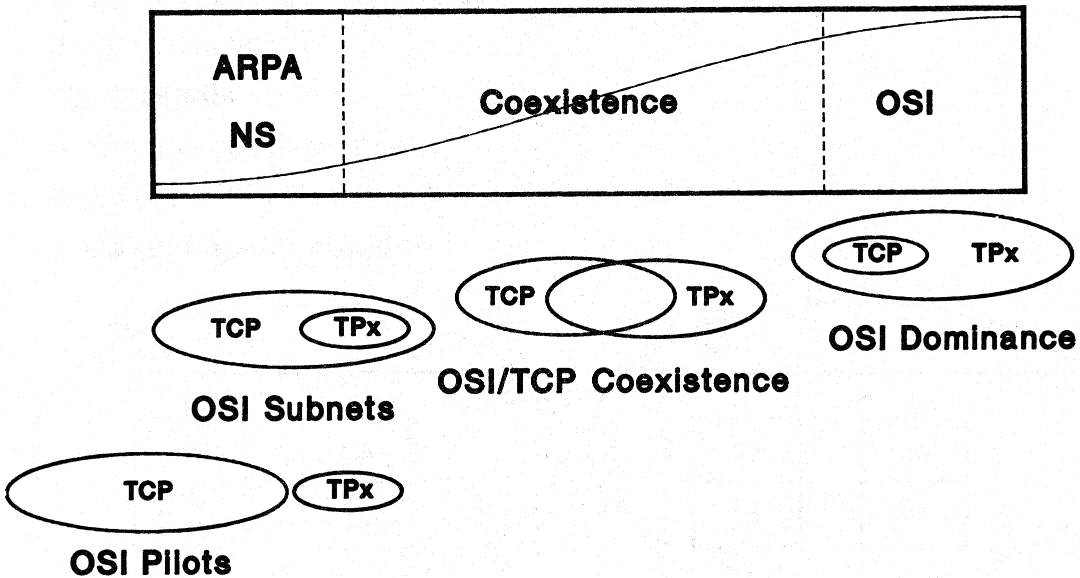
Applications

- New applications – OSI/ARPA
- Existing applications – ARPA
- Application modification is expensive
- Application modification may be impossible

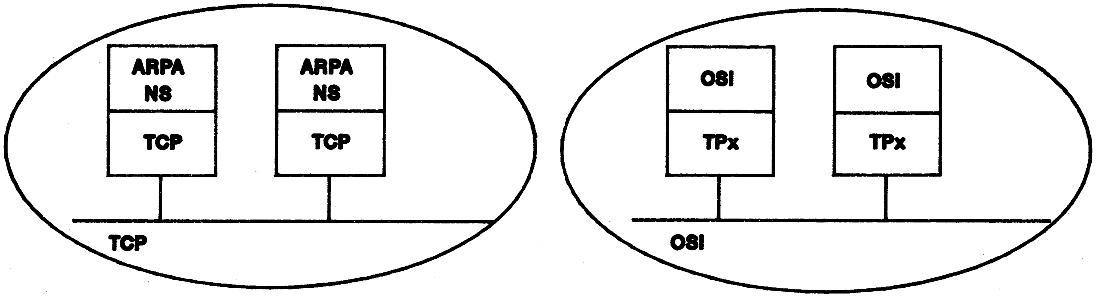
Systems

- New Systems
  - OSI recommended
  - ARPA available as alternative
- Existing Systems
  - ARPA is installed
  - OSI is upgrade (added cost)
  - Upgrade not always available

OSI Evolution



HP AdvanceNet: OSI Pilots



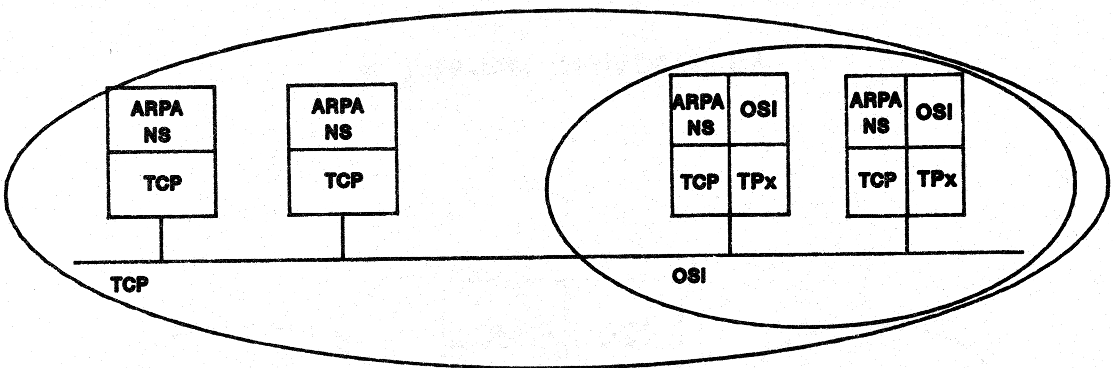
Customer Requirements

- Experimentation with OSI
- Isolated OSI networks

HP Strategy

- OSI architecture
- Aggressive MAP/X.400 program
- Continued support for de facto standards (TCP/IP, ARPA, NFS...)

HP AdvanceNet: OSI Subnets



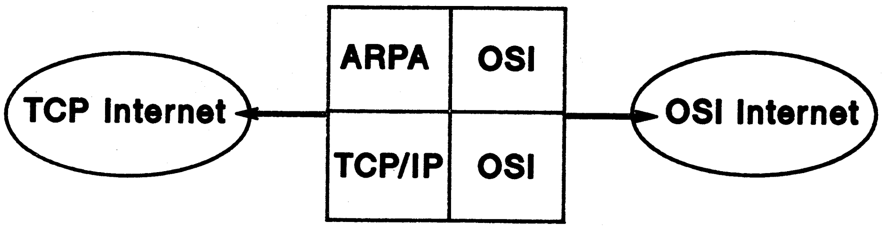
Customer Requirements

- Creation of OSI applications
- Integration of OSI with installed networks

HP Strategy

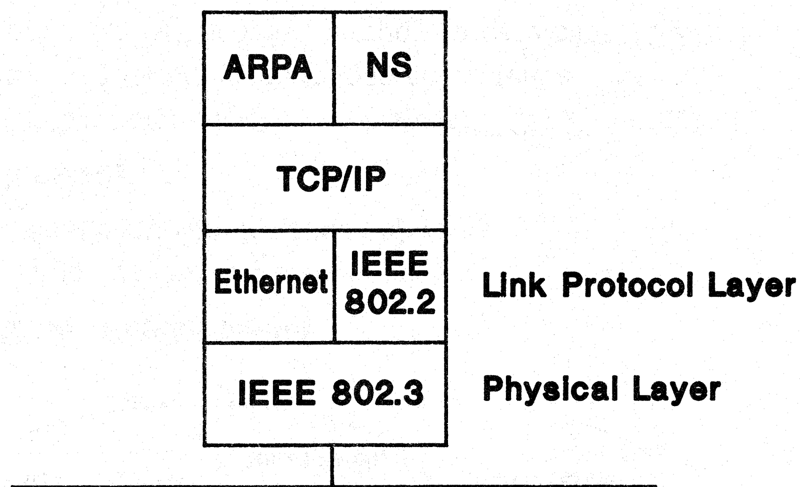
- Early availability of OSI products
- Dual stacks

Dual Stack



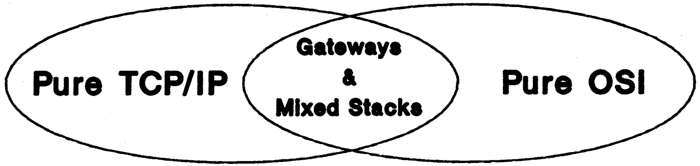
- Internet transparency

HP AdvanceNet Dual Stack



- Ethernet/IEEE 802.3 transparency
- Compatibility with old and new networks

HP AdvanceNet: Later Phases



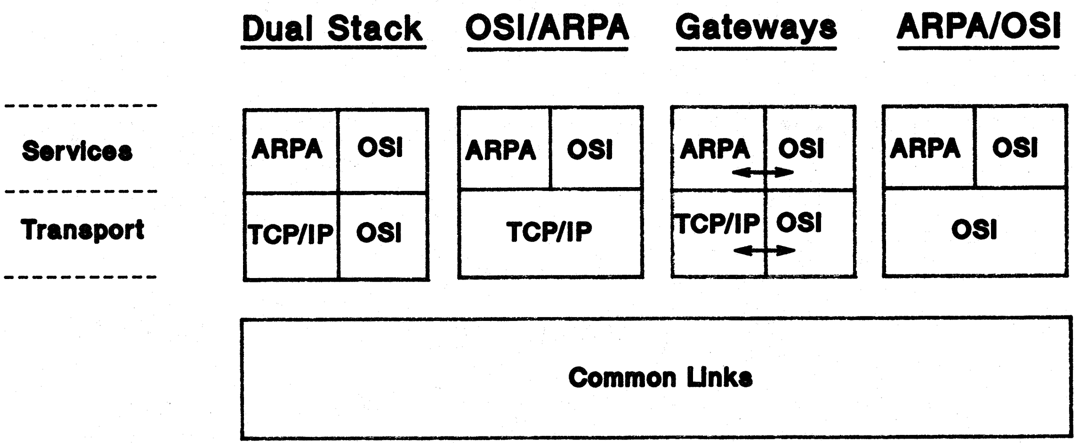
Customer Requirements

- Creation of OSI only networks
- Compatibility with installed networks

HP Strategy

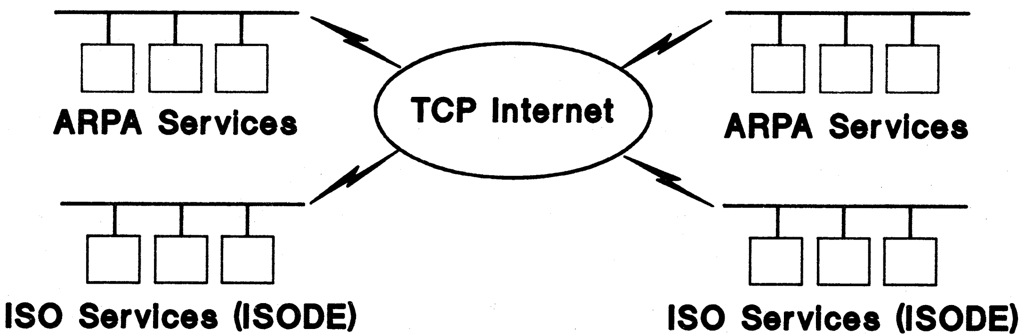
- Maintain compatibility with installed TCP/IP internet
- Provide tools to manage long coexistence
- Cooperate with industry groups to determine TCP/IP internet migration and provide appropriate products

Compatibility Alternatives



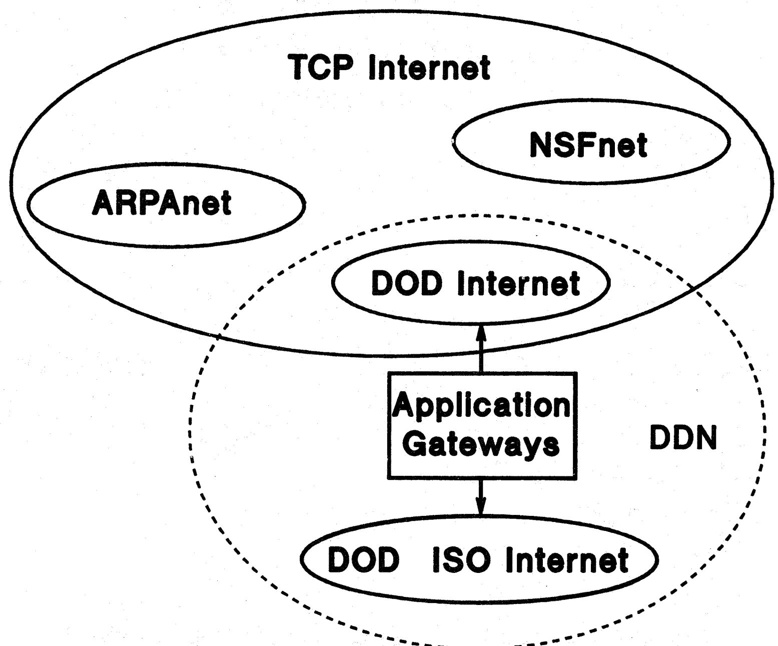
- Applications interface to services, transport is invisible

IETF RFC - OSI/ARPA

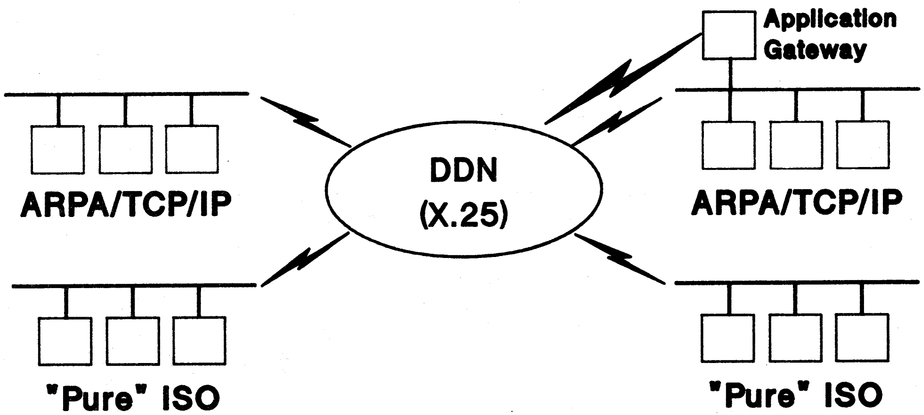


- Maintain single (TCP) Internet
- Mixed stacks

Dual Internet - Gateways

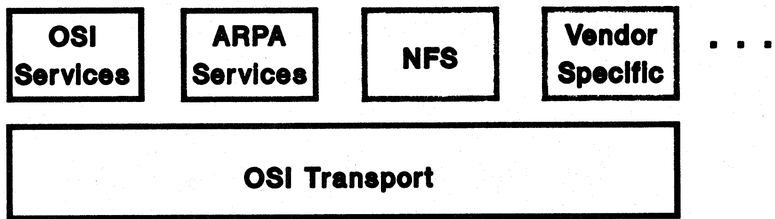


DDN ISO Plan - Gateways



- Dual Internet
- Common transport
- Pure stacks

Supplementing OSI – ARPA/OSI



- Compatibility for applications
- Performance
- Functionality
- Implementation timing

HP AdvanceNet OSI Evolution

- Build on existing OSI architecture
- Early availability of OSI products
- Provide tools to support a long coexistence phase
- Provide gateways and mixed stacks to maintain multivendor TCP/IP – OSI compatibility

Multivendor Network Management
Paradox or Paradigm?

Miriam Block with Martha Weeks
Hewlett-Packard
19420 Homestead Road
Cupertino, CA 95014

#4664

At a meeting of the OSI Network Management Forum, Brian Hewat, Director of Telecom Canada, said,

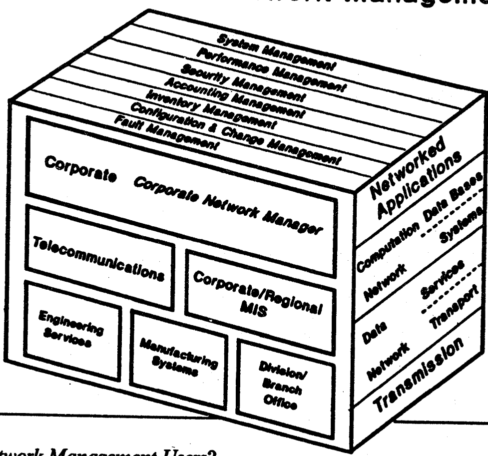
> "Many companies today operate integrated computing and communications networks that combine the products and services of different vendors. They invest thousands, often millions of dollars in these systems and networks each year. It is critical to their business success that these diverse systems work together efficiently.
>
> That's where network management comes in.
>
> Network management is the ability -- through hardware and software systems -- to identify, monitor and control network reliability, configuration, security, accounting and performance."

This definition, while a simplified one, is extremely accurate. Companies are increasingly dependent on their information networks for their everyday business. This paper describes the dimensions of network management, what to look for in a multivendor network management system and a first step Hewlett Packard is taking toward that goal.

DIMENSIONS OF NETWORK MANAGEMENT

First, let's examine what constitutes network management. This includes the network management users, and the various elements, such as applications or equipment, they must manage.

Dimensions of Network Management



Who are Network Management Users?

Wherever new technologies emerge, rarely do you find one person dedicated to managing them. Network management typifies this idea -- few companies have a position called *network manager*. Instead, the tasks related to network management are performed by a variety of users.

Often, the network management tasks fall at the highest level of an organization to individuals with titles such as Chief Information Officer (CIO) or Corporate MIS or Telecommunications Director. These executives' views of network management evolve from concerns about overall network costs, network uptime and strategic planning.

In contrast to the executive network management needs are the concerns of the DataComm Specialists, Distributed Systems Operators and Site Telecomm Specialists who must manage networks in environments such as Business Offices, Manufacturing Plants or Research and Development Labs. These people are responsible for managing Local Area Networks (LANs), the systems on those networks, and for safeguarding the integrity of data flowing through the LANs and out of the site to the regional and corporate backbone networks.

Between the CIO and local site network managers fall the Wide Area Network (WAN) managers whose duties are a hybrid of the two groups discussed above. While these managers are interested in network growth, uptime and planning, they also must implement and operate significant portions of the corporate and regional Wide Area Network.

Obviously, these groupings are simplified generalizations. Networks and network management duties tend to be as varied as the companies that use them. In all cases, an effective network management system must be flexible enough to accommodate the needs of all of these people in a variety of ways. The best multivendor network management system is a modular one which can be tailored to fit a wide range of applications with differing levels of technical detail.

Levels of Network Management

When we look at what the network managers must manage, we find different *layers*. Many of these layers can be seen directly in the seven-layer OSI stack. If we simplify the seven layer model, the network management task can be divided into four general layers:

* On top is the **application network**, comprised of distributed applications such as X.400 electronic mail, HPDesk, EDI, office automation, and administrative applications

* The **computation network** includes networked systems (e.g., UNIX, MPE, OS/2, VMS, MVS, VM) and networked data bases (SQL-based relational, system dictionary, program library, etc.).

* The **data network**, which can be further broken down into *transport* (OSI layers 2, 3 and 4) and *services* (OSI layers 5, 6 and 7), includes items like LANS, X.25, and ARPA, SNA and OSI services.

* The **transmission layer** corresponds directly to layer 1 in the OSI model. There is a need to manage items like T1, modems, broadband, fiber etc.

The number of network components being managed is large, the number of vendors is even larger, and the methods for managing them are disparate. These network layers warrant discussion far beyond the scope of this paper, but it is important to recognize the increasing complexity of network management.

Network Management Needs

The third aspect of network management centers around the functional needs of network managers. OSI has defined five Specific Network Management Functional Areas (SMFAs) for network management. These are:

* **Fault Management**: identify, diagnose and resolve network problems quickly.

* **Configuration Management**: track network and device configurations with the capability to centrally control and change those configurations.

* **Performance Management**: optimize network performance through the collection and analysis of data about the network.

* **Accounting**: provide information on network usage.

* **Security Management**: protect the network and its components from intrusion or surveillance by unauthorized parties.

Additionally, Hewlett-Packard has identified two other areas which complement OSI's SMFAs to round out the functionality necessary to manage the layers discussed above. These two other areas are:

* **Inventory Management:** a complement to configuration management and accounting, to track, monitor and maintain networked assets over a wide geographic area.

* **Networked System Management:** manage networked systems from a central point for consistency and to reduce staffing and costs.

Given the dimensions of network management, network managers face an overwhelming task. To to build integrated, multivendor network management applications, you need an open environment. In this way, true multivendor network management can be achieved.

COMPONENTS OF MULTIVENDOR NETWORK MANAGEMENT

For both designers and users of network management systems, finding an open system presents a major challenge. This section presents an overview of the components that comprise a multivendor network management system.

NETWORK MANAGEMENT STANDARDS

Adherence to standards is integral to multivendor network management systems. ISO has been developing networking standards for years, but network management standards have lagged behind in their acceptance. To accelerate the introduction of network management products capable of operation with each other, the OSI Network Management Forum was created.

The Forum's emphasis is on the *implementation* of OSI standards, and its members are dedicated to reaching fully interoperable, multivendor network management in the shortest possible time.

The organization was founded in July, 1988, by the following eight companies: Hewlett-Packard; Amdahl Corporation; American Telephone and Telegraph (AT&T); British Telecom; Northern Telecom; Telecom Canada; STC PLC; and Unisys Networks. Membership has grown to include over fifty worldwide computing and telecommunications vendors. Forum members have agreed to demonstrate interoperability in September, 1990 at a world-wide event.

According to Forum president John Miller, the Forum "is not a standards body, and we have no desire to create standards. We have a desire to implement in a consistent manner the standards that already exist, and to fill in the gaps as necessary to define a complete specification."

The Forum itself will not create any products; that responsibility will continue to rest with individual vendors. Initially, the Forum's concentration is toward common versions of OSI protocols and message sets to support network management applications. Network management systems which incorporate the Forum's protocol and message sets will build a foundation for interoperable multivendor network management.

Protocols

The communication protocols working group of the Forum, chaired by Hewlett-Packard, presented to the Forum membership a common implementation of the seven-layer OSI protocol stack. The group selected appropriate subsets or *profiles* of features to be used in each layer of the OSI stack. A single protocol stack was created to ensure interoperability between different management products and systems.

Within the first three OSI layers, the Forum plans to adopt the X.25 wide area network standard of the International Commission for Telephones and Telegraphs (CCITT) and the 802.3 local area network standard of the Institute of Electronic and Electrical Engineers (IEEE). In the future, other transport methods may be examined by the Forum.

For the upper layers, the Forum will adopt the draft OSI proposal for Common Management Information Services and Protocol (CMIS/P) which specifies the format of network management messages.

Messages and Services

Another working group of the Forum is establishing the messages and services required within a network for management functions. The first of the OSI-defined Specific Network Management Functional Areas (SMFAs) to be addressed by the Forum are fault and configuration management. The remaining SMFAs, which include security, performance and accounting, will be addressed as the Forum's work progresses.

Interim Solutions for TCP/IP Networks

Although the OSI Network Management Forum work progresses, International Data Corporation (IDC) predicts that "it will most likely be 1991-1992 before there are enough approved standards to make a major impact in the vendors' offerings of true, OSI-compliant interoperable network management products."

In the absence of systems which conform to OSI standards, TCP/IP protocols will remain the de-facto solution for interoperability for the next few years. This creates a demand for TCP/IP network management tools. The Internet Activities Board (IAB), which oversees the technical development of TCP/IP, has recommended two different network management protocols as Draft International Standards (DIS). These interim protocols are:

* Simple Network Management Protocol (SNMP)
* Common Management Information Services over TCP/IP (CMOT): based on the OSI model for management and the CMIS interface

Use of these standards and protocols in a network management system demonstrates true commitment to creating multivendor environments.

SIMPLE USER INTERFACE

Incorporating standards into a network management system provides for interoperability. Taking a step back, let's look at the everyday life of network managers -- satisfying their users' needs. Simply stated, they must keep the network up and running. To meet that goal, network managers must use several software packages and have some technical knowledge of their network.

Some network components have network management built into them, but rarely do two network management systems have the same user interface. The user interface may be difficult to learn and use or may demand in-depth technical knowledge. This adds to the difficulty of the network manager's job. The complexity of the job requires skills the average office worker doesn't have, so network management becomes the realm of highly paid specialists.

This situation demands that network management systems consistently implement a simple user interface. The user interface should be easy for both non-technical and technical users. Network managers must use a variety of network management applications, so the user interface should remain constant from one application to another. Good network management systems will not make the network manager relearn a user interface daily.

ACCOMMODATING DIFFERENT END USER ENVIRONMENTS

Another important characteristic of a network management system is the system's ability to satisfy the needs of network managers in different computing environments.

Low End

By low end, we are referring primarily to local area networks (LANs) used at a departmental level. These small to medium sized networks are based on either PC servers or minicomputers.

Often, the LAN network manager has little technical expertise. In fact, the network manager could be chosen on criteria as indiscriminate as "the person who sits closest to the server."

For this type of network, the price of network management tools is an important decision making factor. PCs may be the most expensive piece of equipment on the network; so requiring these users to purchase additional, costly computers makes network management prohibitively expensive.

High End

A very large network with many remote sites constitutes a high end environment. The network may be a wide area network (WAN) or a combination of a WAN with local area networks (LAN). A company's overall investment in computer systems tends to be higher in this environment. The network may contain mainframes, minicomputers, and workstations, plus systems ranging all the way down to PCs.

Typically, the MIS department is responsible for network management in the high end environment. These users are more technically sophisticated than the users in the office LAN environment, and therefore demand more functionality in their network management systems.

Clearly, a network management system must adapt to a range of computing platforms to accommodate all networking environments.

ACCOMMODATING DEVELOPERS ENVIRONMENT

To create multivendor network management systems requires a developers' environment that meets the diverse product development needs of network vendors. Hewlett-Packard has adopted standard models based on the OSI Systems Management Architecture:

Organizational Model

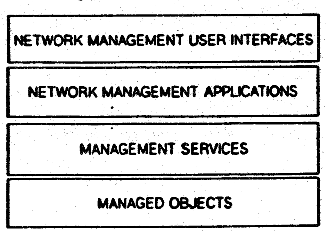
The organizational model assists designers in identifying functional elements and expressing them as components of a management solution. This model breaks down the functional elements into:

* **user interface:** the *exposed* part of the model -- what the user will see. The designer must be able to answer the question "who are the network users?" as described in the dimensions of network management.

* **management applications:** support a specific management activity through a common user interface.

* **management services:** *key component* of Hewlett-Packard's architecture. The decoupling of management services from the applications lets several applications manage the same network object in different ways, thus creating a truly open environment that encourages multivendor network management.

Organizational Model

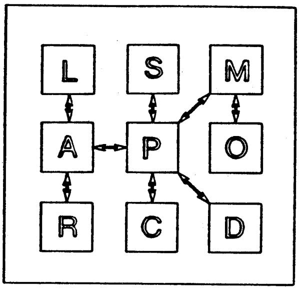


Operational Model

The operational model (or nine squares model) helps the designer illustrate how the components of the organizational model will be used and how various network management solutions will coexist.

The components of the operational model are best shown in the following illustration.

Nine Squares Model



This picture represents a single system. When using the model for illustrating several systems coexisting in a network management system, you would use one of these boxes per system.

The critical component in this model is the [P] representing *postmaster* services. Serving as the integrator for all the other components, the [P] adds flexibility to the model.

Just as with the SMFAs, detailed discussion of these models is well beyond the scope of this paper. You should simply note that these two models are a foundation for multivendor network management systems. Adherence to these models ensures consistent application programming interfaces (APIs) for network management programmers and consistent applications for network managers.

EMERGING TECHNOLOGIES

A network management system that employs all of the correct methods and standards on the right computing platform provides with you a usable system today. But, where will that system be in a year or two? Technology is evolving at amazingly fast rates. You must ensure that your network management platform will allow you to take full advantage of future technologies.

Following is a discussion of three emerging technologies that will play an important role in the network management systems of tomorrow. These technologies are object-oriented programming, expert systems and distributed peer-to-peer management.

Object-Oriented Programming

In object-oriented programming, everything that is part of the system is called an *object*. Objects can be text, graphics, spread sheets, etc. This provides modular design, extensibility (the ability to easily add components), fixed levels of granularity (grouping of several objects into a single new object), and ability to inherit (or incorporate) types of objects and application level code into other objects or code.

For the network management system developer, object-oriented programming lends flexibility and the ability to leverage code from one application to another for faster product development.

Expert Systems

As networks become more complex, network managers will need even more tools to help in their day-to-day activities. For example, adding a node to the network may involve many time-consuming steps. Now, using a network management system, the network manager must also add that node to the network management system.

Expert systems, which build on object-oriented technology, aid in these situations. A network management system might represent the network with a graphical map. The expert system would configure a node simply by adding it to the map. Going one step further, it might assist in "what if" scenarios for planning and designing additions to a network. Expert systems have the potential to play an exciting role in isolating and diagnosing faults in a network.

Distributed Peer-to-Peer Management

Since network management systems must supply an increasing amount of information to varying users in different geographic areas, completely centralized network management activities may prove difficult. Distributing the data across the network allows for more efficient management.

Coupled with that management need is the requirement to distribute information appropriately within systems. While a machine might be a collector and repository of network management information, it might also be the provider of information to another network management system. By designing systems with distributed peer-to-peer management in mind, we are beginning to break down the barriers to multivendor network management.

OPENVIEW WINDOWS -- A FIRST STEP

Hewlett-Packard's OpenView Windows takes a first step toward offering a multivendor network management platform. During the development of OpenView Windows, the OSI network management standards were in a preliminary state. Understanding that standards take time, the OpenView Windows team determined that a common user interface would allow them to focus on multivendor consistency from the user's viewpoint. As the standards evolve, they will easily be incorporated into OpenView's modular architecture.

The OpenView Windows standard is based on Microsoft (TM) Windows with its easy-to-use and easy-to-learn user interface. Additionally, OpenView makes use of Hewlett-Packard's NewWave technology.

At the heart of OpenView Windows is the network map drawn by users to represent their network as they intend to manage it. Symbols with easily identifiable shapes represent the nodes or devices on the network. Once drawn, OpenView applications report the status of the network through colors on the network map. The color scheme is simple: red represents a critical state, yellow indicates a warning, and green shows normal status. Developers can use an API to report the status of their equipment to the map consistently with other OpenView applications.

As the OSI network management standards become better defined, the use of the nine-squares architectural model will allow for easy integration. The current OpenView platform serves well for the low end environment; and future platforms will include solutions for high end environments.

CONCLUSION

Looking for one answer to network management seems to be the paradox. HP's OpenView solution lies in finding an environment that designers can use to systematically develop network management applications which coexist with applications from other network vendors.

We have highlighted a few of the elements that form the paradigm for multivendor network management: a simple user interface, adherence to standards, modularity to allow easy incorporation of emerging technologies, and accommodating developer and end user environments.

With integrated, multivendor network management systems, companies will be more effective at managing information as a strategic asset to improve their overall competitiveness.

Distributed Application Services In Todays Business Environment
#4665

Todd Goldman
Hewlett-Packard
Information Networks Division
MS 43LW
19420 Homestead Road
Cupertino, CA 95014

Today, business is more complex than ever before. Competition, occurring on a worldwide basis, is becoming fiercer daily. Companies have decentralized their national and international operations to better meet their customers' needs. For example, manufacturing facilities are relocated to reduce costs, while sales offices are moved closer to customers to allow for better service. The quick and accurate exchange of information is essential to survival in such an environment. Future competitiveness demands a cooperative computing environment which will put computing power where it is needed, while providing increased access to remote information and resources. HP's Distributed Application Services (DAS) are a framework upon which applications can be built for companies with distributed facilities and information systems.

The Evolution of Distributed Computing

As the trend in business moved toward distributed management, computers also became distributed. A few years ago, companies operated from a single corporate mainframe which handled all applications and information. Once their organizations began to expand geographically, it became necessary to move some of the computing power to local offices. Users found that remotely logging on to a corporate mainframe left them vulnerable to the reliability of the telephone system and lacked the speedy response they required.

Because of this situation, companies moved to minicomputers and used batch file transfers to send information back to the mainframe. Eventually, users on minicomputers sought improved performance, increased access to information and additional flexibility. The development of the microprocessor answered this demand as information and applications were moved to workstations and PCs linked together via a Local Area Network (LAN). PCs and workstations allowed users to run applications quickly, without having to contend with other users on the system, while still retaining access to the minicomputer when it was necessary.

During this transition, information and productivity tools made their way out of the centralized system and into the local environment, closer to actual users. While this significantly increased the productivity of the local environment, it did little to unify the global environment.

Computing Today

Today, engineers are able create better designs and accountants can perform more accurate financial analysis on their own systems (which may use different operating systems,) using software tools designed for their specific needs. As we move towards the 1990s, competitive pressures will force the accountant and the engineer to work more closely together to meet customer needs effectively and efficiently. Computer facilities will need to be localized to keep applications and information close to the user, yet they will also need to be available to everyone as if they were still centrally located.

Computing in the 1990s will require distributed applications which allow for increased communication in this diverse environment. Companies will continue to purchase a variety of computers from a variety of vendors for a variety of specific tasks, but they will need them to work together as if they were a single computer. End users on PCs will need quick access to their own applications and will also need transparent access to lesser-used applications and information on other systems. Furthermore, applications will need to be distributed, with different procedures running on various CPUs, in order to take advantage of the available resources throughout a company while minimizing the cost of computing.

To build such distributed applications, a foundation of services, Distributed Application Services (DAS), must be created. These services will provide the tools and the underlying networking which will make a geographically disperse, heterogeneous and multivendor computing environment appear as if it were a single computer. DAS will provide seamless, integrated and invisible networking as a framework for building distributed applications.

Networking and Automobiles: Getting you where you want to go

Networking should allow someone to extend their computing power without requiring knowledge of the network, in the same way a car extends a person's mobility. Drivers may not know exactly how a car works, but they can get into almost any car and easily get where they wish to go.

Networking should be that simple. Fundamentally, it is simple--this is what Distributed Application Services are all about. DAS is based on four basic elements which are required for computer networking to work. These can be referred to as the key networking functions: **Data passing, Data sharing, Application Access,** and **Execution Sharing.**

Data Passing is the ability to send information back and forth from one location to another. Data passing could be used for electronic mail or Electronic Data Interchange (EDI). It also allows users to share peripherals by allowing files to be passed to remote printers or hard disks.

Data Sharing allows multiple users or applications in a variety of locations to access information at one location without actually moving that information. Data sharing gives a user the ability to remotely access a database or a file. It allows them to alter the data but doesn't allow them to move the information in the file to another location.

Application Access gives a user the capability to run applications in a remote location. It provides a window into a remote computer so multiple users in a variety of locations can share an application.

Execution Sharing, which can also be viewed as a sophisticated combination of data passing and application sharing, allows an application to be separated into pieces and run, in an integrated fashion, among various locations. The ability to perform parallel processing in multiple locations is the result of execution sharing.

These four basic network functions provide the building blocks upon which distributed applications can be built. They give the solutions provider, application developer or systems integrator a basic set of tools for communicating across the network in a variety of ways. All a person needs to know is which fundamental tool they wish to use.

This is similar to the car driver, whose rudimentary knowledge allows them to choose between various types of cars, based on what they need to do. For instance, a Lincoln Town car might be good on a long trip, but a Honda Civic would certainly be better for finding a parking space in Manhattan. In the same way that a driver can choose an appropriate car without understanding exactly how it works, the application developer must be able to select and use the appropriate networking function without having to understand exactly how it works.

The Infrastructure: Tying it all together

The four basic network functions, in and of themselves, are not enough to provide seamless, integrated and invisible networking. They are the engine, the frame and the wheels of the car. The next step is to add the capabilities which will make the car easy to drive. These capabilities, which help the network functions work together, are functions such as **Directories**, **Network Management** and **Security**. These abilities, called the **Infrastructure**, provide a method of linking the various design elements and making them work together more closely and simply. The Infrastructure makes the basic network functions easier to use.

Directories provide information about who is on the network, what CPUs are on the network, and what peripherals are on the network. Directories allow someone to mail an electronic message to someone else without knowing the recipient's address. More importantly, they also allow access to printers, disk drives, applications and any other resources on the network without having to know where those resources are located.

The four basic network functions all use the directory to "know" what resources are available on the network. Without an integrated directory, the basic network functions would need to be "told" what resources were available on the network, rather than just looking in a common place.

In the car analogy, a directory would automatically tell a driver to make a right hand turn at the next light, then a left at the light after that, and so on, until the destination was reached. The directory would get the driver to the destination even if the driver did not know where it was.

Network Management lets a network administrator centrally control and monitor the objects and applications in the network. It ensures that the basic network functions and the other network infrastructure capabilities are working properly, and it also allows the administrator to fix the network if something goes wrong. Network Management is much like the speedometer, odometer and oil light on a car. They allow a driver to monitor how the car is doing from a central location without having to look under the hood. But Network Management goes one step further, it allows one person to monitor all the cars on the road and fix them without leaving the driver's seat.

Security determines who is permitted to use the network. Security is the key, and the alarm system to your car.

Finally, solution providers, application developers and system integrators must also have **Application Programmatic Interfaces** (APIs) in order to make full use of the basic network functions and the Infrastructure. These APIs establish consistent interfaces into the network at various levels. The APIs provide access to a variety of easy-to-use tools which can be used to connect existing and future applications to the network.

Together, all of these pieces provide the services necessary to fundamentally build applications in a distributed manner, and most of these pieces exist today. The idea behind HP's Distributed Application Services is to integrate these pieces, creating a strong foundation upon which distributed applications are built. The goal of DAS is to make networking easy for the solutions provider, systems integrator and applications developer. DAS will allow programmers to quickly develop networked applications which will distribute resources while integrating ideas. With Distributed Application Services, Hewlett-Packard is uniquely poised to solve your networking and computing problems in increasingly complex and competitive business environments, today and tomorrow.

Setting Up a Small Network
(Getting Started)

Marcia Clary
Hewlett Packard Company
Phoenix, Arizona

1. INTRODUCTION

Computers were introduced into business to allow people to become more productive. Many users could take advantage of a single multi-tasking computer it was easy to share data between users on this system. Since that time computers have become smaller in size, more capable in the tasks they perform and less expensive.

Many users have a computer at their desk which allows them to be more productive but with a distributed computing environment, information and data have become disbursed. In order to boost productivity, users have a need to share data in a quick efficient manner. This is why networks are playing a bigger and bigger part in industry. Getting data, in a timely manner, to those who require it will truly enable a company to be at a high level of productivity.

For someone new to networking, learning about it is not the easiest task to accomplish. There are many books and articles published on specific areas of networking but few discuss all the topics and many are more complex than would be reasonable for a beginner.

During the course of this paper we will discuss how a network operates, how to put a network together and how to grow a network.

2. TOPOLOGY

There are several types of network topologies available and in use today. This section will define some of the more typically topologies in industry.

STAR

The star topology is shown in figure 1. All nodes in this type of topology are connected directly to a server and all communication between computers passes though this server.

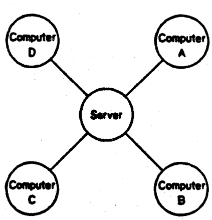


Fig. 1 Ring Topology

RING

With the ring topology the LAN cable forms a continuous circle as shown in figure 2. Each workstation is attached to this ring.

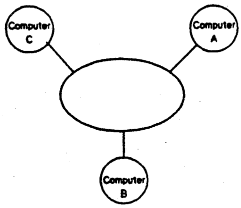


Fig. 2 Ring Topology

BUS

Each station in a bus topology is connected directly to the LAN cable in the manner shown in figure 3.

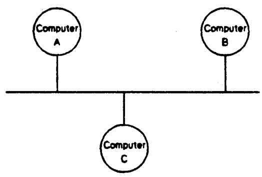


Fig. 3 Bus Topology

3. CABLING

There are many different types of cables used in Local Area Networks. These cables include coaxial, twisted pair and fiber optic. Let's take a closer look at coaxial cable shown below.

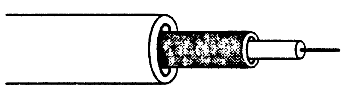


Fig. 4 Coax Cable

The inner cylinder is the conducting core. The next cylinder is the insulation. Then comes the Conducting Mesh or Sleeve. Finally, the protective jacket. Coaxial cable is frequently used in LANs.

Typically, there are two types of names for different standard cables. The first name is the name given to the cable by the Institute of Electrical and Electronics Engineers (IEEE) Computer Society Local Network Committee. The other name is a common name which is usually used in every day conversation. The following table displays the IEEE name and its associated common name.

IEEE NAME	COMMON NAME
IEEE Type 10BASE-T	StarLAN 10
IEEE Type 1BASE5	StarLan
IEEE Type 10BASE5	ThickLAN
IEEE Type 10BASE2	ThinLAN

As you can see, all of the above cable names have the word BASE in them. This refers to baseband transmission which means that a digital signal will be transmitted on these cables. As you get further in to networking, you may run into cable names with the word BROAD in them. This refers to broadband transmission which is an analog signal. An analog signal is sometimes used to transmit video as well as data, whereas a digital signal is used for data.

Although the common name refers to a particular type of cable, the IEEE name actual gives more information about the capabilities of the cable. From the IEEE name we can determine the type of transmission, the maximum length of a segment and the maximum speed of transmission.

```
IEEE Type  10BASE5
           |  | |
           |  | |Maximum Length of segment (500m)
           |  Baseband transmission
           |Speed (10Mbps)
```

THICKLAN -- IEEE TYPE 10BASE5

The IEEE name for ThickLAN specifies it has a maximum speed of 10Mbps, a maximum segment length of 500 meters and that it uses a digital signal.

Here are some other specifications regarding ThickLAN:

- -- The ThickLAN coax cable is 10mm in diameter.
- -- 100 connections per 500 meter segment.
- -- Each connection must be 2.5 meters apart.
- -- The cable must be terminated at both ends by 50 ohm resistive load.
- -- The cable must be grounded at one point.

In order to connect a computer to a ThickLAN, these components are required: a LAN interface card, an AUI (Attachment Unit Interface) cable and a MAU (Medium Attachment Unit, also known as a transceiver). The AUI cable connects directly to the MAU and to the interface card, as shown in the diagram. The MAU usually contains a pin inside of it that will be inserted directly into the cable down to the conducting core. In order to install a MAU, a special installation kit is required to pierce the cable before inserting the MAU pin.

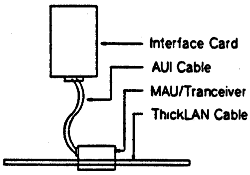


Fig. 5 ThickLan Attachment

THINLAN -- IEEE TYPE 10BASE2

The IEEE name for ThinLAN specifies that the maximum speed for ThinLAN is 10Mbps and it uses a digital signal. The maximum length per segment with this type of cable is 185 meters.

Here are some other specifications regarding ThinLAN:

- -- The ThickLAN coax cable is 4.9mm in diameter.
- -- 30 connections per 185 meter segment.
- -- Each connection must be .5 meters apart.
- -- The cable must be terminated at both ends by 50 ohm resistive load.

Connections to ThinLAN are done through BNC type connections and it may be done in one of two ways. First, some interface cards have the AUI and the MAU built onto the card. In this case, when the interface is installed into the computer a BNC connector will be visible. Then a T-connector is attached to the exposed BNC connector and the LAN cable is connected to this T-connector.

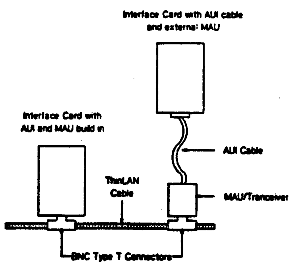


Fig. 6 ThinLAN Attachment

The second way to connect to a ThinLAN is similar to the connection made to a ThickLAN. The interface card, the AUI and the MAU will all come separately. In this case, the user will connect the AUI cable to the card and to the MAU. Then a T-connector is attached to the BNC connector on the MAU and a cable is attached to the T-connector. Unlike connecting to ThickLAN, connecting to ThinLAN needs no special installation kit.

4. OSI MODEL

Thus far we have examined the hardware in a LAN network. Let's now begin our investigation of the software for the network.

In 1984 the International Standards Organization (ISO) developed the Open System Interconnection Reference Model. The creation of this model was the first step toward international standardization for the process of communications between computers. The model allowed existing network protocols to be place in perspective within an overall model. It provided a framework with which standards for the purpose of systems interconnection could develop and it helped to identify areas for development and improvement.

This model was not intended to serve as an implementation specification, but was intended to serve as guidance for development of international standards.

The OSI Model divides the communication process into seven interdependent layers. There is nothing magical about the number of layers in this model, however, there were a few things that were strived for in the creation of these layers. First, there should be a layer for each level of abstraction. Next, each layer should perform a well defined function. Finally, layer boundaries were chosen in such a manner that a minimal amount of data would be passed from layer to layer.

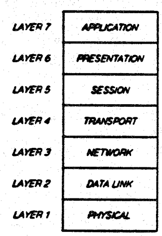


Fig. 7 OSI Reference Model

LAYER 1 -- PHYSICAL LAYER

The lowest level of the model is the physical layer which is concerned with transmitting and receiving bits. This layer describes the physical and electrical characteristics for communication. It defines such things as the speed of communication, whether the transmission will be analog or digital and if digital, at what voltage level a logical 1 or 0 will be detected. In addition, it describes the actual physical connector. This includes the number of pins and the purpose of each pin.

LAYER 2 -- DATA LINK LAYER

The data link layer is responsible for getting data to and from its destination intact. The data link layer breaks up the data into what is known as a data frame (a few hundred bytes). When the data link layer is sending packets, a parity field is added to the data. When the data link layer is receiving packets, the parity field is check to see if any errors occurred during transmission. If errors did occur during transmission, it is the responsibility of this layer to request that the packet be retransmitted.

LAYER 3 -- NETWORK LAYER
The network layer is responsible for getting information from one machine to another. This process may entail routing the packets between networks. This layer is also responsible for packet flow.

LAYER 4 -- TRANSPORT LAYER
The transport layer accepts data from the session layer and breaks it up into smaller lumps if necessary before passing the data to the network layer. It is this layer that establishes the connection to the destination computer and then makes sure that all packets are received in the correct order without omissions or duplications of packets.

LAYER 5 -- SESSION LAYER
The session layer allows sessions to be established between machines. This can be thought of as a conversation between two people. It is the session layer that synchronizes and controls the conversation so that one machine is able to finish it's transmission without being interrupted by the other machine. This layer establishes a session when a user wished to transfer files from one computer to another. A session will also be established when a user wishes to logon to a remote computer in order to have an interactive session.

LAYER 6 -- PRESENTATION LAYER
Layer 6 is the presentation layer. This layer is responsible for converting data from machine format to network format before it is forwarded to its destination. When the packet arrives at its destination, the presentation layer will then convert the data from the network format to the machine format. This layer may include encryption or compaction if necessary.

LAYER 7 -- APPLICATION LAYER
It is the application layer that allows transfer of data between machines and allows communication for distributed databases. Also dealt with in the application layer are electronic mail, virtual terminals and user program callable procedures.

5. <u>DATA MOVEMENT THROUGH OSI MODEL</u>
Now that we know there is an OSI model and we understand the function of each layer, what really happens to data as it moves from one machine to another machine?

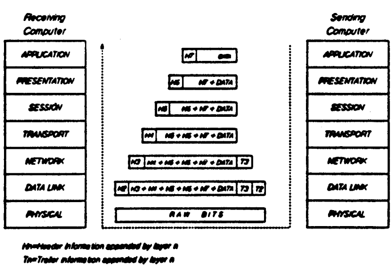


Fig. 8 Data Flow between Computers

In order to get data from a program on the sending machine to a program on the receiving machine, data is passed from the program to the application layer on the sending machine. At the application layer some header information is attached to the data which is then passed to the presentation layer. The presentation layer also attaches some header information and passes it down the stack. This process continues with each layer adding header (and perhaps trailer) information until the packet is finally complete and ready to be transmitted across the physical media.

It is important to understand the OSI model, as well as, how data flows from one machine to another. This information is the basis to understanding networking and it will help when trying to understand the different types of equipment used in the network which will be discussed later.

6. PACKETS

Thus far we have seen how LANs can be arranged, we have seen different cables used in LANs and the type of transmission used on these cables. Basically, all of these things take place at the physical layer. In this section we will move to the Data Layer in order to discuss the packets formed to transmit data.

The standards most commonly used at the Data Link Layer are X.25 and the standards done by IEEE Project 802. Since the X.25 standard is typically used in Wide Area Networks (WAN), we will be examining the IEEE Project 802 Architecture which is typically used in LANs. The IEEE 802 Project subdivided the Data Link Layer into two parts. The first part, known as the Logical Link Control (LLC), is described by IEEE 802.2 and provides communication with the network layer. The second part, known as Media Access Control (MAC), is described by one of the following: CSMA/CD (Collision Sense Multiple Access with Collision Detection) --IEEE 802.3, Token Bus--IEEE 802.4 or Token Ring--IEEE 802.5.

DATA LINK	LLC	IEEE 802.2	IEEE 802.2	IEEE 802.2
	MAC	IEEE 802.3	IEEE 802.4	IEEE 802.5
PHYSICAL		Broadband Coaxial Baseband Twisted Pair Baseband Coaxial	Broadband Coaxial Baseband Coaxial	Broadband Coaxial Baseband Coaxial

Fig 9. Link Level Control and Media Access Control

In the diagram above you can see that the LLC layer is the same no matter which MAC definition is used. The MAC definition actually describes how access to the cable is made as well as how the packet will look when it is transmitted to another computer. In both 802.4 and 802.5 a token is passed from station to station and as long as the computer has the token it can transmit. The 802.3 definition uses a protocol which is CSMA/CD. This means that each computer on the network listens to the network. If the network is free from traffic, a computer may begin to transmit data. If two or more stations are transmitting data at the same time, a collision occurs. At which point all stations will stop transmitting for a random period of time before trying to transmit again.

Each of the above mentioned protocols create a packet in order to ready the data for transmission. It is important to understand that although these packets can be very similar, computers that use different packet formats cannot communicate directly with each other. In addition, computers that use different methods to access the cable (ie Token Bus vs CSMA/CD) cannot communicate.

Let's take a closer look at an 802.3 packet.

PREAMBLE	START FRAME DELIMETER	DESTINATION ADDRESS	SOURCE ADDRESS	LENGTH COUNT	DATA	PAD	FCS
7 BYTES	1 BYTE	2 OR 6 BYTES	2 OR 6 BYTES	2 BYTES	1500 BYTES		4 BYTES

Fig. 10 802.3 Packet

The 802.3 packet is made up of eight fields. The first field is known as the preamble. This field is seven bytes long and it consists of 10101010 repeated seven times. This field allows the sender and the receiver to synchronizes clocks before the data is actually transmitted.

The start frame delimiter field is one byte long and consists of 10101011 and it specifies the start of the actual frame.

The next two fields are the source and the destination addresses, respectively. These fields may be two to four bytes long, but whatever the length, they must be consistent throughout the network. For 10Mbps baseband an address of six bytes is standard.

Although the source field always has the address of the sender of the packet, the destination field has a couple of special cases. First, the high order bit in this field is normally a zero which means that the address in this field is a normal address and that the packet will be delivered to a single destination. If the high order bit is a one, however, it signifies a group address and everyone assigned to the group will receive the packet. This is known as multicasting. In the next case the entire field may be filled with ones which signifies everyone on the network will receive the packet. This is known as broadcasting.

The next field is two bytes long and it specifies the length of the data.

The Data and the PAD fields together may have a maximum total length of 1500 bytes. It is in these two fields that the data is contained. Although these two fields together can be from 0 to 1500 bytes long, 802.3 specifies that the frame must be 64 bytes long from destination address to the check sum which means if the data is less than 46 bytes in length the rest of the length must be made up with pads.

The last field is 4 bytes long and it is for the CRC (cyclic-redundancy-check) value. This field is used to check the accuracy of the packet when received.

Although the 802.3 packet is a standard, many systems in use today use the ethernet packet. The ethernet packet is the predecessor to the 802.3 packet and the two packets are very similar. The only difference is that instead of the length field shown in the 802.3 packet, the ethernet packet has a type field which specifies the network protocol.

There is one other point that should be noted. Computers that use ethernet packets and computers the use 802.3 packets can reside on the same cable because both definitions use CSMA/CD but computer that use different packets cannot communicate with each other. Only those computers that can understand both 802.3 and ethernet packets can communicate with all machines on the network (provided the upper level services are compatible).

7. EQUIPMENT USED IN LANs

From the previous sections we have learned that there are many physical limitations to LANs. These include cable segment lengths, number of connection that can be made to a segment and even the speed at which transmission occurs on the cable. In addition to physical limitations we have learned that different LANs use different packets. This section deals with overcoming these limitations so that we may have all computers at our site communicating.

REPEATERS

A repeater is the simplest device used in a network. Its main purpose is to connect two cable segments of which one or both cables are at the maximum length.

After a signal travels a long cable it can become weak and distorted so the function of the repeater is to accept a signal regenerate the signal to its original state and retransmit the signal. This procedure will enable us to expand our cable length.

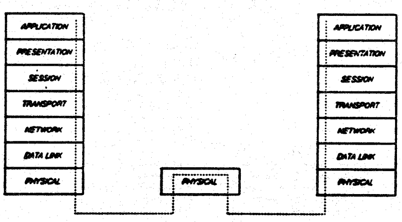


Fig. 11 Repeater

BRIDGES

More complex than a repeater is a bridge. In most cases a bridge is used to connect two LANs together, each utilizing a different physical media, however, two identical LANs may be connected together using a bridge. A bridge may also be used to connect different speed LANs. For example, a bridge may be used to connect several workstations running ARPA/Berkeley Services on a thin coaxial network to several PCs running ARPA/Berkeley Services on a twisted pair network. Essentially all layers of the OSI model are compatible on both networks except the physical layer.

A bridge will physically reside on both networks and it's function is to read all packets on both LANs and then determine if the packet is to be forwarded to the other LAN.

Because a bridge does not forward all packets it reads it becomes useful in other ways. First, a bridge could be used to split the traffic load on a heavily traveled LAN. This means packets that have the source and destination on the same LAN would not be passed through the bridge causing needless congestion on the other LAN.

Second, a computer on a LAN listens to all the traffic to determine if the packet is address to it. If a system find that the pack is address to it, the packet is processed. What this means is that if someone wanted to write a program to read all the packets on the network, it could be done. This can cause security problems, so a bridge could be used to localize sensitive data\traffic to a particular LAN.

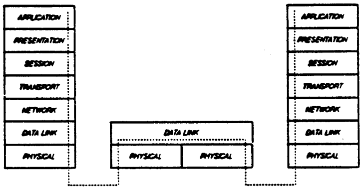


Fig. 12 Bridge

ROUTERS

Routers are used between networks in which different protocols are used at the first two levels of the model.

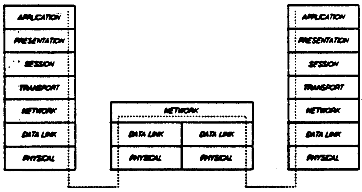


Fig. 13 Router

When many LANs are connected together, how a packet gets from a computer on one LAN to a computer on another LAN may become an issue. Normally the packet will be sent to a computer on an adjacent LAN and then that computer will forward the packed to a computer on a LAN that is adjacent to it. This process continues until the packet arrives at its destination. The point to be made here is that we would like to see the packet be routed to as few LANs as possible in order to get to its destination. The following example will better illustrate exactly what can happen.

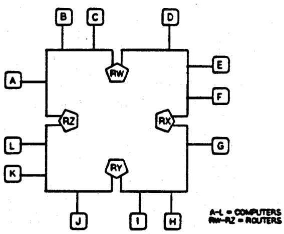


Fig. 14 Router Usage

In the diagram above there are four LANs, all connected by routers (RW, RX, RY, RZ). Let us say that computer A would like to send a packet to computer E. There are basically two ways this packet can go. The first way would be to travel through router W and then on to computer E. It is possible, however, for the packet to travel through router Z to router Y then to router X and on to computer E which is not very optimal. So it is the job of the router to make sure the packet gets from it's source to it's destination and the route is optimal.

When using a router the data link layer and the physical layer may be different between networks but the network layer is the same. This typically means the packet format may be different so the router may have to modify the packet to satisfy this need. An example of this would be sending an 802.3 packet though an X.25 pad to another computer that accepts 802.3 packets.

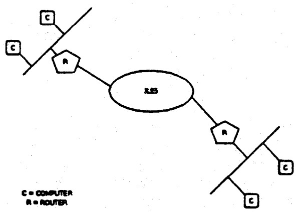


Fig. 15 Router Usage

A computer in a network may perform the function of a router, however, there are special boxes design especially to fill this need.

GATEWAYS

There are basically two types of gateways in use today. The first is used when the first three layers of the OSI model are different, but the transport layer is the same. This type of gateway,sometimes referred to as a "level 3 router", is typically used when each of the LANs connected together have different addressing domains.

The second gateway that is commonly used is a gateway which connects two LANs that are based on totally different architectures. An example of this would be connecting an 802.3 LAN with a SNA network.

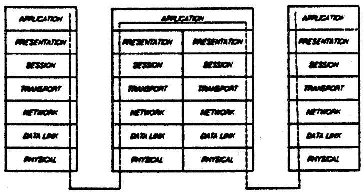


Fig. 16 Router Usage

The above has been somewhat of a textbook definition of the different types of devices used to connect LANs together. However, equipment and equipment nomenclature has been changing in the industry today. First repeaters are still repeaters, however, bridges that connect two similar networks that are on different cables may also be called repeaters because the box acts as a bridge to join the two different cables and incorporated in this box is a repeater function. Next the term router seems to be going away and it is being replaced by the term bridge. These bridges perform the function that has been previously described to be a router function. Finally, gateways are still gateways but watch out for the older term of "level 3 router" which is also a gateway.

8. <u>SOFTWARE SERVICES</u>

In industry today there are many types of networking packages that are available. The two we will look at in this section will be two that are on a wide range of vendor platforms.

ARPA/Berkeley Services

The ARPA/Berkeley services are services that are defined at layer 7 of the model. These services run on top of the Transmission Control Protocol (TCP) which describes layer 4 and Internet Protocol (IP) which describes layer 3. Neither of these protocols is an established formal standard, however, both protocols are so heavily used in industry that they have become the defacto standard.

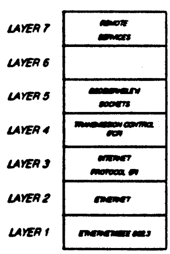


Fig. 17 ARPA/Berkeley within OSI model

The commands referred to as ARPA commands/services were developed by the Department of Defense (DOD) as part of the ARPANET (Advanced Research Project Agency Network). The ARPA services provide file transfer though the File Transfer Protocol (ftp). The ftp program when invoked allows users to connect to another computer and transfer files. In addition to transferring files the ftp program allows users to list remote directories, change directories both local and remote, display contents of current remote directories, create and delete remote files, and change the name of remote directories.

The ARPA service which provides remote login capabilities is the telnet program. This program allows a user to connect to a remote host for purposes of an interactive session.

The mail function under the ARPA services is called Simple Mail Transfer Protocol (SMTP) and its function is to send mail to other machines in the network.

The Berkeley services were developed at the University of California, Berkeley for ARPANET. These services were originally developed to allow communications between UNIX machines. One of the advantages to the Berkeley services is that commands may be executed on remote machines and the login procedure is taken care of automatically.

File transfer for the Berkeley services is done through the remote copy command (rcp). This command allows the user to copy files to/from the remote computer. In addition, it allows a user to copy from one remote computer on the network to another remote computer on the network.

Remote login is done through the remote login command (rlogin). This command allows the user to login to a remote UNIX computer and establish an interactive session on the remote computer.

The remote shell command (rsh or remsh) allows a user to execute a command on a remote host. This command may execute on the remote host using files from either the remote computer or the local computer.

The remote uptime compand (ruptime) displays information about each UNIX computer in the network. This information includes whether the computer is up or down, how long it has been up and what the load is on each computer.

NETWORK FILE SYSTEM

Another popular networking software is Sun Microsystems' Network File System (NFS). Although NFS was implemented on the UNIX operating system it is becoming popular on other operating systems as well. One version of NFS runs on the PC under MS-DOS.

The UNIX operating system has a hierarchical file structure. This is depicted below.

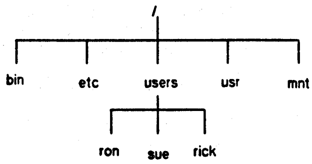


Fig. 18 Hierarchical File Structure

When a user logs into a UNIX system he will usually be put into his assigned working directory which is where users may create and remove file and/or directories as needed. In the above structure "rick" may logon and be put in the directory "/users/rick" .

The function of NFS is to allow users to access directories on other computers in the network as if these directories were part of the local file structure.

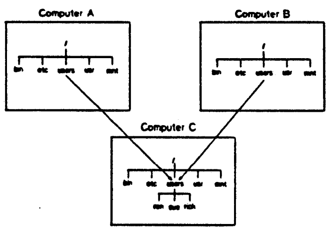


Fig. 19 NFS

In the above example all user directories reside on computer C. Users share computers A, B and C and are never sure which computer will be available from day to day, therefore, a user must be able to access the user files from all machines. In this case, both computer A and B use NFS to attach the "/users" directory on computer C to their local file system.

NFS implements Remote Procedure Call (RPC) at Layer 5. It is this protocol that establishes the calling procedures for commands. It defines the procedure as well as the parameters to be passed. The other protocol which NFS implements is eXternal Data Representation (XDR) at layer 4. This protocol defines the data format into which the sending machine will convert the data before transmitting it. When the receiving machine receives the packet the data is then converted from XDR format to the internal format of the machine.

9. PUTTING IT TOGETHER

DEFINITIONS AND REQUIREMENTS

In order to implement a network, the equipment in place should first be examined. Are there only UNIX workstations? Is there a mixture of workstations? Are there only PCs? Are there PCs and UNIX workstations? Are there other networks that this network may connect into-- now or in the future? This discussion will examine a network of PCs and UNIX workstations.

Next take a look at the users. There are basically two types of users. The first type of user is computer literate and typically take care of a system without much assistance. The second type of user is one who uses the PC to accomplish a task and doesn't wish to know anything about installing software or maintaining the system. The second type of user far outnumbers the first so in this discussion we will assume the second.

Take a look also at the software running on each station. Does everyone on a PC have there own favorite software packages or does everyone typically use the same packages. Is everyone on the same version of software?

Before a list of requirements is specified it may be useful to examine how things are getting done now. How are files getting backed up? How are documents getting printed? How is new software getting installed and updated. When the current system has been closely examined it is truly time to create a list of requirements. This list of requirements should have two section. The first section should be of mandatory requirements and the second section should be of wants that would be nice but not absolutely necessary. The following is a list of common requirements for a network.

1. Facilitate Backup
2. Sharing Printers and Plotters
3. Sharing data
4. Down loading software

The reason for the requirements list is to keep on track as different vendors are interviewed. Vendors have a tendency to show-off the great features of their products and when the features of one product are compared with the features of another product, it can become quite confusing. Therefore, by first defining the requirements and having a clear set of objectives it may be possible to avoid much of this confusion.

SOLUTION 1

Now that our situation has been determined how can it be solved. Since there are many elegant PC networks on the market today, we may decide to use one of these solutions to network our PCs. These types of solutions provide a file server so that all users may store their files in a central location and only one PCs will need to be backed up.

The file server usually acts as a printer/plotter spooler, however, the maximum number of spooled devices may be limited. This printer/plotter spooler may be accessed from each PC just as if it were connected directly to the PC. However, in order to access this printer/plotter spooler from a UNIX workstation it may involve going to a PC, transferring the file from the UNIX workstation to the PC and then transferring the file to the printer/plotter spooler server. In this case it may be better to make a UNIX machine the spooler. This also has its problems but the user would not have to physically move to another machine.

Finally, many of these servers allow the user to down load software from the server into the PC. This would solve the problem about installing and updating software and all users would be sure to be on the same revision of the software.

If a PC network is chosen, here are somethings to keep in mind. First some PC solutions may not run on coaxial cable which is the most common cable used in the UNIX workstation environment. In this case you may have an addition expense of a bridge. In addition, some of the PC networks use Token Bus or Token Ring or even proprietary protocols to transmit data so again you may need a bridge because UNIX networks usually use CSMA/CD. Be careful in this area because bridges are not always available to connect PC networks to UNIX networks.

Many times it is not possible to talk to a UNIX workstation over a LAN unless the ARPA/Berkeley services are on the PC. This means that the ARPA/Berkeley services must be purchased, as well as, the PC networking software. This usually is not a problem if both packages are purchased from the same vendor. However, if each package is purchased from a different vendor, this may create a problem because the two packages may not run together on the same LAN card. If the two packages don't run on the same LAN card, the solution is either purchase two LAN cards for each PC or have the user reconfigure and reboot the computer when access to a different LAN service is required. The first solution is not very economical and the second is not very convenient, however, the point to be made here is to make sure the services to be implemented are compatible.

SOLUTION 2

The other solution is to put all PCs and UNIX workstations on the same ThinLAN and run both ARPA/Berkeley services and NFS on each computer. This would give users the capability to logon to the UNIX machines and do work there as well as have a central location to store files that need to be backed up. All systems on the network will have access to a printer/spooler through one of the UNIX machines, however, in order for a user at a PC to print or plot he must first print or plot to a file and then transfer that file through the Berkeley Service remote shell (rsh) to the UNIX spooler.

In addition, although NFS allows users to keep their files on a centralized disc many of the security methods implement in the PC networks are more elegant than those available in the UNIX environment.

As can be seen, both solutions will probably meet our requirements, however, neither solution does everything we want in the manner that is totally acceptable. The purpose of networking standards is to be able to have a heterogeneous computing environment and eventually all of this will be transparent, however, today many networks are based on the OSI model, but are not standard. One should be informed before setting up a network, especially a heterogeneous network.

Before a network has been decided upon check to see what networks are currently installed. Networks, although planned and implement typically expanded into other departments or into the company network. Once you have determined what networks are in existence, you should review how the network to be implemented will tie into existing networks.

10. <u>CONCLUSION</u>

We have seen that there is an OSI Reference model which is standard, but how many layers in the reference model actually have standards associated with them?

In the first layer, IEEE Project 802 has elected to use the coaxial and twisted pair cables. Another standard at this layer is the fiber optic cable. We have also seen that transmission at on this layer may be either baseband or broadband.

The standards most commonly used at the Data Link Layer are X.25 for WANs and IEEE 802 for LANs. The IEEE 802 Project uses Logical Link Control (LLC), IEEE 802.2. in conjunction with Media Access Control (MAC) which may be one of the following: CSMA/CD (Collision Sense Multiple Access with Collision Detection) - IEEE 802.3, Token Bus - IEEE 802.4 or Token Ring - IEEE 802.5.

Standards in layers 3-6 are fairly new and not in wide use as of this writing, however, there are a few defacto standards that are commonly used in the engineering environment today. One of these defacto standards is the TCP/IP.

The Application layer has many standards associated with it. These standards are also fairly new and just now beginning to be used in a large base. The standards for layer 7 include the following: File Transfer, Access and Management (FTAM), Job Transfer And Manipulation (JTAM), Virtual Terminal Service and Protocol (VTAM) and X.400 which is a mail facility.

Because there will be different protocols at all layers of the OSI model for a long time to come, there will also be a need for internetworking equipment such as repeaters, bridges and gateways.

OSI networks based on standards from layer 1 to layer 7 are quickly coming into existence. The first standard is Manufacturing Automation Protocol and Technical Office Protocol (MAP/TOP) developed by General Motors and Boeing. The second standard is Government OSI Interconnection Specification for Procurement (GOSSIP).

Networks in large use today will not be thrown away in order to jump to an OSI network standard. The question now becomes how will the networks in place today migrate to the OSI standard?

SUGGESTED READING

1. Martin, James
   Local Area Networks
   Prentice Hall, 1989

2. Currie, W. Scott
   LANs Explained, a guide to local area networks
   Ellis Horwood Limited, 1988

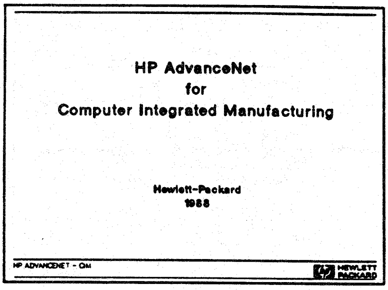
3. Tanenbaum, Andrew S.
   Computer Networks 2nd edition
   Prentice Hall, 1988

4. Knightson, Knowles & Larmouth
   Standards for Open Systems Interconnection
   McGraw-Hill Book Company, 1988

CIM Networking
#4668

Brice Clark
Hewlett-Packard
8000 Foothills Blvd.
Roseville, CA 95678-6598

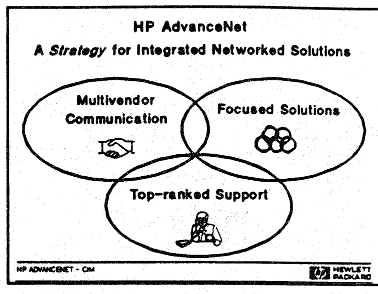
Slide CIM00



HP AdvanceNet for Computer Integrated Manufacturing

(Introductory slide)

Slide CIM01



HP AdvanceNet -- A Strategy for Integrated Networked Solutions

Key Points

- HP AdvanceNet is a strategy for integrating varied computer systems and applications into a complete networked solution. HP AdvanceNet's three strategic "pillars" are:

  - Multivendor communication - because your connectivity needs span a broad range of computing equipment from many different ven dors.

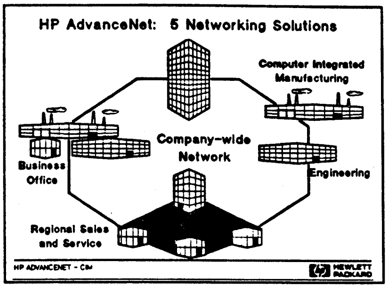
  - Focused solutions - because different kinds of communications problems need custom answers, not just a one-size-fits-all approach.

  - HP's top-ranked support - because when your business depends on fast, reliable access to information, you can't afford to have your network down.

Transition

- Now let's take a look at HP AdvanceNet's focused solutions.

Slide CIM02



HP AdvanceNet: 5 Networking Solutions

Key Points

- There are currently five HP AdvanceNet focused solutions designed to address the special needs and concerns that you, the decision maker, have in building your network. They are:

  - The Business Office - networks supporting applications like office automation, sales and service, administration, marketing, financial, etc.

  - Engineering - networks for the technical office supporting CAD for electrical and mechanical design, simulation, drafting, etc.

  - Manufacturing - networks to support your CIM (Computer Integrated Manufacturing) plans and implementations for all your manufacturing departments.

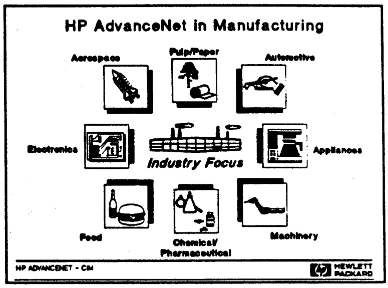
  - Regional Sales and Service - networks for building economical ways to connect regional offices to branch offices, mobile sales representatives, and more.

  - Company-wide - networks for connecting all your operating units together over a corporate backbone.

Transition

- Today we will explore, in significant detail, HP's solution for perhaps the most challenging network of all, CIM.

Slide CIM03



HP AdvanceNet in Manufacturing -- Industry Focus

Key Points

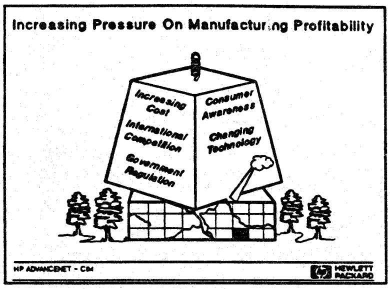
- HP's networking solutions for manufacturing address the needs of many industries. These are industries that HP pays special attention to as it develops products and services.

- Even if your industry is not shown, HP can work with you to select areas of your operation that are a good fit for HP networking solutions.

Transition

- These industries and manufacturing companies in general face common business problems in an increasingly competitive environment.

Slide CIM04



Increasing Pressure on Manufacturing Profitability

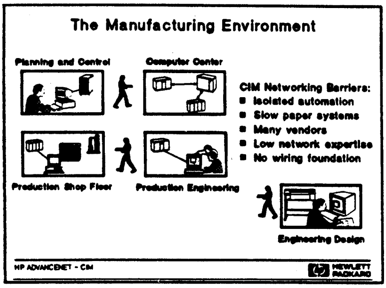
Key Points

- There are many factors that influence a manufacturing company's ability to compete effectively at an acceptable profit level.
- Costs are seemingly ever-rising: cost of physical assets, communications costs in a global economy, legal services, attracting and keeping talented, competent people, to name just a few.
- Few companies enjoy the benefits of protected regional markets. Today, world-class manufacturers must be aggressively penetrating new markets, just as foreign competition is aggressively penetrating local markets.
- In some industries, a high degree of government regulation places an added burden on your company. In the chemical and pharmaceutical industries detailed historical records must be kept to track individual production lots to allow a recall. In the automotive industry, federal safety and emission standards must be met.
- In the 1980s, your customers are making buying decisions with greater care. They place more emphasis on quality and total cost of ownership. Just think about your recent purchases and compare them to those of ten years ago. If you are like most, you'll see a difference.
- On top of all this, technological change is occurring at an ever-increasing rate. Today's automobile is a computer network on wheels, made using a wide range of computer-controlled robots, machine tools, inspection equipment, and support systems. Keeping up with this change is a challenge, with the ever-present threat that if you don't your competitors will. Getting new products to market fast is a key competitive asset but it requires new expertise and that costs money.

Transition

- What leading manufacturers are beginning to discover is that they need to develop a "real-time" management system for continuous improvement in all aspects of their businesses. This means information to the right person, at the right time, in the right form. The electronic information network has become the vital foundation for getting and maintaining a competitive advantage.

Slide CIM05



The Manufacturing Environment

Key Points

- Today's manufacturing environment consists of many interrelated activities. Historically, however, the deployment of information systems has often been fragmented and departmentalized, resulting in what the industry now calls "islands of automation."

- As the human silhouette in the graphic indicates, information often moves between departments or islands of automation via paper. Paper is slow, not easily formatted for assimilation, and usually is not accessible to the right people (the ones who can act on the information).

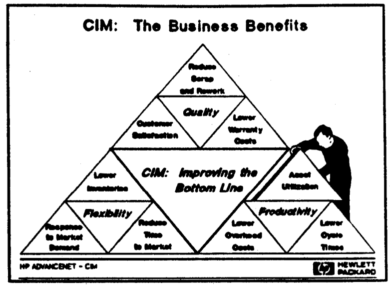
- To speed the flow of information and to ensure that it reaches the right people, you can turn to electronic networks but you quickly encounter a sea of incompatible computers from a variety of vendors. You have networks like IBM's SNA, DEC's DECNet, and many other proprietary schemes, de facto-standard networks like ARPA/BSD, and new international standards like MAP and X.400. You find yourself overwhelmed with options and opinions, and "underwhelmed" by the expertise on hand to tackle the problem.

- Your past investments in computer systems have left you with a spaghetti-like wire maze with many types and sizes, and the whole mess is neither documented nor of much use anymore. In short, you lack a basic wiring foundation to build your network on.

Transition

- You need help, guidance, tools, training.

Slide CIM06



CIM: The Business Benefits

Key Points

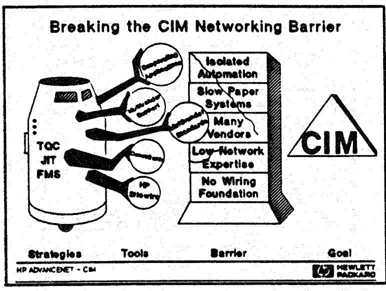
- The business benefits of CIM are many. Ultimately you are interested in improving the bottom line ... profit. At HP we have learned that by using the concepts of process and continuous improvement one can realize benefits in quality, productivity, and flexibility.

- The graphic indicates just a few examples of areas for improvement. And contrary to popular belief, a focus on process and continuous improvement can get you benefits across the board without making tradeoffs like quality versus low cost necessary.

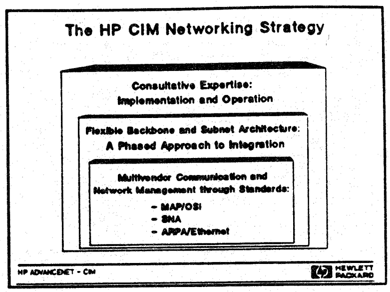
Transition

- So now you're ready to get going. How can HP help and what is HP's approach to CIM networking?

Slide CIM07



Slide CIM08



Breaking the CIM Networking Barrier

Key Points

- Breaking down the CIM barriers begins with a long-range vision of where you want to be. This requires selecting the strategies that best fit your business needs - strategies like TQC (Total Quality Control), FMS (Flexible Manufacturing Systems), TPM (Total Preventative Maintenance), or combinations of them.

- As you implement these strategies with applications and start to link these applications together, you will need the networking tools to break down communications barriers.

  - Standards provide a way to tie together systems from diverse vendors.

  - Applications must interoperate, meaning that two applications will be able to accept data from one another and interpret it.

  - A wiring system that offers all the connectivity you need today and well into the future.

  - Experts who can help you design, implement, manage, and expand your network as your business grows and your needs change.

Transition

- As these barriers fall you will be able to move closer to your ultimate goal.

The HP CIM Networking Strategy

Key Points

- HP's CIM networking strategy is built on three "pillars."

  - Multivendor communications and network management through standards to integrate systems in the multivendor environment.

  - Flexible backbone and subnet architecture for a phased approach to integration, because a CIM network (or CIM) can't be implemented all at once.

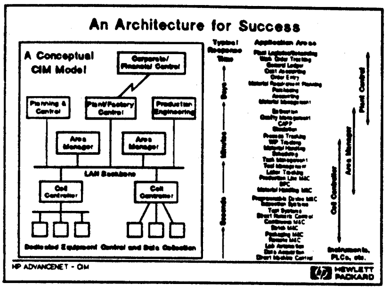
  - Consultative expertise in implementation and operations; HP understands the networking business, so you don't have to.

- HP is committed to providing easy-to-use network products for a full range of computer systems, fully supported on your industry-standard-compatible network. We will provide this high level of functionality, performance, and quality at the lowest total cost of ownership.

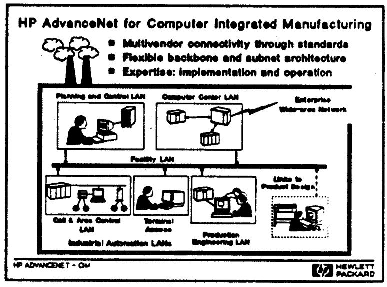
Transition

- To build a corporate CIM networking strategy, we must understand the conceptual model for CIM applications and networks. Let's take a look at that now.

Slide CIM09



Slide CIM10



An Architecture for Success

Key Points

- This very simplified conceptual model of CIM illustrates several concepts. First, notice the response time scale in the center of the graphic. As we move up from the millisecond world of machines, the effective "lifetime" of pieces of information increases until we reach the top of the hierarchy at the corporate EDP center. Our network must take these timing considerations into account.

- Second, notice the wide range of applications that span the CIM hierarchy. Also notice the significant overlap between those applications that might be called cell controller applications and those called area management. The names "cell" and "area manager" are really just conveniences; it is the applications that matter.

- Finally, notice the block diagram showing a "facility" backbone LAN supporting cell controllers, area managers, subnets for planning and control and production engineering, and the plant host. Such an architecture lets you plan big but start small. CIM is a big investment, requiring a lot of expertise. It is wise not to bite off more than you can chew.

Transition

- Now that we have a common framework for CIM networking and the basic elements of HP's CIM networking strategy, it is time to take a more in-depth look at HP's solutions for building your CIM network.

HP AdvanceNet for Computer Integrated Manufacturing

Key Points

- Here you can get a bird's-eye view of HP AdvanceNet for CIM. It has many elements that we have covered in the last few minutes - the manufacturing environment, the three pillars of the HP CIM networking strategy, and the conceptual CIM model. What you see is, in fact, a synthesis of all three.

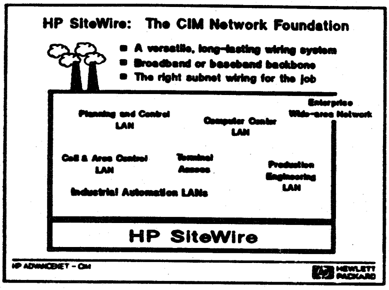
- CIM networking must provide the communications infrastructure to integrate the many functions that make up your manufacturing operation
  - from the shop floor,
  - to planning and control,
  - to computer center data bases and business applications,
  - to corporate systems and field sales,
  - to suppliers and customers via EDI (Electronic Data Interchange),
  - to production engineering designing the manufacturing process,
  - and the important links to product design.

- CIM networking must blend commercial and technical systems to provide timely information to people and machines. It is a critical building block in creating a "real-time" management system to achieve that special edge in an increasingly competitive business world.

Transition

- Let's begin our in-depth look at CIM networking with HP SiteWire, the foundation for CIM networking.

Slide CIM11



HP SiteWire: The CIM Network Foundation

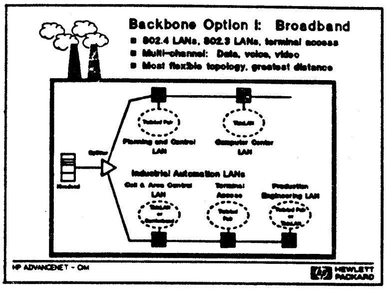
Key Points

- Just as the network is the foundation for achieving integration of machines and information, HP SiteWire is the foundation for your network.
- HP SiteWire offers you the flexibility to select either a broadband or baseband backbone - you can select the optimal subnet wiring for the job.
- Through HP's system integrators program, stringent standards have been established to ensure that the wiring installed meets your standards for quality at the lowest cost of ownership.
- HP SiteWire gives you the ability to integrate your facility departments in a consistent, step-by-step fashion.

Transition

- Now let's explore several HP SiteWire alternatives you can consider for your CIM network.

Slide CIM12



Backbone Option I: Broadband

Key Points

- The first backbone option is broadband. Broadband offers the highest degree of networking flexibility throughout the enterprise.

    - First, broadband can support a variety of connections including 802.4 (token-bus), 802.3, RS-232, HDLC, SDLC, and T1.

    - Second, broadband supports a multi-channel communications scheme, allowing you to transmit data, voice, and video all on the same cable.

    - Third, broadband offers the highest degree of flexibility in terms of topologies supported. Broadband can support twisted-pair and ThinLAN, as well as carrierband subnets. (Carrierband is a 5 Mbps network scheme that is based on a baseband or single-channel signaling method. Similar to broadband, carrierband supports the 802.4 physical layer as well as phase coherent FSK modulation.) In addition, broadband is not sensitive to distance, can be used for full coverage of large sites, and offers high reliability and noise immunity via implementation of the AM/PSK modulation scheme.

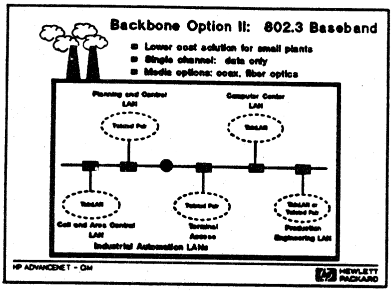
    - Fourth, broadband is a highly reliable medium, having been used extensively by the cable television industry for over 20 years.

- The graphic shows the environments typically considered in CIM networking and HP's recommended wiring scheme for each subnet: twisted-pair for office environments and LAN terminal access, ThinLAN for the computer center, and ThinLAN (ARPA/TCP/IP) or carrierband (MAP) for the industrial automation environment.

Transition

- Now let's look at the second backbone alternative, baseband.

Slide CIM13



Backbone Option II: 802.3/Ethernet Baseband

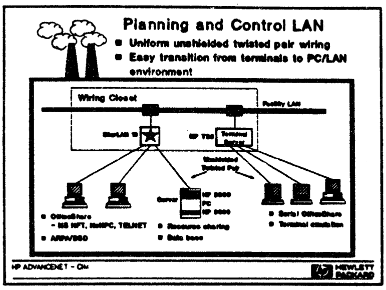
Key Points

- The second backbone option you might consider is baseband. Baseband is based on the CSMA/CD access method and offers a 10 Mbps data rate. Baseband offers several benefits:
  - Lower Cost: Baseband might be an ideal solution for smaller plant networks where the networking requirements call for data communications only.
  - Media Flexibility: With baseband you have the flexibility to choose between either a coax or fiber optic cabling scheme. You can choose the medium that best fits your internal security or distance requirements.
  - Supported Services: Baseband is ideally suited for TCP/IP-based networking services, such as ARPA or HP's Network Services.
  - If your networking requirements grow, your 802.3 subnets can be segmented and connected into a broadband backbone.
- The graphic shows the recommended wiring schemes for LANs typically considered in CIM networking. Subnets supported on an 802.3 backbone are the same as 802.4, with the exception of carrierband.

Transition

- That completes our discussion of HP SiteWire and the two facility backbone alternatives. Next we'll go through an overview of each CIM LAN environment and the networking options available, starting with Planning and Control.

Slide CIM14



Planning and Control LAN

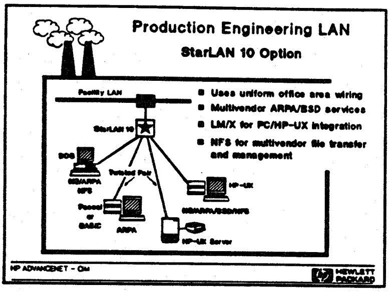
Key Points

- Typically the Planning and Control environment includes functions such as purchasing, order administration, and shipping and receiving. These functions frequently require access to office applications such as IMAGE or HPDESK as well as the ability to share resources such as discs and printers.
- For LAN terminal access, HP recommends the use of the HP TS8 LAN terminal server based on twisted-pair, RS-232 connections. The TS8 allows users to access applications on multiple hosts or to access multiple applications on the same host concurrently. Using PCs in this environment is also possible with HP's terminal emulation packages such as AdvanceLink, or if resource sharing is a requirement, with HP's OfficeShare product.
- As your users' needs become more sophisticated and require the capabilities of a PC, the transition is made easy with StarLAN 10. StarLAN 10 offers 10 Mbps performance over unshielded twisted-pair wire. The transition is almost as simple as unplugging your PCs from your TS8 and plugging them into the StarLAN 10 hub.
- HP also offers several powerful products for the PC/LAN environment. HP OfficeShare offers functionality ranging from simple terminal emulation to network file transfer and resource sharing. In addition, HP supports multivendor ARPA/BSD services in the StarLAN 10 environment.

Transition

- As you have seen here, StarLAN 10 offers a powerful networking solution for the office environment. As we move next to Production Engineering LANs we'll expand on the capabilities and flexibility that StarLAN 10 offers.

Slide CIM15



Production Engineering LAN -- StarLAN 10 Option

Key Points

- The Production Engineering environment typically consists of PCs and workstations that often require access to mainframe computers and peripherals as well as other workgroups such as in product design.

- While there are several networking alternatives available for Production Engineering LANs, the preferred alternative shown here is StarLAN 10. StarLAN 10 uses uniform office area wiring and supports both multivendor ARPA/BSD/NFS services as well as HP's Network Services. For the DOS environment, HP offers ARPA and NS services as well as NFS (PC-NFS from SUN Microsystems). For Pascal or Basic workstations, multivendor ARPA/BSD services are available from Network Research Corporation. Finally, for the HP-UX environment, HP supports multivendor ARPA/BSD and NFS services as well as NS.

- Implementing your 802.3 network with StarLAN 10 offers the following advantages:

  - A consistent office area wiring strategy for both terminal connect and local area networks. This can save time and money in cable management and nodal adds, moves, and changes.

  - Support of up to 100 meters of twisted-pair wiring between a StarLAN 10 hub and a remote system. Each hub can support up to 12 twisted-pair connections.

  - Ability to cascade StarLAN 10 hubs up to three levels to meet your future needs for expansion.

  - Coexistence of voice and data on the same cable bundle.

- For PC/HP-UX integration, HP offers LAN Manager on UNIX® (LM/X), a powerful tool for management and sharing of remote printers and plotters as well as interprocess communication.

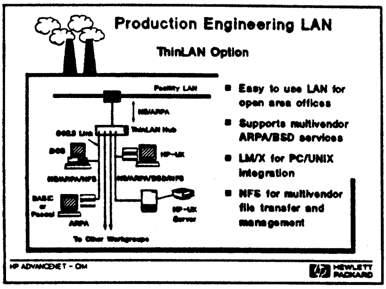
- HP gives you the flexibility to integrate your multivendor or all-HP environment with tools and services that optimize the productivity of your Production Engineering workgroup. The end result: faster time-to-market as a result of an improved production design cycle.

Transition

- That completes our discussion of HP's networking solutions for the office LANs used in CIM networking. We'll now turn our attention to the factory floor and several networking alternatives for Industrial Automation LANs. (Note - there are several backup slides that can be used for the Production Engineering environment. The first slide describes a ThinLAN network and the second describes methods to connect an installed SRM with the facility LAN.)

UNIX is a registered trademark of AT&T in the U.S. and other countries.

Slide CIM15buA



Production Engineering LAN -- ThinLAN Option

Key Points

- A second alternative for Production Engineering LANs is ThinLAN. The ThinLAN baseband coax option allows you to communicate between HP-UX, Pascal, Basic, or DOS workstations with multivendor ARPA/BSD services or HP's NS services. ARPA services for Basic and Pascal workstations are offered by Network Research Corporation.

- ThinLAN offers you several advantages:

  - Standardization on the IEEE 802.3/Ethernet physical layer specification, complemented by support of ARPA/BSD services, helps to ensure communication with other vendors' systems.

  - 10 Mbit/second data transfer burst rate with CSMA/CD signaling protocol provides net work access without centralized control.

  - Each ThinLAN segment (four are supported on each ThinLAN hub) can support as many as 30 separate nodes at a distance of 185 meters.

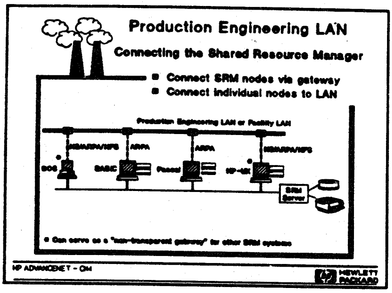
- In addition, HP's new UNIX LAN manager, LM/X, offers support of shared discs, printers, and plotters as well as Network InterProcess Communication for the PC environment. These capabilities are especially important for the Production Engineering environment, where information and resources are frequently shared.

- Finally, if you require file sharing between multivendor file systems, HP offers Network File System (NFS) for HP-UX workstations as well as DOS workstations (Network Research Corporation's Fusion product). NFS allows you to integrate applications, systems, and peripherals in a multivendor environment.

Transition

- A third solution for Production Engineering LANs is the HP Shared Resource Manager (SRM). If you currently have an SRM installed in your facility, let's take a look at some ways to integrate your SRM with other enterprise LANs.

Slide CIM15buB



Production Engineering LAN -- Connecting the Shared Resource Manager

Key Points

- If you already have an installed base of Pascal and/or Basic workstations you may already be familiar with HP's Shared Resource Manager (SRM). The SRM saves you money and improves productivity by supporting shared discs, plotters, and printers, all from a dedicated central controller. Today, the SRM allows you to access shared resources from DOS, Basic, Pascal, and HP-UX workstations. If you have an SRM installed, there are several ways to integrate your SRM with other facility LANs:

  - Gateways: One cost-effective approach to integration is to use either a DOS or HP-UX workstation as a gateway to your production engineering or facility LAN. This approach might be appropriate if you require occasional communication with other departments from a number of your SRM workstations. Communication with other facility LANs can be achieved with multivendor ARPA/NFS services or NS.

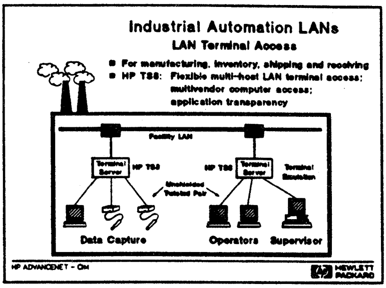
  - Direct Connect: If your workstations require frequent access to other facility LANs, or if that access involves heavy network traffic, another alternative to consider is a direct connection between your workstation and the production engineering or facility LAN. Communication can be achieved from your DOS, Basic, Pascal, or HP-UX workstation with ARPA services or NS. NFS services are also available for HP-UX and DOS workstations.

- With either the direct connect or gateway alternatives you now have several ways to integrate your SRM with other departments within your facility, and you can choose the alternative that best meets your communications requirements.

Transition

- That completes our discussion on solutions for Production Engineering LANs. The next department integrated into a total CIM solution is the industrial automation area. HP gives you a number of alternatives for industrial automation LAN implementations based on industry-standard, de facto-standard, or proprietary communications.

Slide CIM16



Industrial Automation LANs -- LAN Terminal Access

Key Points

- A common problem in CIM networking is implementing a flexible terminal connection scheme that will easily evolve with your increasingly sophisticated networking needs. Consider a LAN terminal connect scheme if you have terminals that require access to multiple applications that reside on different host computers around the factory. Departments that may require LAN terminal connect capability include the factory floor, inventory, and shipping and receiving. HP offers multi-host LAN terminal connect capability with its new HP Terminal Server 8 (TS8). The TS8 provides flexible multi-host LAN terminal access in a multivendor environment.

- As the name implies, each HP TS8 can support up to eight unshielded twisted-pair RS-232 connections, which may be used for a variety of devices including bar code readers, terminals, or PCs. The HP TS8 offers excellent overall throughput at 96 kbps, or 12 kbps per port.

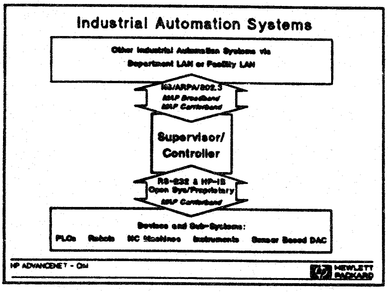
- The TS8 can be used in two configurations. The first is a back-to-back configuration (two back-to-back TS8s) if access is desired to a system that does not support TCP/IP-Telnet services. The back-to-back configuration offers the ability to connect to any computer on the LAN. The second configuration is direct access to a host via a single TS8, provided that the host supports TCP/IP-Telnet services.

- The TS8 is supported on baseband media only. HP is currently evaluating market requirements for broadband connections.

Transition

- An important function in the industrial automation department is provided by the supervisor or controller. Let's look at some key communications requirements for a supervisor/controller on the factory floor.

Slide CIM17



Industrial Automation Systems

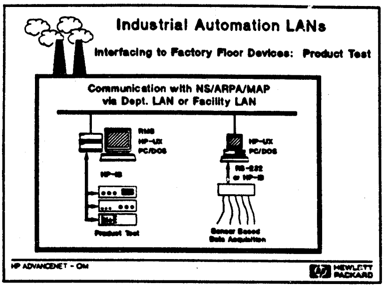
Key Points

- This graphic provides an overview of the communications products HP offers for the Industrial Automation environment. The core system in our communications hierarchy is the supervisor or controller. This system plays a key role in controlling factory floor activity and communicating the results of that activity to other systems around the enterprise via the department LAN or facility LAN. HP offers a variety of system platforms to perform the supervisor/controller function on the factory floor. The following are general guidelines for choosing the platform that best meets your computing and communications needs:
  - HP-UX platforms offer the greatest breadth of networking functionality for factory-floor communications. Today, the HP-UX systems support full ARPA/TCP/IP services for multivendor communications as well as NS for HP-to-HP communications (upstream communications). In addition, the HP-UX systems are the strategic platform for all multivendor OSI communications including MAP 3.0 (upstream and downstream communications). To communicate with non-OSI devices on the factory floor (downstream communications), the HP 9000/800 supports the HP Device Interface System, a software-based tool that facilitates flexible communication to devices that communicate on proprietary factory networks.
  - RTE platforms offer limited communications capability and should be used for niche applications such as real-time data acquisition or device control. Today, the RTE systems support PCIF for custom PLC communication on the factory floor as well as HP-IB and RS-232 (downstream communications). RTE systems also support NS services for upstream HP-to-HP communications and limited ARPA services (FTP) for multivendor communications.
  - DOS platforms are appropriate for several key factory-floor applications, such as end-user access or front-end systems for primitive factory-floor devices. Direct downstream device communications can be established with RS-232 or HP-IB. Upstream, DOS platforms support ARPA services and NS.

Transition

- Next we'll take an in-depth look at HP's upstream and downstream Industrial Automation communications alternatives.

Slide CIM18a



Industrial Automation LANs -- Interfacing to Factory-floor Devices: Product Test

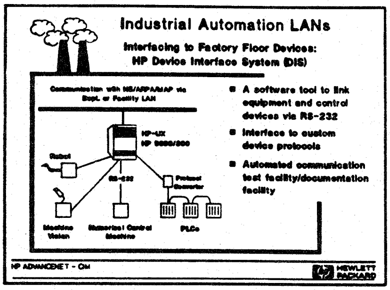
Key Points

- A fundamental communications link for Industrial Automation LANs is the link between workcell controllers and factory-floor devices. These links are used in a variety of situations, from recipe management to alarming to process monitoring and machine control.
- Shown here are several solutions for point-to-point connections that may be required for the product test environment. If your needs call for a low-cost serial interface, HP offers RS-232 interfaces for HP-UX workstations and PCs.
- On the other hand, if your device communications call for HP-IB interfaces, HP offers HP-IB connectivity for both HP-UX workstations and PCs.
- As shown, HP also offers solutions to meet your most stringent upstream communications requirements. If you require multivendor services you can select from ARPA for the 802.3 environment or MAP for the 802.4 environment. NS services are also available for HP-to-HP communications.

Transition

- In addition to the physical connection, customers often require custom protocol interfaces to communicate with specific devices. For the HP 9000/800, HP gives you this capability with the Device Interface System (DIS).

Slide CIM18b



Industrial Automation LANs -- Interfacing to Factory-floor Devices: HP Device Interface System

Key Points

- A common communications requirement for the Industrial Automation LAN is linking workcell controllers and Programmable Logic Controllers via a proprietary network. In these cases the workcell controller often manages a number of intelligent and unintelligent devices via a proprietary LAN optimized for the functionality and high-speed communications needed between devices.

- HP's new Device Interface System (DIS) offers a flexible software-based tool to interface to these proprietary PLC networks via a standard RS-232 interface on HP 9000/800 systems. DIS makes it easy to develop custom communications protocols with the help of an automated protocol test facility as well as an automated protocol documentation facility. User demand has also indicated a need to provide common proprietary protocols as part of the standard package (e.g., Allen Bradley Data Highway). The DIS team is currently evaluating the requirements for a contributed protocol library or a reference list for VABs that have already created these protocols.

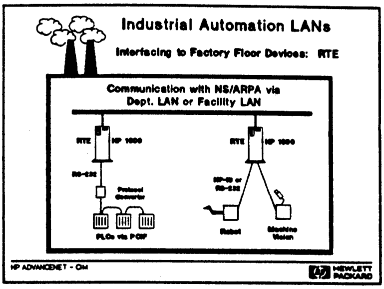
- Once you establish interfaces to your factory-floor devices and PLCs, the HP 9000/800 can be easily integrated with other workcell controllers in the factory as well as other facility departments with multivendor ARPA services or MAP as well as HP's NS services.

- In the near future, HP plans to offer the HP DIS system over the new Real Time Interface (RTI) card recently announced. This enhancement will offer 19.2KB performance (as compared to 9.6KB performance over the MUX card today) for real-time factory-floor communications needs.

Transition

- If you need an HP 1000 at the workcell level, HP also offers a full range of interfaces for connecting to factory-floor devices (backup slide).

Slide CIM18buA



Industrial Automation LANs -- Interfacing to Factory-floor Devices: RTE Systems

Key Points

- Whether you require general point-to-point serial connectivity or access to proprietary PLC networks on the factory floor, HP offers solutions for the HP 1000 that meet your factory communications requirements.

- For general point-to-point connectivity, HP offers a range of EIA-compatible products for the HP 1000 A-Series systems including RS-232 and RS-422 links. If your device-level interfaces call for proprietary protocols on top of a standard link, HP offers the Programmable Serial Interface Card and PSI Firmware Development Package, which offer the flexibility for users to develop any number of customized protocols.

- To interface to Programmable Logic Controllers on proprietary factory networks, HP offers the Programmable Controller Interface/1000 (PCIF/1000). HP PCIF/1000 supports interfaces to a number of factory networks, including Allen Bradley's Data Highway and Gould Modicon's Modbus.

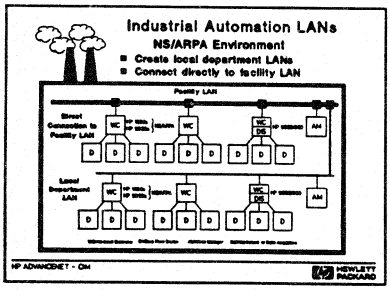
- To communicate with other workcell controllers as well as other departments around the facility, the HP 1000 now supports multivendor ARPA services as well as HP's NS services, allowing you maximum flexibility in designing your integrated networking environment.

- The HP 1000 should be a recommended solution for only those application requirements that cannot be addressed by the HP-UX systems today, such as real-time data acquisition. Strategically, the HP-UX platform will provide the highest flexibility to meet the evolution of customers' communications and computing needs.

Transition

- Now that we understand our fundamental communications requirements for devices on the factory floor, let's turn our attention to the Industrial Automation Department LAN.

Slide CIM19



Industrial Automation LANs -- NS/ARPA Environment

Key Points

- Whether your industrial automation communications requirements call for de facto-standard, proprietary, or industry-standard networking solutions, HP offers a full range of products to achieve your integration needs.

- For the 802.3 environment, there are several topologies that you might consider, depending on your applications requirements.

  - Direct connect to the facility LAN: If your workcells are separated by a great distance, consider connecting your workcell clusters directly to the facility LAN. With this method, you can conveniently place your cells wherever desired around the factory and integrate them with an Area Manager or other department systems as needed.

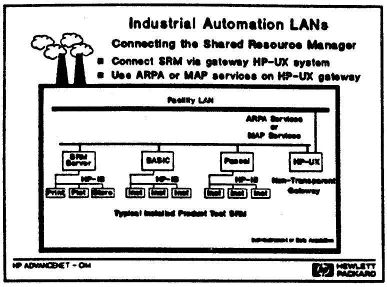
  - Local department LAN: Perhaps the easiest and lowest-risk solution is to create a local subnetwork for your workcell control systems that connects to the facility backbone. The local department LAN offers the benefit of isolating related factory-floor functions from the rest of the facility network, ensuring maximum network uptime and efficiency. As previously mentioned, a baseband LAN is also a very cost-effective solution for departmental communications.

- HP also offers flexibility in the choice of services to meet your networking requirements. Whether you standardize on an HP-UX operating system with the HP 9000/800 or select the HP 1000 RTE-based systems, you can select either multivendor ARPA services or HP's NS services.

Transition

- If you already have an installed base of Basic or Pascal workstations running on an HP SRM, you may also be interested in integrating your SRM with other multivendor LANs around the facility. (Backup slide, integrating the SRM.)

Slide CIM19buA



Industrial Automation LANs -- Connecting the Shared Resource Manager

Key Points

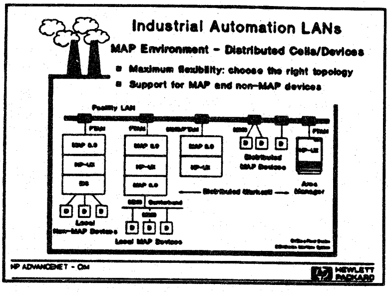
- If you have an installed base of Basic and/or Pascal workstations running on an SRM in your factory environment and wish to integrate these systems with other workcell clusters or facility departments, HP offers gateway support for HP-UX systems on the SRM.

- Through an HP-UX gateway, communications to other workcell clusters or facility departments can be achieved with multivendor ARPA or MAP services. In addition, the HP-UX gateway can serve as an Area Manager for your SRM subnet. HP-UX workstations access the SRM with eight commands supplied by the "SRM Utilities for HP-UX," which allows the HP-UX system to manage files and directories on the SRM.

Transition

- Over the next two to three years, as multivendor factory integration becomes a strategic goal for more and more firms, use of MAP 3.0 communications in a production environment will increase significantly. With HP OSI communications, your multivendor OSI requirements will be met both today and tomorrow.

Slide CIM20



Industrial Automation LANs -- MAP Environment, Distributed Cells/Devices

Key Points

- Similar to the discussion of 802.3-based Industrial Automation LANs, your industry-standard OSI networking strategy may call for a distributed workcell environment or a departmental subnet environment. This slide shows a MAP 3.0 HP-UX-based distributed workcell environment, which would be appropriate to consider for cells or area managers that require communication between one another but may be a great distance from each other within the factory.

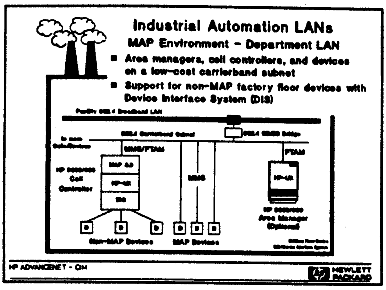
- Since communications requirements often vary between cell controllers and devices and cell controllers and area managers, HP offers several key MAP 3.0 services to achieve multivendor integration on the factory floor. MAP 3.0 FTAM (File Transfer, Access and Management) services would be appropriate for file transfers between two cell controllers or between a cell controller and area manager. On the other hand, if cell-to-device communications are required, HP offers MAP 3.0 MMS (Manufacturing Message Services) for functionality ranging from simple file transfers to complex device control instructions. HP's MAP 3.0 services are supported on 802.4 broadband and carrierband media, allowing you to select the best wiring solution for your factory communications requirements.

- Today, many customers have an installed base of systems and devices that require proprietary communications interfaces. With HP's Device Interface System you can still communicate to your proprietary device-level LANs while using MAP to communicate between multivendor cell controllers and area managers.

Transition

- Whether your Industrial Automation LANs require proprietary or industry-standard multivendor communications, HP leads the way in offering the highest degree of flexibility in implementing a totally integrated solution for your industrial automation LAN. Let's now look at an example of a departmental LAN based on MAP 3.0 communications.

Slide CIM21



Industrial Automation LANs -- MAP Environment, Department LAN

Key Points

- For multivendor workcell clusters that are close in proximity, but separate functionally, consider integrating these cells with MAP 3.0 services over a lower-cost carrierband subnetwork. Carrierband is considered a lower-cost alternative to broadband because it does not require a head end, amplifiers, RF modems, or a complex design and installation strategy. What carrierband does offer is the benefit of the low-noise PC FSK (Phase Coherent Frequency Shift Keying) signaling technique on a single channel similar to baseband.

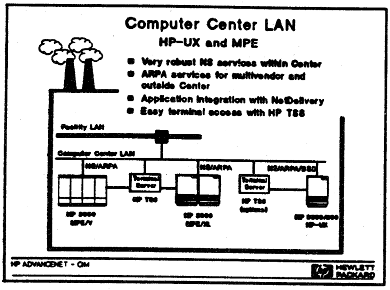
- An 802.4 carrierband subnet supports HP MAP 3.0 MMS and FTAM services for communications between cell controllers and devices, cell controllers and area managers, or between two cell controllers. HP's Device Interface System can be used for applications that call for proprietary communications between cell controllers and factory-floor devices.

- To integrate carrierband subnets with the facility LAN, HP has made a statement of intent for 1990 support of an 802.4 carrierband-to-broadband bridge, rounding out an integrated 802.4 product offering for industrial automation LANs.

Transition

- Any complete discussion of CIM networking must take into consideration information exchange among all facility departments, not just those that deal with the daily activities that occur in manufacturing. Next we'll turn our attention to the networking solutions for the computer center and strategies for integrating the computer center and other facility departments.

Slide CIM22



Computer Center LAN -- HP-UX and MPE

Key Points

- The computer center is typically the central hub of information consolidation for a facility or in some cases the entire enterprise. As the information center for the facility, the computer center must be able to pull information from each department on a regular basis, often many times daily. (An example might be scheduling material requirements on the computer center MRP system based on recent production information.)

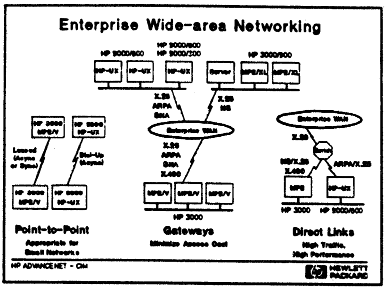
- The computer center establishes communication with other departmental LANs through the facility backbone. For services, HP offers ARPA on MPE and HP-UX operating systems for multivendor communication within the center as well as with other departments. To communicate between HP systems, HP offers NS services for MPE/V, HP-UX, and MPE-XL operating systems. NS offers a robust set of services for distributed applications, including remote file and database access, directory services, and peripheral sharing (HP OfficeShare).

- Operators, supervisors, and other department end users often need terminal connection to several systems within the computer center for database queries, report generation, or electronic mail. The HP TS8 is a flexible solution for connecting users around the facility to multiple systems within the center. The slide shows the TS8 in a back-to-back configuration on systems that may not have Telnet services (MPE/V and MPE/XL). Connecting to an HP-UX system, the back-to-back TS8 configuration can be used as an option for heavy terminal traffic situations, or the TELNET service may be used to establish sessions directly over the LAN from a single TS8.

Transition

- The final component of the facility-wide CIM networking solution is access to the enterprise-wide network. HP offers several solutions for enterprise-wide access including X.25 point-to-point, direct connect, and gateway solutions as well as access to SNA networks.

Slide CIM23



Enterprise Wide-area Networking

Key Points

- For small networks or two systems that require frequent wide-area communication between each other, HP offers point-to-point X.25 communications links for the HP 3000 MPE/V systems as well as the HP 9000/800 systems. Access is provided via leased or dial-up phone lines. Enterprise-wide communications facilitate the consolidation and exchange of financial and operating information among many departments across the company.

- Point-to-point links for the HP 3000 can be used for programmatic access as well as for higher-level user services such as Network File Transfer, Virtual Terminal and Remote Database Access with NS 3000/V software. High-speed synchronous as well as lower-cost, lower-speed asynchronous communications are available.

- Point-to-point links for the HP 9000/800 systems are currently limited to asynchronous links but offer the use of the standard HP-UX networking commands cu (sets up communications parameters) and uucp (file copy command). Layer three X.25 services are also available.

- If you have a number of systems on a LAN that occasionally require wide-area communications, you may want to consider one of several gateway solutions for enterprise-wide communications.

- Gateway solutions for enterprise-wide communications are appropriate for systems tied together on a local area network that require occasional access to other systems around the enterprise.

Gateway solutions can be quite cost-effective over point-to-point or direct connect wide-area solutions because enterprise wide-area traffic is limited to a few systems.

- HP offers a high degree of flexibility in selecting a gateway solution that best fits your communications requirements. For HP 9000/800 systems, layer three X.25 access provides high-speed communications to other enterprise wide-area systems. For multivendor communications, ARPA services are also available over an X.25 wide-area link. If communications are required between the HP 9000/800 and an IBM 370-compatible mainframe, SNA services are available as well. If needed the HP 9000/800 can be used as a router for any TCP/IP traffic over the X.25 network.

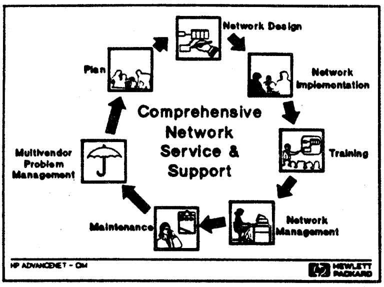
- HP 3000 MPE/V systems also support layer three X.25 access as well as IBM SNA services and HP NS services in a gateway environment. The HP 3000/900 systems access the enterprise wide-area network through a gateway with layer three X.25 services as well as HP NS. If you require multivendor communication with industry-standard OSI services, X.400 is available over X.25 links on MPE/V systems as well. HP's third solution for enterprise-wide connectivity is the direct link approach. Direct links to an enterprise wide-area network offer the cost and overhead savings associated with gateway systems. They can also be cost-effective if traffic requirements can be supported through the use of an X.25 line concentrator.

- Today, direct wide-area links are supported on HP 9000/800 systems as well as HP 3000 MPE/V systems. On the HP 9000/800 direct wide-area access can be accomplished with multivendor ARPA services or X.25 layer three access. If direct wide-area access is required on MPE/V systems, layer three X.25 access is offered as well as industry-standard X.400 services for multivendor communications. NS services are also available for wide-area communication with other HP systems. Another popular solution is the CISCO X.25 router. The dedicated router provides general IP routing in a LAN-connected gateway. There are currently over 100 CISCO routers in use within HP today.

Transition

- Obviously no networking solution is complete without considering the design, installation, and maintenance of the network with a goal of minimizing cost of ownership. HP offers a comprehensive range of network support services to meet the unique requirements of each HP customer.

Slide CIM24



Comprehensive Network Service and Support

Key Points

- HP's network service and support programs are consistently rated best in the industry. To allow the maximum flexibility in meeting your business needs, you can customize your design and support program from a number of modular products:

  - HP Network Planning and Design: An HP Network Consultant will analyze a customer's communication requirements and develop a detailed design based on customer needs.

  - HP Network Prepare: HP works with the customer to develop a network implementation plan that contains a schedule of critical activities and recommendations for network staffing, training, and operations procedures.

  - HP Network Startup: HP helps the customer get the network up and running quickly, by providing coordination assistance for installation activities and resources, connection verification testing, and complete network documentation.

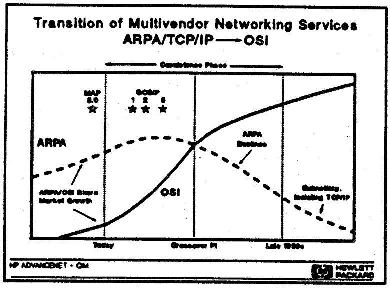
  - HP NetAssure: Maximum network uptime is assured through network problem isolation and problem management in a multivendor environment.

  - Customer Education: HP provides a range of standard and customized training for network users, operators, and managers.

Transition

- Today and into the future, OSI services will offer customers increased functionality based on industry-standard protocols. Let's take a look at how the transition from de facto to OSI standards will take place and the HP de facto/OSI protocol coexistence strategy.

Slide CIM25a



Transition of Multivendor Networking Services -> ARPA/OSI

Key Points

- Just as ARPA was in its infancy in the marketplace 7 to 10 years ago, OSI is in that same place today. Use of OSI in the marketplace today is mostly done by innovative users who are on the leading edge of technology. These innovators set the stage for the general market to follow based on the examples they set for technology usage. Following a standard market penetration strategy, full penetration of the OSI marketplace will occur sometime in the late 1990s.

- Several events that will push the adoption of OSI protocols in the marketplace are the completion of OSI specifications, such as MAP 3.0 for manufacturing and GOSIP for government applications. The event that aided the takeoff of ARPA/TCP/IP in the marketplace and is expected to do the same for OSI was the government mandate in 1980 to implement only TCP/IP networks. As the GOSIP specifications evolve through phases one through three, it is anticipated that OSI protocols will be adopted as standard solutions for all government projects and the market will see a steady increase in OSI network purchases.

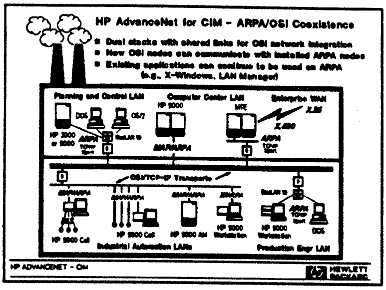
- As a result, the increasing adoption of OSI will result in a decrease in ARPA network purchases, arriving at a crossover point most likely occurring in the mid-1990s. The crossover point can be attributed to the overall growth in market demand for integrated networking solutions as well as the replacement of ARPA networking purchases with OSI solutions. Overall market growth can also be considered the logic behind the shared market growth for ARPA and OSI networking over the next few years.

- By the late 1990s, early 2000s, it is anticipated that OSI networking will be the solution of choice for all applications. The market trend will be to isolate existing ARPA networks in subnets; communication to those subnets, if at all will be accomplished with coexistence solutions to be described in the next slide.

Transition

- Now that we understand the evolution of OSI and TCP/IP one point is clear - in years ahead OSI and TCP/IP networks will need to coexist in the enterprise. Next, we'll discuss the HP strategy for allowing these two networks to coexist during the critical transition period from TCP/IP to OSI.

Slide CIM25b



HP AdvanceNet for CIM -- ARPA/OSI Coexistence

Key Points

- Today, the multivendor networking solution of choice is ARPA/TCP/IP. Many customers, particularly the government, have a large installed base of ARPA networks that will remain in existence for many years to come. To smoothly manage the transition from ARPA-based networks to OSI, HP is leading the way in developing tools that will allow customers' ARPA networks to coexist with new OSI-based networks. It is recommended that customers make the decision to migrate to an OSI environment from an existing ARPA environment (rather than proprietary) so that the issues involved in achieving multivendor interoperability are clearly understood. In addition, a single multivendor networking strategy establishes a common stack from which coexistence can be achieved.

- There are several networking concepts that are crucial to the coexistence of ARPA and OSI networks. The first concept is dual stacks over shared links - a fundamental method for achieving communication between applications written over ARPA services and new applications written over OSI services. The shared links concept is based on the ability to run ARPA services over an OSI stack and/or conversely the ability of OSI services to run over an ARPA stack. The benefits of this capability are important:

  - New applications can be written using OSI services and integrated with existing ARPA networks via the use of a common TCP/IP transport as well as a common link. One might consider this a backward migration strategy. However, this is not a recommended strategy because it further delays the full implementation of OSI networks.

  - Once standardization on a common OSI transport is achieved, applications that were originally written for an ARPA network can still operate in the OSI environment. This strategy protects the customer's investment in ARPA while allowing full OSI network implementations to be achieved. One might consider this a forward migration strategy.

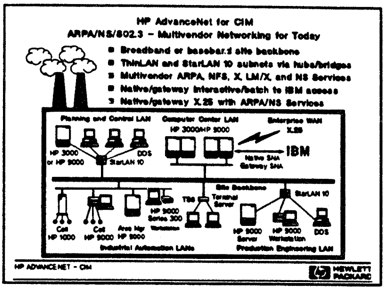
- Another key coexistence concept is the application layer gateway. Application layer gateways will be used to perform full seven-layer translations of information between OSI networks and ARPA networks. Although slower in performance, these gateways may be used for certain applications such as electronic mail, although they are not recommended as a general method to achieve inter-protocol interoperability due to incompatibilities between protocol feature sets.

- HP is committed to providing customers with tools to facilitate the coexistence of multivendor networks. As shown, HP is planning to deliver the dual-stack shared-link technology to facilitate the integration of new OSI nodes with established ARPA nodes. HP is also evaluating market needs for the use of application-layer gateways for certain applications.

Transition

- Now that the relationship between OSI and ARPA networks has been described, let's review HP's CIM networking products for each of these environments.

Slide CIM25c



ARPA/NS/802.3 -- Multivendor Networking for Today

Key Points

- Today, more and more customers are demanding multivendor network services to allow them to select the best computing solutions for their business problems throughout the enterprise. The most widely used de facto standard multivendor networking services available today are ARPA services over Ethernet/802.3.

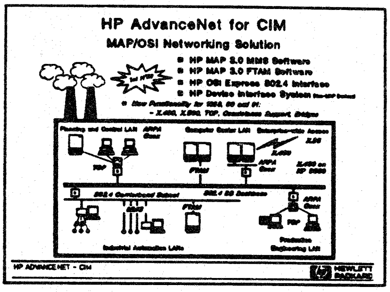
- HP provides a comprehensive ARPA services offering for each department that plays a part in Computer Integrated Manufacturing, from planning and control to enterprise-wide access. ARPA services will be available on the HP 1000 over 802.3 and the HP 9000/800 over 802.3 and X.25 in the first half of 1989. ARPA services on the HP 3000 are now available from Wollongong.

- Multivendor system integration for the PC environment is enhanced with LM/X. LM/X will be available for PC/UNIX integration in the first half of 1989. ARPA services for the PC can also be purchased today.

Transition

- As manufacturers move into the 1990s, OSI will become more and more predominant as the solution of choice for multivendor networking. HP, the leader in multivendor communications, is committed to satisfying customer needs for OSI products both today and tomorrow.

Slide CIM26



Migrating to MAP/OSI with ARPA/NS Coexistence

Key Points

- Most manufacturing enterprises today have chosen to implement a single proprietary network throughout the facility, or a combination of proprietary and industry-standard networks. As more and more multivendor OSI networks are installed, customers will need tools that allow their proprietary or ARPA-based networks to coexist with their OSI-based networks. Understanding customer needs, HP will support ARPA/OSI coexistence tools that allow customers to utilize both ARPA and OSI services over common transports as well as application gateways for certain applications.

- In addition, HP offers a full MAP 3.0 product offering today for broadband and carrierband industrial automation LANs, as well as X.400 communications for enterprise-wide access on MPE/V systems. HP will expand the breadth of its OSI product offering through 1990 and 1991 with additional X.400 products for the HP 9000/800, TOP products for commercial systems, and X.500 directory services for optimal resource utilization in the OSI environment. To complete the OSI product offering, HP will offer industry-standard network management services as well as bridge support to integrate departmental OSI LANs.

- The foundation for HP's OSI communications on technical systems is the HP OSI Express 802.4 interface card. The HP OSI Express interface offers a full seven-layer OSI stack implementation on a plug-in card set. Based on HP VLSI technology, HP OSI Express provides a high-performance network interface with low impact on the CPU. By design, OSI Express has the flexibility to support both broadband and carrierband interfaces as well as other physical layer connections in the future.

Transition

- We have covered all aspects of HP's networking solution for the Computer Integrated Manufacturing environment. I'd be happy to answer any questions on the products or strategies you've seen here today.

Slide CIM26buA

Intersystems Services Summary

	HP 1000 A-Series	HP 3000 MPE	HP 3000 MPE/XL	HP 9000 300-UX	HP 9000 PA-UX	Vectra	Other Vendors
HP NS *	✓	✓	✓	✓	✓	✓	✓ DEC
ARPA	✓ CF	✓	SOI	✓	✓	✓	✓ SUN, DEC APOLLO
Berkeley				✓	✓	✓	✓ SUN, UX
MAP 3.0					✓ CF	CF	✓ COS
SNA	✓ S	✓	✓ G/N	✓ G/N	✓ G/N	✓	✓ IBM
Bisync	✓	✓					✓ IBM

G/N = Gateway/Native S = Special Quote CF = Consult Factory

COS = Corporation for Open Systems Conformance and Interoperability Support

SOI = Statement of Intent

* Services supported varies by processor. Refer to Intersystem Matrix.

HP ADVANCENET - CIM

HEWLETT PACKARD

Intersystem Services Summary

Key Points

- This slide provides a summary of networking services available for each of the HP systems discussed in the CIM networking solution. Use this matrix as a guide in determining the best solution to meet your departmental or enterprise-wide communications requirements.

Designing Multivendor LANs That Work
#4669

Robert S. Yori
Hewlett-Packard Co.
P.O. Box 152030
Irving, Texas 75015

Designing Multivendor LANs That Work

Local area networks (LANs) are one of the most rapidly changing fields within the data communications arena today. Announcements of new products are a daily event, with players in the LAN market coming and going just as quickly.

How can these local area network technologies be molded into one network supporting multiple communication protocols and common software applications? What are the advantages in creating this kind of environment? And, what are some problems that can be encountered?

These issues are addressed in this article. Initially, the focus is on the technology available now that makes it possible to design LANs with hardware and software from various manufacturers. Next, tips for designing these multivendor networks are discussed. The last section outlines emerging developments that will make it possible to manage these multivendor networks.

Before discussing the details involved in designing such LANs, let's consider an example of how these LANs can be constructed. Refer to figure 1 for a picture of this example.

Here we have a Hewlett-Packard Co. OfficeShare PC network, composed of HP minicomputers and IBM-compatible PCs. Another department within the company has a Novell NetWare LAN, because of requirements for particular software packages. Now, the Novell users would like to run applications on the HP 3000 computers. Plus, the HP 3000 applications must transfer files to the Novell PCs.

In order for the Novell LAN users to communicate as remote terminals over the Novell NetWare LAN to the HP OfficeShare LAN, the following is required:

A. The Novell PCs in the network must run a terminal emulation software program such as HP AdvanceLink or TELNET. This would be in lieu of other Novell user applications. Note that the defacto industry standard TELNET does not support all terminal protocols, such as DEC VT-240 or HP block mode. Character-mode applications will permit the use of either TELNET or AdvanceLink; full-screen applications will require HP AdvanceLink.

B. Physically connect the two LANs with a PC gateway running a protocol-translation program. This is required because the NetWare and OfficeShare LANs use different network communication protocols. This gateway PC must also be configured with both Novell and HP LAN cards .

In order for the HP 3000 applications to transfer a file to one of the Novell PCs, use either of the software applications HP AdvanceLink or the defacto industry standard FTP (File Transfer Protocol). In both cases, the same file transfer application must be running on the Novell PC and the HP 3000. A PC gateway would still be necessary, since the communication protocols used by the two LANs are different. Figure 1 pictorially represents this scenario.

Several software packages were referenced in this example: TELNET, FTP, and HP AdvanceLINK. TELNET stands for "Teletypewriter Network" protocol, and is a defacto standard providing for remote terminal access between computer systems. FTP stands for "File Transfer Protocol", and is a defacto standard for transferring files between computer systems. Both TELNET and FTP are part of a set of software services called ARPA (sometimes referred to as DARPA). ARPA stands for "Advanced Research Project Association", and is an organization within the U.S. Department of Defense - hence the alternate acronym DARPA. AdvanceLink is a software program developed by Hewlett Packard Co. that supports file transfer and terminal emulation of HP block-mode and DEC VT-100 terminals.

TECHNOLOGY

The software technology making it possible to connect different LANs is the theoretical model developed by the International Standards Organization (ISO). ISO is an organization of the United Nations with a charter to develop and promote worldwide communication standards.

This model is called the Open System Interconnect (OSI) model. It is a software methodology providing for communication between computer systems of manufacturers implementing the model. Figure 2 shows this 7 layer OSI model. Within the definition of the 7-layer OSI model, any given layer communicates or passes data only to adjacent layers. For example, layer 2, the Data Link layer, passes and receives data only to and from layers 1 and 3.

Within the structure of the OSI model, the job of transferring information from one user's computer, across a communication network, to another computer is subdivided into 7 tasks or layers. At each layer, the OSI model defines the specific activities performed, plus the rules for communicating with adjacent layers. Each layer is a separate set of rules - in effect a separate protocol. These 7 layers make it possible for different computer systems, PCs, or desk-top work stations to communicate, regardless of the operating system used by each computer.

Given that one particular layer of the OSI model interacts with only adjacent layers, the theory behind the OSI model states that layers are "removable". This feature allows, for example, the substitution of software for coax LAN communication (layers 1 and 2) to be replaced by software for communication via telephone lines. This substitution does not affect layers 3 through 7. This allows communication protocols to be easily adapted for various vendor and communication environments.

Even though all layers are "removable" within the OSI model, certain layers work more closely with each other. These closely coupled layers are:

Layers 6-7

Layers 3-4

Layers 1-2

Note that layer 5 is a pivotal layer, connecting the application-interface layers 6 & 7, with the routing layers 3 & 4 (see figure 2). Examples of protocols used at layer 5 include NetBIOS - developed by IBM, and NetIPC - developed by Hewlett-Packard. In all instances, layer 6 must know which protocol is being used at layer 5. If an application is to run on communication software (protocol stacks) supporting both NetBIOS and NetIPC, then the application must be written in such a way that there are either two versions, or the application detects the layer 5 protocol and uses the appropriate interface.

Also, situations occur where LANs using different protocol stacks must communicate. For example, a PC on an Officeshare network may need to share files with a PC on an IBM token ring network. This may be necessary when users select software packages that run only on one particular PC LAN. In this case, a gateway can be used to translate all 7 layers of one OSI protocol stack into the 7 layers of a different OSI protocol stack. This gateway can be a PC running special software, or a hardware device built specifically to handle the translation tasks. In either case, the gateway must be configured with LAN interface cards for both networks. Also, the software running on the gateway must translate one LAN's protocol stack into another LAN's protocol stack. In effect, the gateway supports dual protocol stacks. Figure 1 pictorially represents the functioning of the gateway referred to in the first example. The HP-Novell gateway is an example of such a gateway between two PC LANs. DECs SNA gateway is another example, translating DECNet protocols to IBM SNA protocols. IBM's SNA-to-token ring gateway is another example.

Referring again to figure 1, from the perspective of the OSI model, we have substituted AdvanceLink, TELNET, or FTP for the Novell user application at layers 6 and 7. Also, the PC gateway translates layers 1 through 7 of the Novell LAN protocol stack to a corresponding 7 layer protocol stack for the HP OfficeShare PC LAN. This gateway function is necessary since NetWare and OfficeShare use different software protocols or procedures at the routing layers 3 and 4, even when layers 1 and 2 use the. same physical connection and protocols. Note that NetWare runs over several LAN hardware topologies, including Ethernet and Token Ring.

The feature of layer removability within the OSI model makes the construction and maintenance of a multivendor LAN running a common application possible!

One major advantage of designing a multivendor network based on the OSI model is that a common user software interface can be used on all the computers and PCs in the network. This use of a common application interface on all computers throughout the network will ensure that users can operate any of the workstations in the network. The network designer is free to choose the appropriate technology best suited for layers 1 through 5 in a particular segment of the network. The network can truly be designed for both the convenience of the user, and the flexibility and performance needed by the network designer.

Referring to the first example in this article, the ability to remove layers allowed the substitution of HP AdvanceLink for the Novell application software. Since HP AdvanceLink will communicate with NetBIOS or NetIPC protocols at layer 5, it can then be run on the\ Novell protocol stack using NetBIOS.

DESIGN SUGGESTIONS

Remember the following when designing a multivendor LAN that must support common applications.

When software and hardware components are supplied by various companies. it may be difficult to ensure the interoperability of the components. To avoid these problems, choose network components that conform to international standards, specifically the OSI communications model. This will be the easiest way to implement and support the network, since the ability to "mix and match" software components (remove layers) will be inherent in the products. But, one word of caution. Do not assume that all software and hardware advertised to conform to the OSI model actually does. Testing is still required. This is one of the charters of the Corporation of Open Systems (COS) - to test selected software for conformance to the OSI standards. Software not already tested for interoperability must be tested by the designer of the network.

When designing networks, one common misconception is that Ethernet and IEEE 802.3 are the same. Many times, the terms Ethernet and 802.3 are used interchangeably. In reality, IEEE 802.3 grew out of the Ethernet definitions. Both of these protocols define the same physical cabling schemes. But, the format of the data packets sent over the communication network differ. This difference is in the part of the data packet after the destination and source addresses. The end result is that systems talking Ethernet and 802.3 can use the same cabling scheme, but cannot talk to each other. Packet conversion is required for Ethernet and 802.3 systems to communicate. Two companies that make hardware and software products that perform this conversion are The Wollongong Group (Palo Alto, Calif.) and Cisco Systems (Menlo Park, Calif.). For a more detailed discussion concerning Ethernet and 802.3 differences, see the "Letters" section on age 58 of the September, 1988 issue of "Interact" magazine, an HP installed-base publication.

Also, when designing a LAN, assume that at some point the network will need to accommodate different PC communication protocols (layers 1 through 4), along with various application packages (layers 6 and 7). These abilities will be inherent within the network if protocols and applications that conform to the OSI standards are specified. As the designer, you will then be able to take advantage of the concept of layer removability to install applications throughout the network, using whatever communication protocol is appropriate within a department or work group.

Finally, the best way to control LAN performance is to isolate departmental data traffic from the main LAN backbone. This is accomplished using a device called a "router", or IP router. A router manages the flow of packets through the network by reading the level-3 IP address, hence the term IP router. If the address is for a device on the departmental LAN, the packet never gets sent onto the backbone. The router isolates the departmental traffic, thereby increasing the performance of the entire network. Note that a gateway can perform the same functions as a router, but the gateway reads all 7 layers of a protocol stack, where a router reads only the first 3 layers.

One additional note on gateways and routers. Since gateways and routers can perform the same tasks at layers 1 through 3, do not assume that a gateway should always be used instead of a router. Routers are usually hardware devices designed for a particular task, many times with the layer 1 through 3 software implemented in read-only memory (ROM). Gateways are usually PCs or minicomputers which run software that translates the different 7-layer protocol stacks. Therefore, gateways will tend to be more expensive than routers. Plus, gateways are slower, since their software applications are not ROM-based, and they must translate 7 layers of protocol.

NETWORK MANAGEMENT

Network management is one aspect of network design that is overlooked more often than not during the network planning phase. When designing a multivendor network, make this an issue of paramount importance - designed into the network before any vendor is selected or equipment is purchased.

Network management implies different capabilities to different designers. In its basic form, network management provides for the detection of component failures in the network. The emerging concepts of network management also allow management of the PCs and other computer systems on the network, including performance.

The emerging OSI standards define software procedures for managing networks from the perspectives of determining sources of failure, tracking network and system performance, device inventory management, and security. On the TCP/IP platform (layers 3 and 4), one standard is Simple Network Management Protocol (SNMP). For the corresponding layers of the OSI platform, the network management scheme emerging is Common Management Interface Protocol (CMIP). One of the advantages of CMIP over SNMP is that CMIP allows for greater control of network security issues.

As the OSI model becomes fully defined, TCP/IP protocol stacks will be migrated to OSI protocol stacks. The emerging standards also call for the migration of SNMP to CMIP.

When designing a network on the TCP/IP platform, specify that the network management software not only use SNMP now, but that the vendor will migrate the software to the CMIP standards.

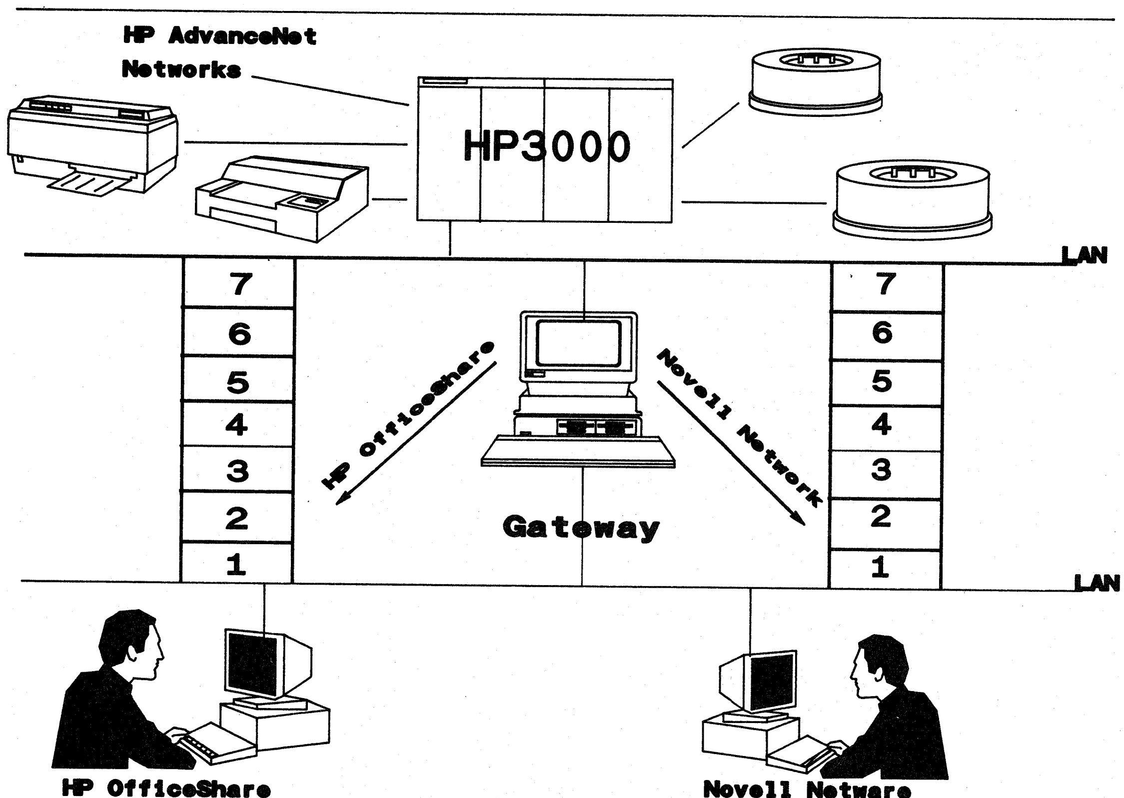
CONCLUSIONS

We have discussed techniques that can be used to create an environment where multi-vendor LANs not only communicate, but also run common applications. These techniques use gateways to translate different network protocols, plus application software that conforms to the OSI standards.

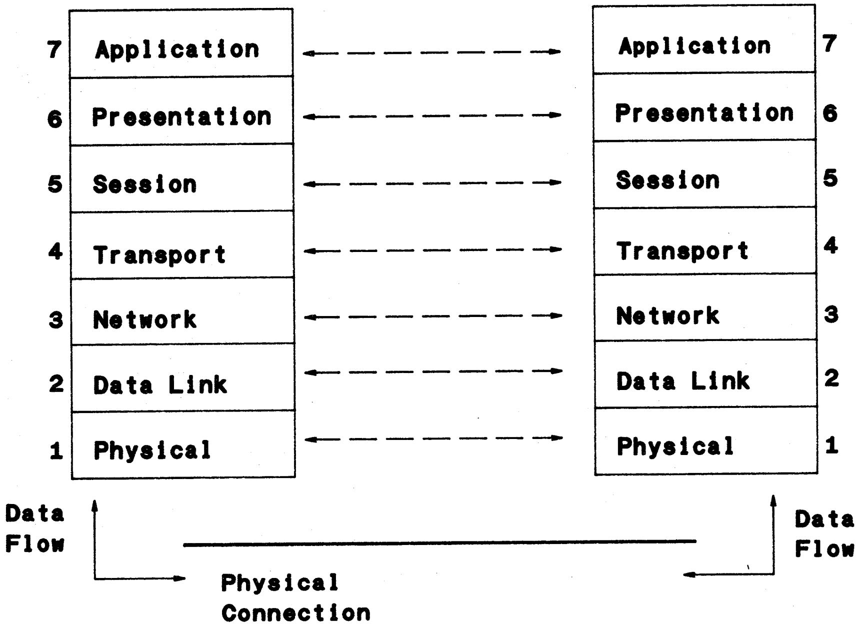
By using LAN hardware and software components based on OSI, vendor interoperability and support issues can more easily be resolved. Also, software applications can be purchased from various suppliers. Purchase criteria should depend on whether the application can ommunicate with the layer 5 protocol being used in the network. This gives users not only a larger choice of software products, but provides the network designer and corporation economic flexibility. Companies now have the ability to purchase applications for the network, and network hardware components, from vendors other than the original supplier.

Another advantage of specifying conformance to OSI standards is that, in the near future, network management will inherently be designed into the network, using the emerging CMIP standards. This will allow centralized management of a multivendor network. From one central console, the network manager can determine the operational state of all components. Also, network performance can be centrally monitored. These are future capabilities. As the CMIP standards evolve in the next several years, this will become reality.

Adherence to international standards in the network will result in one successfully running for many years. Economies of scale are realized for both the designer and the corporation. And, as the international standards evolve, and enhancements are encorporated into OSI, the network can also evolve. Adherence to international standards ensures that the network can be maintained, supported, and enhanced.


HP AdvanceNet
Networks
HP3000
LAN
7
6
5
4
3
2
1
HP OfficeShare
Novell Network
Gateway
7
6
5
4
3
2
1
LAN
HP OfficeShare
Novell Netware

OSI 7-LAYER MODEL



Planning Your Communication Infrastructure

Karen Dudley
Hewlett-Packard Company
Roseville Networks Division
8000 Foothills Boulevard
Roseville, California 95678

Summary

In the past there has been much controversy surrounding the selection of a particular data networking technology. Today, the importance of networking is better understood and users are planning networks that will more effectively support their data communication needs. A new area of concern is how to extend the networking foundation to other buildings in a campus environment or to geographically remote sites. The role of Local Area Networks (LANs) has expanded to provide more than just peripheral and file sharing for local workgroups. LANs have become transport platforms to other corporate resources.

As networks grow in size and complexity, proper planning is essential. The key planning aspects include network structure, selection of networking media and communication products, and integrated management. The use of a common backbone and communication servers will provide a seamless foundation for enterprise-wide communication. This paper covers key issues and technologies you should consider when planning a communication infrastructure for a single site or multiple dispersed sites.

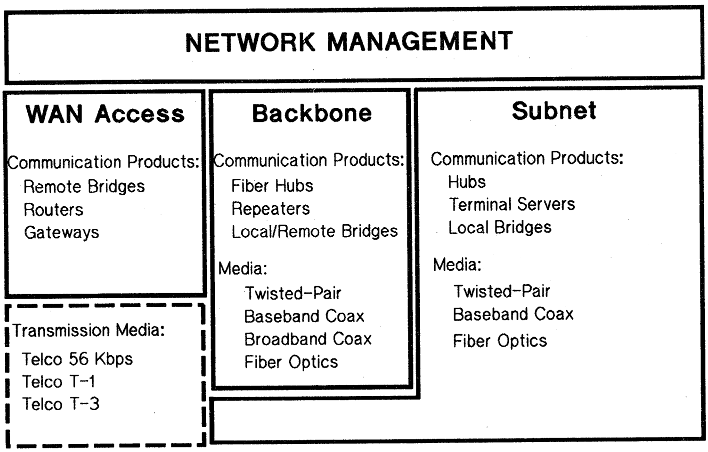
The Communication Infrastructure

The communication infrastructure includes three networking environments that require different planning considerations.

- Subnet or workgroup networking includes PC and terminal connection, data center LANs or manufacturing floor LANs.
- The campus or facility backbone network is the information pipeline tying together workgroups on multiple floors or in multiple buildings.
- Wide area network (WAN) access is the interconnection of LANs using WAN transmission media.

Network management is the common element providing integrated management of communication devices in each networking environment. The key components in the communication infrastructure are the wiring system and the communication products that connect, distribute, bridge or route network data. The media alternatives and communication products are shown below in each environment.

The Communication Infrastructure



The objective of a communication infrastructure is to provide users with quick, transparent and secure access to any computing resource within the entire organization regardless of where that resource is located. This global connectivity allows: separation of physical and logical workgroups, productivity enhancements through file sharing and distribution of E-mail, and the ability to build distributed, transaction-based databases.

User Requirements for a Communication Infrastructure

Transparent Connectivity	– LAN to LAN locally and remotely – Interconnection of equipment from multiple vendors
Network Management	– Integrated and unified management system – Fault isolation, performance, configuration and accounting – Security to prevent entry or tampering
Guaranteed Network Performance	– Available capacity from local and wide area transmission media to minimize user response time – Box reliability and link redundancy
Flexibility	– Ability to grow the network to support more users and traffic – Effective migration strategy to future networking requirements
Cost Effectiveness	– Initial purchase, administration, support – Protect investment in current LAN and WAN technology with modular growth
Ease of Use	– Documentation, installation, operation and support

There are a number of market and technology trends affecting each of the networking environments. In the subnet environment, users now recognize the importance of building wiring and are investing in structured wiring systems (e.g. AT&T's Premises Distribution System or the IBM Cabling System). Twisted-pair wiring is clearly the winner for horizontal wiring in an office environment. The increasing use of workstations instead of terminals is driving the need for higher speed networks and more comprehensive connectivity.

The increasing move towards company-wide networking is influencing both the backbone and WAN access environments. As more users need to communicate, higher speed networks are required within a building or campus and across wide area links. This expanded connectivity brings a greater need for security and asset management. Trends specific to WAN Access include the use of common transmission media for voice and data integration, emergence of LAN technology as a wide area networking alternative and increasing availability of reliable, low-cost digital bandwidth.

Planning Considerations

Planning your communication foundation begins with understanding your business activity and your physical environment. Your business activity determines how and where information moves throughout your facility. Specific applications or tasks that are performed in a given area establish the information needs. Information needs help determine the network technology which can most economically deliver that information. Your business goals and objectives help establish future information requirements.

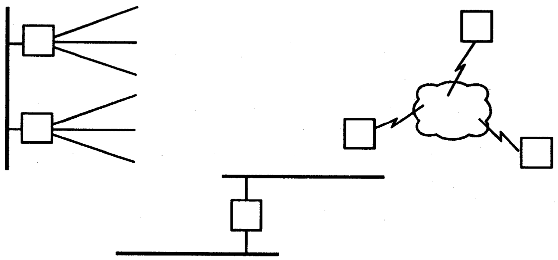
Your physical environment will have a significant influence on the choice of appropriate networking technology. Certain environments can be expected to already have an existing "network" for common communications systems like the telephone. Office environments tend to be clean and quiet, and have phones on every desk. Computer centers are custom environments designed for computers at the outset. Factory production areas are electrically noisy, often physically dirty and can be quite large, even spreading through several buildings. Unlike the office, production areas have few phones.

The second step is to understand the alternative media that are available to you. The media (or transmission media) is the physical wiring over which voice, data and video signals are transmitted. Various types of transmission media are available, each having its own information-carrying capacity (bandwidth) and suitable applications. Implementation of a structured wiring system allows you to select the most cost-effective and best suited media for each subsystem. The key evaluation criteria are price/performance tradeoffs, flexibility, expandability, administration and environmental requirements.

The third step is the selection of communication products to connect, distribute, bridge or route network data.

Communication Products

to Connect, Distribute, Bridge or Route Network Data

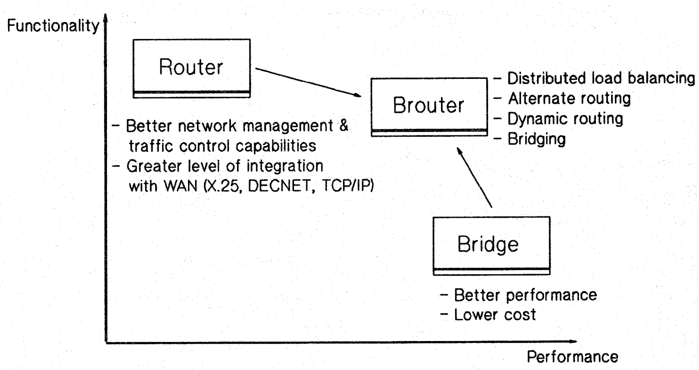


Communication Products

Hubs	– Provide a central connection point for LAN attached workstations/systems – Provide fault isolation within a subnet – Allow star topology – Easy moves, adds, and changes
Terminal Servers	– Provide connection for terminals or other RS-232/422 devices to a LAN backbone – Eliminate dedicated wiring between terminal and computer system – Allow connection to multiple hosts
Repeaters	– Connect two similar LANs – Enhance data signal – Extend distance
Bridges	– Provide transparent interconnection of LANs at the media access control protocol level (protocol independent) – Data packet filtering – Extend distance and number of nodes – Local and remote
Routers	– Connect LANs with protocols in common at the network layer and above (protocol dependent) – Allow connection of multiple LANs with multiple paths – Provide redundancy – Determine least cost or shortest path – Generally used with remote links
Gateways	– Provide a communication path between two LANs using different LAN types and protocols – Protocol conversion up to layer 7

The differences between bridge and router products are beginning to blur as vendors begin to offer "brouters". Routers provide better network management and traffic control capabilities. Routers allow a greater level of integration with wide area networks (X.25, DECNet, TCP/IP), but bridges tend to offer lower cost, higher performance and protocol independence.

Bridges and Routers



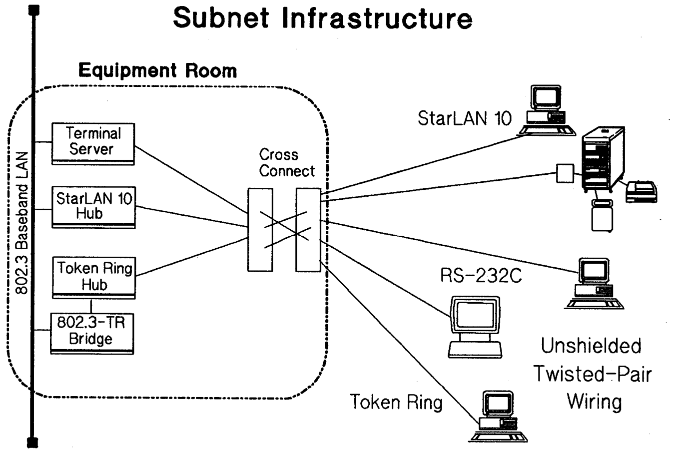
Subnet Planning

Subnet or workgroup networking includes PC/workstation LANs, terminal clusters, data center LANs or production floor LANs. The use of subnets in network design is critical to the development of an effective, manageable communication infrastructure. As LANs grow, performance, reliability, security and cost-effectiveness may be severely compromised. Networks segmented into manageable subnets are more reliable and easier to maintain.

Benefits of Subnetworking:

- Improved response time by limiting traffic congestion on the facility backbone
- Network performance optimization by segmenting traffic types and using separate backbones (e.g. for terminal-to-host and host-to-host traffic)
- Use of the most cost efficient media in a particular subnet
- Easier fault isolation: network is more reliable and easier to maintain
- Partitioning of LAN traffic onto physically separate media to improve security

The diagram below illustrates several key elements in the subnet infrastructure. The media is unshielded twisted-pair which supports voice and most data requirements (StarLAN 10, Token Ring, RS-232, ...) found in a typical office environment. Hubs, terminal servers and local bridges are located in an equipment room that provides a concentration point for subnet wiring. Hubs and terminal servers provide a central connection point for LAN attached workstations/systems, terminals or other RS-232/422 devices and connection to the facility backbone. Hubs and terminal servers allow implementation of a star wiring topology and the associated benefits of fault isolation to a single node, and easy moves, adds, and changes. Terminal servers eliminate dedicated wiring between terminals and computer systems and allow connection to multiple hosts. Local bridges allow connection of different media (twisted-pair to baseband coax) or different LANs (802.3 to 802.5) and provide data packet filtering.



When selecting your network media, include the following considerations. What are your physical connectivity requirements (which devices) and your bandwidth requirements (number of users, type of applications)? What is the physical environment like: office versus production floor?; open versus closed offices? Are there any existing media?

When selecting communication products, there are additional considerations. Which workstations, systems and/or terminals (vendors) need to be connected? Is most communication within the workgroup or out of the workgroup? How often do users move or change their communication requirements? What is the concentration of users and what are the distances to the equipment room? How will your requirements be changing in the future?

Subnet Planning Considerations

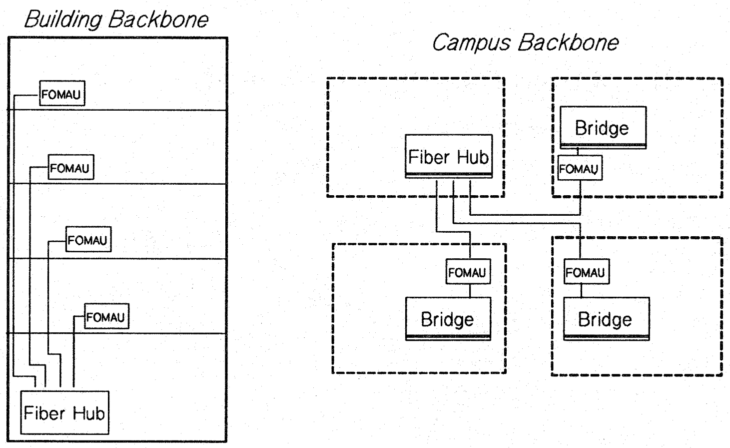
Business and Applications:	**Physical Environment:**
- Connectivity requirements	- Office versus production floor
- Bandwidth requirements	- Closed versus open offices
- Information flow	- Existing media
- Logical versus physical workgroups	- Density/concentration of users
- Easy moves, adds, and changes	- Location/availability of equipment rooms

Backbone Planning

The campus or facility backbone is the information pipeline tying together subnets on multiple floors or in multiple buildings. The backbone should be considered a networking utility for site communication requirements. A backbone facility may include several networking media to meet all site communication requirements.

Today most building backbones use a combination of coax (baseband or broadband) and twisted-pair cable. The use of fiber-optic cable is increasing in new installations. Campus backbones are primarily broadband coax or fiber optics due to distance and environmental considerations. When planning your backbone facility the most important consideration is understanding your capacity requirements. Installing a campus backbone is very expensive and it is more cost-effective to to pull extra cable or to install fiber even if it is not required today. Some backbone alternatives to consider are fiber optic 802.3/Ethernet, broadband, and FDDI.

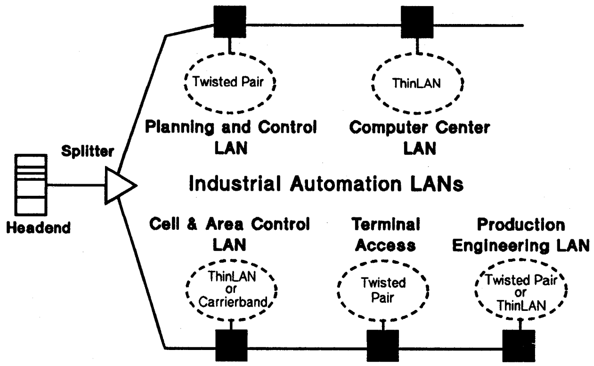
Fiber Optic Backbones (10Mbps 802.3/Ethernet)



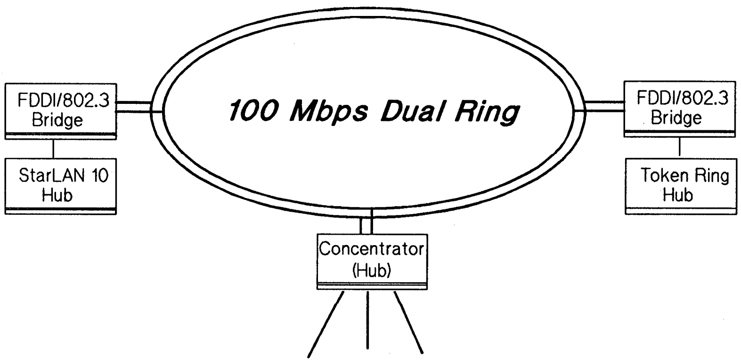
Fiber optic 802.3/Ethernet systems are now available from many vendors worldwide. Declining costs, simpler design and installation techniques, and standards activity are contributing to the growing acceptance of fiber optic 802.3/Ethernet. This alternative is very cost-effective today and allows migration to higher speed LANs for future requirements.

Broadband provides a very robust backbone solution for the facility or campus backbone. Broadband offers a number of advantages: very flexible topology; greatest distance; multi-channel (data, voice, video), supports 802.3 LANs, 802.4 LANs, RS-232 and T1.

Broadband Backbone



Fiber Distributed Data Interface (FDDI)



Fiber Distributed Data Interface (FDDI) is a counter-rotating token ring LAN with a data rate of 100 Mbits per second. FDDI will support 500 dual attached stations linked by 100 kilometers of duplex cable. A single station can support either a host computer or a subnetwork of hundreds of users. FDDI provides a high-performance backbone alternative to link lower speed baseband LANs (802.3, 802.4, 802.5) supporting a greater number of users and larger geographical distances.

Backbone Planning Considerations

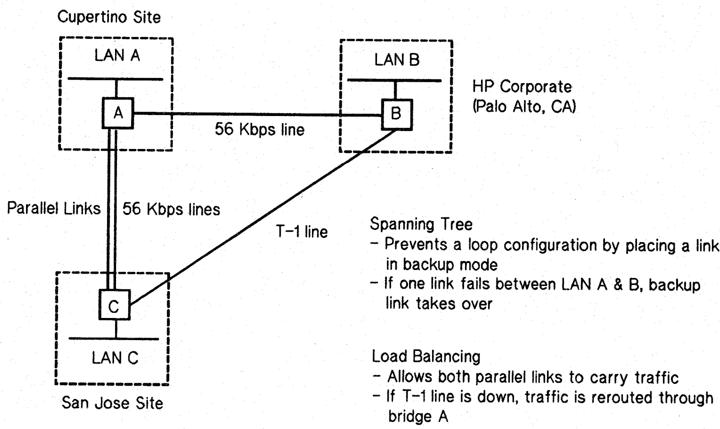
Business and Applications:	**Physical Environment:**
- Bandwidth requirements	- Distance requirements
- Connectivity requirements	- Environmental conditions
- Information flow between multiple floors/buildings	- Existing media
- Logical versus physical workgroups	- Right of way

WAN Access Planning

WAN access is the interconnection of LANs using WAN transmission media. The need for LAN-to-WAN interconnection is a result of the increasing adoption and installation of LAN-based systems. Companies are replacing centralized, low-speed terminal-to-host systems with distributed, high-performance LAN-based systems. Also increasing is the need for individuals to communicate with others located in different geographic areas. LAN managers must implement networks that interconnect multiple dispersed LANs through high-speed wide area networks.

WAN access planning considerations require a detailed understanding of LAN attributes. This includes what the LAN traffic looks like, how much of the LAN traffic will be forwarded over the WAN transmission media, and at what rate should that traffic be forwarded. Choice of communication products for WAN access will depend on the size and complexity of your network. Remote bridges are beginning to play a larger role for linking geographically separate LANs. Bridges have always excelled at transparency, but have not provided the reliability that can be achieved with the use of routers. Developments in spanning tree and load balancing technology allow remote bridges to provide more robust internetworks.

WAN Access with Remote Bridges

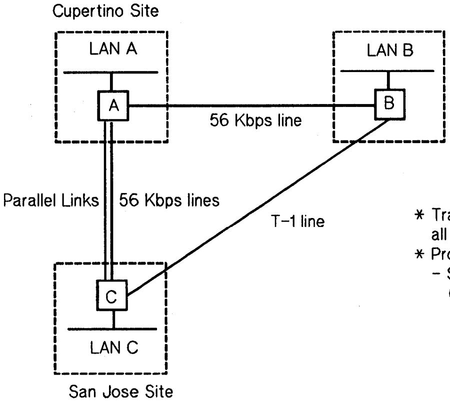


Routers still provide the preferred solution for interconnection of more complex networks. Routers have the ability to use WAN links efficiently and to build large networks composed of multiple subnets.

WAN Access Planning Considerations

Business and Applications:	Physical Environment:
- Bandwidth requirements	- Existing wiring structure (LAN-based versus direct connect)
- Connectivity requirements	- Access to digital lines application?
- How distributed is your application?	- Network size and complexity
- Frequency of traffic	

WAN Access with Brouters



* Traffic that can be routed can use all paths (select least cost/fastest path)
* Protocols not supported are Bridged
  - Same features as Remote Bridges (Spanning Tree, Load Balancing)

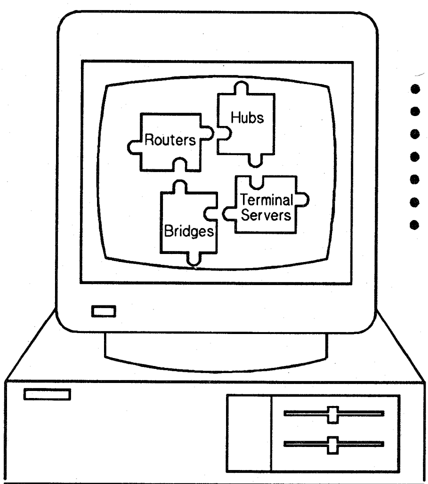
Growing Importance of Network Management

Network management is the ability—through hardware and software systems—to identify, monitor and control network reliability, configuration, security, accounting and performance. Network management is growing in importance as companies become more dependent on their information networks for everyday business. There is tremendous pressure to keep networks up and running, while controlling administration and support costs. As LANs grow and span multiple buildings and geographical boundaries there is an even greater need to create a reliable, manageable, and secure network foundation.

Network Management Planning Considerations

- Importance of integration of multivendor devices into one management scheme
- What level of management is required for each device?
- Is there a requirement for remote device management?
- User sophistication—usability requirements
- Compatibility with existing tools or management systems

Integrated Network Management



- Fault Mangement
- Configuration Management
- Inventory Management
- User Accounting
- Security Management
- Performance Management
- System Management

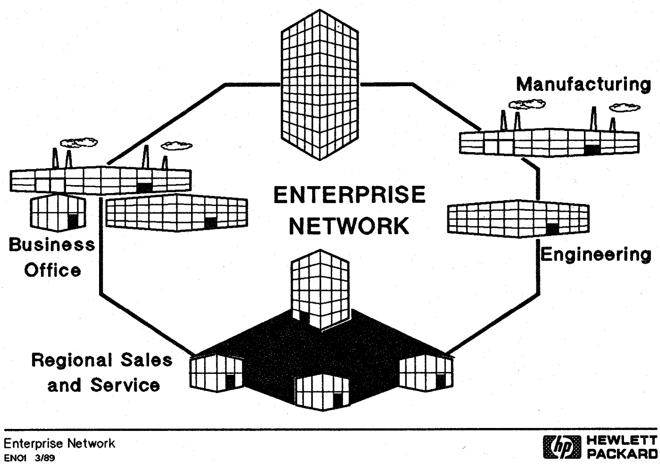
ENTERPRISE NETWORKS
ENHANCING GLOBAL COMPETITIVENESS

Robert Emerson

Hewlett-Packard Company
19490 Homestead Road, MS 41T
Cupertino, CA 95014

INTRODUCTION

Enterprise-wide networks are being utilized by businesses as a key competitive weapon in their day-to-day operations. Corporations are discovering new ways to use communications and computing technologies to improve their profitability - both by reducing expenses and by increasing revenues. This paper explores the trends in enterprise networking, discusses the major components of an enterprise-wide network solution, and reviews the decision criteria being used in planning and implementing such networks.



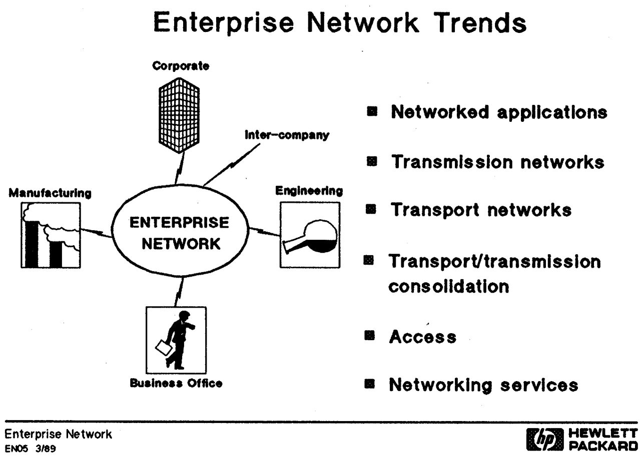
Enterprise-wide networks are extremely broad in scope - both geographic and functional. Such networks usually span an entire country and often are global in their reach. Multiple functional units within the corporation are supported - engineering, finance, sales, etc. - and many different applications are supported. In addition, the network supports communications with other companies - usually suppliers and customers.

The perspective used by customers when planning enterprise-wide networks is end-to-end. Not only is the wide-area transport considered, but also the LANs, systems, and applications which are being networked.

The motivation for deploying enterprise-wide networks is the enhancement of the corporation's profitability by implementing networked applications which are critical components of the company's day-to-day business operations.

KEY NETWORKING TRENDS

MIS Directors and Telecommunications Managers are increasingly able to express their networking requirements in terms specifically related to the business of their companies. A few years ago it was common to hear networking requirements expressed simply in terms of the number of terminals to be connected or the number of sites to be connected. The focus of such requirements was connectivity. While basic connectivity is still a critical requirement, it is usually only a partial solution to a much broader problem. Today, we more often hear a customer expressing his network requirements in terms like "We are deploying a corporate-wide electronic mail system," or "We need to transfer design information between different engineering groups," or "We are implementing an automated stock trading system." This kind of thinking represents a significant step forward in the sophistication with which we plan the use of a network in our businesses.



Enterprise-wide networks are increasingly being motivated by the deployment of specific networked applications which are contributing to the profitability of the corporation. We are seeing a shift in the justification for such networks from saving expense dollars to generating revenue dollars.

Historically, network costs represented an expense to be controlled and reduced if possible. Purchase decisions were motivated largely by the reductions in communications expenses which would result from the deployment of a particular piece of networking equipment. As distributed computing technology has advanced, however, corporations are finding new ways to use networked applications to enhance their competitive position and to GENERATE REVENUE. This shift from an expense-only to an expense-and-revenue perspective has caused a simultaneous shift from a focus on the cost effectiveness and performance of the transport network alone to a more complete perspective on the end-to-end performance of the applications being networked. The transport network is now being evaluated TOGETHER with the systems and applications which it supports.

A second major trend in enterprise-wide networks is rapid ramp-up in the speeds at which these networks operate. During the 1980's, we saw the predominant line speeds for enterprise-wide networks move from 9.6 Kbps to 56 Kbps to 1.544 Mbps. This move was driven by reduced costs for high-speed transmission facilities, availability of transmission resource managers (e.g., T-1 multiplexers) which allowed consolidation of voice and data traffic on these facilities, and the emergence of applications (e.g., video conferencing) which required higher bandwidths. We are currently at the stage where technology is almost getting ahead of the needs of the users. T-3 technology, operating at 44.736 Mbps, is becoming available in many parts of the U.S. and has been deployed by approximately 100 large corporations in their networks. Most corporations, however, do not have the traffic demands required to justify the use of such high-speed facilities. The widespread availability of T-3 facilities and the glut of bandwidth caused by the enormous amounts of fiber optics deployed over the past few years will continue to drive down the cost of these facilities and stimulate the deployment of new image and video applications. It is these new applications which will drive the deployment of T-3, SONET, and broadband ISDN in the 1990's.

The third major trend that we see in enterprise networking is toward increasing network complexity. As networks extend their geographis coverage, we see more vendors, more carriers, and more applications being supported. In many ways, this is positive development, giving the customer many more choices in how he implements his network - the real advantage of a competitive marketplace. On the other hand, the real price of network complexity is the difficulty faced in managing these networks.

In summary, the key trends in enterprise networking are the move toward application motivation, rather than simple transport efficiency, a several-fold increase in the amount of information being transported (and the associated speed of the transmission facilities) and the increasing complexity of the management of these networks.

THE NETWORK AS A BUSINESS TOOL: NETWORKED APPLICATIONS

Before going into a more detailed discussion of the major components in an enterprise-wide network solution, we will elaborate on the strategic uses to which corporations are placing their networks. In a global economy, the movement of information is critical to establishing competitive supremacy. To explore this concept further, we will focus on applications networking in two different industries: Manufacturing & Financial Services.

Manufacturing

Growth and global competition are forcing increased geographic dispersion of the manufacturing enterprise. Competitive pressures require the reduction in time-to-market for new products and the scarcity and cost of R&D resources are stimulating the deployment of automated tools for both hardware and software design. Just-in-time inventory management techniques are reducing the cost to manufacture products. Each of these developments require the efficient and timely transport of ever-increasing volumes of information, within the enterprise and with other companies. Specific uses of applications networking in support of a manufacturing enterprise include the following:

- transfer of chip designs, software code, etc between engineering teams working on different portions of an overall product design. The end systems being networked are usually engineering workstations and the files being transferred are quite large.

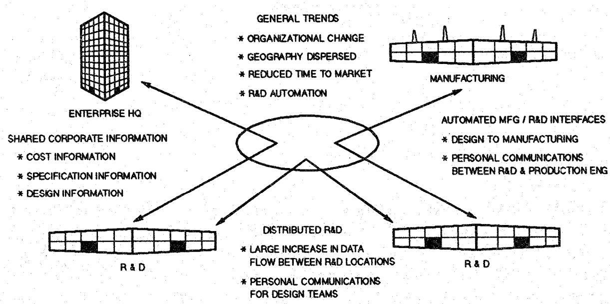
- transfer of completed chip designs, software code, etc from the controlling engineering group to various manufacturing sites. The volume of information being transferred is the same as in the preceding example and the integrity of the information transfer is absolutely critical, since manufacturing processes are being driven directly by this information.

- communication of forecasts, work orders, and shipping information between manufacturing sites and subcontractors. The ability to link the enterprise network to other private or public networks is a critical requirement for implementing such EDI applications.

- communication of production rates/costs and inventory levels from manufacturing sites to headquarters, in order to have real-time visibility on the efficiency of the manufacturing process and the overall profitability

The movement of electronically generated and stored information in such applications provides the manufacturing enterprise with the ability to more quickly design and introduce new products (increasing its competitive position and revenue stream) and operate more efficiently (descreasing its operating expenses). The bottom line result is an improvement in the bottom line.

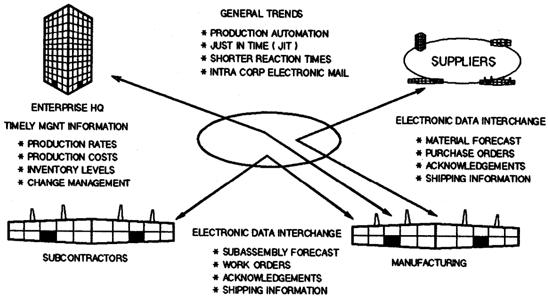
Manufacturing Industry
Major Trends In R&D



Enterprise Network
EN07 3/89

HEWLETT PACKARD

Manufacturing Industry
Major Trends In Manufacturing



Enterprise Network
EN08 3/89

HEWLETT PACKARD

Financial Services

In the financial services industry, the enterprise-wide network can be thought of as the "distribution channel" for financial products. The deregulation of the banking industry in the U.S., strengthening of financial trading centers around the world (e.g., Tokyo, Singapore), and widespread deployment of computing technology are all stimulating the use of application networking in the financial services industry. Specific uses of applications networking in support of a financial services enterprise include the following:

- networked Automated Teller Machines (ATMs) extend the reach of the enterprise in its ability to deliver services to its customers.

- transfer of buy/sell orders for securities directly to various stock exchanges around the world.

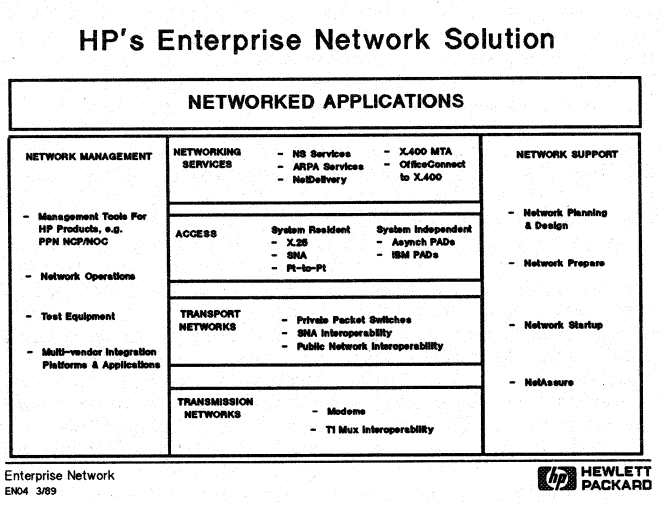
- electronic funds transfers to other institutions.

- access to financial services from PCs located in a customer's home or business establishment.

The use of such networked applications provides the ability to more easily reach a larger number of customers (enhancing competitive position and revenue) and to reduce the coper revenue dollar of delivering these services (reducing operating expense). Enterprise-wide networking is a natural means for delivering "soft" products like financial services.

ENTERPRISE NETWORK: THE INFRASTRUCTURE FOR NETWORKED APPLICATIONS

The implementation of the types of application networking described in the preceding section requires the corporation to think of the enterprise-wide network as a critical part of its infrastructure - similar to the manner in which its factories, office buildings, and people are considered. We view the infrastructure for networked applications being comprised of six principal modules. The following paragraphs describe the general content of each module along with some specific examples of Hewlett-Packard's approach to providing the products and services required to implement this infrastructure.



The foundation for the enterprise-wide infrastructure is the TRANSMISSION network. The transmission network is comprised of the actual transmission lines - such as 56 Kbps DDS, T1, and T3 - together with the devices and systems used to access and manage these facilities. A key component in many transmission networks is the Transmission Resource Manager, commonly referred to as the T-1 multiplexer. The TRM provides the consolidation point for multiple transport networks as well as the consolidation point for voice and data networks. TRMs ensure the efficient utilization of high-speed transmission facilities and the ability to maintain physical paths through the network in the event of a transmission facility failure. Major trends in transmission networks include the migration from T1 to T3 as the dominant line speed for very large networks and the extension of the T1 locations to much smaller locations in the enterprise via fractional T1 services and products. TRMs are already appearing with the higher and lower capacities to deal with this emerging environment.

Hewlett-Packard, as a provider of networked computer systems, has chosen to add value to the product offerings of the major TRM vendors by providing unique services which are complementary to those of the TRM vendors. HP's value-added includes the performance of end-to-end testing of the operation of the TRM products in a networked applications environment, providing customers guidance on how to configure the TRM together with the transport network, LANs, and end systems involved in the application. In addition, HP provides a service known as NetAssure for our T1 partners, in which we provide a single point-of-contact for coordination of the service for networks in which HP products are deployed with selected TRM products.

The second major module in the infrastructure for networked applications is the **TRANSPORT** network. In reality, we should probably say transport networks, because most companies have multiple voice and data transport networks which are physically consolidated onto a single transmission network. Each transport network provides reliable routing and end-to-end transport for a particular set of networked applications. X.25 transport is the most widely deployed multivendor transport in the world today, supporting a broad range of interactive applications, including electronic mail, real-time POS, and automated order input/tracking. SNA transport is used for providing remote users and systems with access to applications requiring large IBM mainframes. Finally, TCP/IP based transport has emerged in recent years as a de facto standard for applications requiring LAN-like speeds across the transmission network. Today, products in each transport area are distinct and are provided by different vendors. One of the key trends that we see is the blurring of the boundaries between the different types of transport and the emergence of new, multi- protocol communications servers which provide the flexibility to support multiple transports simultaneously.

HP's current offering is the HP Private Packet Network (HP PPN) product line, providing a powerful, industry-standard transport which can support a wide range of systems and applications. Although other computer systems venors rely on industry standards to network their systems over anyone's X.25 transport products (and HP can certainly support this approach), HP has gone a step further by also offering its own transport products. This gives our customer the option of using any industry standard X.25 network or reducing the number of vendors by giving a more complete end-to-end solution.

ACCESS to the enterprise-wide network is the next major module in the infrastructure for networked applications. There are two principal categories of access - systems resident and systems independent. Systems resident access is native networking software in the end system, and systems independent access refers to the use of separate products such as PADs and LAN-based servers. Major trends in the access arena include the emergence of low-cost, but multi-protocol PADs as an alternative to today's protocol-specific PADs. In addition, we are seeing the introduction of very high-performance LAN-based servers to satisfy some of the high-bandwidth applications such as transfers of large files or images.

Most computer systems vendors, including HP, now provide native X.25 software on their systems. You'll note that we provide X.25 support on both the HP3000 and HP9000 product lines along with X.25 access from our Vectra PCs. In addition, HP offers a family of X.25 PADs supporting asynchronous, SDLC, and bisynch devices.

NETWORKING SERVICES provide the systems-resident, high-level networking protocols for file transfers, remote database access, file sharing, and message transport - which are required for full interoperability across multiple systems. In the commercial and scientific application environments, de facto standards based around the ARPA services are predominant today. In the manufacturing environment, OSI services (MAP 3.0, FTAM, MMS, etc.) have already been defined and products implementing these services are being introduced. The key trends that we see over the next few years include the rapid DEPLOYMENT of the MAP 3.0 services and the emergence of the OSI services for the commercial and scientific environments. A key challenge during this period will be the migration from the de facto standards in use today to the OSI standards.

HP provides a broad range of support on its systems for all of the major de facto and OSI standards. In addition to our proprietary NS services (which are optimized for HP-to-HP communications), we support ARPA services, MAP 3.0 services, and are evolving to support the other OSI services.

NETWORK MANAGEMENT represents the tools and services to control and operate the enterprise-wide network on a day-to-day basis. Included are network management tools specifically tailored for controlling specific network elements, integration systems for consolidating the network management information for multiple categories of network elements, and support services from third parties. Key trends in this area include the establishment of standards for network management systems and specific product platforms for the integration of network management information. In addition, several firms are now offering facility management services to complement the internal staff of the customer.

HP's OpenView network management strategy embraces a range of product offerings and is based on strong support for network management standards. An advantage of having a consistent end-to-end network management strategy such as OpenView is that the customer will have a consistent user interface and display environment for managing the different modules in their networking infrastructure, including the management of the end systems on which the applications reside. In addition, HP offers worldwide facility management services, allowing customers to get the economic advantage of a private network without having to staff up a large organization to run the network on a day-to-day basis.

Finally, **NETWORK PLANNING & DESIGN** represents the services provided to plan, size, and plan the implementation or expansion of the network. These services are provided by either the dominant vendor, a third-party consulting firm, or the customer's internal communications staff. Trends in this area center on increasing sophistication of the network modeling design tools to result in more cost effective and robust networks.

HP offers a complete range of these services, provided out of three Customer Network Centers in Atlanta, Georgia; Bristol, England; and Singapore, at which we concentrate our network consulting expertise.

The infrastructure for networked applications is much more than just computers or packet switches. It is a coordinated, interconnected set of modules - **TRANSMISSION, TRANSPORT, ACCESS, NETWORKING SERVICES, NETWORK MANAGEMENT, AND NETWORK PLANNING & DESIGN** - which can be considered individually but must be planned implemented and managed in a highly coordinated fashion.

BUILDING THE ENTERPRISE-WIDE NETWORK: KEY DECISION CRITERIA

In the preceding section, we made the case for viewing and planning the enterprise-wide network as an end-to-end, unified entity. Yet, the realities of the marketplace are such that no single vendor builds and sells everything from transmission to transport to services. The key to successfully selecting vendors and building an enterprise-wide network is striking a balance between uniformity and diversity.

Diversity is inevitable because of the geographic scope of the enterprise-wide network and the fact that different vendors predominate in the different modules of the network. Diversity is good because this allows the customer to take advantage of the competitive marketplace and select from the broadest range of products and services from vendors and carriers. Uniformity should be maximized in network management, network planning/design, and overall service and support, to ensure a holistic structure for the network and to ensure optimized end-to-end performance on an ongoing basis.

The key criteria for building the enterprise-wide network are as follows:

- The network planning should be driven by the applications being networked. It is only through a thorough understanding of these applications that the traffic demands and performance requirements can be accurately specified.

- Vendors should have an understanding of applications networking on an end-to-end basis, even if they do not provide a complete solution. If a vendor does not understand how his equipment contributes to the end-to-end performance of your application, he will not be able to provide the necessary level of support to your network.

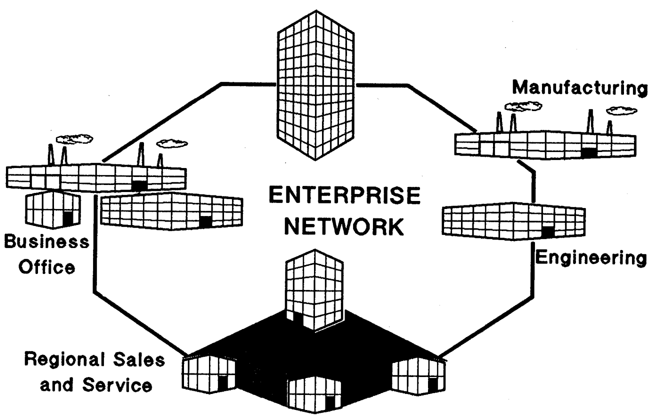
- Multiple network management systems will be required to provide specific tools for managing specific network elements. A consistent user interface and display environment across these systems can reduce training costs and provide maximum flexibility in assigning the management of different network elements to different organizations.

- Service and support should be structured to reduce the number of vendors with which the customer must deal. This can be accomplished by having some vendors coordinate the support activities of others. This reduces the complexity of the support process and places responsibility on the vendors for isolating problems between their respective systems.

Building an enterprise-wide network is a never-ending process. With the continuing deployment of new networked applications, the network must be continuously evolved. The basic principles described above for the initial construction of the network also provide solid guidelines for its ongoing evolution.

SUMMARY

Enterprise-wide networking is driven by much more than connectivity. The objective of enterprise-wide networking is to provide an infrastructure for networked applications - applications which are increasingly critical to the day-to-day operations of the business. It is only by considering the entire infrastructure - transmission, transport, access, networking services, network management, and network planning/design - that network planners and vendors can deliver quality solutions to the networking challenges of the enterprise.



Enterprise Network
EN01 3/89
HEWLETT PACKARD

ADVOCATES OR ANTAGONISTS

A Strategy for Disaster Preparedness

JAMES A DEPP
UP TIME DISASTER RECOVERY, INC.
643 WEST STADIUM LANE
SACRAMENTO, CA 95834
916-648-1282

The development of a disaster recovery strategy is a process of education and commitment. To be successful, our clients confirm the process takes time, and involves a large number of people. The determination of need is critical to assessing the value of recovery, the justified costs of recovery service, and to developing commitment to the process. Several of your associates will have information which should be part of this determination of need, and the suggested investigations which follow should reduce the time for them to assist you. The assistance they give you at the outset also prepares several of them to assist you later, either as the project is approved, or during the planning stages.

A word of encouragement and of caution is appropriate. The results are worth the effort - for some companies the payoff has been in dollars saved/productivity increased, as well as planning completed. For others, the payoff has been the survival of the company. Begin your efforts with top management because it is their interest in or willingness to examine disaster recovery preparedness which will enable you to achieve responsiveness from the many other people who can make this effort effective.

Topics for investigation are grouped by professional who might be knowledgeable or who might most easily investigate the area. The specific professional may well be involved in implementation of the recovery strategy. Specific areas of involvement are suggested.

Development of recovery strategies will require information accumulation, assessment, and strategy approval. Information accumulation includes the impact of disaster, the methods of continuing operation, the costs. Assessment will be a selection process including the weighing of cost versus benefit. The approval includes not only administrative approval and funding, but also the legal and purchasing activities to establish contracts for the services needed.

ADMINISTRATION

The administration must provide not only encouragement and support, but also insight into the larger corporate activities which may affect the plans being made for recovery. This corporate direction will make the initial effort an even better investment.

- Sensitivity to and perception of risks - which risks and how significant they are
- Corporate direction in new products, controls, applications
- Funding availability and timing for this project

It is likely that the processes of drawing information from and giving information to upper management will overlap and intertwine. As the resource person investigating/developing disaster recovery objectives, you will likely be training and educating others during most of this process. Be prepared to educate whenever information is requested.

ACCOUNTANTS

Accountants should determine the financial impact of disaster. The assets lost due to the disaster is one cost, but the business impact may be much greater. This task is not a precise one, and will change as more is learned about recovery, and as assumptions are made about the details of disaster and recovery.

- Replacement costs for assets destroyed
- Continuation costs for business operations
- Lost revenue due to order cancellation or delay
- Reduced labor efficiency during recovery
- Training costs
- Increased labor costs during recovery
- Provision of alternate facilities
- Penalties due to delivery delays (JIT)
- Cost of restoring records
- Insurance costs (even if self insured)

ATTORNEYS

Attorneys should assess the legal exposure of the organization to business interruption due to disasters. This varies with industry and your particular situation. The list which follows gives a number of regulations and laws which may affect your situation.

- The Foreign Corrupt Practices Act of 1977.
- Banking Circulars for 1983, 1984
- Armed Forces regulations and procurement policies, which vary by branch of the services and commodity
- National Bureau of Standards publications (87)
- Office of Management and Budget Circulars (A-123,A-130)
- United States Department of Agriculture (DM3140-1)
- Transportation directives (1600)
- Treasury directives (81-41)
- Department of Energy orders (1360.2,5636.2,5636.4)
- Employment Regulations
  - Age Discrimination in Employment Act of 1967
  - Employees' Retirement Income Security Act of 1974 (ERISA)
  - Privacy Act of 1974
  - Economic Stabilization Act (Wage/Price Controls)
  - Federal Unemployment Tax Act
  - Freedom of Information Act
  - Fair Labor Standards Act
- Occupational Safety and Health Act of 1970
- Executive Orders and Federal Regulations
- Public Utility Holding Company Act of 1935
- Credit/Investments
  - Equal Credit Opportunity Act
  - Investment Advisors Act (1940) or Company Act (1940)
  - Securities Act of 1933
  - Truth in Lending Act
- Contractual or service obligations to customers or vendors
  - Just in Time ordering (JIT)
  - Control records for quality, integrity or lot

The attorney's review of contracts for recovery services may be required. It will help in the early discussions for service to know the legal requirements for your company, and for the attorney to fully understand the nature of the services, and the objectives to be met by the service.

AUDITORS

Auditors are great advocates of recovery preparation, and are both a source of information and of access to higher levels in the administrative heirarchy. While auditors frequently cause the investigation of recovery needs and alternatives, they should not be viewed as adversaries. They can provide insight and options.

- Business practices which support recovery
- Ways others have overcome specific weaknesses
- Checklists of specific needs in the recovery process
- Applications which need specific handling
- Corporate policies
- Audit expectations for strategies, plans, tests

The auditors may or may not be willing to assist in the writing of the recovery plan, or its testing, but they will probably have to approve the strategy and plan, and review the test results. Find out early the needs they see, and objectives which they feel need to be met.

ACTUARIALS

Actuarials and insurance specialists are the source of supportive information, and may contribute to the financial justification for the recovery services. Look to them for both exposure to hazards as well as opportunities for savings in insurance charges as a result of the planning and protection provided.

- Assess exposure to various hazards
- Evaluate the likelihood of specific events
- Develop risk management options
- Provide loss coverage and business continuation coverage
- Establish rate reductions based upon recovery preparation
- Determine which professional adjuster will be used to determine damages to protect against delays in negotiating the claim

ARCHITECTS

Architects, contractors, and engineering staffs should be asked for assessment of the degree of damage and process of rebuilding which might be expected for the physical facilities.

- Probable damage due to the hazards known or anticipated
- Possible injury level which might be expected, and steps to reduce, if possible.
- Prevention activities which will reduce or delay the damage
- Availability of equipment and materials required
- Permit requirements and the time necessary to acquire them
- Temporary relocation options and their costs
- Identification of contractors and subcontractors for preliminary discussion and development
- Estimates of time delays which might be experienced

During actual plan development and preparation for test or recovery, these people can take a number of steps which will speed up the recovery process. This may include drawings, permits, preapprovals, training of subcontractors. In short, reconstruction will be the longest path in the recovery plan, and any steps which can accelerate this step will be of benefit.

AUTHORITIES

The authorities - police, fire, building inspectors, health department - may be the ones dictating who has access to the facility following a disaster, and what is required or allowed for temporary or permanent restoration of activities. It is worthwhile to contact them for insight and preparation.

- Inspection requirements for reoccupying the facilities
- Access to the facility or area
- Acceptable interim options which might involve the facility or grounds
- Restrictions which might apply, and their reasons for them

ASSOCIATES

Associates within your company can identify the real world impact of the disaster, and determine the steps required to operate without the computer support, or with limited support. When the line operations of the company are impacted by the loss of the computer, the value of recovery solutions go up.

- Alternate strategies for operation
- Minimum configuration requirements
- Schedule options for crews
- Opportunities to operate without the computer or with delayed input

If the adverse impact on the line activity of the company is great, these will be the people who are key advocates for recovery preparedness.

CONTINGENCY PLANNING -- THE AUDIT PROCESS

Leslie A. Virgilio
CooperVision Ophthalmic Products
3495 Winton Place
Rochester, New York 14623
(716) 427-7960

Disaster Recovery is the ability to continue your information processing when your facilities for doing so are unavailable. Situations requiring recovery can be natural disasters, industrial accidents, human relations, and hardware failure. None of these are recoverable without a Contingency Plan. The Disaster Recovery Strategy protects against the improbable. Contingency Planning prepares you for the inevitable.

Companies should insure their computer hardware for the replacement costs involved. Along with this policy they may also have an "ability-to-operate" clause which guarantees them some income should they have to close business due to data processing failures. Most insurance policies protect against immediate financial loss due to disaster. Other losses such as client base, reliability of service, cash flow, payroll calculations, and reporting capabilities are not recoverable on insurance policies.

Auditors are now requesting that part of their clients corporate profile be a Contingency Plan and Disaster Recovery Strategy. Some organizations may even be pressured by government agencies to prepare such a plan. Yet, when companies are asked whether or not they have a Contingency Plan for their Data Processing needs, the answer is often, "Yes, we back up our system and store our tapes at a remote location." The problem with this answer is what do you do with those tapes if your machine is unavailable to use due to a disaster situation? Several options are available.

One option is a Private Backup Site. These sites are owned by the business involved. To be of full benefit in the case of a disaster, this site should be in a different location than the original. There are two types of private backup sites: "cold" and "hot". A "cold" site is a fully equipped computer facility, without the computer. Only electrical power, air conditioning, and telecommunications equipment exist. When disaster strikes, the computer and required peripherals must be obtained, installed, and tested. Although relatively low in cost, the "cold" site has the disadvantage of a lengthy implementation. A "hot" site is a fully equipped computer facility with an identical or very similar computer system to the original, already installed. Obviously, the most desirable system from an operations standpoint, this alternative is extremely expensive. Another drawback to this alternative is the easy justification for using the system/facility for other uses. This eliminates the 100% availability for disaster recovery.

A second option is a Mutual Backup or Reciprocal Agreement. A Mutual Backup Agreement can be between two businesses, or between two different computer sites within the same business, with similar system configurations. They agree to back up one another should a disaster occur. The businesses are usually located near each other. To eliminate competition, the companies are usually in different industries. Although there is little or no cost to the agreement, there are several drawbacks. Few corporations will allow a second or third level manager to form a binding contract on a handshake. There are very few sites with sufficient excess capacity to operate a second business without curtailing their own operations. Will your CEO allow his business to fall behind to allow another company to use his hardware? Further, the most critical phase of Disaster Planning is testing. This is the step most commonly omitted from mutual agreements. For these agreements to <u>really</u> work, companies would have to have far more computer capacity than their businesses require.

A third option is the "Cold" Backup Site, which is similar to the privately owned cold backup site. It is an "empty shell" facility owned and operated by a company in the business of data disaster recovery. Cold Sites bring up the term "Allowable Downtime". How long can you be without a computer? How long will it take to "warm up" a cold site? Is it conceivable that a hardware configuration sufficient to support operations can be delivered, installed, and made operational within a sufficient amount of time to keep the company running efficiently? An open purchase order with a vendor for delivery of a complete configuration only guarantees purchase of the equipment, not that it will be delivered when needed.

A fourth option is a Remote Site. This type of facility depends on telecommunications. Dialing into a computer system can create problems for the users. The number of external forces working against the ability to exchange information are staggering: weather, traffic accidents, power failures, and load switching just to mention a few. Also, you must be sufficiently supplied with modems, at reasonable working speeds, and terminals to be able to use the Remote Site.

A fifth option is a Mobile Site. The Mobile Site again brings up the term "Allowable Downtime". Alot of computing power can be put in the back of a truck, but how soon can you get it where it is needed? And at what cost?

Another option is the Hot Backup Site. This situation is probably the most acceptable solution to disaster recovery. It is a fully equipped computer facility, owned and operated by a company in the business of disaster recovery. Although there can be competition for its use, disaster recovery companies can often compensate by having multiple CPU's and/or multiple hot site locations. Often, the disaster recovery facility can also accommodate users by having terminals/workstations for people to use.

The choice of where to recover must meet the needs of the company. Hewlett-Packard has provided us with computing hardware compatibility unsurpassed in the industry; an interchangeable operating system. For the most part, any software will run on any machine just by using two MPE commands: RESTORE and RUN. Therefore, any of the above options will work.

Assuming you have solved the hardware problem, what about your users? A workable Disaster Recovery Strategy and Contingency Plan requires not only hardware to continue operations, but also a transferable set of software and users.

What then should the approach to Auditing and Contingency Planning be? Ask yourself "What's wrong with the existing methods of preparing for a disaster?" The answer is simple. We write up a set of procedures, document systems, define requirements, ignore the users, put it on a shelf and never look at it again. For a Contingency Plan to work, the document must become a useful tool; something that will be a part of our daily operations and decision making. If it is used daily, it will be updated. Having current information is the only way any Contingency Plan can work.

Basic DP Audits are offered by most public auditing firms as part of the annual Financial Audit. These audits cover procedures and data flow, usually tracking specific portions of information in order to understand their source. This information should be incorporated into the Contingency Plan. However, a complete plan must also include the mechanics of operation. It must be developed by individuals who know and understand the computer systems being utilized as well as the information processing needs and methods of the organization. The only way to truly accomplish this is through the Data Processing Audit. This Audit includes complete definition of the Data Processing System, both manual and computerized. Identification of each application and the subsets of these applications are also defined. Within the subsets, key personnel, special requirements such as source documents and output forms, as well as the relationship between applications, are revealed.

It is not good enough for the Contingency Plan to tell us only what to do when a system fails. It must guide us when the individual component parts of the system go astray. These component part failures come in a variety of ways. The most common in any organization is key user vacation time and extended sick leave. Moreover, from a computerized standpoint, if data becomes corrupted or application software fails, which related applications will no longer function? As mentioned earlier, we are dealing with the inevitable. People will change jobs. Hardware systems will fail. Processing will need to be stopped. By understanding the interrelationships and needs of the data processing function, it becomes possible to prepare for these inevitable situations.

Contingency Planning cannot be restricted to the computerized flow of information. It must include those manual procedures required to supply the flow and support those which are computer dependent. In the event of a complete systems disaster, such as fire, it is also necessary for the Contingency Plan to identify which applications are critical to daily business; which applications need to be put into place first. Fortunately, the Data Processing Audit identifies the applications most critical to the organization and, of these, what other applications are dependent and which are related. We now have the ability to put into place portions of the overall system versus restarting of the entire process.

There are several factors that should be considered when doing a Data Processing Audit. These include: hardware resources utilization & requirements; primary & secondary systems support equipment; vendors; forms; software applications; personnel -- duties, responsibilities, back-up and schedules; emergency calling; subordinates; risk analysis -- resources, environment, personnel and software; critical processing timetable; allowable downtime. Let's look at each of these critical areas independently.

HARDWARE RESOURCES UTILIZATION AND REQUIREMENTS

When evaluating hardware resources for disaster recovery planning, we need to know what the minimum requirements needed to be able to function in a contingency mode are. In order to determine that, we need to know current hardware configurations, including: operating system MIT; computer series; megabytes of main memory; number of printers and LPM speeds; number of modems and bawd rates required; number of Mux. channels; number of INP boards; number of modem links; number of tape drives and BPI speeds; disk space utilized; number of terminals needed and if any special terminals are needed for added memory or graphics capabilities; special equipment such as bar code readers and optical scanners.

PRIMARY & SECONDARY SYSTEMS SUPPORT EQUIPMENT

For each site location, we need to know about the environment. What type of power control equipment do we have? What type of environment control equipment? Who are the vendors, the contacts? What company provides our power source? Do we have fire protection? If yes, what type? Halon, water sprinklers? What type of structure is the computer room? Are there fire walls? If there is a fire outside the computer room, how much time do we have before the computer room catches fire as well? All this information is vital to be able to rebuild the type of facility you currently have and/or to be able to salvage what currently exists. These factors also determine the disaster risks and survival abilities.

VENDORS

Computer supplies and other equipment needed to run your systems may also be inaccessible in a disaster situation. A list of vendors with purchase order numbers, inventory lists, and other information is crucial to facilitate replacements. Information about software vendors is needed as well. Does the company provide telesupport and/or site support? Who is the primary contact? Has the vendor given approval for the use of their software on an alternate machine? You want this information easily accessible.

FORMS

Identification of forms must also be done. What are the forms used in the applications? Who is the vendor? What is the order unit of measure? What is the monthly usage? What is the order lead time? Where are the forms stored? What applications use the form and what is the consumption by the application? Forms identification not only applies to preprinted output forms. It should include manually prepared source documents that are needed.

SOFTWARE APPLICATIONS

Software applications can have one of three characteristics. They can be dependent, independent or associated. Independent applications are those which will function as self-contained units regardless of the existence of any other applications. Dependent applications are those which require interaction between two different applications for the purpose of decision making. Associated applications are those which utilize portions of other systems in a passive manner. For each application, dependent and associated applications must be identified. Each application user must be identified as well as their duties and responsibilities. Back up personnel must be assigned to each user. Who are the in-house technicians? Is there a vendor software engineer? If yes, who is it? Where can he/she be reached and at what hours? What type of application are we running? Finance, order entry, etc. What languages is the application written in? What types of files does it require? If it's a purchased application, have any of the programs been customized? How many terminals are needed to run the application? How many megabytes of disk spaces? How many people? What is the Allowable Downtime? Which computer installation is used for this applications processing? For each subset of the application, the transaction volume and required transaction turn-around time must be defined. The critical processing times of each subset must also be defined, as well as the duration. We also need to know what special equipment, whether computer or non-computer, is needed to run each application successfully. For instance, when running accounts payable, the checks may need to be printed on a special printer, bursted by a burster, folded and sealed by a folding machine and then stamped by a postage meter. We need to define if the equipment is critical or merely useful to the processing. A most critical question...has the vendor approved use of the software at an alternate site in a disaster situation? Does the application require special forms? If yes, are they critical or just useful?

PERSONNEL -- DUTIES, RESPONSIBILITIES, BACK-UP, SCHEDULES
Who are the key people in the information flow? They know who they are, but how many of us have assumed responsibilities, out of necessity, that our direct supervisors are unaware of? The relationships between people and their work are similar to the relationships between hardware and software and between software applications. Knowing how the people relate to the work performed is just as important as knowing how the hardware and software relate to each other. This is the mechanics of operation, the manual process required to support the electrical process.

EMERGENCY CALLING
An Emergency Calling List is a list of all key people, in call-priority order. Supervisory personnel should be given the highest call priority since they should be the first to be notified in the case of a disaster.

SUBORDINATES
All key personnel need to be listed by their supervisor. In the case of a disaster, each supervisor needs to know who they need to contact and what the appropriate phone numbers are.

RISK ANALYSIS -- RESOURCES, ENVIRONMENT, PERSONNEL, SOFTWARE
This area is very critical. There are several ways of reducing the risk of a disaster from protection of computer data to protection of data center operations; from protection of vital user records to insurance. A successful risk analysis will identify areas that are lacking. Areas that, if not taken care of, could be partially responsible for a data processing disaster.

CRITICAL PROCESSING TIMETABLE
Critical processing periods are designated for each system and/or subsystem. This information indicates how long an application can be unavailable before it is needed again. It will also indicate how long the application needs to run to successfully complete. Since this information varies from day to day, it is more or less represented in calendar form. Special processing periods (end of

month, quarter, etc.) are also specified. All this is taken into consideration when making judgments about data recovery.

ALLOWABLE DOWNTIME

How long can the company survive without a computer system? The allowable downtime can be dependent on the day of the week, day of the month, and/or time of day. Typically, allowable downtime is the length of time between the running of critical applications.

With all this related information pulled together, the Contingency Plan emerges. But more than just a Contingency Plan, you also have an Audit Report that defines the mechanics of operations, the relationships of applications, the key users and their schedules, and the special requirements required to support the electronic flow of information. A document that, because it is used on a daily basis, will be kept current.

Preparation for the inevitable must begin with foresight. If we are to protect our business, and ourselves, against catastrophe and limit disaster, our data must be sound, our plan current, and our resources assured. The steps that must be taken are: making provisions for the replacement costs of data processing equipment, including the facility itself; auditing and documenting the data processing situation; updating the document as required; selecting a recovery site and binding it with a contract; and testing, at least once a year, to make sure the plan works. Anything less, and the preparation for the inevitable could turn to disaster.

It Can't Happen to Me!
(Disaster preparedness - A Real Life Adventure)

Jim Pashak
Hewlett-Packard Company
7887 Washington Village Drive
Dayton, Ohio 45459

Disaster recovery is a topic often preached and whole heartedly subscribed to but rarely is it placed high on the project list. A system manager's or programmer's time is too precious and costly to work on something that, in all likelihood, will never be used. It's difficult to tell the sales manager you cannot get his report for two weeks because you are writing a disaster recovery plan. I agree that it is difficult to justify the time when so many other things need to be done ASAP. If your company uses the computer to run its business, how can you not take the time?

I have been around the HP3000 since 1977. Back then, disaster recovery procedures were to restore from your partial and full backups. More than any other reason at that time, they were done to protect yourself from the fatal head crash, or the overzealous CE who demanded to realign your heads every three months during PM (Preventive Maintenance - arch., CE scheduled time to work on system to prevent failures and unscheduled downtime, but many times resulted in same). Backup and emergency sites were covered by finding a user with a similar configuration and agreeing to share unused computing power, during 8 PM to 4 AM for example. Reciprocal agreements are not bad. I am just not sure how workable they are and, until recently, had not had a customer that required one.

Hardware reliability has gone up in quantum leaps since 1977, but the same disasters that were with us back then are still around - fire, floods, earthquakes, tornados, hurricanes. We are all aware of these and know they cannot be prevented. What can be prevented is a major loss of data and unreasonable delays in getting another system installed. Of the two sites in the Dayton area that had disasters recently, neither had formal disaster recovery plans. In fact, neither had written data recovery plans to explain how to recover system and data files, should the system manager be on vacation.

The first site is a construction firm with its headquarters north of Dayton. This firm specializes in luxury resort hotels and condominiums in the Carribean Basin. The building is a one story brick structure with a flat roof. Their HP3000/42 with one 7933 disc drive and a 7970 tape drive was installed in May, 1985. They use third party software for job tracking, financials, and payroll. The computer room is an interior office with separate air conditioning, power, and has a raised floor. The DP staff consists of two people, one of which is a contractor. Backups have always been done at this site using the standard weekly full and partials the remaining days. Also on site is a concrete, fireproof safe where the weekly backup tapes are kept. No formal written procedures existed for data recovery and system recovery was considered covered by insurance.

Both the system manager and the contractor had been to a user's group meeting in October, 1988 with the main speaker's topic disaster recovery. The first thing to be implemented as a result of that meeting was to start labeling their backup tapes better. Data recovery was not considered a problem.

The roof was almost done. Work had been on-going for about a month, starting in mid-October, and was expected to be done soon. All that remained to be completed was about 8 feet around the perimeter of the building. The weather forecast called for rain so the roofing contractor covered the unfinished portion of the roof with plastic and weighted it down. A procedure used often and effectively. Unfortunately, at 2 AM on 10 November, 1988 it not only rained but winds gusting 50 to 60 mph accompanied the rain. The plastic put down temporarily was blown off. The permanent roofing already in place was lifted by the wind and rain blown under it. Ceiling tiles absorbed water until the weight of the water caused them to come crashing down. Making everything look like it was covered with wet snow. The carpet was soaked with water. That was the scene that greeted the building manager when he arrived on the scene at 4:30 AM after being awakened by the storm.

The system manager was called at 6:00 AM and told to come down immediately. The computer room has a different ceiling than the rest of the building and at first look did not look too bad. When he saw the water dripping from the air conditioning vent directly onto the disc drive, he suspected there may be a problem. The disc drive (an HP7933) had spun down. He pulled up the floor tile and noticed that water was about one-eighth inch from going into the outlets and started vacuuming the water with a wet/dry vacuum. His better judgment took over and he went to the main breaker and shut off power to the computer room before continuing. Later that morning he placed a service call to HP.

The CE arrived on-site early that afternoon but couldn't really tell how extensive the problems were going to be. The disc drive obviously had problems because it had gotten the most water. What kind of problems and the extent would have to be determined later when things were dried out and power could be restored to the system.

11 November - The CE checked out the system over the course of the day. The Sales Rep called the customer and volunteered a site at the HP office if they needed to get some processing done.

14 November - The estimate for parts and labor to repair the disc and cpu was $10,000. In addition, there may be additional charges for the cpu for problems that would surface when a good disc was installed and all tests done.

15 November - Customer met with insurance adjusters. This was a preliminary meeting to determine what direction to take with hardware. They were told to take their pc's to a pc store for repair and not to do anything with the HP3000 at this time. The customer also started getting preliminary prices on new and remarketed equipment from HP.

16 November - Customer's management said to start ordering equipment from HP. The insurance company seemed to be dragging its feet and something needed to be done. The HP sales rep started arranging for a demo system Series 52 from the Dayton office to be sold to the customer. At about 3 PM I was told the customer was buying the demo system and to get it ready for shipment. I did a full backup on the system and started de-installing the parts the customer didn't need (Lanic, INP's, ADCC's). I had the system ready to head out the door at 6 PM. That evening they met with the insurance adjusters again. The insurance company said we don't owe you a 52, we owe you a 40 and we know where we can get one with a disc drive for $13-14K. The system manager's response was ''I want to know where the system came from and the particulars about it.''

Management to system manager - ''What do you think about it?''
System Manager - ''I want guarantees that it will work.''
Insurance company - ''It's the same computer you have now.''
System Manager - ''I know the history of our system and it's reliability and have no idea what this one would be.''
Since it was after 7 PM, they could get no information on the system proposed by the hardware broker.

17 November - The system the insurance company was proposing was from two different sites. This was not considered the optimum solution by the system manager. HP was called that morning and told not to ship the system. That afternoon the customer agreed the difference in price between the broker's system and HP's would be worth the peace of mind and they would also get an upgrade. Call came in to HP that afternoon to ship the system. A truck showed up at HP's loading dock at 5:00 PM and arrived at the customer's site before 6:00 PM.

18 November - The system was installed and reloaded. No data was lost because a partial was done the evening before the rain storm.

19 November - This was a Saturday. The data entry people came in to enter the past week and a half's transactions.

21 November - Everything back to normal. Reports that were damaged by the water were re-printed (Spool files from period close are placed on tape prior to printing for recovery purposes.).

Why did it take so long to replace the system? What might be done differently to get the system faster and with less hassle? Part of the reason it took so long is because of the nature of the customer's business. Resort hotel construction doesn't happen overnight. Material required had been either on site or ordered for the next several weeks work. Payroll is processed every two weeks and had just been completed prior to the rain storm so they had about a little less that two weeks before it had to be done again.

What might be done differently to get the system faster and with less hassle? Make sure the equipment list the insurance company has is up-to-date. In this case the insurance rider that reimburses a company for lost time had been dropped, so the insurance didn't have any great reason for speeding things up.

What did you learn?

- The value of backups.
- Insurance companies sure don't move as fast as TV commercials show.
- Not to keep dump listings lying around.
- Would be nice to have hot site some where, but until costs come down (or until we get richer), it won't happen.

The second site is a small manufacturer that had a Micro 3000 with a HP7914 and HP7937 disc drives and a HP7974 tape drive for running a manufacturing package. Since the Micro 3000 requires no special environment for operation, it was placed in an office designated as the computer room in the main manufacturing site. The total DP staff is a part-time system manager and a part time operator to do backups. This site is a three story brick building built in 1890 with the newest addition built around 1920. A new addition for administration was added recently but no one had moved into it. The interior floors and floor joist are wood. About 250 people are employed at this site working two shifts so people are normally on site from 6:00 AM to 12:00 midnight.

Backup procedures call for a full backup Friday evening and partials Monday thru Thursday evenings. Backup tapes are kept in a fireproof safe on-site. A typical backup was around 14 tapes.

On Wednesday, 11 Jan 1989 one of the shop foremen came in at 5:45 AM. and discovered the fire. The fire department was called immediately and the fire was extinguished a short time later. The fire started in electrical conduit in the ceiling about 30 feet from the computer room. It spread over the top of the computer room and burned through the ceiling joist, producing a lot of smoke. The sprinkler system, of course, went on spraying water over the entire building. It appears the temperature sensing power shunt was tripped because the printer stopped in the middle of a printout. The computer system was not burned but was completely covered with thick, black soot.

The system manager was never alerted so he came rolling in at his normal time (about 8:00 AM). He found out that for some reason the operator did not get the tapes put in the safe that night. Convincing the fire department he needed to go in the building, they escorted him to the computer room to pick up his very sooty, but not melted tapes. The insurance company and HP were notified immediately. The insurance company brought in a hardware broker that said they would provide a re-marketed system that would be eligible for HP maintenance. The system manager balked at this. They had purchase new hardware from HP and wanted the system replaced with new hardware from HP.

While the negotiations were going on between the customer and the insurance company, the HP sales rep was on the phone to California lining up a replacement system. Because Micro 3000's are manufactured in Mexico, it was decided that going through customs would cause too much of a delay. A remarketed system from HP was available immediately and was acceptable to the customer. As soon as HP got the OK, it would be shipped.

On Thursday, negotiations took place with the insurance company. The customer decided to take the opportunity to upgrade some of the equipment with them paying the difference between the insurance settlement and the actual price.

On Friday, management decided something had to be done and a purchase order was cut for the new system. The president of the company also told the insurance company that if they did not settle he would shut the company down for two weeks while they were deciding; forcing the insurance company to cover wages for 250 employees. The order was called to the local HP office that afternoon and relayed immediately to California.

The system was air-freighted to Dayton on Saturday and installed by the CE Sunday (in the new office addition). The tapes, although very dirty, were not damaged by the fire or water. The heads on the tape drive had to be cleaned often but went through fine. No data was lost and the system was up and operational Sunday evening. New data lines were pulled and new terminals installed. By Wednesday everything was 100 percent operational.

This site was as prepared or possibly more prepared than many sites to handle what happened. Had it not been for the operator not taking the tapes to the safe, the data recovery portion would have been trivial. As it was, it was a bit tense until all of the tapes were read. They were a bit luckier than most might have been because the new site was already there waiting for them to move in. The fire speeded up that process a bit. It wasn't all luck though. They have a company that comes through periodically to inspect the building and equipment and make recommendations on safety. That company also evaluates the equipment on site and has an inventory of what is on site.

What have they learned? Since they were doing things about as well as could be expected before the fire, not too much has changed. The HP7974 was replaced with a HP7980 so the operator has to take 3 or fewer tapes to the safe (as opposed to 14), so it is not likely the tapes will be left out again. They are looking at having their new insurance policy written to cover replacement costs as opposed to the current value. They still have not written any procedures yet.

Some simple suggestions to make your disaster recovery easier:

- Follow your backup procedures;
- Label your tapes clearly;
- Have either off-site storage or a fire proof safe for storing backups;
- Check your insurance policy periodically so you know what is covered;
- If you have no written procedures, write them;
- Determine if a backup computer site is necessary and have it ready if possible (it could be a store room with power run to it);
- Test out your procedures periodically - at least to the point where data recovery is done logically;

There are a lot more, I'm sure. These are but a few to just get you thinking about disaster recovery. Its something that we hope never happens, but there is no substitute for adequate planning.

Using the 950: A Practical Perspective

Gary Sogar
Litton Integrated Systems
2300 Litton Lane
Hebron, Kentucky 41048
606-334-2991

I. INTRODUCTION

This paper is intended for new users and users-to-be of Hewlett-Packard's 900 series computers. It should answer the question "What is it really like to migrate to a 'Spectrum' and use it daily?" Other questions that will be answered include "Is it really that fast, what's so different about it...I mean really," and "what's this I hear about 'job starvation'?"

The buyer of a 900 series is faced with a difficult task. There are many questions, but the answers are difficult to find. The machine is so new the local HP representatives have a difficult time getting complete answers. Also, there is not a well-known network, either formal or informal, of "Spectrum" users one can turn to for the "real dirt." Rumors are frequently heard from "usually reliable sources," but they are difficult to confirm.

While this paper doesn't have all the answers, it should serve as a good overview and starting point for those newly initiated into the 900 family. After surveying the planning and implementation process, I will address day-to-day issues, including a collection of miscellaneous information, notes, observations, "gotchas" and recommendations.

One note about the possible obsolescence of the technical information contained in this paper. It was written in May of 1989. Some of the problems and information presented may be outdated and inaccurate by the time you read this. Please check with your local HP representative.

II. PLANNING

Many areas need to be addressed before actual installation and implementation of the new machine can occur. Planning and preparation will guarantee a successful and problem-free (as much as humanly possible) migration. The following paragraphs briefly describe the main areas of concern with recommendations.

<u>CPU, MEMORY AND DISC SPACE:</u> After deciding to buy a 900 series, you now must decide exactly what to buy. Which CPU to buy can only be determined by each site's particular needs. There are good rules of thumb for the purchase of the other two main ingredients: disc and memory.

DISC SPACE: Increase your current disc capacity by forty to fifty percent. This is due to three hard facts: MPE XL and system files take more space than on the classic machine, native mode program files take more space than analogous compatibility mode files, and optimized compatibility mode program files take ten or even fifteen times the space as non-optimized ones (more information later on optimized program files).

MEMORY: About 0.8 to 1.2 meg for each interactive user, depending upon how much code and data are shared. For example, if you average seventy to eighty users during normal business hours you should buy 96 meg. You might be tempted to buy more, since you've heard about memory mapped files and that jobs eat up memory. More memory can be helpful, but try the recommended amount first. If you don't experience any problems, additional memory probably won't significantly increase your performance. Too much memory can even cause performance problems, but more about that in the later part of the paper.

PRINTERS AND TERMINALS: Some older printers and terminals may not work on your new machine. Older HP terminals may need to have their ROM's upgraded or they won't be able to communicate with the 900. Any terminal must have XON/XOFF capability. In the case of printers, older ones may be on the unsupported list. You may try to use them, see that they print, and assume they work. Data will be lost, however, if they go off-line while printing. No upgrades for these printers: they must be replaced.

TELECOMMUNICATIONS: Make sure all your multiplexors have settings for XON/XOFF. Be wary of "black boxes" or any other device that may alter the signal, especially handshaking control. Telecommunication is a bit more sensitive on the 900. For example, if XON/XOFF is not set correctly at your printer and both multiplexors, your printer will not perform properly, if at all.
Terminals can communicate with the 900 at 9600 baud with no problem as long as they are set to XON/XOFF transmit and receive pacing and have parity set to "none."

LDEV NUMBERS: Make sure you have an accurate listing of logical device numbers, including terminal/printer types and baud rates. Label you cables, too. The accurate ldev and cable numbers are essential when you do the massive re-plugging into the DTC's. The terminal/printer types and baud rates are needed when you run NMMGR to configure the DTC ports. There are fewer terminal/printer types on the 900, and the list will help you determine the correct settings.

THIRD-PARTY SOFTWARE: Obtain native mode versions of your software if possible. Some compatibility mode packages run horribly on the Spectrum or may even crash your system. The closer the package is to working inside the operating system and system utilities the higher the possibility that a compatibility mode versions might cause problems.

Obtain your software as soon as possible. This will permit you to load-and-play on your new machine as early as you can. This will also mean the vendor will send you update tapes as they are released, which is very important for software newly converted to native mode

An additional note on third party software. Most vendors' prices are based on the CPU you own, so be prepared for an increase is cost. Sometimes this increase is substantial. Sometimes it's criminal.

<u>TRAINING</u>: The importance of training cannot be over emphasized. The three MPE XL related classes (programmer, operator and system manager) are essential. They are also very good and full of information both necessary and useful. The student manuals for these classes are useful to keep. We have used ours several times as reference sources. If possible, send more than one person to each class, but send each to a different instructor (which usually means a different site). Each instructor has his or her own expertise and areas to emphasize. Different training sites also mean different student populations, which could potentially yield a variety of interesting information.

<u>STAFF:</u> As migration comes closer, and especially after your new box is delivered, your staff will become more and more involved in making the new computer operational. Planning their time and duties is important, since they will also be maintaining your current system until migration is complete. Appoint one person as the migration coordinator. This person's responsibilities would include order verification of hardware and software, scheduling delivery, being the primary HP contact, planning hardware and software installation, requirements planning, etc. In short, someone to plan the major tasks for migration, but not to do all the work.

<u>CURRENT SYSTEM:</u> Do a thorough cleanup of your system, especially your databases. Downsize your datasets. Masters can (and should) get somewhat fuller on the Spectrum (90% or even more). Because of the "mapped file" situation, I/O is not as much as a problem (since the system does as much as it can in memory), so migrating secondaries aren't the problem they were. Don't bother to pack your details yet. When you bring them over to the new machines they'll be split up all over the disks anyway. It is important, however, to pack them after they're on the new system.

HP provides several programs that run on your current system that will aid in migration. For example, one will list all programs that have privileged mode (a potential death knell on the 900). These programs use little overhead, and most definitely should be used.

UDC'S: Take a close look at your UDC files. This is for two reasons. First, some of your more creative UDC authors might be doing some processing that would be dangerous on the Spectrum, such as the PM taboo. Second, the new command file feature can be an excellent alternative to the difficult to maintain UDC files. For the purpose of this paper, a command file could be described as one UDC that has been kept as a flat file that is not SETCATALOGed. That isn't totally accurate, but it will suffice here.

III. MIGRATION

If you have planned and prepared well, the actual physical migration to the new system doesn't have to be a headache. The one area that may cause you grief is converting your in-house code to the new versions of the compilers. The following paragraphs will address this and other issues.

GOING NATIVE: It is best if all the software on your new XL system is in native mode. As of this paper's writing, even HP can't offer you that, so you do what you can. Each of the new compilers (Fortran, Pascal, Cobol) have their own set of incompatibilities from the classic version to the XL. How difficult converting your own code will be is a function of how many of these incompatibilities exist in your current code. At its simplest, going native is recompiling and linking (instead of prepping) your code on the new machine. New skills are required, so the going may be slow at first. The new procedures are similar enough to the classic ones so that "getting the hang of it" does not take too long. However, this is another reason the training classes are so important.

Two of the new compiler directives are very important to mention. One is "$HP3000_16 ON." This lines up real numbers on 16 bit boundaries instead of the new IEEE standard. IT IS ESSENTIAL THIS OPTION IS SET IF YOUR PROGRAM ACCESSES A DATABASE OR UNCONVERTED FILES THAT CONTAIN REAL NUMBERS. OTHERWISE YOUR DATA MIGHT BE READ/WRITTEN INCORRECTLY. This is because TurboImage XL databases are not on the IEEE boundaries. Compiles dictionaries must be checked for this "gotcha" also.

The other compiler directive is "$LOCALITY." It is analogous to the "SEGMENT =" directive on MPE V based machines. Program units of the same locality will be put in the same physical area of the program file. If these units have a high degree of co-usage (such as subroutines that call each other) fewer memory management page faults will result if they are in the same locality.

INSTALL...THEN REINSTALL: Actually installing your system and loading your files can be as simple as advertised: a RESTORE after the install. However, this procedure scatters and fragments your files (including databases) throughout all discs. At your earliest convenience take a full backup, re-install your system, then restore from the full backup. This effectively organizes your discs, which helps i/o and memory management operate more efficiently. While it's true that the mapped file system means disc fragmentation is not a problem from a file building viewpoint, "messy" discs can cause excessive i/o and memory management (due to more frequent page faults).

TIME TO PLAY: The Spectrum is not a machine that's difficult to learn how to use. However, it does require some "getting used to" for those of us reared on the MPE V systems. Besides, this might be the only time that the system is all your own. A virgin system is a DP person's dream come true!

It is highly recommended that you do plenty of SYSGEN's, UPDATE's, UPDATE CONFIG's, running NMMGR...but only one (or two) INSTALL's (since they take so long). Learning these and other system procedures now will save time when you REALLY have to do them later.

MASTERS AND DETAILS ON DIFFERENT DISCS?: The jury is still out on this one. Not enough information was available at press time. While the mapped file access feature seems to indicate that you could put a master and its associated details on the same disc, you still have to do the actual i/o, don't you? What can be said is that if putting connected masters and details on the same disc is a bad idea, it isn't as bad as it was.

USER TRAINING: From a user's perspective, using an XL system is nearly identical to using the classic system. Very little formal training, if any, is necessary. There are a few aspects that might need to be addressed. The XL system has a definable connect prompt that can be different than the standard colon prompt once a user signs on. For example, when you turn on a terminal and hit the return key, "MPE XL" might be displayed. After signing on, the familiar colon prompt will appear. If you want to avoid confusion, you can change the "MPE XL" to be a colon (you do this in SYSGEN).

Some new commands will prove helpful to many users. COPY is perhaps the most useful. It does a simple file copy, which is what most people use FCOPY for. COPY is much faster. CHGROUP is another useful command. It eliminates multiple HELLO's. LISTREDO is another (aren't you getting tired of being frustrated when you type in "REOD?"). It permits you to DO or REDO any of your previous MPE XL commands. PRINT will, as you might guess, prints a file to a device or any file specifications. Writing and using command files will be helpful to some users, especially your own department.

PRINTERS: As mentioned earlier, telecommunication is more sensitive on the XL machine. Printer configuration, including the DTC port, multiplexors and the printer itself, has become more important. Local printers should operate fine at 9600 baud, but remote printers might have to be set lower. This is related to the 900's speed. Internally, the computer handles the information faster than the multiplexors and the printer itself. This results in the 900 expecting handshaking before the other devices are ready. This will not necessarily be a problem with all remote printers, since it is a function of various pieces of hardware.

P.S.: Don't forget to set your multiplexors and printers to recognize XON/XOFF.

IV. DAILY OPERATIONS

While the users may not see a difference in working with the XL machine, you and your staff most certainly will. Not only are system management and operations quite different, the machine performs differently and these differences must be addressed. This section describes these areas of concern and how to handle them.

PERFORMANCE: Whole papers can be (and probably are) written on this topic. Managing performance on the Spectrum is the proverbial "new ball game." This is an area with few, if any, experts. There are few standard practices and procedures. Useful information is starting to filter in from the Spectrum community. Some of this information, along with our experience at Litton, is contained in the paragraphs below.

O.K., SO HOW FAST IS IT REALLY? One standard comment on the XL machine is that "it screams at night when running jobs, but it acts like a model 70 during the day." The increase in speed is most dramatic when the machine is running only jobs. Throughput increases of two to ten times are common. This is easy to understand. Fewer jobs run concurrently at night than sessions during the day. Memory management is very effective in the job environment.

So where does that performance go during the day? Improvement in session throughput is discernible but not dramatic. This situation is a direct result of three areas: the way the system runs jobs (i.e. DQ and EQ), the dispatcher, and tuning.

JOBS: Whenever a process has the cpu, the system tries to bring as much data that the process needs as possible from disc into memory. For DQ processes this is typically a large amount. Since the DQ process has the cpu, inactive areas of memory are swapped out to disc. The word "inactive" is from a system viewpoint. It could be the memory used by someone who has just stopped to take a drink from their coffee cup. So not only will a job grab the cpu, it will grab memory too. When other processes get the cpu the memory manager has to work harder. Limiting the number of concurrent jobs to one or two helps this performance problem, but many shops (like ours) need more concurrent jobs during the day.

Another, more serious reason for jobs degrading response time is "job starvation." If there are no interrupts (such as a process requesting i/o), the dispatcher's cycle time is about four seconds. Tuning quantums don't come into play. Therefore, a process that has the cpu and doesn't pause, it could keep the cpu for four full seconds! If a job's data is in memory it won't pause for i/o, so it "starves" the other users. The dispatcher finally will kick in and see the CQ screaming for service and will give it to them. This problem is rumored to be corrected in an upcoming release, so check with your HP representative.

THE DISPATCHER: Simply stated, the dispatcher was written for MPE IV and hasn't been touched since. The dispatcher was designed around architecture that is primarily constrained by i/o and small main memory. These are no longer the concerns for XL systems. How the dispatcher does its work makes for an interesting read, if you can get hold of it and understand it. Performance would probably improve markedly if HP would redesign the dispatcher.

TUNING: Throw out everything you've learned on the classic machines about tuning. Because of the current situation with jobs and the dispatcher, coupled with the new file mapping and memory management schemes, effective tuning must be relearned from scratch. There are no rules of thumb here. Settings that work well for one shop might be horrendous for another. Size of memory, the number of concurrent jobs and sessions, and the type of processing going on effect the XL system in a more detailed way than on the classic machine. Generalizations or useful default tuning values don't exist since the new machine is more sensitive to subtle changes in the processing environment.

The best advice is to watch tuning carefully and change it frequently, not only to get a feel for what works for you but also to help the machine operate most effectively under its current load.

"THE WALL" AND "THE PLATEAU:" Many sites report that their new systems perform dramatically better, even during the day when sessions are running...until they reach a certain number of sessions. The number for this "wall" varies, but on the 950 with 96 meg of memory it seems to be around sixty. Performance and response time then fall, but do not plummet (you could say it goes from great to good or fair). Once this wall is reached, further degradation doesn't seem to occur (as it would on the classic machine) as the number of sessions increase. Our site is a typical example. When we migrated our final company to the 950, we reached "the wall." When end-of-month rolled around, a real killer on the 70, we noticed no degradation of performance: it looked like any other week.

I have heard no explanation for this phenomenon.

V. MISCELLANEOUS NOTES

This section is a random collection of observations, information and experiences on running a 900 system.

There's an easy and logical method to numbering logical devices. Make each ldev number three digits: the left digit is the DTC number (an arbitrary single digit you define), the center is the board on the DTC (0-5) and the final digit is the port on that board (0-7). To further aid in organization you could make the first one or two ports on each board printer port. Then you would know that all devices ending in a zero were printers.

Compatibility mode programs can bog down the system. Some run horribly and can totally hog the cpu. Many sites complaining of performance problems have excessive CM processing going on of which they are unaware. Most performance monitoring tools designed for the XL machine will give you the percentage of processing time in CM. If native mode versions are unavailable, at least OCTCOMP them.

OCTCOMP is the command that runs OCT, the Object Code Translator. It appends native mode procedures to the program file. The result is not a native mode program, but a compatibility mode program that runs much faster than the same program would if it were not OCTCOMP'ed. This is covered in the XL programmer's training class.

Be careful about running programs on the console. We ran one that locked out all users while it was running. This only occurred once, and I have heard of only one other occurrence, so the reason for this remains unexplained.

XLTOOLS is an account on the XL machines containing some useful programs, such as SURVEYOR. However, BE VERY CAREFUL about the programs you find here. Some of them use privileged mode (which can cause horrors), and some are designed for HP technical support only.

Are your users complaining that there is sometimes a pause for a few seconds after they hit the return key before they get response? Their data and/or code pages have probably been swapped out of main memory onto disc. As already mentioned, the system will try to bring as much information into memory as the current process needs. If a user is inactive as far as the system is concerned, the user is swapped out. Hitting the return key is a request for service, and it takes some time to bring the pages back into main memory.

There are occasional "black holes" when the system seems to lock up for seconds or even minutes at a time. Some factors causing this are process that unexpectedly are running in BQ (this usually occurs with programs using PM), job starvation (explained earlier), improper tuning and compatibility programs.

Some useful performance tools: Glance (HP), Surveyor and Scout (XLTOOLS account), PROBE/XL (SSI), and a SYSVIEW type product (Carollian).

LASERROM and LASERRX are all they're cracked up to be, though LASERRX wasn't available on the XL machine as of May.

Occasional INSTALL's ("reloads") can be helpful for memory management. Organizing files to be less fragmented on disc will decrease memory page faults. When to do it and how much it helps is a judgement call.

You can actually have too much memory on an XL machine. The system will continue to read data into memory as long as there is free memory. Once memory is full, it starts to look for swap out candidates. This can take a lot of time.

Additional memory beyond the recommended amount (about 1 meg per session) will not necessarily help performance. We experimented with 128 meg on our 950 (60 to 80 sessions) with no noticeable effect. This result has been substantiated by other reports.

Database management has become more important. Dataset size is important because of the file mapping feature. Lean and mean databases will operate best. Detail packing is important.

Be sure to read the software status bulletins HP mails you. System bugs are constantly being found and fixed.

RL modules must have an entry point. For example, a module containing only Fortran data blocks will not be recognized.

Unsupported printers will loose data being sent to them if they go offline.

When you install your XL system be VERY sure you don't bring over any MPE V system programs (such as SPOOK5). They can crash your system.

All extents of system files must be on ldev 1.

If a port or a board is down on a DTC don't forget to take a dump of it (SYSDIAG). HP will use this for diagnosing the problem.

About device classes: a separate profile (created with NMMGR) is required for each class.

Change the HPPATH system variable so you can create command files that are limited to certain sign-ons. HPPATH contains the path that will be searched for commands files. This also permits you to decrease the number of UDC's and UDC files.

VI. ACKNOWLEDGMENTS

I would gratefully like to acknowledge the following people without whose help this paper and presentation would not have been possible:

Dennis Hille, Litton Integrated Systems, Hebron, Kentucky

Michael Hornsby, Beechglen Development, Cincinnati, Ohio

Chris Haucke, Hewlett-Packard, Cincinnati, Ohio

AXLUG (ASK XL Users Group)

Advances and Trends in Mass Storage Technology

Andrew Patterson
Husni Sayed
Deborah Cobb
IEM, Inc.
P.O. Box 8915
Fort Collins, CO 80525
(303) 223-6071

Introduction

Since the introduction of magnetic tape as a storage medium, mass storage technology has been advancing steadily in an attempt to keep up with the demands of users. The development of Winchester disk technology introduced the computer world to higher capacities, more reliable storage, and faster access to stored data. Following this revolution in data storage techniques, advances in mass storage technology slowed considerably: until the last decade. Recent breakthroughs in the technology, popularizing Removable Winchesters, Optical disk drives, and high capacity tape systems, are defining the shape of the future.

Looking Back

Magnetic tape systems and Winchester disk drives, developed not too long ago, are already beginning to fade into innovations of the past. While these technologies will certainly have a role in mass storage for many years to come, the push for faster, smaller, more reliable systems is already starting to shoulder these solutions out of the limelight.

Magnetic Tape

Magnetic tape, still in use in many places, has been used as a backup and archival mass storage medium for decades. Magnetic tape systems in use up until recently are essentially the same as when they were first introduced: the technology has advanced very slowly, especially in comparison to other mass storage techniques. Although tape storage capacities have increased dramatically with the introduction of thin-film media, access times (which are a function of tape speed) have not changed much over the years. Tape densities have increased, but slowly.

Before the advent of Winchester disk technology, tape drives were used (out of necessity) as primary mass storage devices. Though not highly efficient, the only alternatives at the time, such as paper tape, were even worse. In an attempt to produce acceptable random-access times, multiple tape drives were used to access a single logical file. Today, with the wealth of faster, more efficient primary mass storage alternatives, tape drives are used almost exclusively for backup, archival, and data interchange purposes.

In addition to being slow, there are other limitations that are inherent in tape media. Due to problems with media wear, a single tape typically cannot be rewritten more than 300 times. Because of problems with stretching, breaking, and print-through, the data retention time for magnetic tape is less than 3 years. Magnetic tape also has a limited storage capacity in comparison to newer mass storage technologies. A $10\frac{1}{2}$" 9-track tape reel can store only about 180 MBytes at maximum density.

9-Track Tape

9-track tape uses large, open reels to hold the $\frac{1}{2}$" tape. The term "9-track" refers to the manner in which the data is stored: in nine distinct tracks across the width of the tape. One track is used for each of 8 data bits, and the remaining track is used to store a parity bit.

Advances in 9-track tape technology have focused primarily on increasing the density of stored data. The first 9-track tapes had densities of 550 bytes per inch (bpi), while today's high-end 9-track tapes boast densities of 6250 bpi.

The single greatest advantage to 9-track tape is its pervasiveness. Virtually every computer installation has at least one 9-track drive somewhere. In addition, the data formats have been highly standardized: all 9-track tapes of equal density write and read data in an identical manner. These two traits have made 9-track tape the medium of choice for data interchange tasks. For this reason, 9-track tape drives will continue to be popular until another standard of data interchange emerges.

Cartridge Tapes

With cartridge tape, a small plastic case is used to enclose two reels of tape. The tape is automatically loaded when the cartridge is inserted into the drive. Cartridges have several advantages over 9-track tape, including their compact size, ease of handling, and resistance to the environment. There are two types of cartridge tapes commonly in use today: the 3284 cartridge and the $\frac{1}{4}$-inch cartridge.

The IBM 3284 (or, simply 3284) cartridge is named after the IBM computer for which it was developed. The small cartridge, approximately 4" by 4", uses $\frac{1}{2}$" wide tape. It stores data in a manner similar to 9-track tapes, except that multiple 9-bit tracks are stored across the width of the tape. Currently, 3284 cartridges use two 9-bit tracks, though it is expected that this track density will double several times in the future.

QIC, or Quarter Inch Cartridges, are roughly 6" by 4" and (as the name implies) use $\frac{1}{4}$" wide tape. Smaller sizes are available for personal computer systems. QIC tapes have capacities ranging from 40 to 135 MBytes, depending upon the length of the tape.

QIC tape drives employ serpentine serial recording. Data is written in 9-tracks down the entire length of the tape. At the end of the tape, the direction of tape motion is reversed and the data is then written parallel to the first set of tracks, but in the opposite direction. A major problem with QIC tape drives is their lack of standardization, as each manufacturer implements storage schemes in their own way.

Magnetic Disk Media

Magnetic disks can be divided into two general categories: Removable disk media, and Winchesters. Magnetic Winchester hard disks were originally developed in response to an outcry for *faster* access to stored information. These random access devices, much faster than the sequentially-accessed tapes, finally gave users access to data in under a second.

Early in their evolution, Winchester disks acted mainly as "middlemen" between tape systems and main memory: users continued to rely on magnetic tape for storing data, and transferring information between computers. As Winchesters became faster, smaller, and more reliable, they gained more widespread acceptance and use. Today, these mass storage devices are found virtually everywhere, offering capacities anywhere from 10 to 2000 MBytes. Though more compact storage methods have since been developed, no other solution has been able to beat the Winchester where speed is an important consideration. Manufacturers have recently developed "Remvovable" Winchester drives consisting of a main drive unit, with a hard disk unit that can be inserted into the main unit and removed as necessary. This gives the user the ability to remove the disk for storage in a secure location, or purchase additional media without the need to buy a whole new disk drive.

Removable disk media are relatively small, thin magnetic disks or disk cartridges that are inserted into a drive unit. This class of magnetic disks includes the ever popular "floppy" disk, and other technologies such as the Bernoulli disk. Removable disk media do not offer as high a capacity as Winchesters, and most have slower access times and data transfer rates.

Current Trends

Helical-scan tape and optical disk drives are transforming mass storage technology. Helical-scan tape is inexpensive, and has a large enough capacity to allow unattended backup of on-line systems. It would take almost 13 reels of high density (6250 bpi) 9-track tape, or 12 IBM 3284 cartridges, to store as much information as a single 8mm tape cartridge. Furthermore, the traditional tape system would require the presence of an operator to change every one of those reels or cartridges: a process that may take hours. This additional labor cost makes helical-scan tape even more attractive. Decreased storage space is another incentive that favors helical-scan tape. 8mm tapes can store 326 MBytes per cubic inch, compared to 3 MB/in^3 for 9-track tape, or 27 MB/in^3 for QIC tape.

Optical disks have many of the advantages of helical-scan tape, such as large capacities in small volume, and also offer faster access times. They are serving to fill a previously empty niche that existed between large capacity, low cost mass storage tape systems, and fast access, high cost Winchester disks. The removable nature of optical media makes it ideal for storing large software systems and data bases that can be swapped in and out as needed. Erasable opticals allow for instant access to on-line removable file systems. This makes them ideal for large, less frequently accessed data bases. Winchesters will still be in demand where fast access times are crucial however, such as in virtual memory systems.

Helical-Scan Tape

Helical-scan tape stores data using technology that was originally developed by the video recorder and digital audio tape industry. The name is derived from the method by which the tape travels over the head. Previous tape technologies used a fixed head, with tape passing (relatively slowly) over the head. Helical Scan, however, uses a head that is mounted on a rapidly spinning drum aligned diagonally to the track. As the tape passes over the drum, the head writes tracks of data in a diagonal pattern corresponding to the pitch of the head. This method produces track densities on the order of 1,000–2,000 tracks per inch.

Although helical-scan drives used in the computer industry are very similar to a VCR (Video Cassette Recorder) or a DAT (Digital Audio Tape) player, they require a much higher reliability. VCRs and DAT players typically have an error rate of 1 in 10^6. When an error occurs during a VCR recording, a small extraneous spot may appear on the screen. Similarly, in the case of DATs, a timeout may occur for less than a millisecond. These types of errors are virtually undetectable to human senses. An error rate of 1 in 10^6, however, is unacceptable for mass storage applications. Because of this, error checking and redundancy must be implemented to produce an error rate more in the neighborhood of 1 in 10^{13}.

8mm Tape

8mm helical-scan tape systems, derived from commercial Camcorder recording technology, are a very recent addition to mass storage applications. With a maximum storage capacity of 2.3 GBytes per tape, 8mm media has the single highest storage capacity-to-volume ratio of any mass storage device currently in use (326 MB/in^3). Access time, as with all tape systems, is relatively slow: in the tens of seconds. Burst transfer rates are on the order of 10 MBytes per minute.

The media used for 8mm tape in the mass storage industries is the same lightweight, plastic cartridge used in the entertainment field. Any high quality metal tape from a Camcorder can be used in an 8mm helical-scan tape drive. The 6" by 4" by $\frac{1}{2}$" cartridge fits easily into a shirt pocket. The volume of sales for the entertainment industry has drastically lowered the price of these cartridges to under $10.00 each. This factor, plus the huge storage capacity of these tapes, has driven the cost of storage to less than one penny per MByte.

Digital Audio Tape (DAT)

DATs, or Digital Audio Tapes, are just now being marketed in the computer industry. They are very similar to 8mm tapes, the main difference being that the medium is 4mm wide instead of 8mm (resulting in a maximum capacity that is half that of 8mm tapes). In addition to being half as thick, DATs are also smaller than 8mm tapes (about 3" by 2").

DAT storage devices use the same tapes as are used by digital audio tape recorders. Commercial DAT recorders for entertainment purposes, however, are currently banned from import due to the intense lobbying efforts of the audio recording industry, which wishes to preserve the current demand for compact disks. These restrictions have effectively reduced the availability of DAT cartridges, so they are generally more expensive than 8mm tapes.

Optical Disks

Optical recording devices were first developed as an alternative to the Video Cassette Recorder. In 1978, the first optical disk system — the Laser-Disk Video Player — appeared on the consumer market. This read-only device used a 12" platter and a laser read-head to play back digitally encoded video signals.

Since then, optical recording technology has been further developed by the mass storage industry, and split into two distinct branches: WORM (Write Once, Read Many) and Erasable optical. A third optical technology, CD-ROM (Compact Disk Read Only Memory), is not commonly used for mass storage since information can only be written during the manufacturing process. CD-ROMs are used largely to distribute and reference large amounts of relatively static data such as on-line encyclopedias, legal citations, and (of course) musical recordings.

WORM and Erasable optical systems use a removable disk enclosed in a plastic cartridge. The main expense involved in optical disk systems is the read/write head, which uses lasers, beam-splitters, lenses, and mirrors to access data. For this reason, most optical systems are single-sided: the disk must be removed from the drive and manually turned over to access the other side of the disk cartridge. An optical disk system that could access both sides of a disk cartridge without removing the disk would require two read/write heads, effectively doubling the cost of the drive.

Jukeboxes, or auto-changers, offer users automated access to a number of optical disk cartridges. Jukebox systems typically contain two optical disk drives, and a mechanical arm used to select and load one of many optical disk cartridges that are stored in the jukebox. These types of systems usually have 5%–10% of their capacity on-line at any one time, and have a maximum capacity approaching one hundred GBytes.

Because the head in an optical disk drive weighs much more than the head in a Winchester disk system, access times for optical drives tend to be much higher than for Winchesters (50–150 ms as opposed to 10–20 ms). Data transfer times, which are dependent solely upon rotational speed, are comparable between the two systems.

The media used in optical disk drives also have numerous advantages over magnetic media. First, since the density of an optical disk is limited only by the wavelength of light used by the laser writing the information, optical disks are capable of tremendous track capacities. Most current systems use a near-infrared laser with a wavelength of 8,000 – 10,000 angstroms, resulting in track densities on the order of 16,000 tpi. A single $5\frac{1}{4}''$ optical disk cartridge with such a track density can hold roughly 800 MBytes of data. Second, the distance between the head and the surface of the disk is much greater than that used by traditional Winchester technologies. A distance of 0.4 mm is typical for optical disks, while Winchesters commonly use 0.0002 mm. This increased separation between the disk and the head makes head crashes very rare. A stray smoke or dust particle, which would cause a head crash in a Winchester disk, will have no affect on an optical disk drive. Finally, the optical disk cartridge itself is very durable. Encased in plastic, it is immune to fingerprints and resistant to heat and humidity.

Optical disk cartridges are expensive in comparison to tape (about $200–$250 each), but their huge capacity makes them cost-competitive with all but helical-scan tape systems.

WORM Optical Drives

WORM optical drives write information by burning small pits in the surface of the disk cartridge, using a laser. Once written, a pit (which represents a binary "1") cannot be restored to a normal flat surface (which represents a binary "0"), so data can be written only once to the same disk sector. To read the data, the same laser is directed at the surface, but at a much lower power setting. The laser is reflected off the surface, and this reflected light is gathered into a photocell. The light reflected by a pit is easily distinguished from light reflected by a flat surface: this difference in light reflections is used to read bit patterns.

WORM optical drives have one major "snag" that is not encountered with other mass storage technologies. Most existing file systems are structured so that some space on the disk is reserved for a directory. This directory must be updated each time a file is added, edited, or deleted. Since WORM optical disks cannot be rewritten, such directory maintenance is impossible. One solution to this problem is to employ special software drivers that use an entirely different file structure involving linked directories. Other solutions involve using a flexible disk to store the directory entries, while the actual data is stored on the optical disk. Directory maintenance limitations, combined with the write-once nature of the media, make WORM optical disk drives useful primarily for backup and archival tasks.

The write-once "limitation" of the medium, however, gives the WORM optical disk drive one unique advantage: once data is written, it cannot be altered. This characteristic makes WORM drives excellent for storing information that must be maintained for legal and audit considerations.

Erasable Optical Drives

Erasable optical disks have only recently been introduced as a viable mass storage technology. Several manufacturers, including Sony, Ricoh, and Hitachi, are now shipping erasable optical systems. A host of smaller companies will be introducing units within the year.

There are three separate technologies associated with erasable optical disks: magneto-optical, dye-polymer, and phase-change. Magneto-optical is the only technology that has reached the production stage, as various problems with the other technologies will require further research before they can be made into marketable products.

Magneto-optical technology, as the name implies, uses a combination of lasers and magnetic field effects to store and retrieve data. The disk is composed of a magnetic material, highly stable at room temperature, encased in a plastic cartridge. To write to the disk, a laser heats a spot on the disk to above 140 degrees Celsius, at which point the magnetic flux can easily be changed by a magnetic head. After the disk cools – only microseconds later – the magnetic flux once again becomes nearly impervious to magnetic fields. It is estimated that, at room temperature, a two ton magnet would be required to change the data on a magneto-optical disk (MOD) cartridge.

The properties of the Kerr effect are used to read data stored on an MOD cartridge. The Kerr effect states that light will rotate in a particular direction if influenced by a magnetic field. An MOD drive uses this effect by directing a low-power laser at the surface of the disk. The light reflected from the surface will rotate in a clockwise or couterclockwise direction, depending upon the orientation of the magnetic flux of the surface. The read head detects the rotation direction, and sends a corresponding value of "0" or "1" to the computer.

Erasable optical disks, however, do have their disadvantages. MOD systems must write zeros to the surface before data may be written to that spot. This means that the disk must rotate twice to complete a write operation: once to write zeros, and once to write the desired information. This quirk effectively increases the write-access time of an MOD drive by 40% over read access times.

The removable MOD cartridge is similar in shape, size, and capacity to the WORM cartridge. A 6" by $5\frac{1}{2}$" by $\frac{1}{2}$" hard plastic casing encloses the disk, and the read/write area is protected by a metal shutter. Unlike WORM disks, MOD cartridges are currently being standardized by both the ISO and ANSI.

Interface Standardization

Manufacturers of interfaces, used for communication between computers and peripheral devices, are just now beginning to recognize the need for standardization. In the past, most large companies used their own interface to connect mass storage devices to their computers. The problem with these interfaces is that they can be used only to connect peripherals to the equipment of a particular manufacturer: forcing manufacturers to equip their peripherals with a different interface for each brand of computer, or limit their market to a small number of computer types.

Computer manufacturers are now beginning to equip their computers with a standard interface, so that any peripheral supporting that standard may be connected. There are currently two major standard interfaces supported by the computer industry: SCSI (Small Computer Systems Interface) and IPI (Intelligent Peripheral Interface).

SCSI, pronounced "scuzzy", is the most widespread standard peripheral interface. Although the term *small* computer systems is used in its acronym, SCSI is widely used for both small and large computer systems.

SCSI uses an 8-bit data path, and has a maximum transfer rate of 2.5 MBytes per second in asynchronous mode, or 5 MBytes per second in synchronous mode. Unfortunately, there is a lot of leeway in the SCSI standard, and many manufacturers have been taking advantage of this. Consequently, a mass storage device that implements SCSI may not always work on every computer system with a SCSI interface.

SCSI is about to be replaced by SCSI–2. This new interface can use a wider data path (up to 32 bits), and has a much faster data transfer rate. SCSI–2 is also downwardly compatible with SCSI.

IPI has been around for only the last five years. It uses a 16-bit data path, and can transfer data at up to 10 MBytes per second. In spite of this seemingly high performance, IPI is not as widely used as SCSI. IPI is used primarily by manufacturers like IBM, and is sold to captive customers rather than to OEMs. IPI is used in fewer than 1% of disk drives sold today.

As more computer manufacturers switch to standard interfaces such as SCSI and IPI, the goal of "plug and play" peripherals will become more of a reality. Hewlett-Packard, for example, has recently announced that their new MOD drives will be equipped with a SCSI interface instead of HP-IB (Hewlett-Packard Interface Bus), which has been a HP staple for years. This is the first step in their effort to move toward a more standardized interface.

Comparing the Technologies

With all technologies, there is some trade-off between speed, capacity, cost, and use. Figure 1 compares the Burst Transfer rates for the technologies discussed so far.

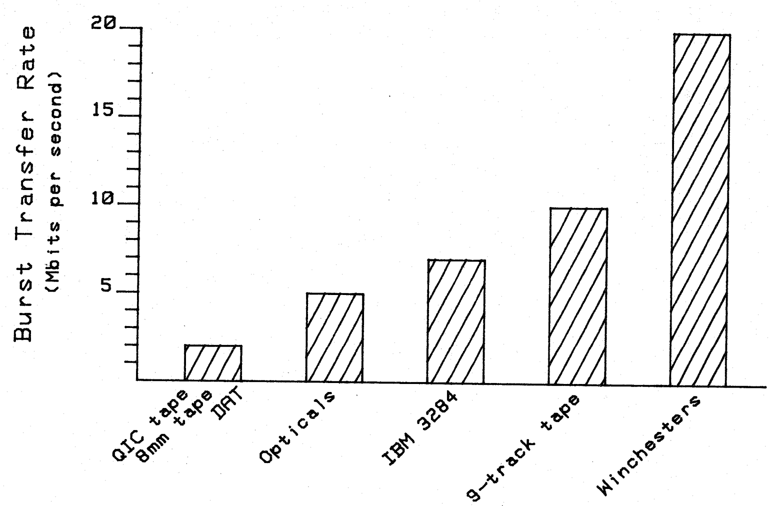


Figure 1: Comparison of Burst Transfer Rates

Figure 2 below shows a range of average access times for Winchesters, Removable disk media, Optical Disks, and Tape drives.

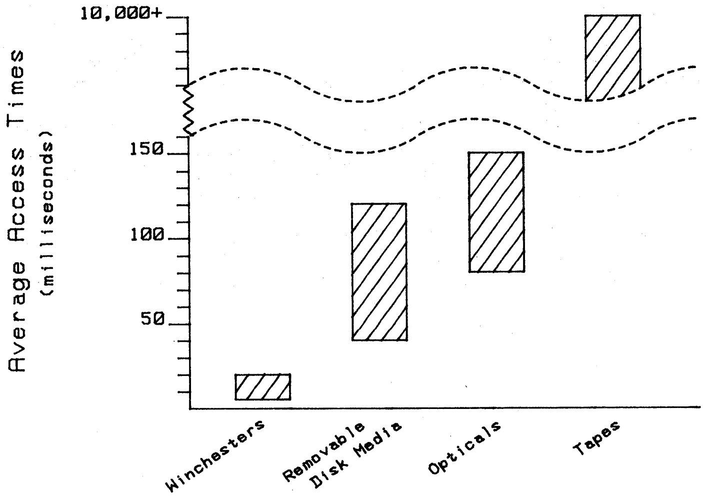


Figure 2: **Comparison of Average Access Times**

Figure 3 presents an analysis of typical storage costs (per MByte of data) for a variety of mediums. The costs shown were figured using the costs of the media only: drive costs were not included.

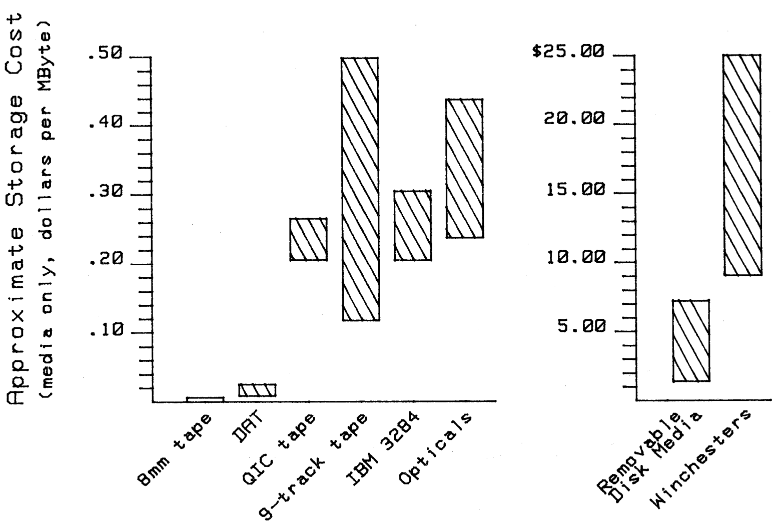


Figure 3: **Cost Comparison for Storage Methods**

Looking to the Future

Helical-scan tape has started the 9-track tape drive on the road to obsolescence. In the coming years, all manufacturers will be using one of a few standard interfaces for their computers, which will make everyone's lives easier. Helical-scan tape, already gaining in popularity, will be the medium of choice for backup and data interchange tasks.

The WORM drive will be supplanted by the more flexible erasable optical disk drive, except for a small niche where data permanence considerations are crucial. Also, the cost of optical media should drop dramatically as more manufacturers enter the market.

A major innovation to look for in the future will be the widespread use of integrated mass storage systems. These systems use a combination of Winchester, optical, and tape technology to store files: where they are stored depends upon the frequency of their use. A particular file will automatically migrate from fast on-line Winchester storage to slower optical and tape systems as the frequency of its use decreases.

Finally, as optical and helical-scan tape systems are refined, their cost, storage capacities, reliability, access times, and transfer rates will all improve.

An IBM Screen on Your HP Terminal

Carlos da Roza
Canadian National Railways
1100 University Street, 4th floor
Montreal, Quebec H3B 2G7
(514) 399-7025

INTRODUCTION

Living with the IBM

Canadian National Railways is an IBM shop. We have some three thousand IBM peripherals connected to five IBM 308x mainframes located in two sites 1500 miles apart. The terminals are spread from coast to coast. All of this is linked together under SNA. We have a staff of about 800 in Information Systems developing, maintaining and controlling the network and its applications.

In comparision, we have a staff of 40 performing the same task for six HP/3000 computers servicing 400 peripherals also spread out across the country with the CPUs located in the same two computer sites. With an IBM to HP ratio of 9 to 1, it is safe to say that the HPs will not displace the IBM systems as the repository of the corporate database.

As our HP network grows, requests to connect with the corporate database grow more frequent. Until now, we have been doing this by building a transactional transport facility between the two networks. We have gone through two such facilities, but still the desire to integrate the systems grows. Each type of connect means writing code to use the transport system to retrieve and display information. We will never be able to keep up with the changes that the IBM systems force upon us as they evolve. What we find we really need is a way to get that IBM screen and put it on every HP terminal we have in the field.

Eliminating the Second Terminal

There are several ways to get IBM access. The most obvious way is to put a real IBM terminal at the field workstation. Each workstation will then have two terminals. Each workstation cluster will also have two printers. All circuits are duplicated. The distasteful aspect of this solution, other than the cost involved, is that at each workstation, one of the two terminals will remain idle at any time.

Another solution is to use some hardware to permit an HP terminal to operate as an IBM or an HP terminal. This saves on terminal and office costs, but does nothing about the duplicate communications circuits. It also means having to outfit every workstation with some new hardware.

HP and other vendors have software which will allow IBM access. HP's offering, IMF (Interactive Mainframe Facility), is meant for occasional access and is CPU and cost intensive. Third party software solutions tend to be quite steep in price.

Evolving Standards - The OSI Model

On the positive side, the industry is moving towards standardization of communications. The Open Systems Interconnect (OSI) model has been built splitting communications into various levels, each level being responsible for certain aspects of the link, and each level resting on the foundation of the lower levels.

The down side of this model is that standards are still evolving. No generally accepted standards are yet defined for the upper levels of the model, particularly those governing session, presentation and application information. It will be several years yet before we have manufacturer-independent devices on the market for full interconnect.

THE PASSTHRU FACILITY

What we are building at Canadian National is a facility which will allow our four hundred terminals to continue serving our HP systems, and yet have the capability to connect with the IBM systems and get IBM screens. After we achieve this, we intend to integrate our software so that our users see a seamless whole when they sign on to a multi-system application. When they need information residing on the HP, they will get an HP application screen. When they need mainframe information, a function key will get them to the proper IBM display screen. In many ways, this facility will operate similarly to HP's IMF, but there will be several significant differences.

Architecture

The pass-thru facility consists of a trio of programs which runs only on the HP. The main pass-thru program conducts a session between the HP terminal and an ATP (Advanced Terminal Processor) port to which is connected a commercially available protocol converter. The protocol converter performs most of the conversion work. The pass-thru program directs traffic, responds to a controller program's directives, and performs some transmission optimization. Each active IBM session gives rise to one pass-thru program process.

The pass-thru controller is a batch process which manages the various sessions, acquiring and disconnecting ports and terminals, responding to the pass-thru console's requests, reporting status and exceptions to the pass-thru console (and log file), and assuring that the facility is functioning.

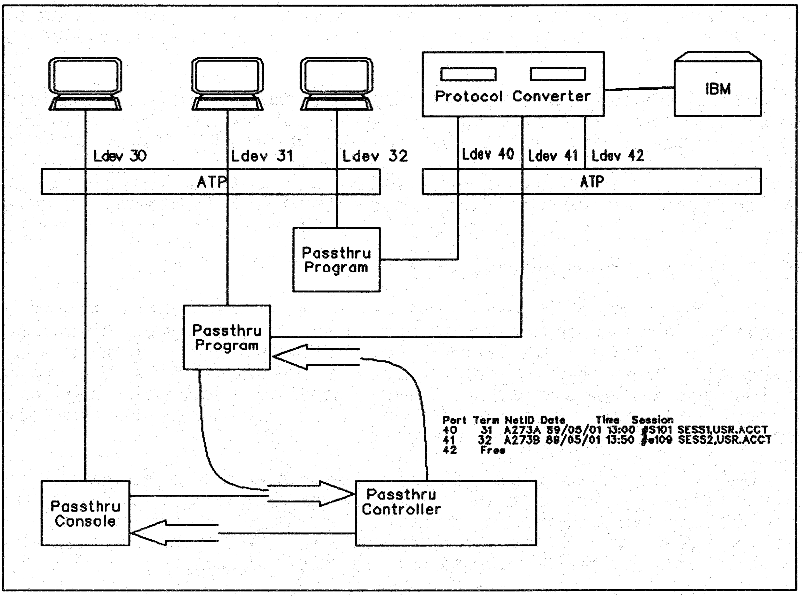


Figure 1: Architecture

The pass-thru console is a program which interacts with the pass-thru controller. Through it, the facility can be launched or disabled, directives to disconnect or connect sessions can be sent to the controller, and status of the facility and of each session can be requested. Our controller also connects to the other various subsystems we have so that virtually all of our systems on our full network can be controlled from this point.

Eliminating Real-Time Processing

The protocol converter we are using acts as an IBM 3270 controller. It manages the SDLC (Synchronous Data Link Control) line, and has up to 32 asynchronous ports to which it expects to have ASCII terminals connected. It has a variety of customizable parameters, particularly those specifying which characters are used to control the terminal.

By having the converter manage the SDLC line, the heavy high priority CPU processing is offloaded from the HP/3000. In fact, the pass-thru facility operates completely in standard CS queue. Most of the translation to and from HP-compatible screen control escape sequences is done on the converter. It operates in block mode and will send to the ATP port in blocks generally no greater than 3000 characters, representing one screen. It also typically receives blocks no greater than 200 characters from the ATP port.

The protocol converter will perform some transmission optimization. For example, it will not transmit the entire screen just because a field has changed. Nevertheless, further optimization is performed by the pass-thru facility. A typical transmission of a full screen to the HP terminal consists of about 1000 characters. Screen field updates usually require no more than 300 characters.

<u>The Passthru Program's Design</u>

The pass-thru program opens up a port to the terminal several times, once for write, and one or more times for read. It does the same for an available port to the protocol converter. It opens a message file for input (this serves as a channel for commands from the pass-thru controller) and one for output (for messages to the controller). All opens are no-wait.

Space for two maps of the IBM screen are reserved and initialized. One serves as a pre-processing map (what was on the screen before processing an input data stream) and one serves as a post-processing map (what the screen has evolved into after processing the data stream).

Reads are posted on all input files, then it waits for the first read to be satisfied. When an input stream comes from either the HP terminal or the protocol converter, a copy of the pre-process map is copied to the post-process map and the input stream's data are used to change the post-process map. At its conclusion, a comparison is made between the two maps, and an optimized output stream is generated to be sent to the opposing port, i.e. input from the protocol converter generates output to the HP terminal and vice-versa.

Input from the control queue is a command from the pass-thru controller and acted upon, with results reported to the report stream to the pass-thru controller.

All required reads are reposted after processing.

<u>Data Overrun</u>

Occasionally, the protocol converter will send more characters than expected in an input stream. This will cause the read posted on the protocol converter's port to complete before the end of transmission and data will be lost, as it is not possible for the pass-thru program to process the data and repost the read before the next character is sent by the converter. Even at high priority, this is not possible as the stack and/or code may have to be swapped in from disk. To eliminate this possiblity, several reads are stacked back to back on the port.

As the I/O system sees a read completion, it handles it, storing the data in its various terminal buffers (TBUFs), and upon seeing another I/O (read) pending on the same port, immediately enables it. Since the pass-thru program

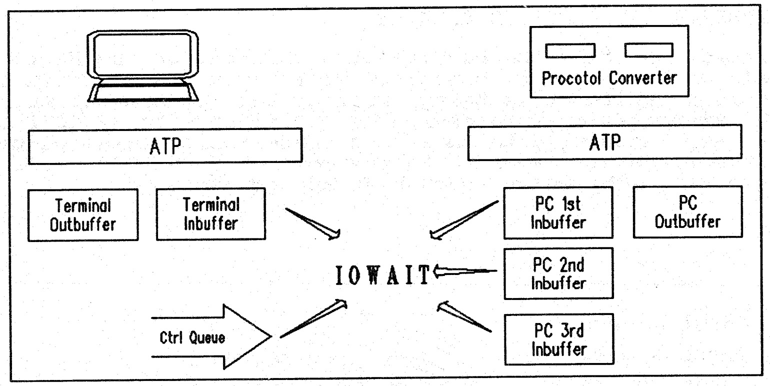


Figure 2: No-Wait IO

is not involved in this process, no swapping is required and no data is lost. The system can take its time in reallocating the pass-thru program in memory to handle the first read completion while the ATP is gathering the remainder of the message.

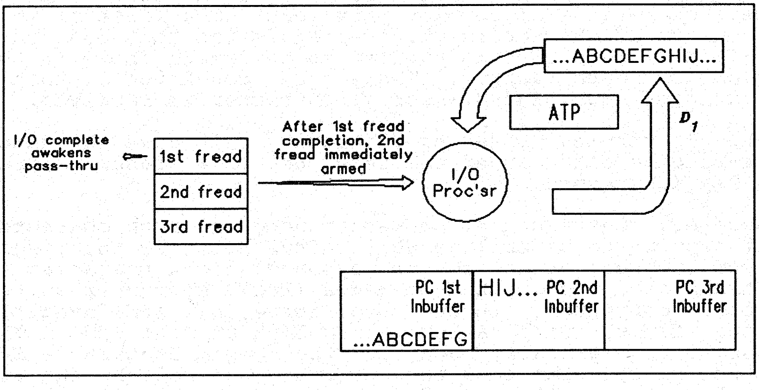


Figure 3: Stacked reads prevent data overrun

The pass-thru posts three reads on the protocol converter and one read on the HP terminal (the HP terminal is far more predictable in the volume of transmission - it never exceeds 1920 characters).

Preventing Input During a Write

There is still the possibility that either the HP terminal or the protocol converter will begin transmission of a screen while it is being written to. To prevent this, each transmission to a device is prefixed with an XOFF to halt any incoming data while the transmission takes place. As the reads are reposted to the port, the HP/3000 sends a DC1 (a.k.a. XON) which re-enables any transmission.

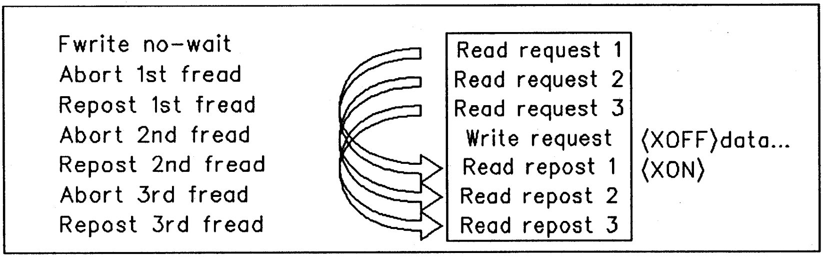


Figure 4: Preventing input during a write

Read Triggers

The HP/3000 ATP port requires one of two conditions to be met before it will complete its read. The first is that the number of characters received satisfies the read count specified. This case is handled as described above in the section on data overrun. The second condition is that a read trigger (EOR: End-Of-Record) character is received.

The HP terminal poses no difficulty. It uses the usual block-mode handshake when the Enter key or a Transmit function key is pressed.

However, the protocol converter sends no such character as it expects to control an HP terminal hot. To force such a read trigger, we define the keyboard unlock character to be translated to an End-Of-Media (EOM) character on the protocol converter. In our experience, all IBM formatted screens finish with a keyboard unlock at the end of the transmission (probably due to the Begin Bracket - End Bracket pair). The pass-thru program just defines the EOM character as an alternate EOR on the protocol converter's port to ensure that the read is triggered.

The Status Line

The status line is also transmitted by the protocol converter. The pass-thru program keeps it in a separate area and sends it to the HP terminal upon reception on the last line while the keyboard is locked, i.e. during transmission, or while input is inhibited due to error. When the reset key is hit, or as soon as keyboard entry is enabled, the status line is wiped from the HP terminal and the last line of the screen is redisplayed.

There is a minor problem we have encountered. The protocol converter sends a status line as the last line of a transmission, after the keyboard unlock is sent, and it has proven difficult to configure the protocol converter to alter its status line. Nevertheless, there are various possibilities still open, including using the packet switching feature available on the protocol converter to force a data forwarding character to be transmitted as the last character. In any case, it is not a serious flaw preventing the pass-thru's operation.

Field Definition Differences

An interesting difference between HP terminals and IBM terminals is that on HP terminals, unprotected fields terminate at the end of the line. IBM terminal fields may span several lines. This difference must be taken into account when building output streams to the HP and the protocol converter. For example, a multiple line field sent from the IBM will force the generation of several unprotected fields on the HP terminal, each starting from column one. Upon reception of these fields from the HP terminal, the fields must be concatenated before transmission to the protocol converter.

PF Keys and PA Keys

There are quite a few more function keys on the IBM terminal (which IBM refers to as Attention IDentification (AID) keys) than on the HP terminal, particularly the PF (Program Function) and PA (Program Attention) keys. To accommodate that, the pass-thru program assigns one HP function key to the PF and PA keys, and when that key is struck, it prompts for key number on line twenty-four, overlaying the display. After responding to the prompt, it redisplays the last line.

The other HP function keys are mapped to Clear, Attention, Reset, etc.

Block Mode on the HP Terminal

Block mode is used on the HP terminal. When the IBM screen is formatted (has protected and unprotected fields), block page mode is used, otherwise, block line mode is used.

Block page mode is straightforward. The HP screen is put into formatted mode and all unprotected fields are transmitted (more on this in the next section). In block line mode, only the information since the last cursor position is transmitted. For some HP terminals, a non-displaying character may be laid down on the screen, and when enter is struck, all characters from the last non-displaying character to the current cursor position are sent. For other HP terminals, this search for the last non-displaying character only takes place up to the beginning of the current line. And yet for others, the entire current line is transmitted.

To resolve this, the pass-thru program detects the terminal model used and, remembering the last cursor position, takes whatever steps are necessary to retrieve the input data.

Modified Field Tags

Some HP terminals permit modified field tags. Only fields that have been changed are transmitted. If the HP terminal being used permits that feature, the pass-thru will use it. This eliminates unnecessary transmission.

A Last Word on Screen Optimization

While building an HP screen, the pass-thru attempts to use the shortest sequence of characters necessary to travel to the location of the screen it needs to modify. Sometimes, the best way is to wipe the screen and repaint it. Sometimes it is better just to wipe one or several lines, and finally it may be quicker just to address the field and overwrite it. It also takes into account the possibility that the use of a carriage return, line feeds and/or a cursor control sequence may be quicker. All of these calculations are done on stack and are quite fast and do not degrade performance, particularly when compared to the limited throughput of a circuit.

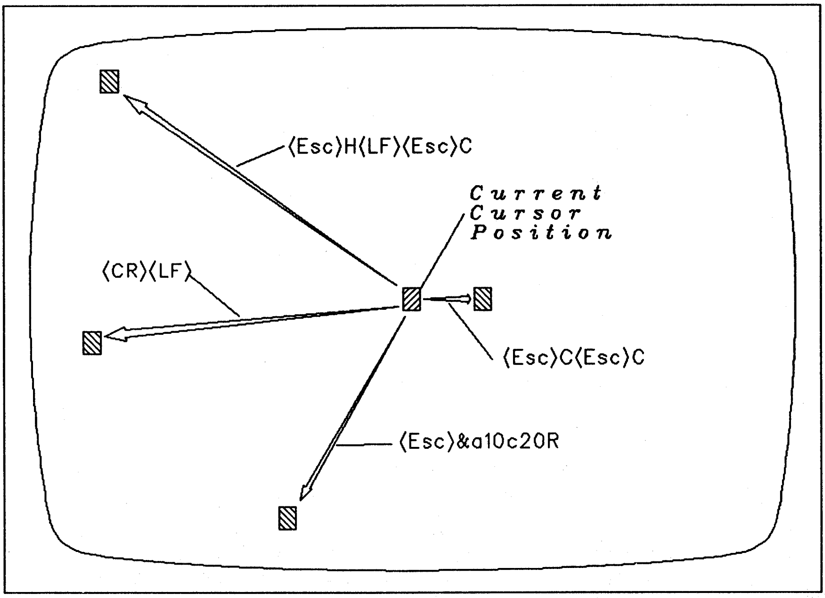


Figure 5: Traversing the screen efficiently

CONTROLLING THE PASSTHRU FACILITY

The Controller

The controller was written to control the pool of ATP ports connected to the protocol converter. It also serves to check on the viability of a session and to provide control services to a pass-thru console. It runs in background as a job under standard CS priority.

The pass-thru console initiates the facility by launching the pass-thru controller job. The pass-thru controller reads its configuration table to see what ports are to be used as pass-thru ports, and establishes communications with the pass-thru console by using a pair of message files. It also dynamically acquires OP capability and REFUSES every pass-thru port.

The pass-thru program, upon invocation, sends a demand for access to the controller through its input message file. The controller allocates a port and sends the information back to the pass-thru which initiates the session. Upon establishment of the session, the pass-thru will relay session information such as IBM LU and status back to the controller, which updates its own tables, flagging the pass-thru port as acquired. Upon termination, the pass-thru also forwards appropriate information to the controller.

```
PASS-THRU STATUS REPORT AT 89/05/01 14:02
=========================================
Port Term  NetID  Date     Time  Session
---- ----  -----  -------- ----- --------------------
 40   31   A273A  89/05/01 13:00 #S101 SESS1,USR.ACCT
 41   32   A273B  89/05/01 13:50 #S109 SESS2,USR.ACCT
 42  Free
```

Figure 6: Pass-thru status report

Commands received from the pass-thru console are interpreted and executed with results returned to the console. Commands include those to begin and end a trace, reset a session, request status on one of all of the sessions, and to shut down the facility.

The Heartbeat

One operation that the pass-thru controller performs is that of monitoring the viability of the pass-thru session. Every minute, it polls the sessions it believes are alive. If no answer is received by the next poll, it presumes a lockup and clears the port, eliminating the session.

```
PASS-THRU DETAILED REPORT AT 89/05/01 14:03
===========================================
Passthru Port: 40              Terminal Port: 31
    CRT Model: HP2624B                Net ID: A273A
      Session: #S101 SESS1,USR.ACCT   Status: X WT
Sess Acquired: 89/05/01 13:00
               Time     Data
   Last Input: 14:02:09 PACSONL
  Last Output: 14:02:05 SYSTEMES D'INFORMATION DU CN
```

Figure 7: Detailed pass-thru status report

Port Pooling

By using a pool of ports for the pass-thru facility, HP terminals may request a port and be attached to any available ports. This makes for relatively efficient use of IBM LU's. The limitation is that a specific LU access cannot be guaranteed. There are future plans to enhance this facility to permit port grouping.

ADDITIONAL FACILITIES

Multipoint Terminal Systems

The current pass-thru facility works on HP terminals connected via ATP. A majority of our HP facility uses MTS. This is an integral part of the pass-thru facility which has yet to be built. There are some differences, particularly in controlling the terminal, but they do not seem insurmountable.

Alternatively, we will likely to be going to X.25 as a terminal network. This will present a different yet similar set of problems.

Scripts

One facility which we intend to incorporate is that of execution scripts. These will permit a series of predefined actions to take place upon session acquisition. Logging on and navigating to the proper screen would be greatly enhanced as would automatic security clearance. The degree to which these connects will be automated will depend on user demand.

Hot Connects

Another useful feature planned is that of having a hot connect into a running session. If a particular IBM session and screen is required frequently by a group of HP applications, it would be far more efficient to have this screen logged on and enabled. An HP application requiring this screen would attach onto this port directly, perform its function, and release upon completion.

Timeouts

To prevent negligent users abandoning their screen and occupying a pass-thru port, the controller would monitor session activity. The heartbeat transaction reports the time of last input and output for that session. After a configurable idle period, the controller would ask the pass-thru to perform a disconnect with the necessary logoffs, leaving the IBM LU in a normal disconnect state.

Forced Disconnects

One last feature we would incorporate is that of forced disconnects. Upon receiving a command from the console, the controller could ask the pass-thru to perform a timeout disconnect. This could be automated so that when a higher priority request for IBM access is made, the lowest priority session existing, or the one with lowest activity could be bumped off after the requisite warnings have been sent. An extension would queue all such requests and notify the user when a port comes free.

COST COMPARISONS

Direct Hardware Costs

Duplicating the workstation hardware runs into the thousands of dollars. An IBM or clone terminal costs upward of $500. Each workstation would also need a portion of the IBM 3270 controller (1/32nd) which runs into the tens of thousands. This cost would be replicated for each workstation needing access. Communications costs depend on distance and bandwidth, but they are significant.

Indirect Costs

Some of the costs usually not accounted for are those indirect costs such as added office space, inconvenience to the worker and the costs of idle resources. Certainly, given two terminals in a single workstation, there will be a high incidence of one of the terminals sitting idle at any one time.

Alternative Software Solutions

Canadian National issued an RFP (Request For Proposal) in 1988 to integrate the HP and IBM network of terminals to about twenty vendors. The quotations that came in ranged from $110,000 for 32 port access, to $5 million for full interconnect. In every case we found the cost/benefit ratio too high to be palatable.

Passthru Costs

The pass-thru concept relies on the existing physical plant with a commercially available protocol converter as the only additional piece of equipment. A 32 port unit costs $16,000, resulting in a unit cost of $500 per concurrent access. ATP costs were not included in any of the estimates. With a minimum cost ratio of 7 to 1, we decided to pursue this avenue which, although it does not give us full bidirectional access, does solve half of our problem at an acceptable cost.

CONCLUSIONS

At this point, we have a working model of the pass-thru using ATP ports. Development costs have been minimal, four man-months have been expended in coding and testing, and the hardware costs have been partly offset by additional projects sharing the protocol converter. At this time next year, we should have a fully functional implementation providing our entire HP network with IBM access.

A Moving Experience

Scott A. Moulton
PPG Industries, Inc.
95 Columbia Court
Barberton, Ohio 44203
216-848-4161

Introduction

One of the challenges of computer systems management is moving your site. It is difficult enough just to move a computer room, but add to that moving an entire R&D facility. The information presented is from the view of a computer system manager. Topics cover pre-move planning, what was learned from the move and what could have been done differently.

PPG Industries, Inc. is a diversified manufacturing corporation with 1988 sales of $5.6 billion. PPG is a leading global producer of flat glass and fabricated glass products, continuous-strand fiber glass, original and refinish coatings, and industrial and specialty chemicals. Products of the Pittsburgh-based company serve manufacturing, building, processing and numerous other world industries.

PPG operates 73 major manufacturing facilities throughout the world. In addition, the company conducts research and development at 12 facilities worldwide. PPG has four business groups, Chemicals, Coatings and Resins, Glass and Biomedical. Each of these groups has Research and Development facilities. Up until now, the R&D facility for Chemicals was located in Barberton, Ohio, which is 40 miles south of Cleveland, Ohio. The decision to move Chemicals R&D was announced on December 15, 1988. It is being done in order to improve the synergy with other R&D groups and corporate headquarters.

In January of 1989 we started planning the move of the Chemical R&D Facility from Barberton, Ohio to Monroeville, Pennsylvania, a distance of approximately 120 miles. The new facility was an existing R&D site for another company. The issues addressed were the following: identify what to keep from the previous tenants; identify what would be moved; define when it would be moved; and support both locations for an interim period.

The project included moving 90 people in offices, laboratories, a library, and two computer rooms. The equipment moved included an HP1000A analytical system, 75 personal computers, lab equipment, furniture, and files. Finally, there was a migration from an HP3000 supporting Office Automation to a DEC VAX 3400 and PC based solution.

Requirements

The requirements for the relocation were simple; they were as follows:

1. Only provide for current capabilities.
2. Minimum amount of disruption.
3. Relocation done over a two month period, August - September.
4. Keep changes to a minimum, thereby reducing the possibility of problems.
5. Provide maximum amount of uptime for computer systems.

Equipment

The computer center consists of two separate computer rooms housing an HP3000, HP1000A and an HP1000E. In addition to the central computers there is a Micom 6600 data communications switch, 60 IBM compatible PC's and 15 Apple Macintosh computers.

HP3000

Hardware: Model 68
5 Mbyte memory
3 7933 disk drives
1 7935 disk drive
1 7978 tape drive
96 ATP ports

The HP3000 supports Office Automation. It runs HPWord, HPDesk, HPList, HPDraw and EZChart. Also, it provides access to our corporate IBM via SNA and NRJE software. Active users total approximately 15 daily sessions and 120 authorized users. The greatest use of the HP3000 is from our secretarial staff.

HP1000A

Hardware: Model A900
3 Mbyte memory
1 7912 disk/cartridge tape drive
1 7933 disk drive
1 9144 cartridge tape drive
3 12040 MUX cards
3 18651 LAS loop controller cards

The HP1000 is an analytical system used for data acquisition in chromatography applications. It runs the LAS 3350 software from HP along with STATIT and GRAFIT from Graphicus. Communications for this system is composed of three MUX cards supporting 24 RS232 ports and three loop controller cards supporting up to 45 instruments connected through A/D converters.

HP1000E

This system was the previous analytical system. It has been installed since 1981 and is being phased out of use. In addition to the LAS 3357 software, it also supported physical testing software from Instron. In February of 1989 the LAS application moved to the HP1000A and due to the relocation the Instron physical testing software is now supported on a PC based system.

Micom 6600

The Micom is a data communications processor which is similar to your phone system only for data. With it we are able to access any computer from the same terminal or PC. Terminal and computer lines are connected to the Micom providing access on an as needed basis. This gives each terminal or PC connectivity to any computer.

Personal Computers

Throughout the facility there are a variety of IBM PC's, IBM compatibles and Apple Macintosh computers. The total is approximately 75 computers. They are used primarily as stand alone devices. However, they also provide communications through terminal emulation software to the HP3000 and HP1000's.

The Old Facility

The old facility was initially spread out among four buildings. Due restructuring we reduced that down to two buildings. All communications was via RS232 wiring. The Micom data communications processor was used to provide high speed transmission from remote locations back to the computer rooms.

The personnel moving in the first phase all reside in one building having three floors and a basement. The computer rooms were located in the basement and the labs were on the main floors. Each office and lab was wired to support at least two RS232 ports.

All scientific personnel have their office in the lab, therefore their terminal or PC was also located inside the lab. The space allocation was well planned and access to the building was easy.

The New Facility

The new facility in Monroeville, Pennsylvania is actually an existing research facility previously occupied by Koppers. There are five buildings connected by glass enclosed passageways. Each building is classified as a wing, designated A through E.

Wing A is a four story building which houses the administrative offices. A typical office module is 10 feet wide by 15 feet deep. There are 69 offices in this building not including the executive office area on the top floor.

Wings B, C, and D are the three laboratory wings. Each wing consists of laboratory and office modules on opposite sides of a common corridor. There are 72 laboratories and 144 offices in the three wings. The typical laboratory module is 20 feet wide by 28 feet deep, while the typical office module is 10 feet wide by 15 feet deep.

Wing E houses facilities such as lobby, auditorium, cafeteria, library, and conference rooms.

Preparation

Old Facility

One of the first steps in the preparation process was to identify special equipment and its requirements. For the computer group it meant identifying each piece of computer equipment and the following requirements: electrical, dimensions, weight, and heat dissipation. This is an excellent way to provide a complete list of requirements to each of the contractors.

After identifying the computer equipment, we then defined the amount of office space needed. Being a support group requires an additional amount of space for storage, training and instrument repair. Therefore, never give up space, you might never get it back.

Finally, we identified the items in each area that needed to be moved. This might sound strange, and you might be asking yourself the question, doesn't it all get moved? The answer to that is yes, if you are a pack rat. Moving is a good time to clean house and get rid of the things you don't need.

New Facility

Even though it is a new facility to PPG, the building is 30 years old and required extensive renovation to make it acceptable to PPG. During the renovation process the computer facilities were also upgraded to meet current requirements while laying the foundation for future growth.

When working with an existing facility many of the tasks required for a new facility still apply: computer room layout, power conditioning, air conditioning, security and accident prevention still need to be addressed. The differences are in how they are covered. If you need to perform new or existing facility design companies like HP are more than willing to help with this task. For us, there was already a computer room, and to build a new one was not cost effective. Therefore the location stayed the same, however we were able to expand the size.

The computer group's key item of interest was the wiring. In early February we began defining what we required in the new facility. The way we narrowed it down was to get input from a variety of people. We did this by walking the facility with PPG Corporate Communications, Hewlett-Packard, Digital Equipment and an independent consultant.

The consultant toured the facility in mid-March. Since he does a large amount of work for PPG his opinion carried a large degree of credibility. Also, we were able to have him take responsibility for having the wiring done. This included preparing a request for quotation and evaluating the bids once they were received. Wiring bids were let on June 7, with responses due back by June 14. Afterward a selection was made and work began the week of June 26. We found having this task done by a consultant freed us up to concentrate on other activities.

After evaluating their recommendations we identified twisted pair wiring from the closet to the office as the wiring of choice. Two bundles of 4 twisted pair each were run to each lab and office area. This provided us wiring for voice and data communications. Once inside the closet they connected to a multiplexer that sends the data to the Micom 6600 located in the computer room.

The wiring presented a challenge due to the layout of the new facility. There needed to be more wiring and it was spread out over a greater area. In addition, the cable trays were small and inadequate for holding the amount of wire we needed. Therefore conduit was run along the walls to carry the wiring to the labs and offices.

Since one of the requirements was only to provide current functionality, we could not address a LAN without separate justification. Our budget did not provide for this expense. Therefore, with the advent of twisted pair for many new LAN's our approach would provide us with the foundation for migrating from RS232 to a LAN in the next 6-18 months.

In addition to the regular wiring we also had to consider two additional wiring needs. The first was the LAS loop wiring and the second a Novell network based on Arcnet. The latter is used for accounting purposes. Both of these are separate from our twisted pair wiring. This was done to reduce the number of changes made in the move. In the future we look to merge the Arcnet and twisted pair into one 802.3 type network accessible by everyone.

Moving Time

After the announcement of the move on December 15, 1988, we had roughly seven months to complete all planning and scheduling. The move of the R&D facility was to occur over a two month period, from August to September, 1989.

The first people to move were the administrative groups on August 8. Packing of these two areas was done over a two day period. The materials were then shipped and finally unpacked. The whole process covered approximately 5 days.

The computer group was also scheduled early on in the time period, August 18. The move of the computer equipment was done during this time frame. The computer group took responsibility of the central systems and data communications, while the individual was responsible for their terminal or PC. Backup of PC's was done to the HP3000 and when the PC was brought up at the new facility, the backup was purged from the HP3000.

Our secretaries migrated from the HP3000 based office automation tools to PC based systems. Files were identified on the HP3000 that would be needed on the PC's. Then, they were transferred to disk and stored on their respective machine. Most of the secretaries did not move. Therefore new personnel received training and assistance at the new facility.

Other groups will continue to move through the first part of October. Once complete, we will have moved approximately 90 people and hired an additional 30-40 people.

Summary and Conclusions

The computer group consisted of three people. One chose not to relocate but is still with the company through the duration of the move. Due to the amount of time required to plan the move, other activities will take a back seat or you will find yourself putting in a large amount of extra time. For the first five months the move occupied 35% of the groups time. The last two months that number increased to about 75%. During the August to September time frame all of our time was dedicated to the move.

As of this date the move is still on-going. The completion should be around mid-October. The most important task that can be done is to plan and be flexible. Proper planning produces better results. Document everything and know what will be done, who will do it and when it will be done. Word of mouth and memory are fallible.

In conclusion, even if you are an optimist, which I am, build into your schedule additional time. Challenges will arise and delays are always possible. The most important thought to carry throughout the whole process is, be willing to do whatever it takes to get the job done.

If you attend the session, updated information and handouts will be available.

Using MPEX to Solve System Management Problems
Five Case Studies

Bill McAndrew
Donnelly Corporation
414 East 40th Street
Holland, Mi. 49423
(616) 394-2405

MPEX is a utility program sold by Vesoft, Inc. of Los Angeles, California. Many HP3000 shops are using MPEX to solve a variety of problems ranging from the altering of a single file to the global management of all files on their systems. If you are not familiar with the MPEX package, suffice it to say that it is an extremely powerful utility that addresses deficiencies in several areas of MPE. You can think of it as a dozen of your favorite contributed utilities rolled into one program.

Vesoft provides complete documentation with the package and there is an online help facility in the newer versions that is truly excellent. However, despite the 300 or so pages of documentation (or perhaps because OF it), it is quite easy to miss some of the "finer points" of the package. Even though the author covers each command exhaustively, and tries to provide meaningful examples of each command in action, it would be impossible to document every possible usage of them. It is quite easy to miss some of the "hidden beauty" in the package.

This paper is not a reference manual on MPEX, nor is it a tutorial. It is simply a collection of some unique applications of the products MPEX and STREAMX. A basic understanding of these programs will be necessary to fully understand the five case studies. The cases are arranged in order of increasing complexity.

All five case studies have been included on the San Francisco swap tape under the name MPEXSTDY.

CASE 1: PURGEWRK

Problem Description:

Several different systems running on our computers use program created workfiles, usually used for recovery purposes. Over time, these workfiles can accumulate due to aborts, wasting a lot of disc space. Since they weren't explicitly created by the users, the users typically don't clean them up. I want to find these files and delete them weekly before our full backup. A complication is that some of the workfiles have a file code of zero, making it difficult to distinguish them from user files.

The Solution:

The jobstream PURGEWRK provides the needed solution. It is listed in Appendix A. As you can see, it goes a little beyond the normal "K file cleanup" job. This job is streamed automatically each week before our full backup and typically reclaims between 10,000 and 30,000 disc sectors.

Important points in the example:

Notice that the bulk of the work is done inside an MPEX USE file called PURGEWRK.SE.SYS. This is done to be able to use question marks freely in the fileset specifications. The jobstream will be streamed using STREAMX, which interprets question marks as a parameter prompting and substitution device.

The EDITOR step in lines 10/18 of the job creates a small user template that is used later on in lines 36/37. The template takes advantage of the target fileset ability of MPEX. The graph data files need to be purged, but they have a file code of 0 and no other "identifying marks" except for beginning with DF and DL. The user template finds and deletes these files by finding their graph file counterparts first. These CAN be identified by file codes and are used to point to the corresponding data files.

CASE 2: LOCKACCT

Problem Description:

A mechanism is needed to temporarily shut out all users from a given account while repairs or updates are taking place, or at least give them a customized warning message before allowing them to proceed. System Welcome messages are fine, but some users don't seem to read them or heed them. Welcome messages are also not account specific. If locking out users, the facility must allow Systems personnel (people with AM capability) to freely access the account while keeping other out.

The Solution:

The problem is solved by the jobstream LOCKACCT, which is shown in Appendix B. It uses STREAMX to prompt for several variables used by the stream and accomplishes the lockout using a little "MPE programming".

Typically the account manager for the problem account streams LOCKACCT and specifies either a custom message to be displayed to the users, or the default message. The jobstream will construct a special lockout UDC, which differs according to whether you are locking the account or simply displaying a message to users, and puts it in its own UDC file. It then catalogs this file along with the normal account level UDC file. The lock/message can be applied at any time; existing users in the account are allowed to complete their sessions, but all new signons are subject to the LOCKUDC. The lock/message is removed by simply doing a SETCATALOG command with the normal account level UDC file(s).

Important points in the example:

STREAMX makes the prompting for parameters very easy and user-friendly. The ECHO command allows you to be very complete when prompting for input, and even allows the variable defaults to be displayed (for example, line 52 displays the actual default to be used in this run of the program). Sophisticated editing of accounts (lines 54/56), user ids (lines 61/63), and file names (lines 155/158) can be done.

A message file of record length 72 should be used. Line 194 prepares the output file for this. A record length of 72 is the default for an unnumbered Editor file. The record length of the message file is checked in line 157. The message file will be displayed using FCOPY, and unless the record lengths of the input and output files match exactly, a confusing warning message will be issued to the user trying to sign on.

Lines 192/209 make up the UDC that will be executed. STREAMX intercedes at line 199 to conditionally add the code to bye regular users off if that option is chosen.

The lockout is accomplished by seeing if the user can successfully execute a LISTUSER command (lines 200/205). No output is generated because of the $NULL in the command, but the CIERROR JCW is set appropriately.

At the tail end of the constructed UDC is a call to an ACCOUNTLOGON UDC. Since this UDC may or may not exist, there is a continue statement before it. As its name would imply, this UDC handles all logon UDC processing at the account level. Because LOCKACCT works as an account level logon UDC, and is cataloged first, it would disable any existing account level logon UDC. To correct this, LOCKACCT issues the call to this UDC name to do any further account level logon processing, should the user make it past the account lock. This means that account managers must name their account level UDC ACCOUNTLOGON in order to use LOCKACCT. This has never been a problem as these UDC names have always been arbitrary in the past.

The code in lines 213/240 keep the newly constructed UDC in either the file LOCKUDC or LOCKUDC2. This handles the case of having an existing message or lockout in place when trying to issue a second one. Current users may have one of the files locked, but the other should not be accessed. The stream can't handle the case of more than two different messages being active at the same time. STREAMX also handles the situation of no regular account level UDC file in lines 227/231 and 235/239.

Lines 241/246 inform the account manager of the successful completion of the stream. In the case of an account lock, a message is sent to the console operator as well.

CASE 3: QCOMPILE

Problem Description:

Many of our applications are written using PowerHouse, a fourth generation application development system available from Cognos, Inc. An application's screens and reports are dependent on a

PowerHouse directory of files and fields within those files called a QSCHEMAC. Whenever this directory is changed by the addition of a new file, or a field in the Qschemac is modified, all application screens, reports and batch processing runs should be recompiled. Some automated method of recompiling is necessary. One complication is that the program source resides in normal ASCII files (file code = 0) and uses no particular naming convention.

The Solution:

The jobstream QCOMPILE provides an effective solution to this problem. It is listed in Appendix C. Once streamed it informs the developer of its status or its completion. It first recompiles the Qschema source file into a new Qschemac. Next it finds all of the QUIZ report objects in the account and uses them to find their corresponding source files so they can be recompiled. Similarly, the QUICK screens and the QTP objects are also recompiled. Optionally, the source can be listed during the compiling, giving a full listing of the whole system.

Important points in the example:

This jobstream is fairly old (some 4-5 years), but despite its older syntax, is pretty straightforward. Lines 251/257 are an old way of putting alphabetic variable input into a jobstream. Note that until recently STREAMX could not alter the number of lines of code to be streamed, although it could manipulate the contents of any given jobstream record. Because of this, MPE IF statements had to make our jobstreams decisions for us, rather than constructing a dynamic jobstream at stream time.

Lines 260/261 handle the case where the Qschema source did not contain an EXIT statement. Since this jobstream is used on several different applications written by many different programmers, it must accommodate different "styles" of programming. If there was an EXIT in the file, the EXIT on 261 will be an error in the jobstream, hence the CONTINUE statement. On the other hand, if there is no EXIT in the source file, QDD will continue reading the jobstream as input. In this case, the CONTINUE statement is programmatically executed by QDD, which just ignores it. The EXIT statement on 261 then terminates QDD.

The "^[" strings in line 263 and elsewhere in the examples should be replaced by the escape character. Line 263 sends a highlighted message to the streamer telling him that the Qschema file has been recompiled.

The LISTF commands on lines 272 and 274 give a "before and after" picture of the compiled Quiz objects in the account. An intentional byproduct of the recompile process is the deletion of all objects which do not have a corresponding source file. In most cases, these are temporary fix programs, or the object has been named differently than the source. In the latter case, the programmer will find out quickly (either from this LISTF, or from the users) that his object is no longer there.

Line 273 is the key to the example. It uses the MPEX user template (listed in lines 309/339) to do the recompiling. Once again, the powerful target fileset capability of MPEX is used to find the source file by first locating the object files. Locating the object files is easy because of their unique file codes. This eliminates the need for any naming standard on the source files, although the system requires all of the source files to be in the group SOURCE, and to be named the same as the object files. Note that although the source fileset is the target fileset in the USER command, there is no requirement in the template that the files be used in that order. The LIST variable, which controls whether or not the source will be listed was prompted for in line 255, passed to the template in line 273, and used in template line 334.

The template enters Quiz in the start section (lines 309/324), does the compile in the file section (lines 325/335), and exits Quiz in the finish section (lines 336/339). Note that the entire application's Quiz objects are recompiled in one run of Quiz, rather than entering and exiting Quiz with each source. This is much more efficient, but requires that Quiz reinitialize itself before each new compile, hence the statements in lines 329/332. The old object is purged in line 333, and then recompiled (if the source exists) in line 334. This not only cleans up the stray objects as mentioned before, but eliminates any delete confirmation prompts from Quiz.

The remainder of the jobstream (lines 279/307) use the remaining two templates to recompile the Quick screens and the QTP objects using the same methodology as with Quiz.

CASE 4: USEQPROD

Problem Description:

Application development using Powerhouse usually consists of writing code in Editor, saving it, exiting Editor, running a Powerhouse program such as Qdesign, finding an error, exiting Qdesign, running Editor, texting in the source file, make code changes, saving it, etc. A significant amount of time can be spent in just traveling between utilities. A better applications development environment is desired where the editing-to-testing cycle can be shortened.

The Solution:

USEQPROD is an Editor /USE file which quickens this development cycle by using labeled function keys to implement popular key sequences and takes advantage of the MPEX hook to run the Powerhouse products from within Editor. The entire text of USEQPROD is listed in Appendix D. The procedure dates itself by having a name that goes back to the time when Cognos was known as Quasar, and the Powerhouse products were not collectively named. The developer enters Editor once, and executes the USE file rather than /TEXTing in the source file.

After making changes, the file is kept with the /KEEP command. A function key is then pressed which invokes the desired Powerhouse utility by using the MPEX hook to call the appropriate UDC from within Editor. F1 through F5 are used for this. Function keys 6 and 7 have been programmed with USE commands for the Powerhouse utilities, making the compilation a two button operation. If there are errors, the developer exits back into Editor with his file ready for more changes. Typical Powerhouse development can be shortened by about 50% using this technique.

Important points in the example:

Most of the USE file consists of the little used /Q command of Editor which simply echoes strings back to the terminal. This command only seems to be of value when used in /USE files.

Lines 398 and 400 show that comments can be inserted into /USE files if surrounded by double angle brackets (<< and >>). As far as I know, this is not documented in the Editor Reference manual.

The strange Z::= command is used to accept a variable from the user. Editor will accept a variable length string at line 399 which is substituted inside later strings using the Z:: construct. This is how the source file name is encoded into the function keys.

Function keys 1-8 are each set twice using /Q sequences. One sequence sets the key with a function key label, the other without the label. This allows the /USE file to be used successfully with 2645A terminals, as well as newer ones supporting function key labels. The 2645A will abort the escape sequence at the "16d" subparameter, so it needs a separate escape sequence. On the newer terminals, the keys will be set twice.

The function keys are set with the function key setting escape sequence. This is documented in each terminal's reference manual. The "^[" string should be replaced with the escape character. The <esc>&f starts the sequence off. The "2a" indicates that the key should be set to "transmit" the sequence to the host directly. The string will be echoed back to the terminal by the host. This eliminates the need to encode return characters at the end of each key value. The "k" parameter tells which function key the sequence is for. The "16d" parameter indicates that a 16 character function key label will follow at the end of the sequence. The final "L" parameter sets the number of characters in the key value string. Since the L is in caps, it terminates the escape sequence. The 16 character key label follows immediately, and that is followed by the key value string. Note that the key value starts with a "%" which invokes the MPEX hook.

The four Powerhouse products set in the keys, as well as SPOOK, support process suspension and reactivation. This is implemented in Powerhouse by using the SUSPEND program parameter in the UDC call. MPEX supports this process handling environment quite well, and the result is the elimination of the process loading overhead normally associated with the repeated running of these programs.

In lines 412/413 the Powerhouse USE command is set up, embedding the source file name in the key value string. The key value string is set only once, in line 413. Line 413 will execute on terminals with and without function key labels. Since 412 sets the function key label, it only applies to the newer terminals.

Embedding the source file name in line 413 is fairly tricky, as the Z:: string can be of variable length, yet the escape sequence demands an exact length for the key . value string in the L parameter. To make this work, the string containing the Z:: value is padded with trailing blanks. When Editor dereferences the /Q command on line 413, the actual value of the Z:: string will stretch the string out. Only the first 44 characters of the key value string will be used in the function key, however. The "^@" at the end of the string represents a null character (dec 0) that is required to keep Editor or the terminal driver (I'm not sure which) from stripping off those necessary trailing blanks from the sequence.

Function key f8 is set to use the use file again. This allows you to easily switch to a new source file.

The escape sequence in line 420 is used to force display of the user keys on the terminal, in case they were not already displayed.

Lines 421/423 use Editor's extremely confusing method of setting tab stops for the source file. Lines 421/422 use escape sequences to set the physical tab stops on the terminal, while line 423 sets the logical tab stops in Editor and the file. Note that the physical and logical tab settings are completely independent of one another, making it extremely confusing when the two aren't done in concert. The newer terminals have a feature called TAB=SPACES. This feature should be used whenever possible to implement tabbing.

The /TEXT command in line 424 enables the /USE file to be used instead of a /TEXT command.

CASE 5: PSELECT

Problem Description:

In our company, computer users are spread out over many different geographical locations. Each location is served by at least one remote spooled printer, each of which is connected to one of the computers in our network. When a user wishes to stream a job, s/he signs on to the computer that contains the application and streams it. The user also wishes to send the resulting report to the printer at his location, however, this printer may or may not be connected to his computer. A method is needed to allow a user to send jobstream output to any printer on our network dynamically. The user should not have to know which printers are on which computer, nor even be familiar with the device class (or worse yet, ldev number) of the desired printer. From the programmer's standpoint, the method must be easily implementable, and be as free from maintenance as possible.

The Solution:

PSELECT solves the problem by using some advanced features of STREAMX. The code is listed in Appendix E. The programmer inserts two lines of code into his jobstream either once, or as many times as desired to allow redirection of report output. It is possible for each report in the jobstream to be directed to a different printer. The two lines inserted will cause the streaming user to be presented at stream time with a menu of printer choices. The user selects one of these printers and the jobstream continues. At this point a STREAMX variable called PRINTER contains the last half of a file equation for the selected printer. The Printer variable can then be substituted in a file equation under any formal file designator. The specifics on printers (computer node names, device classes, environment files, etc.) are contained in a file called PRDATA, which is accessed globally from the SYS account.

Important points in the example:

As shown in lines 430/433, only a few additions are necessary to a jobstream to implement PSELECT. The ::PSELECT command is the crux of the whole example. In the STREAMX documentation it is mentioned that any MPE command can be executed at stream time by preceding it with a "::". What is not mentioned, however, is that your command will not be fed immediately to the COMMAND intrinsic, but will be processed by a Vesoft command interpreter, similar to CI.PUB.VESOFT. This means that MPEXL commands are available from within STREAMX (even if you are on a classic 3000!) UDC's are not available here, but command files are, so ::PSELECT is a call to a command file residing in the default path (it's in PUB.SYS). The commands in the command file are executed sequentially by STREAMX. The command file will set the desired variables and then return control to the jobstream.

One of the variables set in the command file is called REMOTE. This variable is necessary to set up a REMOTE HELLO, if the desired printer is not on the local computer. If the printer is local, the variable is a COMMENT command. Either way the variable becomes a record in your jobstream.

From this point on, the variable PRINTER can be used in any file equation to direct output to the chosen printer. The variable contains all of the file statement parameters necessary starting with the DEV= clause. This way, PSELECT does not have to know what your formal file designator for the output is. The Printer variable can be substituted as many times as desired.

The command file itself, listed in lines 427/438, looks deceivingly simple. A separate program called Prselect.pub.sys is run to display the menu of printer choices contained in Prdata, and then prompts the user for his choice. The data for his printer along with either a REMOTE HELLO or COMMENT command is output into a temporary file as ::SETVAR commands. The 3 record temp file (2 SETVARs and a ::EXIT) is then input in a nested execution of STREAMX. Any output from this run of STREAMX from within STREAMX is suppressed by the ;STDLIST=$NULL.

The nested run of STREAMX was necessary because until recently (STREAMX version 2.2) we had no way of having STREAMX execute :: statements that were generated dynamically within the jobstream. Version 2.2 gave us the long-awaited ::USE command which makes this a simple task.

I don't know where ::SETVAR variables are stored--all I know is that they are there when I need them. Also, variables that are set by the nested execution of STREAMX are available to the outer STREAMX process as well. Consequently, the ::SETVAR PRINTER done in the nested process provides the variables for the outer process (your jobstream).

The Prdata file is pretty straightforward. It allows for comments, as well as information to be displayed to the user. The actual printer definitions begin with a ".". This file must exist on each computer in the network. The contents of each file are machine specific.

PSELECT is due for some extensive rewriting. Although it works quite well, and is very simple for the users to understand, Systems people have a hard time understanding it. With STREAMX 2.2, it should be possible to code PSELECT entirely in STREAMX code, eliminating the menu program and the nested run of STREAMX.

Conclusions

This paper has shown five varied examples of how the Vesoft products MPEX and STREAMX can solve some real life Systems problems. Although these solutions can be used "right out of the can", the techniques they use may be of more benefit to you in solving other problems your shop may have.

If you do not own a copy of MPEX or STREAMX, you may want to look into purchasing these products. The examples in this paper show a small subset of the capabilities available to you. If you are a current Vesoft user, this paper may have convinced you that it will be worthwhile to read (or reread) the full documentation that comes with these products, especially since the 2.1 versions of MPEX/3000 and SECURITY/3000. They continue to be one of your best buys in utility software.

APPENDIX A

Jobstream PURGEWRK

```
!JOB PURGEWRK,MANAGER.SYS;OUTCLASS=LP
!COMMENT
!COMMENT   JOBSTREAM TO SEEK OUT AND PURGE ALL NON-ACTIVE K FILES
!COMMENT   FROM EDITOR (BOTH DEFAULT FORMAT AND COBOL TEXT), ALL
!COMMENT   RECOVERY CHART FILES AND DATA FILES FROM EZCHART, ALL
!COMMENT   RECOVERY DRAWING AND FIGURE FILES FROM HPDRAW.
!COMMENT
!EDITOR
ADD
FILE !A,!B
PURGE !A
PURGE !B
***********
//
KEEP $NEWPASS,UNN
EXIT
!MPEX
USE PURGEWRK.SE.SYS
EXIT
!EOJ
```

MPEX use file PURGEWRK

```
COMMENT *****  LIST AND PURGE EDITOR K FILES *****
LISTF K#######.@.@(CODE='EDITQ'),LISTFXX
LISTF K#######.@.@(CODE='EDTCQ'),LISTFXX
PURGE K#######.@.@(CODE='EDITQ')
PURGE K#######.@.@(CODE='EDTCQ')
COMMENT *****  LIST AND PURGE SLATE WORK FILES *****
LISTF P#######.@.@(CODE='SLATE'),LISTFXX
LISTF P#######.@.@(CODE='SLATW'),LISTFXX
PURGE P#######.@.@(CODE='SLATE')
PURGE P#######.@.@(CODE='SLATW')
COMMENT *****  LIST AND PURGE CHART WORK FILES *****
LISTF CF#?????.@.@+CL#?????.@.@(CODE='GRAPH'),LISTFXX
USER $OLDPASS,CF#?????.@.@(CODE='GRAPH'),DF#?????.@.@
USER $OLDPASS,CL#?????.@.@(CODE='GRAPH'),DL#?????.@.@
COMMENT *****  LIST AND PURGE HPDRAW WORK FILES *****
LISTF ??######.@.@(CODE='DRAW'),LISTFXX
PURGE ??######.@.@(CODE='DRAW')
COMMENT *****  LIST AND PURGE FIGURE FILES *****
LISTF ??######.@.@(CODE='FIG'),LISTFXX
PURGE ??######.@.@(CODE='FIG')
```

APPENDIX B

Jobstream LOCKACCT

```
::ECHO  LOCKACCT is used to display an account specific message
::echo  on a user's terminal at logon time.  Optionally, access
::echo  can be denied to all users without AM capability.
::echo
::ECHO  Enter the account you want to lock or set a message on.
::echo  You must be an Account Manager of the selected account.
::ECHO  The default is {hpaccount}.
::echo
::prompt string account = "Which Account";default="{hpaccount}";&
:: check=(mpe("listacct " + account + ",$null") <> 909);&
:: checkerror="Invalid Account"
::echo
::echo  Enter the userid you wish to sign on with in account {account}
::echo  The default is {hpuser}.
::echo
::prompt string user = "Userid";default="{hpuser}";&
::  check=(mpe("listuser " + user + ".{account},$null") <> 910);&
::  checkerror="This is not a user in account {account}"
!JOB LOCKACCT,{USER}.{ACCOUNT};PRI=DS;OUTCLASS=LP,1
!SETJCW JCW = 0
!IF JCW <> 0 THEN
 *
 ***************************************************************************
 *   Bill McAndrew   6/1/86   Donnelly Corporation Technical Services
 *
 * This jobstream can be used for two different purposes:
 *
 *        1.  Issuing a variable length message to all users of an account
 *            at logon time (in addition to the system welcome message)
 *
 *        2.  Keeping all users out of an account while recovery or
 *            maintenance is taking place.  A custom, variable length
 *            message can be created explaining the lockout, which is
 *            issued before the user is automatically byed off.  Users
 *            with Account manager capability are not byed off.
 *
 * The jobstream accomplishes this by adding an additional account level
 * logon UDC (in its own UDC file) to the existing UDC's of the account.
 * The user streaming the job must have AM (account manager) capability.
 *
 * The account lock and/or message applies to all new signons into the
 * account.  Current users of the account will not receive the message
 * or be byed off until they sign on again.
 *
 * The stream requires answers to 4 questions:
 *
 *     1.  Which Account?
 *
 *          The jobstream is launched in the specified account with the
 *          user name being your current logon user id.  This ensures that
 *          only account managers have the power to stream this job.
 *
 *     2.  Override message file?
```

```
*
*         This parameter specifies an Editor file containing the
*         message to be displayed to the users at logon.  The message
*         should be created as an unnumbered file in editor (record
*         length of 72) and can be as long as desired.  If the record
*         length is not 72, Fcopy will issue a warning message to the
*         user requiring a response before printing the message.
*         If a file is specified, make certain that the subject account
*         can read it, i.e., place it in the account or release the file
*         after keeping it.  If no file is specified, the stream uses
*         the default file LOCKACCT.SE.SYS.  The contents of that file
*         follow:
*
*
*
*       ***************************************************************
*       *  Sorry, this account is temporarily unavailable for use.
*       *
*       *                Please try again later.
*       ***************************************************************
*
*
*     3.  Current Account UDC file?
*
*         The existing account level UDC file for the subject account is
*         a required parameter.  This UDC file will be re-set with the
*         newly created LOCKUDC file.  The UDC filename may be fully
*         qualified.
*
*     4.  Bye off at end of message?
*
*         This question determines whether the message is for informational
*         purposes only, or if an account lockout is desired.  The default
*         is account lockout or "Y".
*
* If the account is locked (bye option selected) a message is issued
* to the system operator informing him of the locked status, and the
* user who has issued the lock.
*
* If the stream completes successfully, a message is sent to the streamer.
*
* To remove the lock [and message]:
*
*    -- Log on to the subject account.
*    -- Type   :SETCATALOG (udc file);ACCOUNT  specifying your account UDC
*
*******************************************************************************
*
!ENDIF
!
::echo
::echo  Enter the file containing the message you wish to set.
::echo  Just press return to use the default message file.
::echo  The file must be an unnumbered ASCII file with record
::echo  length = 72.
::echo
::PROMPT STRING MSGFILE = "Override Message file";&
```

```
:: default="lockacct.se.sys";&
:: check=(finfo(msgfile,0) and finfo(msgfile,14)<>-72);&
:: checkerror="File does not exist, or record length not 72"
!comment Message file: {msgfile}
!
::echo
::echo Enter the account level UDC file for this account.
::echo The filename may be fully qualified (file.group.account).
::echo If there is no UDC file, press return.
::echo
::prompt string udcfile = "Current Account UDC file";default="*";&
:: check=(udcfile = "*" or finfo(udcfile,0));&
:: checkerror="File does not exist"
!comment UDC file: {udcfile}
!
!SHOWCATALOG
!setjcw cierror = 0
!continue
!setcatalog ;account
!if cierror <> 0 then
! tell {hpuser}.{hpaccount} ^[&dJ You are not an Account Manager of {account}
^[&d@
! eoj
! endif
!
::echo
::echo Answer Y to lock the account preventing all but Account
::echo Managers from signing on. Answer N to just set a message
::echo on the account.
::echo
::prompt string byeoff = "Bye off at end of message";&
:: check=(ups(byeoff) = "Y" or ups(byeoff) = "N");&
:: checkerror="Answer Y or N"
!
!EDITOR
ADD
LOCKUDC
OPTION LOGON,NOBREAK,NOHELP
FILE STD=$STDLIST;REC=-72
CONTINUE
RUN FCOPY.PUB.SYS;&
INFO='FROM={msgfile};TO=*STD'
RESET STD
::if byeoff = "Y" or byeoff = "y" then
SETJCW CIERROR = 0
CONTINUE
LISTUSER @,$NULL
IF CIERROR <> 0 THEN
BYE
ENDIF
::endif
CONTINUE
ACCOUNTLOGON
***************************************************************
//
KEEP $NEWPASS,UNN
EXIT
```

```
!SETJCW CIERROR = 0
!CONTINUE
!PURGE LOCKUDC
!IF CIERROR = 384 THEN
!   COMMENT   CIERROR 384 means "unable to purge file"
!   COMMENT   LOCK OR MSG CURRENTLY SET IN FILE "LOCKUDC"
!   SETJCW CIERROR = 0
!   CONTINUE
!   PURGE LOCKUDC2
!   IF CIERROR = 384 THEN
!      TELL {HPUSER}.{HPACCOUNT} Error! 2 Lockaccts set!
!      EOJ
!      ENDIF
!   SAVE $OLDPASS,LOCKUDC2
::if udcfile <> "*" then
!   SETCATALOG LOCKUDC2,{UDCFILE};ACCOUNT
::else
!   SETCATALOG LOCKUDC2;ACCOUNT
::endif
!ELSE
!   COMMENT   LOCK OR MSG SET IN FILE "LOCKUDC2" OR NOT SET AT ALL
!   SAVE $OLDPASS,LOCKUDC
::IF UDCFILE <> "*" THEN
!   SETCATALOG LOCKUDC,{UDCFILE};ACCOUNT
::ELSE
!   SETCATALOG LOCKUDC;ACCOUNT
::ENDIF
!ENDIF
::IF BYEOFF = "Y" OR BYEOFF = "y" THEN
!   TELLOP  {HPUSER} HAS LOCKED {ACCOUNT}
!   TELL {HPUSER}.{HPACCOUNT} {ACCOUNT} is now locked.
::else
!   TELL {HPUSER}.{HPACCOUNT} Message set on {ACCOUNT}
::endif
!EOJ
```

APPENDIX C

Jobstream QCOMPILE

```
!JOB QCOMPILE,?$VARIABLE=S$ WHAT'S YOUR SIGNON (user.acct)?
!SETJCW Y = 1
!SETJCW N = 0
!SETJCW YES = Y
!SETJCW NO  = N
!COMMENT  LIST or NOLIST:  ?$VARIABLE=LIST$ LIST or NOLIST?
!SETJCW ANS = ?RECOMPILE QSCHEMA.SOURCE INTO QSCHEMAC.PUB (Y/N)?
!IF ANS = Y THEN
!QDD
USE QSCHEMA.SOURCE  ?$VARIABLE=LIST$?
!CONTINUE
EXIT
!CONTINUE
!TELL ?$VARIABLE=S$?; ^[&dB QSCHEMA has been recompiled. ^[&d@
!ENDIF
!SHOWTIME
!SETJCW ANS = ?RECOMPILE ALL QUIZ REPTS (Y/N)?
!IF ANS = Y THEN
!MPEX
COMMENT   ***********************************
COMMENT   *   COMPILE ALL QUIZ PROGRAMS
COMMENT   ***********************************
LISTF @.@(CODE=642),*LISTFXX.LISTF.VESOFT
USER QUIZCOMP.USER.VESOFT,@.@(CODE=642),=.SOURCE,"?$VARIABLE=LIST$?"
LISTF @.@(CODE=642),*LISTFXX.LISTF.VESOFT
EXIT
!CONTINUE
!TELL ?$VARIABLE=S$?; ^[&dB QUIZ reports are recompiled. ^[&d@
!ENDIF
!SHOWTIME
!SETJCW ANS = ?RECOMPILE ALL QUICK SCREENS (Y/N)?
!IF ANS = Y THEN
!MPEX
COMMENT   ***********************************
COMMENT   *   COMPILE ALL QUICK SCREENS
COMMENT   ***********************************
LISTF @.@(CODE=641),*LISTFXX.LISTF.VESOFT
USER QUICCOMP.USER.VESOFT,@.@(CODE=641),=.SOURCE,"?$VARIABLE=LIST$?"
LISTF @.@(CODE=641),*LISTFXX.LISTF.VESOFT
EXIT
!CONTINUE
!TELL ?$VARIABLE=S$?; ^[&dB QUICK screens are recompiled. ^[&d@
!ENDIF
!SHOWTIME
!SETJCW ANS = ?RECOMPILE ALL QTP OBJECTS (Y/N)?
!IF ANS = Y THEN
!MPEX
COMMENT  ********************************
COMMENT  *   COMPILE ALL QTP OBJECTS
COMMENT  ********************************
LISTF @.@(CODE=643),*LISTFXX.LISTF.VESOFT
USER QTPCOMP.USER.VESOFT,@.@(CODE=643),=.SOURCE,"?$VARIABLE=LIST$?"
LISTF @.@(CODE=643),*LISTFXX.LISTF.VESOFT
```

```
EXIT
!CONTINUE
!TELL ?$VARIABLE=S$?; ^[&dB QTP objects are recompiled. ^[&d@
!ENDIF
!EOJ
```

<u>MPEX user templates QUIZCOMP, QUICCOMP, and QTPCOMP</u>

```
START
:COMMENT   *************************************************
:COMMENT   *   RECOMPILE ALL QUIZ REPORTS
:COMMENT   *
:COMMENT   *  EX.  %$USER QUIZCOMP.USER.VESOFT,
:COMMENT   *           @.@(CODE=642),=.SOURCE,
:COMMENT   *           "NOLIST"
:COMMENT   *
:COMMENT   *  RECOMPILES ALL COMPILED QUIZ REPORTS AS
:COMMENT   *  SPECIFIED IN THE OBJECT FILESET.  COMPILED
:COMMENT   *  OBJECTS MAY RESIDE IN ANY GROUP BUT THE
:COMMENT   *  CORRESPONDING SOURCE MUST BE IN GROUP 'SOURCE'
:COMMENT   *  ALL SOURCE FILES MUST END WITH 'BUILD' COMMAND
:COMMENT   *************************************************
:QUIZ
************************************************************
FILE  !PUB,!SOURCE,!LIST
;****************************************************************
;*********   COMPILING REPORT:  !PUB   **********
;****************************************************************
SET DEFAULT
SET SAVE CLEAR
:RESET QUIZLIST
SET REPORT DEVICE TERMINAL NOSTATISTICS
:PURGE !PUB
USE !SOURCE   !LIST
************************************************************
FINISH
:COMMENT   ======= END OF QUIZ COMPILES =======
EXIT
************************************************************

START
:COMMENT  **********************************************
:COMMENT  *  COMPILE ALL QUICK SCREENS
:COMMENT  *
:COMMENT  * EX.   %$USER QUICCOMP.USER.VESOFT,
:COMMENT  *              @.@(CODE=641),=.SOURCE,
:COMMENT  *              "NOLIST"
:COMMENT  *
:COMMENT  *  SCREEN STATEMENT DECLARES WHERE OBJECT GOES
:COMMENT  *  ALL SCREENS END WITH "BUILD" STATEMENT
:COMMENT  *  SOURCE FOR ALL SCREENS MUST BE IN 'SOURCE'
:COMMENT  **********************************************
:QDESIGN
***********
FILE !PUB,!SOURCE,!LIST
;****************************************************************
```

```
;*******   COMPILING SCREEN:  !PUB   ********
;*******************************************************************
SET DEFAULT
:PURGE !PUB
USE !SOURCE  !LIST
CANCEL
***********
FINISH
:COMMENT    ======= END OF QUICK SCREEN COMPILES =======
EXIT
***********

START
:COMMENT ***********************************************
:COMMENT *  RECOMPILE ALL QTP OBJECTS
:COMMENT *
:COMMENT * EX.   %$USER QTPCOMP.USER.VESOFT,
:COMMENT *              @.@(CODE=643),=.SOURCE,
:COMMENT *              "NOLIST"
:COMMENT *
:COMMENT *  BUILD STATEMENT DECLARES WHERE OBJECT GOES
:COMMENT *  ALL PROGRAMS END WITH "BUILD" STATEMENT
:COMMENT *  SOURCE FOR ALL PROGRAMS MUST BE IN 'SOURCE'
:COMMENT ***********************************************
:QTP
***********
FILE !PUB,!SOURCE,!LIST
;*******************************************************************
;*******   COMPILING PROGRAM:  !PUB   ********
;*******************************************************************
SET DEFAULT
:PURGE !PUB
USE !SOURCE  !LIST
CANCEL
***********
FINISH
:COMMENT    ======= END OF QTP COMPILES =======
EXIT
***********
```

Editor use file USEQPROD

```
Q " "
Q "Key in your source filename and press return"
     << File entered can be fully qualified: name/lock.grp.acct >>
Z::=
     << Set function keys 1-5 to the Q-products via MPEX          >>
Q "^[&f2a1k13L%QDD ,SUSPEND    "
Q "^[&f2a1k16d13L  Qdd           %QDD ,SUSPEND   "
Q "^[&f2a2k18L%QDESIGN ,,SUSPEND"
Q "^[&f2a2k16d18LQdesign         %QDESIGN ,,SUSPEND"
Q "^[&f2a3k17L%QUICK ,,,SUSPEND "
Q "^[&f2a3k16d17L Quick          %QUICK ,,,SUSPEND "
Q "^[&f2a4k17L%QUIZ ,,,SUSPEND  "
Q "^[&f2a4k16d17L Quiz           %QUIZ ,,,SUSPEND  "
Q "^[&f2a5k6L%SPOOK  "
Q "^[&f2a5k16d6L Spook          %SPOOK  "
     << Set function keys 6-7 to Use statements                   >>
Q "^[&f2a6k16d1L  Use    List  U "
Q "^[&f2a6k44LUSE Z:: LIST                                          ^@"
Q "^[&f2a7k16d1L  Use    Nolist U"
Q "^[&f2a7k46LUSE Z:: NOLIST                                          ^@"
     << Set function key 8 to USE USEQPROD                         >>
Q "^[&f2a8k19LUSE USEQPROD.SE.SYS  "
Q "^[&f2a8k16d19L  Use   UseqprodUSE USEQPROD.SE.SYS  "
     << Display User keys                                         >>
Q "^[&jB"
Q "          ^[1  ^[1  ^[1  ^[1  ^[1  ^[1  ^[1  ^[1  ^[1  ^[1  ^[1
 ^[1"
S TABS=(4,7,10,13,16,19,22,25,28,31,34,37)
TEXT Z::
```

APPENDIX E

Command file PSELECT.PUB.SYS

```
comment    Command file for printer selection program
comment     WSM   7.6.88
comment
comment   These are the commands needed to use PSELECT in your job!
comment          ::PSELECT
comment          !{REMOTE}
comment          !FILE xxxxxx=xxxxxx;{PRINTER}
comment
comment
run prselect.pub.sys
file strmfile=prchoice,oldtemp
run streamx.pub.security;parm=1;stdlist=$null
```

Sample PRDATA file

```
*
*    This file contains data about the printers on our network.
*
*    Records with an asterisk in column 1 are comment cards and
*    are not processed.
*
*    Records with a "D" in column 1 are printer description lines
*    and are displayed by the program.
*
*    Records with a "." in column 1 are printer definition
*    records and have the following layout:
*
*    Node names, if present, should be right justified.
*
*<--Node-><-Class><--------Other additional file specs--------------->
DHere is a list of printers you can send your report to:
D
.           LP
DThe system line printer of the Series 70 (Computer room - LP).
.REMPRINT#LP
DThe system line printer of the Series 950 (Computer room - LP).
.           LASER
DThird street laser printer in report mode (LASER).
.           LASER    ;ENV=QUIZPORT.HPENV.SYS
DThird street laser printer in document mode (LASER;ENV=GOTHIC).
```

System Managers guide to MPE/XL migration and installation

Isaac Blake
City of Tempe
Information Systems
120 East Fifth Street
Tempe, AZ 85281
(602) 731-8218

After years of waiting and wondering, the HP Precision architecture machines and MPE/XL operating system is here. These new systems provide a challenge and a new frontier for both experienced and new System Managers. The last time such a changed occurred was when HP left the design of the Series III's to adopt the HP-IB systems that we know of today. A interesting piece of history is for those of us who were around for both evolutions, the cycle of both "generations" were the same. I had to chuckle reading the media critics shoot down the "RISC" concept, because several years prior they were doing the same with the HP-IB concept. Basically the media was saying that HP-IB was going to be the downfall of HP, and the users had the same worried look.

Granted, the early machines like the series 30 and 33 were slow and had initial problems, but if you take alook at the machines that came after those each one grew until the top of the HP-IB systems, a series 70. If you look at the HP-PA path, the same is true. I was fortunate to be working for HP when the first HP-PA system, a 930 was available. Granted the problems with the 930 was inline with the series 30 (maybe systems with series of @30@ have this tag?), but remember one key point; this system was to assist users and vendors with getting their software over to MPE/XL and for testing. It was NEVER meant for production. In fact, for the MPE/XL operating system, release 1.1 was always slated as the first release for users to do production on.

I'm not trying to change my career to being a historian, however, I feel that it's important for users to realize and put in proper PERCEPTION, the events and it's "cycle". Most of the bad press, comments, and concerns originate from pre-1.1 days, and the vast majority of users on 1.1, and beta testing 1.2 are quite happy. The failure rates on hardware have been almost non existant, and if you have been good at keeping on top of the patches and releases, the software has settled down as well.

The City of Tempe currently has 10 HP3000's, 2 of which are HP-PA systems. We have both a 950 and 925/LX, for the 950 we migrated from a 3-bay HP3000 series 70, and the 925/LX was a new installation. The 950 was the first to be installed and was not to be delivered until MPE/XL 1.1 was available. Both systems are currently on MPE/XL 1.1 version A.10.17 with various patches.

The series 70 had our largest and most diverse user base, therefore it was a good challenge for migration. The configuration of the system was three bays, nine megabytes of memory, 244 ATP's, LANIC, six 7933's, two 7937's, 7980 tape drive, INP, 2680 printer, Support Link modem, and a pool of dialup modems. Software on this system was very diverse, and consisted of just about all HP Software and third party software. The system was on MPE/V version V-delta-2.

Below is a list of the system software on the system:

DataComm:	ThinLAN, OfficeShare, NS/3000.
HP Utilities:	HP Security Monitor, OPT, Rspool, Sampler, HPTREND, Predictive Support, IDSCHAR, IDSFORM, IFS, LPS, TDP, and HPMENU.
Languages:	Basic, CobolII, Fortran, Pascal, RPG, and SPL.
Office Automation:	HPDESK, HPDRAW, HPEZCHART, HPWORD, HPSPELL, HPSCHEDULE/3000, Visicalc, HPWORD intrinsics, and DSG/3000.
Other Utilities:	INTEREX CSL programs, QUAD.
Third-Party Software:	COGNOS, DBGENERAL, INDEX/PLUS, MPEX, PAL/GENTRY, COGO, JobRescue, OCS, REPORTER, SPEEDEDIT, SPSS, S/Compare, JetLink, and DOC/3000.

As you can see, it was a very extensive list of software!

MIGRATION BEGINNINGS

It is important that as soon as you start looking at migration, that you adopt and share with others that this is a "PROJECT". That means that you should have a project team, regular meetings, action items, timelines, the whole works. The project leader for the City was my boss, Lew Dalrymple. Lew did no less that the best job I've ever seen or heard of with a project. He has also completed a paper and presentation on migration from a project management viewpoint. I strongly suggest that you get a copy of the paper, entitled

"Managing a HP3000/950 Migration, a Workbook approach", and see the presentation.

Make sure that your project team includes the appropriate HP counterpart. That is, a HP SE for the System Manager, the HP Sales Rep for the Project Leader, the HP CE, and any speciality fields such as Datacomm, Office Automation, Personal Computers, etc... This will allow for good interaction and acts as a "check and balance" with each area.

HP offers the "FASTLANE" service to help users migrate from a MPE/V to a MPE/XL system. This service and support will successfully help you with all the migration issues that you may have. It will start with a Migration Overview for you that highlights what Precision Architecture is, the differences between Compatibility Mode (CM), Native Mode (NM), and Mixed Mode (CM+NM). Also included is a excellent section on Migration Options and Solutions. This will cover Program, Datacomm, and Operational issues to migration, and their solutions.

The migration itself should be broken down into six area, which are:

* Education
* Analysis and Planning
* Preparation
* Installation
* Compatibility Operation
* Native Mode Operation

Education is key to the success to the project. Make the committment upfront to send people to the Migration classes that HP has. Also, make use of the consulting time that's available to you thru "FASTLANE". Although most things will be familiar, there is enough differences to bother you. Compare it to being bilingual, but instead of difference being Spanish to English, it is more like English verses American.

Analysis and planning will be made simplier by using the Migration tools which are OCA (Object Code Analyzer), RTM (Run Time Monitor), and MPT (Migration Planning Tool). These tools need to be run on a MPE/V system that is on U-MIT or later. OCA will scan program and SL files looking for incompatibilities, and generates a detailed report. RTM on the other hand will monitor the actual run time errors that occur. For example, OCA will issue a warning that a program uses the COMMAND intrinsic (since some MPE/V commands are not supported under MPE/XL), however RTM will generate an error if you issue a truly unsupported command like :SHOWCACHE. MPT is used to combine the output from all the migration tools for a consolidated report.

Preparation will be any and all conversions that you need to do. These would include FORTRAN-66/V to FORTRAN-77/V, BASIC/V to HP Business Basic/V, COBOL68 to COBOLII/V, IMAGE to TurboIMAGE, DS/3000 to NS/3000, and the SPL issue which entails either translating the code to PASCAL, or, use the SPLASH compiler from SRN. Remember, most programs will run in CM with very little problems. Look for any conflicts with your accounting structure (MPE/XL has certain reserved accounts and groups like CONFIG.SYS), identify all your User-Logging ID's and global RIN's, and study your usage of UDC's. Also, if needed layout any switch stubs you may need.

Installation is made easy with the DIRMIG utility. This utility will migrate your MPE/V operating environment to MPE/XL. This will involve having your MPE/V SYSDUMP tape and running DIRMIG. It will then create your directory structure, global RIN's, User-Logging identifiers, UDC environment, System Table information, and Private Volume information. Once this is done it will then restore your user files onto the MPE/XL system. If you do not desire a full migration, DIRMIG will allow you the option to select any or all of the above.

Compatibility Mode Operation is what you should be in for a few weeks after installation to allow the system to settle in. You should come up with a test plan to ensure that any incompatibilities that were detected are working correctly on MPE/XL. After you are satisfied with CM operation, then look at using what I consider "the best" migration utility, OCT (Object Code Translator). OCT will take a MPE/V program file, read thru it, and generate NM code to be executed in it's place. This is fantastic since you can now take a CM program and get better NM performance without have to recompile the program. Further, it offers an approach that allows you to get NM performance on programs that you do not have source for. Even though OCT does generate good NM code, and the performance is better than CM, you will have to recomplile the code in NM to get the best performance.

Native Mode Operation will be your final step. This will require you to make changes to any NM incompatibilities, recompiling the source into NM programs, and converting data files to NM format (32-bit alignment, and IEEE floating point format). This is also where you can start taking advantage of all the outstanding MPE/XL commands and intrinsics, along with features like Mapped files. Another nice feature of the NM compiliers is that they will allow you specify a level of optimization. This means that the compilier will inspect your code, and if it find a better way of doing what you requested, it will do it that way. I feel that this is the beginning of Artifical Intelligent compliers, which is long overdue.

MIGRATION REALITIES

The first reality that I would like to deal with is my earlier comment about being "bilingual" between MPE/V and MPE/XL. This is especially important if your site is going to have a mixture of both systems. The problem is that you can get confused going back and forth, and this problem will magnify with the number of users that you have on your systems. I strongly advise that you "modularize" functions on your system. For example, develop a job stream that does a FULL and PARTIAL backup for MPE/V and MPE/XL. The operator will then only have to remember that he needs to do :STREAM FULLDUMP.JOB.SYS on any system to do a full backup. Make use of job streams, UDC's, and especially command files.

MPE/XL by it's nature uses more disc space. The "standard" rule seems to be that you will need 2-7937 disc drives to compensate for the increase. This increase is for both the Operating System, and the larger NM program files. Since we are using RISC, this means that there will be more instructions needed to do the same thing, thereby causing more disc space.

Take the time to layout your system, deciding which device goes where, what the difference is between a Mid-Bus/Channel Adapter, Device Adapter, etc. Also follow a good suggestion that was made to us, make your DTC LDev numbers the same as the DTC/Board/Port Number. For example, LDev#234 would be DTC #2, board #3, port #4. It really works out and makes tracking alot easier.

A common mistake is hooking up the Support Link modem. The modem is connected from the system to the modem, and the system to the DTC. The cables that are supplied by HP are marked "COMPUTER" and "MODEM", which works fine from the system to the modem, however will NOT work from the system to the DTC! The reason is the difference of DCE verses DTE, so you will have to reverse the cable going to the DTC.

MIGRATION CONCLUSIONS

We had a very successful migration, and the number of "surprises" that we had were almost non-existant. Therefore my recommendation to you is to "come on in, the water is fine!". You will really enjoy MPE/XL and the HP Precision Architecture systems, and find a whole new frontier to explore.

BIOGRAPHY

Isaac Blake is currently System Manager and Technical Support Analyst for the City of Tempe, Arizona, where he supports 12 HP3000's, ranging from Micro3000's to a Series 950.

His experience covers 17 years in the computer industry, with the last 13 years working on HP3000 systems. During this time he has gathered extensive broadbase knowledge of the HP3000 family most notably in System Management, Performance, Security, Operations, MPE/XL, Migration, and User Support.

After working for HP as a Senior Support Engineer at the HP Western Response Center, he returned to the user community in 1987. Currently he is on the INTEREX Board of Directors, and is a Reserve Police Officer who has dealt with investigation and prosecution of computer related crimes.

He has successfully installed MPE/XL systems and the migration of those systems from MPE/V to MPE/XL, along with the installation and support of the Precision Architecture hardware. Also, he has addressed and resolved many of the issues involved with supporting both MPE/V and MPE/XL systems concurrently.

System Manager's Guide

to

MPE/XL Migration and Installation

by: Isaac Blake, City of Tempe Arizona

4806-7

MIGRATION HISTORY

HP/3000's are mostly upward compatable

SIO (series III) to HP-IB (series 30/33)

Classic Hardware upgrades

MPE software upgrades

HP Precision Architecture

MPE/XL Operating System

8-9084

CITY OF TEMPE SYSTEM CONFIGURATION

1 - HP/3000 series 950 (70 Migration)

1 - HP/3000 series 925/LX (installation)

2 - HP/3000 series 70's

2 - HP/3000 series 58's

1 - HP/3000 series 42

2 - HP/3000 Micro/XE's

1 - HP/3000 series 37/XE

6-9084

MILL series 70–950 migration

Hardware:

HP/3000 series 70 with 9MB memory

244 ATP's (12 modem ports)

LANIC and INP

6 - 7933 disc drives, 2 - 7937 discs

7980 tape drive

2680 Laser Printer

4806-10

MILL series 70–950 migration

Software:

HP general products and utilities

Languages and 4GL products

Office Automation products

Third party software

COT developed applications

Other utilities

4806-11

STARTING MIGRATION and/or INSTALLATION

Manage and approach as a project

Create a project team with authority

Have, and utilize, HP counterparts

Develop a plan, timeline, and milestones

Invest in HP's "FASTLANE" service

Keep expectations and tasks reasonable

4806-12

MIGRATION PHASES

Education

Analysis and Planning

Preparation

Installation

Compatibility Mode Operation

Native Mode Operation

4806-13

MIGRATION PHASES

Education:

IS the key to the success of the project!!!

Complete all the HP Migration classes

Read articles and proceedings

Contact other users who have migrated

Train all users

Learn to be bilingual

4806-14

MIGRATION PHASES

Analysis and Planning:

Good time to learn & document the system

Object Code Analyzer (OCA)

Run Time Monitor (RTM)

Migration Planning Tool (MPT)

Develop standards and classes

Make functions and modules

4806-15

MIGRATION PHASES

Preparation:

Language Conversions

IMAGE to TurboIMAGE

DS/3000 to NS/3000

Correct Third-Party software for MPE/XL

Laying out entire system configuration

Resolving conflicts and incompatabilities

91-9064

MIGRATION PHASES

Installation:

Partial, or complete migration?

Take your time, and have some play time!

DIRMIG and SYSGEN

DTC, LAN, and Network Installation

Support Link and Synaspe (Mux)

Training and Verification

4806-11

MIGRATION PHASES

Compatibility Mode:

CM for awhile to allow system to settle

Object Code Translator (OCT)

OCT CM programs (especially HP's)

STORE/RESTORE/SYSGEN issues

Verify disc space usage

NM planning and testing

4808-18

MIGRATION PHASES

Native Mode:

Programming MPE/V vs MPE/XL systems

Data alignment (16 bit vs 32 bit)

Switch stubs

Increased disc space for NM programs

Utilizing new features

Misc issues (i.e. floating point)

4806-19

MIGRATION REALITIES and CONCLUSIONS

The Project and Team approach works

HP is an invaluable resource

Being bilingual is a must

Keep up on current releases and patches

Modularizing functions keeps your sanity

"Come on in, the water is fine!"

Managing A System When Users Are Programmers

James F. Davis
Cognos Corporation
2301 E. Lamar Suite 250
Arlington, TX 76006
(817) 649-1944

INTRODUCTION

Typical current problems that many system managers deal with daily are: deteriorating machine performance, shortages of equipment, overworked data processing (DP) personnel, and hardware capacities pushed to the limit. However, most system managers have well organized and maintained systems because they know how to handle these current problems and are dealing with educated and trained professional data processors. With the proliferation of end user computing, the system manager is presented with a set of challenges that had not been imagined a few years ago. These challenges include programmers increasing ten fold, all new end user programmers with no knowledge of proper programming standards and techniques, business decisions being made using reports developed without the proper testing or controls. All this with no real increase in hardware "horsepower" or storage capacity!

The phenomenon that has created these challenges is the introduction of fourth generation languages (4GLs). These 4GLs are user friendly, English like, and allow the separation of data updating and reporting functions. Some even allow "point and shoot" reporting. From the end users' point of view, they can access their data without dependence on DP and can be more productive. From the system manager's point of view, the backlog of user requests is reduced. However, their abuse can "break" a previously well maintained system.

It is possible, in a mid-sized HP shop, to go from three to potentially hundreds of "programmers," as each end user is a potential programmer. The effects on the system and even the business can be considerable. Having been involved with system management in several mid-sized shops that made 4GL report writers available to the end users, I have gained some insight into what can happen. Since each shop is unique, it is not possible to offer answers to specific problems. However, there are some areas that warrant attention.

The purpose of this paper is not to push the advantages of 4GLs. It is to sensitize system managers to potential problems that occur when end user computing is introduced and to provide some solutions that I have found useful. In addition I will include a checklist that can be tailored for any shop's use when considering the addition of end user computing.

This paper is organized into five sections. The first four are areas where problems are most likely to develop: data knowledge, performance, capacity, and specialized language idiosyncrasies. The last section is standards, which if used properly can lower or negate the impact of the problem areas.

Logical data knowledge, how the data is constructed and related, seems like second knowledge to most data processors. It can be confusing and entirely new to most end users.

Performance, the throughput of the computer, is already an area of concern for most system managers. The addition of many new "programmers" can degrade almost any system's performance.

Capacity planning, the attempt to predict demand and manage the use of DP resources, can entail much more than disk and memory capacity when end users become programmers.

Specialized language idiosyncrasies, the pitfalls or "features" in each of the 4GLs on the market today, can range from "pesky" to "machine eaters." It is not within the scope of this paper to address specific syntax that must be watched for. However, there are some general concerns that affect all 4GLs.

Standards, written procedures that are followed for and by all 4GL users (DP staff and end users), are the best tool for the successful management of a computer where end user computing is imminent.

Logical Data Knowledge

Problem One

Logical data knowledge is the most important area to be considered when end users become programmers. Without a good working knowledge of the physical and logical relationships within the data structure, inaccurate reports are almost guaranteed. If business decisions are made using these reports, the results could be disastrous.

Understanding data structures is one of the most difficult concepts for most end users to grasp. Think back to your first introduction to the physical file structures - sequential, keyed/sequential, direct, relative, data bases (automatic/manual masters, detail sets). These concepts only became second nature through the specialized training that professional data processors receive or after considerable frustration and trial and error. When the end user is presented with a report writer, they are often expected to understand these structures without any formal training or background knowledge. They must not only understand the physical properties of the files, but more importantly they must understand which files can be logically related.

File relations are often the cause of problems. With many 4GLs it is possible to link files together in ways that were never conceived by the original creators of those files. The only safeguard is the clear understanding of the logical data relationships within the system.

In older systems where data is not normalized the problems can be compounded. Some files are "multi-use." (the actual format of the record is dependent on the data it contains). Some files contain data you would not expect to be there (all salesman information is only stored with the sales order - rather than in a salesman file).

<u>Solution</u>

The "point and shoot" or "report painter" 4GLs solve the logical data knowledge problem with pre-defined "views" of the data. These views are set up by someone in the data processing area or by a knowledgeable end user. This takes the guesswork out of the data structures for the end user. It also limits the user to the views that have been defined. This can place the end user in the position that end user computing was designed to relieve, him being dependent on data processors for the ability to report the data on the system. It can also add to the backlog of user requests within the DP department.

Other report writers leave the ability to make the data views entirely in the hands of the end user. With this type of report writer there are no limits and no need for intervention from the data processors. However, there are no guarantees that the views are being constructed properly.

Should end users have the ability to define all data views or are pre-defined views best? Unfortunately there are no clear cut answers.

For many end users, the pre-defined view approach to the data is the best. Their needs or level of knowledge make the limits placed by views acceptable (or in many cases desirable). In those 4GLs that do not provide point-and-shoot reporting (and data views) it is still possible to provide the same interpretations of the data for the end users. Multi-use files can be described more than once in the data dictionary (one logical description of the file for each use of the physical file with a name that defines that use). For views that will be used often, the code to establish each view can be kept in a file and used when needed (much like the COBOL copylib). The views could also be noted in an end user handbook (e.g. "For a report on Accounts Receivable use the following view statement . . .").

Some end users will need or want reports that require more than can be provided with pre-defined views. In many cases the end users who need that level of reporting will also have the capacity or desire to learn the correct logical structures. Should they be allowed to construct their own views? This decision can only be made on a case by case basis.

Performance

Problem Two

Performance is an area of concern for any system manager. It is amazing how many system managers load a 4GL report writer on their system, make it available to their end users, and then are surprised that the system's performance is degraded. What they fail to realize is that they have added many "programmers" who are compiling and running programs - ON LINE!.

When asked to describe a 4GL, many professionals make some comment on its effect on performance. It can be argued that reports written in some of the 4GLs are no less efficient than the same report written in a 3GL such as COBOL. One thing that many people forget is that allowing end users to run a 4GL report writer on-line is the same as writing a report in COBOL, then having end users type "RUN objtfilename" to create the report.

Solution

In many cases, forcing batch processes to run in the batch queue will solve performance problems. What about point-and-shoot report writers that are made to run on-line? What about end users that are accustomed to running their reports on-line?

Part of what makes 4GLs desirable to the end users is the fact that they are interactive - it is hard to be interactive in batch. The main problem with running batch processes on-line is the way that the HP is tuned. The HP is designed to give on-line users priority over all batch jobs. On-line processes are not penalized as heavily as batch processes when they are CPU intensive. There are two ways to help keep the HP from going to its knees when several 4GL users are working on-line: force those users into the batch queue (DQ or EQ) or overlap the on-line and batch queues (CQ and DQ).

If special groups or accounts are created for those end users who will be running 4GLs, they can be assigned a maxpri of DS or ES. This forces the users who sign on to those accounts into the "batch queue." If it is desirable to allow those users into the CQ during low usage periods (outside of normal business hours), it is a simple matter to create a "high pri" and "low pri" jobstream that signs on as manager of their account and changes the group maxpri levels. An operator can then run these jobstreams as a part of the normal processing day. The only problem with this technique is that it is a "broad brush" approach. Many end users have one sign-on that is used to run application programs and write reports. If the users run any application, 4GL or other, while signed on to the special group, they are forced into the batch queue.

A more refined solution is to use a front end to the 4GL that will place it in the batch queue when it is executed. Some utilities such as MPEX allow you to create son processes in a specific queue. You could use such a utility within a UDC to "spawn" the 4GL in a batch queue. If no such utility is available it is not difficult to write a 3GL "front end" interface that calls the CREATEPROCESS subroutine to start the 4GL in the batch queue.

Another approach is to influence only those users who are running CPU intensive programs on-line. If the bottom of the CQ is moved so that there is a healthy overlap into the DQ, on-line users who place heavy demands on the CPU will drop below the top of the DQ, allowing the batch processes and other sessions to access the CPU.

Some 4GLs offer the ability to limit users to a specific amount of CPU time. For those who don't have that ability, MPE offers an alternative. Again, it is rather a "broad brush" approach, as it can only be applied on the group level. One of the MPE parameters that can be set for each group is a CPU limit. Care must be taken not to set the limit too tight as intervention by the system manager is required when the a user hits his limit. This method is not entirely fool proof as the CPU second count can only be reset at the account level. If one user hits his limit, and is reset, all users will be reset. It is also important to note that the CPU limit is a cumulative counter; it can not be set for a specific process (or report).

It is important to know your system's performance limits. On some systems (especially the Micro/3000s) it may be necessary to limit the number of on-line 4GL users. This is not desirable, but may be the only way to control the system's resources.

Additional Concerns and Remedies

Two other areas that warrant attention are remote data base access and disk caching.

Remote data base access may be necessary on some systems. The overhead necessary to access a data base on a remote system may add more processing time to some 4GL programs. The decision whether to allow remote data base access to 4GL users is one that must be made on a case by case basis.

The addition of disk caching to a system can cut processing time significantly. However, it can also allow a program to work in the CPU for longer periods of time without being interrupted for physical I/Os. This can affect the machine's performance if a batch process is being run on-line. Most reports do quite a bit of I/O. Without disk caching running, each I/O causes an interrupt (which allows other processes to access the CPU). With disk caching on, there are less interrupts, as some of the data is in the cache in memory. If the data is in memory, the process is not interrupted and does not release the CPU. Turning off disk caching can in some instances actually increase overall throughput when multiple processes are running.

Capacity

Problem Three

When capacity planning is mentioned, most system managers automatically think of two areas, CPU/memory and disk capacity. There are areas other than internal capacity where planning and management are important: hardware, human, and MPE capacities can be taxed by adding more "programmers" to a system.

Internal Capacity Considerations

One of the best ways to assess the impact end user computing will have on your system is to "test drive" the software before the purchase is made (and definitely before the product is turned over to the end users). There are resource / capacity planning systems, like HPTREND, available to use after the 4GL is installed.

It is best to phase end users into programming a few at a time. Pick a few advanced end users that can serve as "team leaders" and start them first. Then train more end users a few at a time. This will allow you to control and gage the impact of end user computing on all of your resources.

One standard that can be implemented to maintain your disk resources is a group level limit on file space. This requires assigning a specific sign-on and group to each 4GL user and setting group file space limits. When a user reaches his group limit, they should be required to review the files within his group and purge or archive any unused files before the disk limit is raised.

Hardware Capacity Considerations

In many cases, end user computing generates a greater demand for terminals or workstations. This demand can be met in several ways. The most obvious solution (although not always financially possible) is to buy a new terminal for everyone who requests one. When terminals are limited there are two approaches that can be used. One is to establish work areas with several terminals available for end users to share. Terminals can also be assigned using connect time or cpu time counts to determine the heaviest users. It is possible to determine resource usage with the system log files. The log files can be accessed and reported using programs within the TELESUP account (:LISTF @LOG@.DOC.TELESUP for documentation on these programs), by internally developed programs or with third party systems (some are as inexpensive as $400.00).

There is a finite number of ports that can be added to any HP. The number can be increased by purchasing a larger machine or an additional I/O bay. However, the computer's logical maximum capacity for on-line users may be considerably less than the physical number of ports possible for that piece of hardware. The demand for ports can be managed in the same way that the terminals and workstations are. Another solution is to install a data switch that will allow an almost infinite number of users to contend for a finite number of ports.

When supporting a remote location, data communication problems and solutions are going to relate closely to capacity planning for ports. Another data communication consideration is the use of dial in modems. Often users will have a PC at home that is capable of dialup communication and they will want to use the computer after hours from home. This possibility must be considered and planned for.

A DP manager recently commented that "I gave my end users a 4GL report writer and performance suffered. I upgraded to a Series 935 and instantly became a hero. Now they complain that the printers we have can not keep up with the reports they are running." The addition of end user computing can increase the demand on any system's printers. When adding print capacity, the system manager must consider more than lines or pages per minute. Adding more print capacity in the computer room may only move the bottleneck to the distribution of printed reports. A better solution may be to add distributed printing, printers in multiple locations throughout the company.

Human Capacity Considerations

End user computing, while helping to relieve the backlog of user requests can create new demands for the DP staff. There will be an increased need for technical support. This can be answered by establishing end user team leaders or a help desk in larger organizations. The system manager will see an increase in the demand for almost every resource within his area of responsibility.

The operations staff will be supporting potentially many more "programmers" who are testing reports, requesting special priority in both the print and job queues, demanding throughput and output in a timely manner. Two specific areas within the operator's responsibility must be watched closely when end users become programmers: job run times and spoolfile sizes. Because they often do not know the size of the files that they are working with, it is not uncommon for end users to create reports that run considerably longer or create considerably more output than they expected. The operator must be aware of this possibility and be prepared to check with users when a job has been executing for a long time or has created a hard copy report of monstrous dimensions.

MPE Capacity Considerations

End user computing will increase the number of on-line users and batch jobs on the system at any given time. Because the maximum number of sessions or jobs allowed can not be changed without re-booting the machine, reaching the limit in the middle of the day is undesirable. An easy preventative measure is to set your session and job limit lower than the actual maximum. When the job or session limit is reached, there is a "buffer" that can be used. The actual limit can be reset later during scheduled down time.

Spoolfile size limits are a little more difficult to deal with. The maximum spoolfile size is determined by the extent size that is specified during a system start or load. Each process is allowed a maximum of 32 extents. When the maximum is reached, the process is aborted. The obvious solution is to define a very large extent size. However, this can be detrimental on a system where the disk space is fragmented or limited. The spooler must be able to place each extent on a contiguous piece of free space or the result is the same as hitting the maximum spoolfile size - the process is aborted. If free space is at a premium, another approach can be taken. By limiting reports to smaller amounts of data, or breaking large reports into several smaller reports, the size of the spoolfiles can be controlled. You must know how much free space is on the system and how it is distributed (you can use the program FREE5.PUB.SYS to determine this) before you can decide how best to approach this problem.

Specialized Language Idiosyncrasies

Problem Four and Solutions

While it is not within the scope of this paper to address specific syntax problems, there are some areas that should be addressed with all report writers: construction of data views, sorting, sequential vs. keyed access, and multi-pass programming.

Improperly constructed data views can lead to inaccurate reports. It should be ensured that the persons who construct the views understand the data structures being used.

Unnecessary sorts can adversely affect performance, both of the report and of the system overall. Some end users are not aware of files that are already sorted and apply sorts where they are not needed. Some sorts are unnecessarily complex and can be written more efficiently by someone who understands the data.

A common mistake many end users make is using selection criteria that forces a sequential read of a file when a keyed read could be used. It is important to make sure end users are aware of the keys available. An occasional review of end user written programs can point to data items that may need to be converted to keys.

Most 4GLs are data driven or non-procedural languages, they apply a certain set of commands to each record. Without the records to "drive" the program, no commands will be processed. Many beginning 4GL programmers will try to write programs that do everything in one pass through the data. This is often the proper technique when using a 3GL or procedural language that allows do-while or do-until programming. Most 4GLs lend themselves to multi-pass programming, working through a complicated process one step at a time. Each step is executed in a separate program with the data being passed from one step to the next in extract files. Multi-pass programming can greatly increase a 4GL application's efficiency and throughput.

Standards

Overview

Standards are the best (or maybe the only) way for a system manager to maintain control when end user computing is added to a system. If good standards are in place before the 4GL is in place, all of the problem areas associated with end user computing can be controlled. As presented earlier, standards are written procedures that are followed for and by all 4GL users (both DP staff and end users).

Most of the solutions that have been presented are actually dependent on standards. Specific sign-ons for users, team leaders, pre-defined data views, and end user handbooks are all part of an overall program of standards. A comprehensive set of standards should address MPE, education, the data dictionary and frequently used reports.

MPE

Setting up specific sign-ons, groups and even accounts for the users makes all the power of MPE security and account management available. CPU usage, connect time, disk space, and file access can all be monitored and controlled if users are assigned unique sign-ons.

Teaching end users to release reports in batch jobs can benefit both the end user and the system manager. Batch jobs will allow the end user to better utilize his workstation. When a report is released in a jobstream, the user can perform other tasks which require his workstation. Obviously, the system manager gains a great deal more control when batch jobs are used. When a report is being produced by a batch job, the batch process is being performed in the batch queue. The operator has full control of all batch jobs: when they are released, how many are run at the same time, when and where the reports are printed. Batch jobs can even be suspended if the need arises.

The only problem with teaching the end user how to release reports within batch jobs is that another "language", MPE, must be learned. Some report writers work around this problem by offering a command that will cause a report written online to be released into a batch job. For those 4GLs that do not provide such a solution, it is a simple matter to provide the end user with a skeleton jobstream. Some system managers worry that giving end users access to MPE will promote "hacking." While this can be true in some instances, if the account structure is designed properly, MPE and the file system security prevent most end users from taking advantage of the system.

Education

Required education can be a very effective tool for controlling the impact of end user computing. An effective education program must be more than sending the end users to a class to learn the language syntax. It should include team leaders, classes on syntax and data structures, access to the software manuals, end user handbooks and in large shops an information or help canter.

Team leaders can help to lessen the impact of end user computing on the data processing staff. These leaders should be advanced end users from each area of the company that will be using the report writer. They should be trained in the report writer syntax and data structures that are specific to their work. They will be the first contact for end users who need assistance with the 4GL.

Team leaders should have a specific contact analyst within the DP department. This contact will be their resource when the questions or problems are beyond their abilities. A specific contact is important for several reasons. The analyst will become familiar with the end users and their area. This will lower the possibility of the same problem being solved by multiple analysts. The impact of end user support can be monitored more closely and controlled if the end users do not use the whole DP staff as a resource. Projects that require DP resources can be managed more effectively if end user assistance impact can be planned for.

In larger organizations, it may be easier to provide an end user information canter or help desk. This could be used in conjunction with the team leaders or in place of them. Placing all end user interface in one area allows for the most control.

Most software companies offer some kind of training on their end user computing product. This is an excellent way to introduce the end users to the basic syntax and some of the idiosyncrasies of the 4GL. It is important to remember that in most cases this type of training often uses simple data structures and is aimed only at teaching the language.

When an end user returns from a class and tries to apply what has been learned, without additional training, he is often overwhelmed by the "real life" data structures that he must work with. Therefore, some kind of internal instruction should be provided after the end user has learned the report writer. This instruction should focus on the data bases, structures and relationships that the 4GL user will encounter at work. It should cover possible data views, relationships that are not straight forward, data items that are "hidden," data names that are misleading and known problems with data (e.g. files that contain similar data that may not be in sync and data that is known to be inconsistent).

Technical 4GL manuals should be accessible to all end users. Many people would rather solve their own problems than have to rely on someone else for assistance. This can lessen the impact on team leaders and the DP staff who support them. If 4GL pocket or quick reference guides are available they should be placed at every end user workstation.

An end user computing handbook should be developed. It should contain a listing of the data dictionary or schema and a glossary of data items and files. The glossary should list an English like reference for any data item or file whose use may be difficult to discern from its name (e.g. Employee master - EM103, A/R information - DBAR-DTL, DBAR-MSTR and DBCLIE-DTL . . .). Predefined data views should also be included. If the end users are expected to release reports in batch jobs, a skeleton batch job should be included. It is also a good idea to include some sample reports or specialized code that may be difficult for the end users to write. This will help keep people from "reinventing the wheel." All areas that are presented in the internal training class should be contained in the handbook. It will be most useful if the handbook is designed as a reference document, not a tutorial.

Data Dictionary

The data dictionary or schema that is central to most report writers is in itself a good tool for standardization. If it is designed and properly administered, it can add to the security of the system, act as online help, and make difficult data structures easy to understand. Improper administration of the data dictionary can undermine any other standards that are put in place.

It is imperative that the utility that is used to create the dictionary be kept from the end users. It should be lockword protected and as far as the end users know, it should not even exist. If this capability is given to the end users, they can access any file on the system. The ability to standardize reports, control data views, and support end user computing are at best difficult and most likely impossible if each end user is able to create his own dictionary.

The data dictionary can allow for the standardization of reporting. Most dictionaries provide the ability to format the output characteristics of each data item. Whenever the data item is reported, it will look the same (decimal points are in the right place, the information is separated by dashes between the correct characters . . .).

Some dictionaries include security capabilities that can enhance the MPE and file system security. It is possible to apply security at the data item level with some 4GL systems. Security can be enhanced by using multiple dictionaries when internal dictionary security is unavailable or undesirable. If specific files are to be made available to a limited number of end users, they could be included in a special dictionary. The dictionary access can usually be controlled by file equations. The file equations for dictionary access can be set in a logon UDC. If a dictionary contains files that must be secure, the dictionary itself can be lockword protected.

Difficult file structures can also be addressed in the dictionary. If a physical file has several logical uses, a separate file description can be created in the dictionary for each logical file. Data items that have multiple uses can also be redefined in the dictionary.

Frequently Used Reports

End user written programs that create reports that are utilized frequently must be reviewed. This review should be performed at several levels. The syntax should be reviewed for efficiency, the logic should be reviewed for correctness, the report should be tested using proper testing techniques, and accepted departmental standards should be applied to the code and jobstream. The definition of "frequently run reports" must be created for each shop. A good general rule is, if the report will be used to make business decisions, it should be reviewed.

A review of the report's syntax will ensure that the report is as efficient as possible. This will help overall performance on the system as well as speed the delivery of important information.

End users do not always approach a problem or report from the best angle. Misunderstanding data structures or idiosyncrasies of the report writer can produce an inaccurate report. If business decisions are being based on the information provided, the results can be disastrous. Checking the logic used and running the report using proper testing techniques will ensure that the information being reported is correct.

Applying DP departmental standards to a frequently used report ensures that the report can be maintained. In many cases the end user that creates the report is not the one who runs it. When the creator is no longer with the company, the users of the report will turn to the DP department for support. Reviewing the reports as they are written can save maintenance time in the future.

Conclusion

End user computing can have a positive effect on the DP department's backlog and the user's productivity. The addition of an end user report writer will impact the system manager's areas of responsibility. Machine performance and DP capacities will be affected. Standards are the system manager's best tool for maintaining control when end users become programmers. The end users must be taught to use the language and understand data structures if end user computing is to be successful. A support system must be developed for the new user/programmers. Good standards should be in place and a capacity audit performed before the 4GL is released to the end users. If system impact is not considered and planned for, the addition of end user computing can be a complete new set of headaches rather than the solution to the backlog that all DP shops face.

A System Manager's Checklist

Performance
- Batch processes running on-line
  - group limits
  - front end processor
- CPU limits
  - contained within the 4GL
  - group limits
- Remote Data Access
- Disk Caching

Capacity Planning
- Internal
  - CPU
  - memory
  - disk
- Hardware
  - terminals
  - ports
  - data communication equipment
  - printers
- Human
  - user support
  - system management
  - operations
- MPE
  - job/session limits
  - spoolfile limits

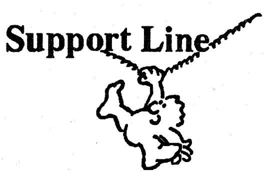
Specialized Language Idiosyncrasies
- Data views
- Sorts
- Sequential vs. keyed access
- Multi-pass programming

Standards
- MPE
  - individual sign-ons, groups and accounts for end users
  - limits on CPU and disk space
  - file system security
  - use of batch jobs

- Education
    - team leaders or help desk
    - syntax
    - data structures
    - software manual access
    - user handbooks
- Data Dictionary
    - secure dictionary creation tool
    - data item formatting
    - security
    - file uses
- Frequently Used Reports
    - syntax review
    - logic review
    - testing
    - install DP standards

HEWLETT PACKARD

Improving Productivity: Putting HP SupportLine to Work for You



Joseph Reves
Hewlett Packard
North American Response Center
Mountain View, California

Hewlett Packard's HP SupportLine is an online, interactive information retrieval and problem solving tool. With its introduction on December 5, 1988, HP offered Response Center Support and Account Management Support customers a tool for accessing the most current, relevant information available about HP products and applications. HP SupportLine was developed to be a window into information collected daily by Customer Support Engineers. HP SupportLine is easily accessed from an HP terminal or PC with HP terminal emulation, and a dial-up modem. Users are uniquely identified by system handle and password.

This paper will present some information and strategies you can use to put HP SupportLine to work for you in your business. Strategies for navigating HP SupportLine, interacting with a Problem Solving Assistant, and submitting electronic calls will be outlined. Formulating a Keyword Search, and using the Menu Search facility to expand your understanding of a topic are also discussed. Additional topics include sections on configuring your equipment to connect, using display formatting options, and printing documents. Finally, a discussion of HP SupportLine's role in helping to avoid potential problems, and in keeping your knowledge of vital support information current will be presented.

OVERVIEW OF FEATURES

```
 ************                                 (Menu) 'MAIN00         '
HP SupportLine
 ************                                 Page 1 of 1
--------------------------------------------------------------------------

                          HP SupportLine Main Menu

   1   HP News Page
   2   Information retrieval
   3   Expert systems
   4   Submit a question to the Response Center
   5   Provide feedback about HP SupportLine
   6   Tutorial
   7   User administration
   8   Leave HP SupportLine



::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1..8 Help Next Previous First Redisplay Menu Topmenu GO PRint
REmember FOrget SHow SET SAve Userlevel Exit
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
Enter selection :
```

HP SupportLine's services are designed to assist in both reactive problem resolution and education to avoid future problems. The News Page provides information about new products and promotions, general HP news items, information and technical tips on existing products, training schedules, and information on current HP PowerPatch releases. The Information Retrieval screen allows users to access the databases which include Known Problems and Resolutions, either by using a keyword search feature or by using a directed search through a series of menus. Problem Solving Assistants allow a user to enter into an interactive dialog walking through a logical trouble-shooting process to resolve a specific problem. The Call Submittal screen allows users to submit calls to the Response Centers online, and receive written replies to their questions. An online Tutorial document provides users with a reference for learning SupportLine which includes sample exercises. A Feedback feature allows users to respond instantly with comments about HP SupportLine.

NAVIGATING HP SUPPORTLINE

```
HP SupportLine provides you with a number of alternative user levels.

   NOVICE    a simple set of HP SupportLine commands
   REGULAR   the full set of HP SupportLine commands
   EXPERT    the full set of commands, with minimal screen information

Commands available are displayed at the bottom of each screen when either
Novice or Regular mode is chosen.

To select the default mode press RETURN at the prompt.  If you are not sure
which mode to use, select NOVICE mode.

Please select user level (default NOVICE; ? for help) :regular

Request completed.


                         Welcome to HP SupportLine
                         =========================


Press the RETURN key when you wish to continue . . .
```

Each time you log on to SupportLine, you will be prompted for your user level: Novice, Regular, or Expert. The default level or mode is Novice. In Novice mode, the command set is restricted to a basic set of commands required to use HP SupportLine. In Regular and Expert modes, additional commands which allow the use of options and user topics are available, as well as some additional commands to navigate. The difference between Regular and Expert mode is the display options; the same commands are available to both users.

HP SupportLine is constructed primarily of menus and documents linked together. As you navigate through SupportLine, think of the menus as branches in a tree, which lead finally to a list of documents to select. The MENU command is used to display the previous menu, or 'branch' of the tree. The TOP command returns you to the Main Menu, the first menu displayed when you logged on. The NEXT and PREVIOUS commands are used to move within a multipage document or menu. Pressing the RETURN key is the same as typing the NEXT command. On the last page of a document or menu, NEXT returns you to the previously displayed menu.

```
 ************                            (Internal) 'SHOW
 HP SupportLine
 ************                            Page 1 of 3
-----------------------------------------------------------------------------
                        User-Defined Topics

COBOL INTRINSICS Calling Intrinsics from Cobol - menu search
2680             2680 Printer Configuration - document
NEW PRODUCTS     New Product Information topic - menu

                       System-Defined Topics

ES               Expert Systems
QA               Questions & Answers
AN               Application Notes
FEEDBACK         Submit feedback comments to HP
PASSWORD         Change password
CALL             Submit a PICS call to the Response Center
REPORTS          Company and user session usage reports
::::::::::::::::::::::::::::::::::::::::::::::::::-->> Press RETURN for more...
Help  Next  Previous  First  Redisplay  Menu  Topmenu  GO  PRint  REmember
FOrget  SHow  SET  SAve  Userlevel  Exit
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
Enter selection :
```

The REMEMBER, FORGET, and GO commands allow you to avoid stepping through the menus sequentially, and go directly to a menu or document. The SHOW command will display both user-defined and system-defined topics to which you can GO. System-defined topics exist for most services available from the main menu, and frequently accessed areas as well. You can use the REMEMBER command as you browse through HP SupportLine to 'bookmark' documents or menus you want to come back to later. You can also use REMEMBER to mark documents for other users at your site to retrieve later. The REMEMBER command is not valid at a Keyword Retrieval menu, but will work within the Menu Search selections. The REMEMBER command can always be used to mark a document, no matter how it was retrieved.

Note above that the first page of output from the SHOW command lists both topic names and a brief description of the topics. You must use the FORGET command to delete a user defined topic. User defined topics are retained from session to session, making it possible to set up a permanent list of topics your site uses frequently.

Setting Display Options

```
 ************                                 (Internal) 'SHOW          '
 HP SupportLine
 ************                                 Page 3 of 3
-----------------------------------------------------------------------------

                   Current Configuration

Screen Layout:                         Screen Control:
   Header Format        FULL              Display One Item Menus   NO
   Trailer Format       FULL              Clear Screen             YES
   Text Page Length     15                Pad Last Page            YES
   Menu Page Length     15

Document Layout:                       Terminal Information:
   Menu Header          YES               Terminal Type            ???
                                          Printer Connection       ???
Edit Control:
   Editor               Screen

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
Help  Next  Previous  First  Redisplay  Menu  Topmenu  GO  PRint  REmember
FOrget  SHow  SET  SAve  Userlevel  Exit
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
Enter selection :
```

The SET command allows you to customize the way a menu or document is displayed on your screen. The display on your screen is made up of several different parts you can chose to display or disable.

The **Header** information displays the * HP SupportLine * banner, and the current menu or document name you are displaying. The page number for multipage documents or menus is also a part of the Header, as well as the total number of entries or pages contained in a menu or document. The Header is four lines.

The **Trailer** information displays the commands you can enter from the current prompt, and prompts you to press RETURN to continue to view a document. The Trailer is four lines.

The **Menu Header** is the title of the current menu, and accounts for three lines. Note that Menu Page Length is the total number of lines in the menu, including the Menu Header. If you turn Menu Header off, the number of items listed in the menu increases by three.

The **Pad Last Page** option is used to pad the **Text Page Length** or **Menu Page Length** with blank lines to display a full page of data. This generates a clean, easy to read display with Headers and Trailers always displayed at the top and bottom of the screen. By turning on CLEARSCREEN, and turning off LASTPAGEFILL, you eliminate the unnecessary blank lines but continue to work with a single screen of data at a time.

You can save some time and eliminate some data communications overhead by turning off the parts of the display you don't need. Here are some suggestions:

New Users: Use the REGULAR user mode with the default display options. This offers a full command set, with enough information to navigate comfortably.

Proficient Users: Set HEADER off, but keep TRAILER for the context-sensitive list of available commands. Increase TEXTLENGTH and MENULENGTH to 19 for a standard 24 line display. Use REGULAR user mode.

Expert Users: Set user mode to EXPERT, then increase the MENULENGTH and TEXTLENGTH to 22 for a standard 24 line display. You eliminate the extra overhead, and work with a full screen of data.

If you are accessing HP SupportLine with a PC command file or script, you can eliminate HEADER, TRAILER, CLEARSCREEN, and LASTPAGEFILL, then increase TEXTLENGTH and MENULENGTH to 32767. This will allow you to capture entire menus or documents without prompting you to press RETURN.

HP SUPPORTLINE FEATURES

Retrieving Information

HP SupportLine Databases

The HP SupportLine databases include several types of documents. **Known Problem Reports (KPR's)** document problems that Hewlett Packard is currently aware of and working to resolve. This information is currently distributed in the **Systems Status Bulletin (SSB)**, mailed to Response Center Support and Account Management Support customers twice monthly. Known Problem Reports may be created by customers, Field Engineers, or Response Center Engineers. **Application Notes (AN's)** discuss in broad terms the application of a particular product or feature, and generally include examples. An **Engineering Note (EN)** states a specific problem, and a resolution for that problem. A **Question and Answer (QA)** is similar to an EN, but may discuss a question in more general terms. Application Notes, Engineering Notes, and Questions and Answers are created by Response Center Engineers . In the case of Engineering Notes, actual Response Center calls are used to create documents which are technically reviewed and then formatted for the database. Each database is updated daily, except the SSB database which is updated monthly.

Browsing the Databases

Obviously, you will want to use the Information Retrieval system when you are seeking the answer to a specific question or problem. You can also browse the databases to retrieve information that can help you avoid potential problems. Application Notes (AN) and Questions and Answers (QA) can give you and your users information which can be used in planning configuration of new products, preparing for the installation of new software, or designing new applications. HP SupportLine offers fast, simple access to previous issues of Application Notes you may not already have. Start by selecting the databases you wish to review, then search on existing products or equipment you have installed at your site. As you browse, use REMEMBER to mark topics you wish to come back to.

```
 ************                               (Menu) 'R
 HP SupportLine
 ************                               Page 1 of 1
-------------------------------------------------------------------------------

                        Information Retrieval

    1   Application menus
    2   Data communications network menus
    3   Hardware menus
    4   Language menus
    5   Operating systems menus
    6   KEYWORD SEARCH of databases and System Software Bulletins
    7   KEYWORD SEARCH of SSBs only




::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1..7  Help  Next  Previous  First  Redisplay  Menu  Topmenu  GO  PRint
REmember  FOrget  SHow  SET  SAve  Userlevel  Exit
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
Enter selection :
```

Types of Database Searches

There are two ways to search HP SupportLine databases. The first type of search is a menu search, which is a directed search through a series of menus. The Information Retrieval menu breaks down the menu search options into broad categories of documents, then narrows the product or area within a product to produce a list of documents which are all related. For novice users, the menu search options offer rapid access to the specific area of the database most likely to provide a solution. While exploring the menu search options, both novice and experienced users are likely to find new ideas or areas of information worth searching. The REMEMBER command can be used as a 'bookmark' during a menu search, to allow you to return to an interesting menu you want to follow a different way. Use the SHOW command after your search is complete, to display these bookmarks.

The second type of search is a keyword search, also accessed from the Information Retrieval menu. In a keyword search, you supply words which are likely to appear in the documents you wish to retrieve. Start with a broad scope, then refine your search string to be more specific. In general, your primary search keys will define the products involved, and in refining your search you will add words to represent the feature or aspect of the product you are interested in. You can also characterize your problem to further narrow your search.

You can specify individual databases to search by using the GO AN, GO EN, GO QA, or GO SSB commands. You will be presented with a menu which will allow you to further narrow your search by product type, then initiate a keyword search against that specific database. You can also enter the type of document in the search pattern. For example, 'AND RCEN OR RCQA' with your search string searches Engineering Notes and Questions and Answers. Note that the GO SR command allows you to search for specific Service Requests by number.

Formulating a Keyword Search

To begin your keyword search, first consider the databases which are likely to meet your needs. For example, if you are looking for general information on a topic, you may want search only the AN database and the QA database. If you are looking for a resolution to a specific problem, searching the EN database may offer the best results. If you are not sure, search all databases and narrow your search later. Use patterns which describe the product your question deals with, or use the text of the error or failure you have experienced. Use the wildcard character '@' to suffix words with several endings. For example, use the pattern 'CONFIG@' to retrieve documents containing the words configuration, configure, configuring, and configured. This first search pass should also contain synonyms for your primary patterns. The pattern 'TURBO@ OR IMAGE OR DATABASE OR BASE or DB@' will retrieve most of the documents that can answer questions about TurboImage databases.

After constructing this initial search pattern, you can further narrow your results by using words that describe the feature, command, or aspect of the product you have defined in the first pass. To edit your current search pattern, type 'SEARCH *' and you will be prompted to make changes with simple line edit commands. You can also enter words to characterize the problem you are having. For example, the words 'ABORT, HANG, FAIL@, ERR@, TERMINAT@' all describe the kind of problem or question rather than what the question is about. Use the 'NOT' logical operator to exclude groups of documents that don't relate to your search. The pattern 'TURBO@ NOT TURBOSTORE' will eliminate an entire product from your search. To simplify your search, always put OR relationships first in your search pattern. The precedence for evaluating logical operators is: OR then NOT then AND. See the HP SupportLine User Manual or the Help facility for a complete description of the SEARCH command syntax.

Once you narrow your search to what you consider to be a manageable list of documents, you can use the PRINT DOCUMENT command to produce a hardcopy of the menu and the search pattern that produced it. This can be valuable for later searches; the reason REMEMBER and FORGET don't work in the keyword retrieval system is that you are building your own menu as you search. Documents are added to these databases daily; a future search with the same pattern may offer different results. With a hardcopy of the complete menu of documents, you can then select the documents you wish to review. Individual documents can be marked with REMEMBER and FORGET. In reviewing the documents your search produced, consider the type of document again against the type of information you are seeking. You may wish to run the same search pattern against a different set of databases. When browsing the documents you have located, use the MENU command to return to the search menu. Using the TOP command will cause you to exit the Information Retrieval system, and you will lose the results of your search. Note that returning to the IR menu will also eliminate the results of your search.

Using Menu Search

Begin your menu search by selecting the appropriate category from the Information Retrieval menu. Later in the menu search, you may have an opportunity to characterize the type of problem you are having. As you progress through subsequent menus, note choices or branches you may want to explore later, and REMEMBER them with a name that will remind you. Also note terms that you may be able to use in a keyword search later. You will probably find that using a menu search will teach you how to formulate a keyword search, and provide you with keywords. When you reach the menu of documents that deal with your question, REMEMBER this menu if you intend to refer to it again. You can use PRINT DOC here as well to produce a hardcopy of this list of documents.

Submitting a Call

The first step in deciding to submit an Electronic Call is to define the problem clearly. As you go through this process, you will find that you also have the information you need to run an Information Retrieval. Before submitting a call, you should always attempt a search of the databases. Not only may you find the answer to your problem immediately, but you may also find information that could help you define your problem more clearly. This can save time later if you need to place a Response Center call, and make your call more efficient.

Defining the Problem

To define your problem, first document the symptoms that indicated to you there was a problem. Try to associate these symptoms with some action or condition that exists to produce them. For example, "When I issue this command, I get this error..." describes the conditions that produce the symptoms of the problem. Document all the products, programs, and subsystems involved, the operating system, and the version or level of each of these. Try to remember when this problem began, and document anything that may have changed in your environment at about the same time. Document anything that you do which impacts on the problem, for example: "When I turn off Disc Caching the problem goes away." Include any opinion you have about what may be causing the problem, then include any information that supports your theory. Finally, state the results you expect in the resolution of the problem. What do you expect to be able to do? Would some type of workaround meet your needs?

An advantage for PC users accessing HP SupportLine with communications software such as AdvanceLink is the ability to prepare a problem description offline, and then transmit the text to HP SupportLine. You can capture error messages, sections of code, or other text and use a feature such as AdvanceLink's &SENDF to transmit this text line by line when prompted for a problem description. Job $STDLIST files, version level displays, or screen print output are other examples of text you can enter this way. You could also use your favorite text editor to produce the problem description, then save the file in an ASCII format and transmit it.

Types of Calls

The two types of calls you can place through HP SupportLine request either a two hour response, or a twenty-four hour response. Calls are logged by the same coordinators that log phone-in calls and are assigned a code to indicate the type of response desired. Engineers working on your HP SupportLine call will contact you by phone if additional information is required to answer your question. If you have requested a two hour response, an engineer will begin working on your call within two hours and will attempt to contact you to either obtain more information or discuss the resolution with you. Be sure to leave any special contact instructions in the text of your call description.

Your problem or question can be resolved entirely electronically if your question is "closed-ended". That is, the scope of the problem can be addressed completely with the information you initially supply. A question or well-defined problem with a single answer or limited set of options can be handled by an engineer without scheduling phone calls or dialing into your system. Even if you submit a call electronically, you can still have a dialog with the engineer to obtain more information or test different solutions. By submitting your call electronically, you are giving the engineer a "head start" much the same way as leaving a detailed voice mail message. The engineer can start right away to seek an answer with the information you have supplied. Submitting a call through HP SupportLine also means that you will receive a written reply to your question. This can allow you to maintain a personal problem resolution history for your site.

Urgent Calls

Obviously, if your problem is urgent you should place a phone call to the Response Center. If your system is down, or your production systems are adversely impacted, place a call and request your call be given a high priority. Often, customers call the Response Center for assistance defining the problem they are experiencing. The Response Center can work with them to develop an action plan to clearly define the scope of the problem, and the steps required to resolve it. These types of calls are probably more appropriately placed over the phone.

HP News Page

```
 ************                                     (Menu) 'NEWSMENU
HP SupportLine
 ************                                     Page 1 of 1
---------------------------------------------------------------------------------

                                  News Page

   1   New products - this month
   2   New products - last month
   3   General News
   4   Promotions
   5   Price changes
   6   Existing product news
   7   Technical tips of the week
   8   Training classes and shows
   9   HP contacts
  10   PowerPatch - available versions


:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1..10 Help Next Previous First Redisplay Menu Topmenu GO PRint
REmember FOrget SHow SET SAve Userlevel Exit
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
Enter selection :
```

News Page offers the most current available information on a variety of topics. New product announcements for the current month and last month are accessed from the News Page menu, as well as new information on existing products. New pricing and promotional information is also available in this section. The Technical Tips of the Week selection offers a menu of new documents in a 'mini-Application Note' format easy to scan quickly. Version information for the current available PowerPatch releases can be obtained at this menu. General news releases, a list of HP contact numbers, and current information on training and trade shows rounds out the selections on this menu.

The Technical Tips selection is one of the more popular features on this screen. Technical Tips are designed to anticipate some of your questions about applications or products and address them before they become serious problems. The format is that of a Question and Answer or a 'mini-Application Note' which can be scanned quickly in a few screens. Reviewing these documents could even give you new ideas for applications you had not considered.

Using a Problem Solving Assistant

```
XPDL Interpreter A.01.04     Copyright Hewlett Packard Co 1988

                        MPE Job Stream Assistant (A.02.00)
                     Copyright Hewlett Packard Company, 1988
------------------------------------------------------------------------------

This assistant acts as an aid to HP3000 operators who are having difficulties
with jobs.

The assistant will ask you to perform operations on the machine you are having
difficulties with and will ask questions about what happens, in order to
diagnose and correct the problem.

Some of the operations need to be done on the system console and require OP
capability.

You may stop the assistant at any time by typing "EXIT" or "E" at a prompt.

If you are unsure what a prompt means type "?", "H" or "HELP" for more
information.

Does MPE give an error message when you use the STREAM command ? >>
```

Several Problem Solving Assistant programs are available within HP SupportLine to help you learn a problem solving process for several types of problems. By entering into a dialog with an Assistant, you can be prompted to gather information and attempt solutions for particular problems. By demonstrating a logical process you can use to solve a problem, the Assistant can help you to generalize and apply the same process to similar problems. A dialog with an Assistant can also provide you with a checklist of items you can use to define your problem more clearly.

You may want to log your dialog with an Assistant to your local printer, to use for teaching problem solving techniques to other users at your site.

ACCESSING HP SUPPORTLINE

To access HP SupportLine you need an HP ASCII terminal or a PC with HP terminal emulation software. A terminal with 'HP Personality' is required to handle some limited terminal control sequences HP SupportLine uses. HP SupportLine does not use HP softkeys or extensive terminal control sequences. The intention is to avoid excessive communications overhead at low speeds, and to support a broad variety of HP terminals. Printers connected to the terminal via serial, parallel, or HPIB connections are also supported, as well as internal printers. Terminal/printer combinations are validated at the time of a PRINT request. Use the SHOW command to display the current printer and terminal settings. Use the SET command to configure your terminal and printer selection, and SAVE to retain the current settings permanently.

You will need an asynchronous dial-up modem which conforms to either the Bell 212A standard, or the CCITT V.22bis. standard for modem interconnection. Supported transmission speeds are 2400, 1200, and 300 Bps. In addition, MNP class 5 data compression/error correction is supported with V.22 connections. Connect the modem to a serial port of the terminal or PC using a 13242N or an equivalent 'straight through' cable. Use these datacomm settings: 2400, 1200, or 300 baud, zeroes parity, 7 data bits, and one stop bit.

One advantage to using terminal emulation software with a PC is the ability to capture data to the disc. Logging data to a disc file for later review can save long-distance charges by minimizing time spent on-line reading the results of a search. The AdvanceLink software package from HP offers this capability. Documents can then be printed or uploaded into HP DESK for distribution to your users. Even if you don't have a PC, logging to a connected printer allows you to review information at your leisure.

Another advantage to accessing HP SupportLine with a PC is the ability of many terminal emulation packages to support scripts or command files. Users can write files of commands to dial up, log on, and issue commands to HP SupportLine. The command file could log the results to a disc file or drop the user into the area they wish to access.

ACCESSING HP SUPPORTLINE WITH A PC

To summarize some of the advantages for users accessing HP SupportLine from a PC with HP Terminal emulation software such as AdvanceLink:

- You can use your PC to 'capture' documents as ASCII text files, then distribute these files to your users via HP DESK. You could also include these files in a manual of procedures for your operators or programmers.

- You can write command files or scripts to automatically connect to HP SupportLine, or to retrieve data from SupportLine. You could issue commands to HP SupportLine within a command file which would customize the user interface, depending upon which user at your site is accessing HP SupportLine.

- You can use a command file to set up your own local softkeys to navigate HP SupportLine. This allows for rapid access to function keys you configure locally, to suit your own environment. Even users accessing HP SupportLine via a terminal can load softkeys locally to make navigating HP SupportLine easier.

- You can prepare the text of a problem description offline, in your favorite text editor, then save the text as an ASCII file and send it to HP SupportLine when prompted for the problem description. You can include any supporting documentation in an ASCII text format, such as error messages or excerpts of program source code.

- You can capture menus and documents to disc to review later, thus reducing your long-distance access time to HP SupportLine. Once you select the documents you wish to review, you can connect to HP SupportLine and go directly to retrieve exactly what you need.

OTHER RESOURCES

HP SupportLine is only one part of the total problem solving resources available to you. Using the correct resources for your question or problem, and using them appropriately, will help you run your operation more efficiently.

The **Reference Manual** set is the first detailed source of information about any product, and is shipped with the product. Extracting the information you need from a stack of several manuals can seem a very formidable task. Start with the user's guide for a product, and read the chapter you will probably find at the beginning of this manual on general information and getting started. Then skim the Table of Contents for the specific information you will need to install the product and begin to use it. Use the Reference Manual to answer specific questions you have about terms or procedures mentioned in the User Guide. Concentrate on learning where information can be located, rather than trying to memorize all the information presented. Use the Index to do your own 'keyword search' to locate important items quickly. Use a highlighter, or insert tabs to help you remember where to find frequently accessed topics.

HP LaserRom is a PC-based product which accesses a Digital Compact Disc to access HP manuals. HP LaserRom offers the ability to use an on line keyword search through the text of a selected manual set to retrieve sections for you to review. The advantages include speed and the ability to search several manual sets at once. Illustrations can be displayed, and the results of a HP LaserRom search can be printed or stored in a disc file. HP LaserRom is updated monthly.

HP SupportLine differs from HP LaserRom in that the HP SupportLine databases contain problem solving information. HP SupportLine is accessed by dial-up modem and also offers the ability to search for keywords in the text of documents. If you are unable to locate an answer to your question by using the resources in the Information Retrieval area, you can submit a call to the Response Center through SupportLine. Problem solving information in the HP SupportLine databases is updated daily.

The **Response Center** is designed to be your central contact for software problem solving assistance. The Response Center can react to your critical problem within 15 minutes and can coordinate all the resources available to resolve your support requirements. Response Center engineers use problem solving data in the HP SupportLine databases to answer your questions and generate new documents to add to this database from your calls. If the information you need is available in HP SupportLine, engineers will tell you they found it there. Our objectives are to help you be more self-sufficient, and to reduce your need to contact the Response Center by providing information to avoid problems.

Finally, your **local Account Team** is your face-to-face contact with HP. Your Account Team can help you obtain any of the resources discussed above, from ordering manuals to obtaining HP SupportLine access. More detailed new product information can also be obtained from your Account Team, as well as customized support services and consulting.

Support Line

HP SupportLine is a tool which allows for reactive problem solving as well as active planning to avoid potential problems. You can use HP SupportLine to keep abreast of new product developments and applications, as well as to answer your questions about existing ones. By extending the Response Center's services with HP SupportLine, we move away from simple response, and toward actively managing the Support Information needs of the future.

Matching Disk Storage Technology With HP 3000 Systems

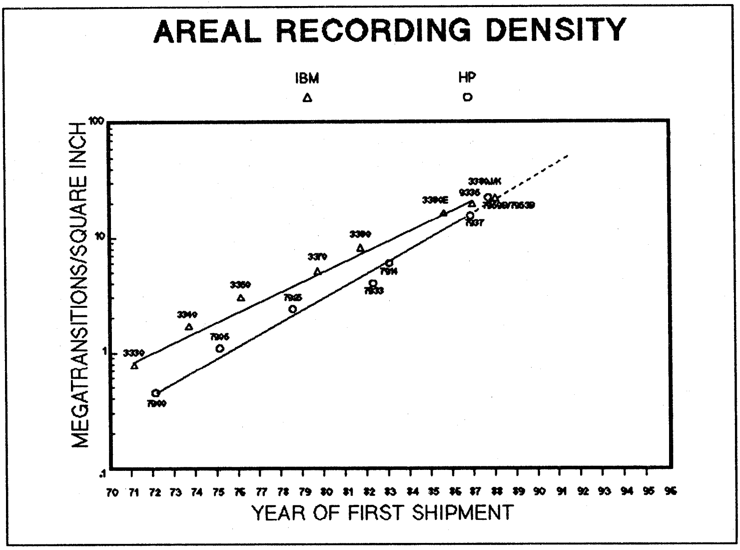
Margo Whale
Hewlett-Packard Disk Memory Division
11413 Chinden Boulevard
Boise, ID 83714

Primary on-line disk storage is an integral part of the total system's capacity, reliability, performance and cost. Whenever a disk drive is purchased, the product chosen is the result of some tradeoff between these factors. System managers will want to make decisions that optimize the capacity, reliability and performance of their system at the lowest cost. This paper identifies the issues by taking a look at industry trends, and identifies the alternatives by a brief review of system architecture differences and HP disk product differences. Finally, it discusses a method of establishing a common cost value for reliability and performance as a means of comparing disk product tradeoffs and provides a simple example with two imaginary products.

Industry Trends--The Issue

The basic principles of magnetic recording technology have remained essentially unchanged since the 1950s. Nonetheless, products have consistently evolved in capacity, price per megabyte, performance and reliability. Areal density (which is how many bits can be stored per square inch of recording surface) largely determines the improvement in capacity, product size, and cost per megabyte. To date, IBM has led the way in developing the recording advances that have sustained the areal density improvement in disk drives. Beginning with the first IBM disk drive in the mid-1950s, the magnetic areal density of leading edge products has consistently increased at 27% per year in millions of magnetic transitions per square inch. This has resulted in a 21% per year decrease in price per megabyte. HP products have followed this trend as shown in Chart 1.

[Chart 1]



Industry performance trends, however, have not matched the areal density and cost per megabyte trends. In a multiuser or multiprocessing environment, the critical disk performance parameter is I/O transactions per second per megabyte, known as "access density." Since a high capacity disk usually has more applications or users accessing its data than a low capacity disk, it must be proportionately faster to give equivalent performance. For example, even though the IBM 3380K with 7.5 gigabytes of storage is one of the fastest drives in the industry, IBM customers have found that they cannot fill the drive even close to capacity without adversely affecting system performance. Similarly, early versions of the HP 7933 had lower access density than the HP 7925 and customer satisfaction was less than optimal until 7933 performance was improved through features such as rotation position sensing. The HP 7937 had similar access density as the HP 7933. Further performance improvements in the 7933 and 7937 were obtained by disk controller caching, which reduces the frequency with which a seek and latency are required in handling a disk transaction. Future improvements in access density within the industry are not expected to keep pace with the 27% per year areal density improvement.

Reliability is another area of high importance to customers. Industry reliability data is generally unavailable because of the lack of a standard and the confidential aspect of actual data. HP has adopted a goal of 10 times improvement over 10 years which equates to an improvement of 26% per year. Although HP disk drives have been able to exceed this rate of reliability improvement and attain industry-leading reliability, this rate in the industry is still slightly less than the 27% per year areal density improvement. As computer systems continue to configure more and more disk devices, this necessitates ever-improving reliability just to keep the overall system reliability constant.

The discrepancy in the rate of industry improvement in areal density with the rate of improvement in access density and reliability should raise some questions for the discerning system manager: "How can I continue to take advantage of lower cost per megabyte trends without affecting overall system performance? If I purchase lower capacity disks to get better access density, am I reducing the reliability of my system?" We need to approach these questions by first clarifying some differences in system requirements and disk drive attributes.

HP System Differences

The HP 3000 MPE XL operating system is different from the older MPE V/E operating system in terms of configuration capabilities and disk I/O requirements. The Series 70 (which is the largest MPE V system available) is compared with the Series 950 running MPE XL in Chart 2.

[Chart 2]

HP SYSTEM DIFFERENCES

	MPE V/E (T-Mit) (Series 70)	MPE XL (1.1) (Series 950)
# Spindles	24	48
Maximum Gbytes (Assumes HP 7937)	13.7	27.4
Typical Gbytes	2.8	4.0
Typical I/Os/second	60	16-24
Peak I/Os/second	>100	48-72

The number of spindles which can be configured on the HP Series 950 is greater than that supported on the HP Series 70. Assuming both systems support the same capacity of disk drive (571 Mbytes assumed), the Series 950 results in a greater number of supported gigabytes. Total gigabytes supported on these systems can be increased through support of higher capacity disk drives and support of more spindles. On MPE XL, the maximum disk storage will be dramatically increased through both methods. Typical gigabytes found on the Series 950 today are 4 to 4.5, much lower than the maximum possible. The typical I/Os per second and peak I/Os per second of the two systems are based on an industry benchmark that resembles actual customer site data in illustrating I/O channel requirements. These data point out that the MPE XL operating system offers an advantage over the MPE V/E operating system in reducing the number of I/Os required of the disk devices.

This reduction in total disk I/O is a result of the following: mapped files, for efficiently reducing the number of disk reads; transaction management, for reducing the number of disk writes; and dynamic reads and gathered writes, for increasing the average I/O transfer size and thereby decreasing the overall number of disk I/Os.

Therefore, since the maximum configuration capabilities are different between MPE V/E and MPE XL and since the typical I/O requirements are different, the tradeoffs between capacity, performance, and reliability must be evaluated on an individual system basis.

HP Disk Product Differences

The disk products which can be selected for your system are dependent upon the products which are supported on your operating system. Chart 3 gives pertinent information as of May 1989 for a selection of HP products. You should check whether these products are supported on your operating system when making comparisons for yourself.

[Chart 3]

HP DISK DRIVE ALTERNATIVES
(5/89)

	7963B/97963B	7937H	7937FL	7937XP
Spindle capacity	304	571	571	571
Price	$8575/$6000	$15,700	$16,250	$16,500
SMMC	$34/33	$50	$50	$50
I/Os/second	35	32	32	66 *
MTBF	50K (1); 24K (3)	70K	70K	70K

*Assumes read hit rate of 70% and read percentage of 70-75%

Hewlett-Packard considers product attributes such as spindle capacity, performance, and reliability when evaluating which products to support and recommend on HP systems. When determining whether a product is appropriate for an office environment, HP also considers acoustic noise to a bystander as measured in decibels. The power requirements and heat generated by products are other factors which affect a disk drive's suitability for a particular environment. This paper will assume that the environmental suitability of products for a MPE data center is already established in order to focus on the tradeoffs between capacity, price, performance and reliability.

Disk Storage Costs

In order to evaluate the tradeoffs between different disk products, we must establish a common means of measurement. This paper proposes to convert all tradeoffs to a dollar value whereby the product with the lowest cost will summarize the greatest overall value for the capacity requirement of your system. When purchase cost, maintenance cost, reliability and performance have all been reduced to a common denominator, we can compute the total cost for a given capacity of disk storage for the variety of products which we may consider as alternatives. A five-year period is used for comparison purposes. In order to generically illustrate the concepts, this paper will compare two imaginary products which would be configured on an HP 3000 Series 950 MPE XL system. The product attributes for Product A and Product B which will be used in this paper are shown in Chart 4. Notice that on a per-megabyte basis, Product A's purchase cost is lower than Product B's, Product A's service cost is lower than Product B's, and Product A's I/Os are higher than Product B's. The Mean Time Between Failure (MTBF) for Product A is the same as Product B on a per-spindle basis. Comparing these two products will help us observe the tradeoffs between disks of two different capacities.

[Chart 4]

DISK DRIVE ALTERNATIVES

	Product A	**Product B**
Spindle capacity	400 Mbytes	800 Mbytes
Price	$7,500	$17,000
SMMC	$35	$60
I/Os/second	35	32
MTBF	70K hours	70K hours

Whereas purchase cost and cost of ownership are objectively provided through corporate price lists and support price lists, performance and reliability costs are subjectively specific to individual customers. This means customers will have to establish Their own values for performance and reliability. This paper will offer some suggestions for computing these subjective costs.

Cost Calculations

Purchase price and maintenance cost of the imaginary products being compared are based on the disk attributes found in Chart 4. The cost of each disk must be multiplied by the number of drives it takes to total 4 gigabytes on the system. To obtain this number we divide the total megabytes of 4000 by the number of megabytes per disk in Chart 5.

[Chart 5]

PURCHASE & MAINTENANCE COSTS

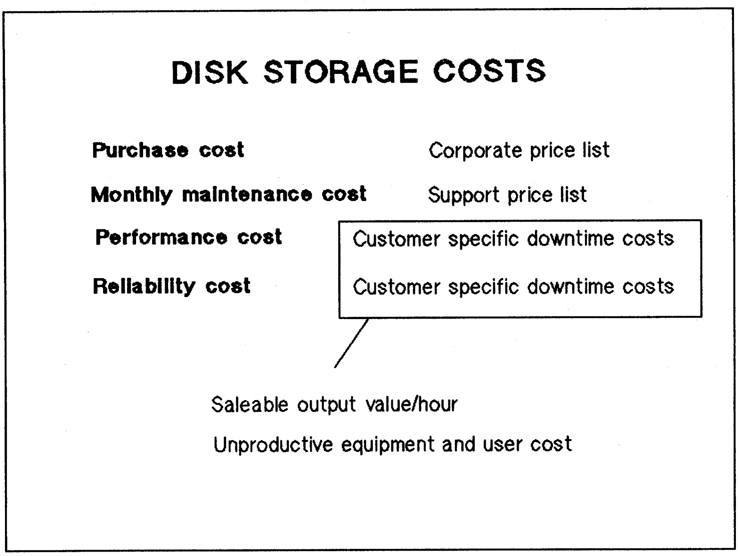
4 Gigabytes = 4000 Megabytes

	Product A	Product B
Total Capacity	4000	4000
Capacity/Drive	400	600
Number of Drives	10	5
Purchase Price/Drive	$7,500	$17,000
Total Purchase Price	$75,000	$85,000
SMMC/Drive	$35	$60
Total SMMC	$19,950	$17,100

It would take ten Product A's (400 megabytes each) and five Product B's (800 megabytes each) to total 4 gigabytes. Only 57 months out of the five-year (60-month) period are used as service months since these products would both have a 90-day on-site warranty. The total purchase and maintenance costs for the two products (Product A=$94,950 and Product B=$102,100) result in an appearance of lowest cost with the Product A alternative. The lower maintenance cost on Product B is not enough to make up for the lower purchase price of product A. Reliability and performance costs are more difficult to quantify; however, these costs can dramatically alter the final cost.

One way of determining reliability and performance values is to consider the cost of lost output in the case of unplanned downtime. These costs may be computed based on the value per hour of lost output in production or services or based on the cost of unproductive facilities and manpower.

[Chart 6]



A survey of 39 Series 950 system managers was made by the HP Commercial Systems Division from October to December 1988. Most were production sites averaging 69 jobs/sessions, 96 megabytes of memory and 4.5 gigabytes of disk storage. Three-quarters of the sample ran 7-day shops and over half had three shifts. The data collected were to determine current system availability, future availability needs and customer preference for data backup. This paper describes only a portion of the information collected in order to provide an example of disk costs. Customer responses on the cost of unplanned downtime were received from 16 customers. Estimates of unplanned downtime ranged from $200 to $40,000 per hour. The cost for 13 customers ranged from $200 to $5000 with a mean of $2,080. For the purposes of this paper, the round number of $2,000 will be used as the cost per hour of down time.

In the same survey, customers were asked to provide an estimate of typical downtime for a disk failure. Only six of the 39 customers surveyed had experienced a disk failure. The average estimate was 10 hours, 41 minutes. Actual estimates ranged from 1 hour, 11 minutes to 25 hours, 5 minutes. Chart 7 provides a breakdown of the downtime estimate according to individual downtime component.

[Chart 7]

TYPICAL DOWNTIME ESTIMATE
(Due to disk failure)

	Hours:Minutes
In-house diagnosis	0:24
Call HP	0:09
CE comes on site	3:26
CE diagnosis	1:20
System reboot	0:29
Data reload	3:07
Data Recovery	1:46
Average downtime per failure	10:41
Range (N=5)	1:11 to 25:5

For the purposes of this paper 10.7 hours will be used as the estimated downtime for a disk failure. If a customer has experienced disk failures, an average of the downtime experienced by that customer may be more meaningful.

Computation of disk reliability cost is the product of the estimate of downtime hours multiplied by the estimate of cost-per-downtime hour. The cost-per-downtime hour has already been assumed to be $2000 for this example. The number of downtime hours must be computed for each product evaluated. The average number of failures during the period, multiplied by the hours of downtime per failure, produces the average number of downtime hours. The average number of failures for each product is the quotient of the sum of total power-on hours per drive divided by the (MTBF) per drive.

[Chart 8]

DISK RELIABILITY COST

Cost due to disk failure = Downtime hours x cost-per-downtime hour

where

Total downtime hours = Number of failures x downtime hours/failure

Number of failures = $\frac{\text{Sum of drive power-on hours}}{\text{Drive MTBF}}$

Cost-per-downtime hour is customer specific

When calculated for Product A and Product B (each with a MTBF of 70,000 hours), this computation results in an average of six Product A failures over the five-year period and three Product B failures over the same period. Assuming 10.7 hours of downtime per failure at a cost of $2000 per hour, the reliability cost for Product A is $128,400 and for Product B $64,200. This example demonstrates that a system with a disk drive which is one-half the capacity but has the same MTBF may cost twice as much in downtime due to disk failure. When estimating disk failures, keep in mind that MTBF is an average, or mean, describing when failures have occurred and may be measured differently by different disk manufacturers. Actual failures may occur either before the mean is encountered or after.

It is clearly evident from the resulting reliability costs that as the number of disk failures increases, the reliability cost rapidly increases, even to the point of exceeding the cost of purchase and maintenance.

Computation of disk performance cost is a more difficult matter. Product A offers greater access density (more

I/Os/second per megabyte) than Product B, which makes it appear that it could offer greater overall performance. However, a loss of performance can only be attributed to disk drives if the system throughput is limited by the amount of disk I/O. That means that the disk drives are incapable of keeping up with the read and write requests which are made by the system. Looking at it another way, it means that the system would have higher throughput if only the disk drives could perform more I/O transactions.

In most tests, the disk I/O is rarely the bottleneck on system throughput. Operating system differences and application software differences affect the number of I/O requests to the disk drives as well as whether I/O requests can be made of more than one disk drive at a time. If the software cannot utilize disk concurrency (many disks reading or writing at the same time), or if the frequently accessed data is not distributed evenly among multiple disks, it is more likely that individual disk performance would become a factor. Whether the disk can handle all the requests from the system is also dependent on the block size of the data transferred and on interarrival time (how close the requests are to one another from the host). For purposes of this paper, we will assume that the total I/Os which are possible is equal to the product of the number of disk drives on the system multiplied by the maximum I/Os (1 kbyte transfers) of which one drive is capable. For detailed information on your system and software, you will need to consult your HP System Engineer.

Based on the above performance assumptions, disk performance cost is the product of cost per lost I/O multiplied by the number of lost I/Os per hour. Cost per lost I/O is the quotient of cost per downtime second divided by the average number of lost I/Os per second. A lost I/O per second is the remainder of total I/Os per second required on average by a system or job minus the total I/Os per second available on disk. Lost I/Os per hour can be figured by multiplying the lost I/Os per second by 3600 seconds per hour.

[Chart 9]

DISK PERFORMANCE COST

Cost due to lost performance = Lost I/Os x cost per lost I/Os

where

System performance is limited by disk I/O

Lost I/O = I/Os per second required − I/Os per second available

$$\text{Cost per lost I/O} = \frac{\text{Cost per downtime second}}{\text{Lost I/Os per second}}$$

For the drives compared in this paper, the total available disk I/Os per second (Product A=280 I/Os and Product B=300 I/Os) exceeds the total required disk I/Os per second (72 I/Os peak on MPE XL), thus there is no disk performance cost for any of these products. Based upon the numbers of disks required to total 4 gigabytes on an HP 950 MPE XL system, it is unlikely that disk I/O capability will be a cost with today's products. As previously mentioned, MPE VE systems require more disk I/O and have a higher probability of suffering disk contention. In addition, as future disk capacities increase, the number of disks required to furnish the required storage will decrease and performance costs may be more of an issue in making tradeoffs.

[Chart 10]

DISK COST SUMMARY

	Product A	Product B
Purchase cost	$75,000	$85,000
Maintenance cost	$19,950	$17,100
Reliability cost	$128,400	$64,200
Performance cost	0	0
	$223,350	$166,300

The overall cost of the two products compared in Chart 10 shows Product B to be the lowest cost solution because of the reliability advantage. If disk performance were a bottleneck, this scenario could be reversed. Therefore, none of the tradeoffs can be taken for granted.

Hewlett-Packard continues to monitor product tradeoffs and reduce the total cost to users of disk storage through improvements in capacity, price, performance, reliability and serviceability of disk drives. Because Hewlett-Packard recognizes that access to data is an opportunity cost in performance as well as availability and that the cost varies among users, further system enhancements through multi-processing and disk-mirroring are planned in the future. In the final analysis, however, system managers must understand their systems and use the concepts in this paper to make the tradeoffs which will offer the greatest value to their users at the lowest overall cost.

DISK DRIVE RELIABILITY
YESTERDAY, TODAY AND TOMORROW

Michael Rusnack

Hewlett Packard Company
Disk Memory Division
Post Office Box 39
Boise, ID 83707-0039

INTRODUCTION

Disk drives have changed quite a bit over the years: they've become smaller in size, larger in capacity, and less expensive to purchase and more reliable in performance.

The first three features -- size, capacity and price -- are simple to recognize and can be compared among products easily. But comparing the reliability between products is complex because each disk drive manufacturer measures reliability differently.

This article explains how Hewlett Packard Company computes MTBF for its disks and uses this calculation as a bench mark for continuous improvement. Plus, it details how HP's progress in reliability means a more dependable disk for your system today and in the future.

HP'S MTBF CALCULATION

A product's MTBF can be a good indicator of its reliability. It can also be misunderstood because of the number of ways it can be calculated (see Figure 1.) Therefore, it's important to know how each manufacturer computes MTBF; then you can make equitable comparisons.

HP's MTBF computation assumes a 24-hour a day operation and includes all components of the disk drive sub system: The head disk assembly (HDA), the power supply, cables and all printed circuit assemblies (PCAs), including the disk controller, servo and read/write boards. The reported MTBF is a weighted five-month moving average and is based on all repair orders that occur during the first 90 days after installation, when the failures are most likely to occur. We count all service calls -- even those where a customer complaint cannot be verified (often referred to as "NTF" or "No Trouble Found"). This results in a more conservative number, but appropriately sets your expectations.

Figure 1 - Mean Time Between Failure (MTBF) Calculations
An MTBF number is a calculation of the product reliability based on a predicted or measured failure rate.
Following are three common MTBF formulas.

1. **Test Sample Method** - This method runs a sample of disks until failures are encountered. An extrapolation of the MTBF is then made.

$$\frac{\text{Number of Units X Number of Running Hours}}{\text{Actual Failures}} = \text{MTBF}$$

This method has implied confidence levels. The sample size and duration of the test must be known in order to derive a confidence level.

2. **Theoretically Calculated Method** - For this method, you must calculate the expected failure rate for each component in the product then sum the component failure rates to arrive at a product failure rate. Most likely used to predict MTBF for a new product.

$$100 \;/\; \text{Annual Failure Rate in \%} \times \text{Hours Used Per Day} \times \text{Days Used Per Year} = \text{MTBF}$$

This formula can be used once the product annual failure rate is determined and assumptions are made for operating hours per day and days per year.

3. **Demonstrated Method** - Same as the theoretically calculated method, but uses actual failure data derived from customer installations. HP uses this method to calculate MTBF for its disks.

$$100 \;/\; \text{Annual Failure Rate in \%} \times \text{Hours Used Per Day} \times \text{Days Used Per Year} = \text{MTBF}$$

There are variables in this equation which take assumptions and have major impact on the results. From the customer data, an annual failure rate must be determined. Some common exclusions are:

- o Dead on arrival units.
- o Installation problems.
- o Controller problems/cabling problems.
- o First 200 hours of operation
- o Firmware problems.

HP uses this method of MTBF calculation and includes all of the above.

Some manufacturers may only specify the MTBF of the disk mechanism or HDA on their specification sheets. The disk mechanism in the HP 7936 and 7937 disk drives, for example have a proven MTBF of over 130,000 hours.

The entire HP 7936 and 7937 disk drive subsystem (with controller, power supply and all electronics) have a documented MTBF of over 70,000 hours. And this continues to improve.

Reliability wasn't this high in the earlier disk drives generations, however. In fact, many lessons were learned in the factory and even at customers' sites to bring the figures to this level.

WHEN RELIABILITY IMPROVEMENTS BEGAN

In 1982 and 1983, HP introduced the HP 7933 and 7935 disk drives. These products were "state of the art" at the time and each had 404 megabytes of storage. The HP 7933 was a fixed media drive; the HP 7935 had removable media.

The drives were well-received in the market place and considered leaders in technology. A short time after introduction, however, it became apparent the reliability of these drives had fallen behind a leadership position. The products were not meeting customer expectations. Moreover, we had indications that other disk drive manufacturers' reliability was exceeding ours by as much as 50 percent.

At this time the managers at HP Disk Memory Division decided to move several of the research and development engineers from the "new" product development to assist in the HP 7933 and 7935 reliability improvement program. The division spent over a year redesigning the products through the use of Total Quality Control (TQC). This important set of tools was the key to increased product reliability and customer satisfaction.

"TELL US WHAT YOU LEARNED . . ."

When the redesign was complete after that year, a team consisting of the division's general manager, members of his staff and two engineers (including my self) went to Tokyo in February 1985 to install 16 "new and improved" HP 7933 and 7935 disks. The improvements included several redesigned PCA's, and upgrade of head cables and some mechanical modifications to extend the life of the actuator mechanism.

The disk drives were installed at four different customer sites and monitored for over a year. By December 1985, the MTBF had exceeded our 1983 MTBF by three times. We were pleased with this increase; our products equaled or exceeded the reliability of our competitors' products. But more importantly, our users were pleased.

Disks that were installed at other customer sites were retrofitted with the upgraded head cables and modified actuator. Old PCA inventory in the service offices was purged and replaced with the upgraded boards for future service calls.

As a result of this reliability improvement, the customer quality assurance manager of HP's Disk Memory Division was invited to speak at the Annual Japanese Quality Conference. (This conference is sponsored by the Japanese Union of Scientists and Engineers, the same organization that sponsors the highly regarded Deming Prize.) In November 1986, HP made its formal presentation on quality and became the first non-Japanese company ever invited to address this prestigious group.

Many engineering improvements were made during the HP 7933 and 7935 reliability improvement period. Other improvements were vendor management and change control. These improvements have been incorporated into the process used to manufacture HP's newest disk drives.

THE NEW PRODUCTS

The HP 7933 and 7935 disks were superseded by the HP 7936 and 7937 disk drives. The design was solid and the component count was much lower compared to the HP 7933 and 7935. We realized the higher number of components, the higher the failure rate. So we designed many components out of these new products and used Large Scale Integration of circuits to replace several packages. By designing many components out, better reliability was designed into the end products.

We also worked closely with our supplier to maintain a minimum amount of inventory at the factory, resulting in a Just-In-Time (JIT) environment. The quality of the components starts at the very beginning of manufacturing a component, not during the inspection station at the end of the line. Changes to the supplier's process (or changes to the part it self) are reviewed at Disk Memory Division to determine the impact on our final product.

In addition, a rigorous process was instituted to create as reliable a product as possible right from the start. Our approach included:

- o Creating the environment. Unqualified pressures on engineers for shortcuts to product release were resisted. They were "permitted" to design reliability into the product and were not forced to release the products before they worked.

- o Discovering design weaknesses. Extensive stress testing was conducted to discover tolerance to temperature, humidity, shock, vibration and line voltage. "Strife" testing (STRess/liFE) was done to diminish the margin between stress and strength. This forced failures to appear; they were analyzed and solved to root cause.

- o Design Defect Tracking (DDT). Once defects were found, they were entered into a database and reliability engineers searched for the root cause by asking "why" five times (see Figure 2).

- o Duane Charts. These charts were used to track and predict the product MTBF. They also measured the effectiveness of the development team and environment by quantifying the rate of MTBF improvement.

By November of 1986, the HP 7936 and 7937 disk drives were ready for customer shipment. The MTBF at time of introduction was about 60,000, far exceeding the original target goal.

CHALLENGE # 1: HOW CAN WE ENHANCE SOMETHING THIS GOOD?

Since the MTBF goal was exceeded and design enhancements were implemented on the HP 793 7937 right from the start, the manufacturing team had a difficult challenge: how can a product this good be improved? This challenge was divided into four objectives:

1. Maintain (or exceed) reliability at customer sites as demonstrated by life testing at the factory.

2. Assist the suppliers from whom we bought components to implement TQC techniques in their processes. This ensures consistent high-quality components are received.

3. Analyze each and every failed component to determine the root cause of failure.

4. Standardize corrections so misinterpretation is eliminated.

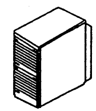
"Ask Why Five Times" Process

As part of the TQC philosophy, HP engineers follow a problem to its root cause. It is then corrected and documented. Here is just one example of tracing a disk drive failure to its root cause.

1

Why did this HP 7937 fail its test?

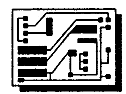
A circuit shorted on a Printed Circuit Assembly (PCA) board.



2

Why did the circuit short on this PCA board?

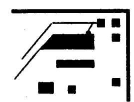
An Integrated Circuit (IC) leg touched the circuit path.



3

Why did this IC's leg touch the circuit path?

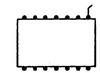
IC leg was bent outward.



4

Why was the IC "leg" bent outward?

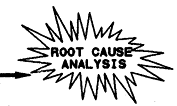
The IC was inserted upside down causing the leg to bend.



5

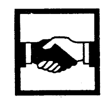
Why was the IC inserted upside down?

Polarity not clearly marked on the IC.



ACTION

*Give feedback to supplier on enhancing the polarity mark. Assist in improving his process.



*Train line workers to question components that aren't clearly marked. Verify polarity before insertion.

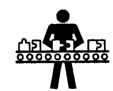


Figure 2

CHALLENGE #2: STEEP PRODUCTION RAMP

The production ramp of the HP7935 and 7937 was rapid; it took only six months to accelerate from producing a few evaluation units to manufacturing full production volumes. Demand climbed for these new disks, resulting in "round-the-clock" manufacturing. The steep ramp and continuous production made up the second challenge for the manufacturing engineering team. Such conditions often result in material availability issues, inconsistent component tolerances and occasional production process bottlenecks.

But that didn't show us down! We developed a field failure database that recorded every failure in the assembly process. This data base allowed us to evaluate each failure on an individual basis and trace the failure to its root cause. Eventually, the database was expanded to include failing assemblies returned from the field. We use this database now to determine the reliability projections.

MEETING THE CHALLENGES

For the last two years (immediately following the HP 7936 and 7937 introduction), the reliability team has meet on a weekly basis. Representatives from R & D, PCA test, materials engineering and manufacturing engineering team attend. Each week, one assembly is reviewed. The manufacturing engineer responsible for a particular assembly presents in detail the enhancements currently in process on that assembly. These checks and balances allow us to focus on the item with the greatest potential. This weekly forum also serves as an informal design review for modifications to the product.

This team has been able to show a number of successes by categorizing disk failures into one of four groups:

1. Process Error at the Factory. Whether the failure is due human error or an oversight in the procedure it is corrected and documented to prevent its reoccurrence. Personnel training and honest feedback are our most important tools.

2. Design Error or Lack of Margin. The R & D engineers work to ensure proper margin is designed into all applications. On occasion, however, failures happen because of a design error or lack of margin. Manufacturing engineers will redesign the circuit with the same care (often calling on the original designer for assistance) used in the initial design.

3. Component Defects. These are isolated and probed to root cause. Oftentimes, the cause of a component failure is due to a handling error. Typical examples of component defects include bent pins, damaged packaging and electrostatic discharge (ESD) damage.

4. Supplier's Process. Component failures due to a vendor's process are relaxed and corrected by the supplier at the supplier's facility.

LOOKING AHEAD

Since the HP 7936 and 7937 product design was solid at the time of introduction, the manufacturing engineering focused (and continues to focus) on all field failures that occur. This ability, coupled with the data maintained on-line using a relational database, allows the team to measure the effectiveness of their work.

Our work on the HP 7936 and 7937 will continue through its production life, end of service life and obsolescence. Future HP disk drives will follow on the same TQC path; root cause analysis for failures found, extensive "strife" testing at the factory and building reliability into each stage of the production process.

CONCLUSION

What does this mean to you? It means you receive the most reliable disk product available for your HP system. Our goal is to make each new product more reliable than the product it replaces. And the more reliable your disk drive, the greater your system uptime and user availability. Plus, the simplicity of the HP 7936 and 7937 facilitates a quicker repair with minimal downtime in the event of a failure.

We can't guarantee an HP disk drive will never fail. But we can guarantee if a failure does occur, we will analyze it to its root cause and work to prevent it from happening again.

Tape Backup for the 1990's - DAT

Author: Peter Messer
Product Manager
Hewlett-Packard
Computer Peripherals Division
Bristol
England.

Hard disk capacities have been increasing rapidly across all systems from PC's to mainframes for many years, and look to continue doing so for the future.

Developments in tape backup have lagged behind that trend. For instance, whilst capacity improvements in 1/4" tape drives now offer 320 Mbytes/cartridge, 760 Mbyte hard disks are becoming available. Backup has become regarded as an operator intensive, time consuming and sometimes expensive process as a result. In many data centres, the ever greater number of tapes to be stored has created new problems in managing the amount of information stored, and increased operating costs.

New technologies based on helical scan are capable of matching the increases in hard disc capacities because they already offer high data densities and could be developed further. Helical scan products are familiar to most of us as the VCR we use in the home, and the video camera.

Digital Audio Tape is the most recent development of these technologies, and was developed for the consumer audio market to overcome the limitations of the analogue recording of audio signals on cassette tapes. By digitizing the audio signal into two channels of 16-bit data, DAT provides a frequency range of 5 to 22 Khz, and distortion of 0.005% with a dynamic range of 96 dB, which is well beyond the capability of the best audio cassette systems available today.

The hi-fi enthusiast would find this specification appealing: DAT for data storage purposes also offers significant advantages, as this paper will show. The areas covered comprise the following:

1.) What is helical scan and how is it implemented in DAT ?

2.) How is DAT modified for computer data storage use ?

3.) What features does DAT offer for the storage of data ?

4.) How can DAT be developed for the future ?

5.) How does DAT fit into the current range of storage products ?

1.) What is helical scan and how is it implemented in DAT ?

Unlike current 1/4" and 1/2" tape drives which record data along the tape length, the heads and media of a helical scan device are aligned such that recording tracks are written at a shallow angle across the tape. This allows for a much larger area of tape to be utilised than with conventional technology, and provides for a greater areal data density, and hence increased capacity.

Another difference is that in conventional tape drives the track locations are held mechanically, by reference to a fixed surface, whereas in helical scan the location of the recorded tracks is electronically controlled. This means that track densities can be can be one or two orders of magnitude greater than conventional tapes, because of the greater precision in defining the track position.

In helical scan recording the tape is slowly moved past a spinning drum on which the read and write heads are mounted. By using closed loop servo control of the relative speeds of the drum and tape, the drive can precisely follow the narrow tracks on the tape.(Ref 1)

However, the helical scan products used for home and professional video recording have a number of drawbacks for computer data storage:
Firstly, the tape is wrapped around the drum enclosing an angle of at least 180 degrees. This is used in these recorders to act as a buffer for the stream of incoming video data. The disadvantage is that the large wrap angle needs a large number of mechanical parts, makes repositioning slow, and could cause early failure of the drive mechanism.

Secondly, data recording requires read after write for data integrity, but on all helical scan recorders today, there are only two heads, one for writing and one for reading. The data tracks must separated by guard bands and so the theoretical areal density of the tape is reduced, and the data capacity of the tape is smaller.

Thirdly,these recorders are basically analogue recording devices, and a need a new format for digital data storage usage. (The development of a suitable format is key to the success of any new tape technology.)

DAT technology offers a number of important advantages over current helical scan technology as follows:

The wrap angle is 90 degrees, which simplifies the mechanical design, such that fewer mechanical parts are needed, and DAT based drives should prove more reliable than current helical scan drives as a result.
DAT drives have two heads opposite each other, which are used for both reading and writing of data; the heads are wider than the tracks which means that they overlap, and thus all of the surface is used for data, except for guard tracks along the tape edges.

Whilst this would normally cause interference between tracks in a conventional recorder, this is overcome in DAT by mounting the two heads at different angles to each other. As each head tries to read data, it will pick up the strongest signal from data written at the correct angle, and be able to centre on the correct track by balancing the stronger signal against the weaker ones on either side. This is known as azimuth recording.

This method of recording means that the standard 200 foot (60 meter) DAT tape stores approximately 1300 Mbytes of digital data. This is a significant improvement on existing tape media, and represents a packing density of 114 Mbits/sq in, on a tape less than 0.2 inches wide (3.81 mm.).

Unlike the VCR, DAT stores information (usually music) in digital form, and sophisticated error correction routines are already well established. It uses a small cassette, measuring only 2.8 x 2.1 x 0.4 ins. The tape format used for consumer applications of DAT was agreed by a committee of over 80 companies in 1987, and lays down the way in which digital audio data is stored. This has provided the basic groundwork for developing a computer storage format. (For a description of the audio format see Ref 2.)

2.) How is DAT modified for computer data storage use ?

The computer version of DAT being jointly developed by HP and SONY, uses the same basic mechanism and circuitry, with some additional hardware and firmware to satisfy data storage applications.

The first of these modifications is the addition of two more heads, so that the drum has two separate pairs of read and write heads set opposite to each other and set at the appropriate angles as discussed earlier. This allows the drive to perform read after write and correct early errors in data recording.(This will be discussed later).

The second modification involves the addition of standard computer interfaces, for example HPIB or SCSI, and appropriate firmware to enable a computer to write data to and read data from the drive.

The third modification is the development of a standard format which defines the way in which the data is laid down on the tape. This will enable users to interchange data or software with other drives, in much the same way as is presently possible with 1600 bpi reel to reel, for example.

The DDS format, jointly developed by HP and SONY, has been adopted by eight other tape drive manufacturers for their future products. Using the audio format as a basic building block, and incoporating computer system needs, it provides high reliability recording with minimum loss of capacity and and transfer rate.

The discussion of the format is beyond the scope of this paper, but more information is available from HP (Ref 3 and 5).

The fourth development is concerned with ensuring that the amount of data error correction in the format is suitably extensive enough for computer storage purposes. Consumer DAT products correct errors by using a number of different techniques, but these drives do not need the same level of error correction as is necessary for data storage. Considerable effort has been applied to develop the format to include powerful error correction routines.

Typical hard error rates for computer tapes are around 1 in 10^-11 bits. For example on a 1/4" tape drive less than 1 in 500 cartridges would have an uncorrectable error. The aim of the HP/SONY development of DAT for computer purposes is to achieve error rates several orders of magnitude greater than current 1/4" or 1/2" tape products.

A number of error correction techniques are used, two of which are useful to explain.
The first of these is read after write, where a set of data is read immediately after having been written. If an error is found, then the drive rewrites that block once more; if an error is still present, the data is rewritten once more, and so on.

The second is multiple group writing, which is an optional feature where each block of data is written two, three or more times, to ensure that at least one good block of data has been recorded. This particular method has some benefits for high speed tape copying requirements such as software distribution. A new method of "contact printing" of tapes will provide a fast copying process and use multiple group writing, to ensure error free copies of the master.

The HP/SONY DDS format offers the user eight further levels of error correction, to achieve a hard error rate several orders of magnitude greater than 1/4" and 1/2" tapes.

A detailed discussion of error correction techniques and error sources is beyond the scope of a short paper such as this one, a more detailed report is available from Hewlett-Packard (Ref 4).

3.) What features does DAT offer for the storage of data?

DAT has a number of other features which make it attractive for data storage uses in software distribution, data interchange and unattended backup, with improved reliability:

Software Distribution:

DAT tapes currently are priced at less than $15 for the standard 200 foot (60 meter) versions. (1/4" and 1/2" tapes cost 10-20 times more for the same capacity). Furthermore, DAT tapes do not need preformatting for use, so that tapes are easily available at competitive prices. The mechanisms currently available have the potential for a built in cleaning roller, thus avoiding the need for a cleaning cassette and subsequent head wear. The compact size and low cost of DAT media adds to its suitability for software distribution.

Data Interchange

Considerable effort has been expended in developing a robust data storage format, with powerful error correction routines. HP and Sony have been working with eight other tape drive manufacturers on development of the DDS format. This has now been submitted to the American National Standards Institute for adoption of as an ANSI standard. The group of manufacturers, including HP and Sony, are working with ANSI to fully specify the DDS format for DAT based streaming tape drives. A standard format, together with the availability of drives from several manufacturers will provide the means for universal data interchange using DAT tapes.

Unattended Backup

The capacity on a standard 200 foot (60 meter) DAT tape is 1300 Mbytes; at a transfer rate of 183 Kbytes/sec it takes about 2 hrs. to complete a backup of this size. (This is 30 times of the capacity of a 1600 bpi 2400 foot reel, and about twice the backup rate of 1600 bpi tape drives.)

Our experiences with the HP 1/4" cartridge autochanger product showed us that many customers liked the unattended backup capability this product provided; this was confirmed by our market research. Being able to backup 1300 Mbytes on a single cassette should make unattended backup even easier,and offers further savings on operator time and storage space.

A special feature of DAT (called Fastsearch), allows for access to any part of a full tape in around 20 seconds. (this can be even faster with a shorter or partially filled tape). This makes for fast and easy retrieval of stored data.

Reliability

The low tape speed reduces tape wear and means tapes should last longer. Powerful error correction routines leveraged from DAT technology means that drive and data reliability will be even better than existing tape drives.

4.) How can DAT be developed in the future ?

One of the most valuable features of DAT is that it offers many capabilities for future improvements. The technology is simple, reliable and easily extendable. Leverage from the high volume consumer industry will provide for lower cost readily available mechanisms to enable manufacturers to produce high quality, easily obtainable drives at competitive prices. Recent press announcements have shown that a large number of computer tape drive manufacturers are committed to producing products in the near future based on DAT technology.

A feature of the DDS format is that it is designed to be forwards and backwards compatible with future drive developments, some of which are outlined below:

Increased speed: Faster tape and drum speeds as well as data compression will be able to increase the transfer rate to around 2 Mbytes/sec.

Increased Capacity: Longer, thinner tapes are likely in the near future to provide capacities up to 2 Gbytes, using new media coatings. Data compression techniques could then increase the capacity per tape up to around 8 Gbytes.

Smaller Size: The current mechanism size is the same as 5 1/4" hard disk drives, making it easy to mount in computer systems.We believe that it will be possible to make the drive even smaller in the future, so that it could be possible to build it into desk top PC systems, for example. This would give PC users easy, convenient backup of their data as an alternative to floppy disc. Smaller mechanisms and more integration of the drive electronics will be able to produce half height and 3 1/2" form factor products.

Although likely to be priced higher than 1/4" tape drives for the immediate future, developments in manufacturing

costs and size could well see future products offered at a similar prices.

Long term developments could also be to add autochanger capability and possibly incorporate audio as well as data

storage on the same tape. The HP/SONY DDS format described earlier does allow for the latter possibility.

5.) How does DAT fit into the current range of storage products ?

Because DAT offers low cost media, high capacity for unattended backup and good transfer rate, it will compete successfully with both 1/4" and 1/2" tape drives in the mid range systems marketplace.

The adoption of a suitable format interchange standard means that DAT based drives will compete with 1600 bpi reel to reel for many applications, unless IBM format is needed by the user.

DAT tapes because of their low cost and compact size could rival floppy disks, 1/4" cartridges and 1/2" tapes for software distribution. This would be the case in applications involving more than 100 Mbytes of software.

Current helical scan products presently appear to offer similar capabilities to DAT, but we believe that the newer technology has more opportunities for future developments than other helical scan products as currently defined.

Rewriteable optical technology is fast developing as a new storage medium between disk and tape, for specific application areas. It seems unlikely that competition between these two different but new technologies will occur, since their areas of application are likely to be complementary. The optical technology offers near hard disc performance but at a higher media price than DAT initially, and DAT offers lower priced, higher capacity media but slower access times. Users of Hewlett-Packard systems will be ideally placed to utilise both of these new technologies in the future.

References:
1."DAT for Data", Jay Young and Eng T Tan, Systems International January 1988.
2."All present and correct", Eng T Tan and Mark Dunstan, Systems International February 1988,
3."DAT data format takes shape", Pete Bramhall and Mark Dunstan.Systems International March 1988,
4. Designing a Data Storage Format for Digital Audio Tape, Hewlett Packard, Revision B October 1988.
5. Digital Data Storage Format Description, Hewlett Packard, Revision B October 1988.

Bootstrapping HP SRC: HP Software Revision Controller

Alan Padula

Hewlett-Packard Company
Data and Languages Division
Cupertino, California

February 7, 1989

> BOOTSTRAP *adj* 1: carried out with minimum resources or advantages
> 2: using its own action to initiate or sustain itself
>
> \- Webster's New Collegiate Dictionary

INTRODUCTION

This paper premieres Hewlett-Packard's new MPE V & MPE XL based version control product, HP Software Revision Controller (HP SRC), from a user's viewpoint and as one of its software developers. An explanation of version control systems is presented followed by the author's accounting of how an internal software tool was bootstrapped into a prototype of HP SRC that was subsequently used:

- as a foundation layer of software to build upon,
- as a platform to demonstrate and refine the product's user interface,
- for managing parallel development of itself from disjunct geographic locations,
- in its own successful product development from daily use of itself on itself!

Features and functionality of HP SRC are highlighted through examples of the command interface and portions of sample output. Focus is also placed on the major benefits derived from bootstrapping, that is, using a product in the development of itself.

VERSION CONTROL SYSTEM BASICS

1. What problems do they solve?

The need for version control systems have been around for as long as programming has. Complete version control systems, coupled with effective project development processes (see **Organization of the Project Files & HP SRC Environment**), address a variety of problems such as:

- **Versioning**
  This refers to storing and retrieving permutations of a program and its associated files,

- **File Access Conflict Management**
  This refers to preventing programmers from changing the same file at the same time and thereby writing over each other's modifications,

- **Audit Trails**
  This refers to identifying what files have changed, who changed them, when they were changed, and what those changes were,

- **Parallel Development**
  This refers to concurrently altering the same set of files with the ability to later merge the changes from the independent development efforts together,

- **Security**
  This refers to establishing who may access files and maintaining the integrity of the environment.

2. Early "Solutions"

Many methods have been employed to solve these problems long before the advent of the first automated version control system. There are some creative "workaround" methods to prevent file access conflicts but often these have serious shortcomings when addressing problems like versioning, audit trails, parallel development, etc. Enumerated below is only a small potpourri of some of these old, more common ways of avoiding file access conflicts on the HP 3000:

- Verbal communication has long been a standby of smaller projects. Through one programmer asking another if they are currently modifying a file or not, the problem of updating the same files at the same time and the subsequent corruption of each other's changes is averted. This method has significant drawbacks including limitations on the size of projects, the required geographic proximity of its members, the presence of its members at any one time, and of course, the quality of the programmer's memories!

- A central log file may be read by programmers to see if anybody is using the file that they need. If not, the programmer would update the log file to indicate that they are using the file. It is debatable whether this method is much better than a paper signup sheet; especially given that two users could access this same log file at the same time and end up writing over each other's log information! It provides no security and its success, as is true with many of the old methods of version control, is dependent on the programmers faithfully following manual procedures.

- A librarian is appointed for the project and when a team member needs to modify a file, they must request it from this person. The librarian keeps track of who has what files, who may access what files, and controls their distribution. This method is actually quite effective but requires a high availability to the librarian and its success is dependent on the librarian's skill as a bookkeeper (and their health!)

- Adding a lockword (such as the programmer's name) to the file to be modified is another creative & "semi-secure" method of dealing with file access conflicts. Users who try to access that same file would fail until the lockword was removed (corresponding to when the original programmer completed their modifications). While it does keep the non-malicious user from updating a file someone else is using, it is still very limited compared to the power of a version control system and again dependent on everyone following procedures.

- Another method is the use of a simple check in/check out tool that does nothing more but control the "copying" of files between the user's work groups and the central shared group. The copy is allowed or disallowed based on whether the file is already checked out. This scheme has the same limitations as the previously mentioned one.

3. General Features of Version Control Systems

There are some basic features inherent to the top version control systems including:

- **Check In/Check Out**
  This refers to registering and copying files into and out of the version control system. When checking in a file: who checked it in, when it was checked in, and what revision number it was assigned, is all logged automatically. A transformation is typically performed on the file being checked in. This results in change information being recorded so that the revision of the file may be reproduced upon a subsequent check out. Inherent to all version control systems is the ability to retrieve prior copies of files which have been checked in i.e. the version control system serves as a repository for previous revisions of files. When checking out a file, the user is first verified and then recorded as the "locker" of the file's revision. Most importantly, under typical operation the system will prevent the user from checking out the same revision of the same file that someone else has checked out. This is key to preventing a file access conflict. On occasion, in the *real* world, a predicament arises where a user has a file checked out but is unavailable to check it in at a critical time. Most systems provide a crucial "escape valve" that allows a person with the proper level of security to check in files that others have checked out.

- **Reporting Of Status Information**
  This refers to the ability to list information about the state of the version control environment such as what files are checked out, who has them checked out, when they were checked in, who checked in a particular revision of a file, etc. This is important in maintaining an active system and in providing a traceable audit trail.

- **Differencing**
  This refers to the ability to list the changes made between two revisions of a file. In tasks such as tracking down a new bug introduced in the software, this differencing ability can be a real productivity booster. In practically "no time" one can find exactly what lines were added, deleted, and modified between two revisions.

- **Merging**
  This refers to the ability to take different revisions of a file and "automatically" combine their changes into a new file. *Automatically* means that whenever possible the merging system will mesh the changes properly. If the system can not clearly determine how to resolve a conflict, the indeterminate area is flagged and the user is left to manually do the merge. While merging does not seem to be a part of all version control products, it is an essential component in the support of parallel development efforts. Merging when coupled with intelligent software development methodologies provides an answer to making quick, "hot" bug fixes while major enhancements are being done on the same files.

4. Storage Methods

Two of the most common storage strategies for version control systems are:

- **Archived Files**
  These files are stored as their full image everytime a revision is checked in. To store 3 revisions of a 1000 record file would minimally take 3000 records plus whatever control records the version control system needed to store revision-to-file mapping and version control history information. Some systems may store them in a compressed format where trailing blanks and such are stripped thus saving some disc space.

- **Delta Files**
  These files, sometimes referred to as "differenced" files, store only the changes between revisions. Thus, to store 3 revisions of a 1000 record file that had a total of 20 changes could take as little as 1020 records plus a relatively insignificant number of control records the version control system requires. Those are significant savings when compared to the previous archived example. Two common delta file formats are "reverse" deltas and "forward" deltas. *Reverse deltas* are organized such that the latest revision of a delta file is stored in its full image format while previous revisions are maintained as change records. The main benefit in reverse deltas is manifested in performance savings when accessing the latest revision of a file. *Forward deltas* are organized such that the first (i.e. the earliest) revision is a full image of the file and any subsequent revisions checked in are stored as change records. This may save time when checking out an earlier revision but a performance penalty is paid when accessing the latest revision.

Some systems utilize both delta and archived file storage formats. HP SRC is such a system and adopted reverse delta format technology for its delta files based on the philosophy that the latest revision will be the most frequently accessed and hence, the most important to be optimized for performance. HP SRC stores most "flat" ASCII files as differenced files; while binary files like program and USL files are stored as archived files.

BOOTSTRAPPING HP SRC: PROJECT DEVELOPMENT PRACTICES

In developing HP SRC, the Lab team gained many advantages by starting with an internal tool that was being used in the company. It was the intention from the start of the project to bootstrap the product; that is, to use the tool in the development of itself. Hereafter, whenever a reference to "prototype" is made in this paper, it will refer to that internal tool or any of its many permutations of user interface and functionality through its development lifecycle into what is now the HP SRC product. A major objective of the development team was to gain insight into the "useability" of the prototype as end users. By using the protoype early in the development lifecycle, REAL data was provided that helped in making and refining major design decisions based on experience rather than esoteric thought.

Quality of the software was further insured through production use by some of the most critical and least forgiving users around --- the actual developers. Anytime a new release was built, the software would soon be put into the HP SRC software development team's "production" account where any problems would quickly be revealed. Naturally, full unit testing and some preliminary system testing was performed along with a "just-in-case" backup (after all, this is software development were talking about!) before *in-production* use and testing began.

1. Management of Documentation

The prototype proved to be as valuable for versioning the documentation as well as the software. A MPE group for the Internal Design and External Design was setup and maintained under HP SRC control. The specifications were partitioned into many files that corresponded to the sections of the documents. As sections were completed and revised, the files were checked in and out. Although, at this obviously early stage of the project, the interface was crude and was being regularly modified, the prototype was still providing a platform to learn and develop from.

As engineers worked on the documents, it was not unusual for them to inadvertently try and gain access to the same file at the same time (i.e. the file access conflict problem). The prototype was very effective in keeping engineers off of each other's toes. The problem of writing over each other's changes was eliminated.

Another key benefit of using version control at this phase in the lifecycle was the ability to roll back to previous revisions of the design. In one instance, an engineer made a proposal that resulted in significant changes in the design. After a time though (and a few revisions!), it became apparent that the changes would have to be backed out. The earlier correct revisions were quickly located in the HP SRC environment and "re-checked" in to become the latest development revisions (and a collective sigh of relief was had by the engineers who would have had to type in all of the changes over again!)

2. Organization of the Project Files & HP SRC Environment

Version control systems can only be as good in a programming environment as the development process that supports them. This conclusion was reached after using the prototype ourselves and observing its use in different scenarios throughout the company. The message was clear: the integration of HP SRC and the development process would be a key element to its success. This led to two goals for the product: to provide a tool easily integratable into an existing process and a form of support to aid users in that integration.

There are probably as many ways to set up an HP SRC development environment as there are ways to organize a project. The Implementation Guide was founded on this tenet. The Guide is provided with the product and helps the customer analyze their current project setup from which recommendations can be drawn for how they might structure their HP SRC environment. Depending on the project development methodology and current project file organization, a project may even have more than one HP SRC environment. The initial setup of the HP SRC environment and the project file organization used by the HP SRC team was simple though. In one MPE account there was:

- **Two HP SRC Environment Groups,**
  One group contained the HP SRC internal files for all our source and the other the internal files for our documentation. (One HP SRC environment could have been used for both but we preferred the clearer separation provided by different groups.) The internal files consist of the stat file, archived files, and delta files. The stat file is similar to a directory that keeps tracks of these files and the HP SRC users (including their access capability). Recall that delta files contain the actual text and text change records for each revision of a file.

- **a Ready Group**
  This contained the latest full text copy of the source files. The files were normal, non-HP SRC files that were accessible by any editor or compiler. The purpose of the group was to provide a general area for files needed by all engineers during their individual compiles. Thus, engineers did not have to incur the performance overhead of a check out of files that they were not modifying but were needed for a compile. Multiple Ready groups for a single HP SRC environment are not unusual. This group also kept the object file results of our master builds. Master builds refer to a full compile and link of the entire product incorporating the latest set of files and changes checked in by the project team's members (typically done on a weekly basis).

- **a Development Group per engineer**
  This contained whatever files the engineers had checked out for their own unit development purposes. It also contained a their own copy of the latest object files that were being used in compilation and linkage.

- **a Test Group per engineer**
  This contained a very small HP SRC environment that was used in the individual unit testing of each engineer's changes.

The initial setup of an HP SRC environment is straightforward. A simple CHKIN of the first file will automatically create the environment. Large projects with many files can be checked in easily by using standard wildcard characters. For example, **CHKIN S@, FROM=SOURCE** would check in all the files starting with "S" in the group SOURCE. If preferred, HP SRC can be executed in batch mode to help accomplish this.

3. Adding Users & Setting Security in the Project

Adding valid users and setting their security class is an easy, intuitive process. User and security access is maintained with the ADDUSER, DELUSER, and CHGUSER commmands. By specifying the user name and security class (see below) in the ADDUSER command, the user is added to the HP SRC environment as an active user with certain access capabilities e.g. **ADDUSER ALAN.TOOLS,LIBRARIAN.** CHGUSER is used to change an existing user's security class and DELUSER of course, is used to delete a user from the environment.

There were only two levels of security in the original prototype, one granted users the capability to do anything in the environment and the other provided a more middle-of-the-road capability. While full capability was set for all project engineers, it was known from the start that the commercial development environment would demand more levels of security. Insights gained from the use of the prototype drove this home and led us to phase in a total of five levels of security over HP SRC's development lifecycle. A high level summary without the details of those capabilities is outlined below. They are, in decreasing order of capability:

- ADMIN
  This user has the full run of the system. Exclusive privileges are granted to add/delete users or to change their security level.

- LIBRARIAN
  Whereas a whole team of programmers may modify files for a project, some programming staffs allow only the Project Lead of the team to move files to their official "build" area. The LIBRARIAN class coupled with the AUTHOR1 class facilitates this type of methodology. The Project Lead is assigned LIBRARIAN security which grants the special capability to check in files that other users have checked out. All other programmers are assigned AUTHOR1 which allows them to check out files, but not to check them back in. Only the Project Lead would be allowed to do that.

- AUTHOR2
  This user can check out files and check them back in; however, they can not check in files that other users have checked out.

- AUTHOR1
  This user can check out files but can not check them back in (see LIBRARIAN above).

- READER
  This user can look at files and history information but can not check out or check in files. This is ideal for a support group or a management approval system where the only capability needed and in fact, desired, is to be able to read the files.

Another security advantage of maintaining files under HP SRC is that it discourages the kind of programmers who feel they can make changes to files in their team's official build area without following standard procedures. How many times has someone made a change and forgotten to tell anyone else? HP SRC logs anyone making changes; dissuading the dubious "on the fly" approach to programming.

4. Handling Conflicts

The HP SRC source files were partitioned and modularized as much as possible in order to reduce contention by engineers for the same files. Some files, like the global declarations file and message catalog, always seemed to be in high demand though. The typical scenario would start with the engineer who needed the file trying to check it out. When this failed they would use a command, LISTREV that reports status information about files in the HP SRC environment (see example under **Code Documentation**) to see who had the lock. Once the engineer who held the lock had been determined, the engineer who needed the file would take one of the following actions (all were repeatably used during the project with success):

- Ask the engineer who had the lock to check it in (many time they were through with it or did not need it at the moment),
- Ask the engineer who had the lock if they could modify their copy of the file directly (a questionable practice!),
- Check out a previous revision of the file they need, make the change they need, then check it back in as a branch that would later be merged in (via ADDDIFF) after the latest revision of the file had been checked back in.

A bootstrapping bonus was achieved with the irritation by project engineers of having to do a LISTREV to find out who had the lock. The succinct error message declaring a revision of a file to be "already locked" was enhanced to include who held the lock. HP SRC was quickly modified and that change remains in the product today.

5. Customizing the Interface

UDCs were utilized in developing an intuitive, command-driven user interface. This type of interface met the objective of running on "vanilla", character-mode terminals which are still in great abundance in commercial shops. Although a menu-driven interface did "demo" better and seemed to be chosen more often by managers, the programmers consistently preferred a command-driven one. This view gained credence in the project as we made daily use of the prototype. The commands had to be descriptive enough in their name and function for easy recall but not so long as to hinder the programmer in a hurry (e.g. us!)

The interface was probably the biggest benefactor from our bootstrapping efforts. Not only were deficiencies immediately seen but they were immediately FELT! The commands in the original internal prototype were much too cryptic and too hard to remember for commercial use. The first design of the interface produced command names that were descriptive enough in their purpose but ACTUAL USE by the team demonstrated them too cumbersome and long. After "demoing" an improved design, integrating user input, and refining the interface as we used it; we came up with the current set of commands:

HP SRC COMMAND SET:

ADDDIFF	CHKIN	DELDIFF	LISTDELTA
ADDSYM	CHKINCOPY	DELREV	LISTDIFF
ADDUSER	CHKINPLACE	DELSYM	LISTREV
		DELUSER	LISTTREE
CHGDESC	CHKOUT		LISTUSERS
CHGLOG	COPYLOG	LOCK	
CHGOWNER	COPYREV	UNLOCK	RECOVERSTAT
CHGPREFIX			SRCHELP
CHGUSER	COPYDELTA		

Initially the number of commands was considered an issue. Actual use of the product by the Lab team diminished the concern as the majority of the time, we found ourselves using only four commands: CHKIN, CHKOUT, CHKINCOPY, and LISTREV. Following as a distant second in the frequent usage category were: COPYREV, LOCK, UNLOCK, and CHKINPLACE.

The final set of commands is a powerful and general one, but the ability to customize those commands has been left in so that fine tuning to specific customer requirements or preference is possible. For example, it would be easy for a user to change the COPYLOG command to a shorthand CL. Just as the Lab team modified the HP SRC prototype over and over again, so can the HP SRC customer! (And naturally, the interface is localizable.)

It was not surprising that the Lab team rarely used the online help facility but when necessary it was usually to determine a command's parameter name or position. Thus the birth of the Quick Reference Card that comes with HP SRC.

HP SRC has special support built into it for COBOL but it can also be used with most any programming language such as FORTRAN or PASCAL as well as in many types of non-programming situations. The main target user was the COBOL programmer but use by the Lab team in programming and documentation helped the product maintain a broader dimension. The greater assurance of its multipurpose use was considered yet another benefit of bootstrapping the product.

6. Resistance to Change and Performance Concerns

A concern of the team was the potential resistance of programmers to use something that might slow them down or something they might perceive as a needless overhead. Any trepidation was overcome quite early by the Lab team as the prototype contributed to their productivity during the development of the documentation as previously mentioned. Further affirmation was found in its rapid growth in use at Hewlett-Packard. It is understandable that programmers might hesitate using any version control system due to the additional procedures required but the benefits are soon realized by those very same people. In working with one potential customer early in the process, it was asked if the concern might be a genuine one. The response was a commanding one from the management present that version control would be used in no uncertain terms! Apparently just the previous week a significant portion of software development had been lost that could have been averted through the use of such a tool. The integrity benefit alone was enough to convince that user of its value.

Good, solid performance has been designed into HP SRC. Providing a high performance tool was thought to be a major factor in the acceptance and useability of the product. Indeed, the reverse delta storage technology was adopted and has proved to be one of the performance enhancers to the product. Performance is another good reason to use the CHKINPLACE command (the first good reason is while you are performing a check in; it guarantees no one else can secure the lock you hold on it.) CHKINPLACE will check in the file, leave the file in the MPE group where you checked it in from, and maintain your lock on the file. Contrasted to a CHKIN followed by a CHKOUT, this saves you the extra step of issuing the CHKOUT command and saves HP SRC the processing time of a CHKOUT which includes obtaining a lock, reconstructing the revision, and copying it back out. These are only a couple of features to speed up development with HP SRC.

Interestingly though, in our use of the prototype, we found that the criticality of a blazing, high performance tool did not figure as heavily as we thought. On the average, the HP SRC project checked in only four to five files per week per engineer. That activity was generally performed by the each engineer at a singular session of time when they would check in all of the files corresponding to their work unit. In other words, instead of intense daily interaction with the product, engineers tended to check out a group of files to work on, and when complete with the necessary work, check them all in as a set. Since most of the programming jobs were scoped to take around a week, there was not a heavy daily use by any one individual engineer.

7. Individual Unit and Master Builds

As already mentioned in the **Organization of the Project Files & HP SRC Environment** section of this paper, engineers accomplished individual builds in their own group and maintained copies of the object files from the last master build. Project methodology prescribed that the engineer check out the set of files that required changes into their own group. From there editing, incremental compiles, preps/links, testing, and debugging took place. The incremental compiles would access the Ready Group for files that had no need of modification but were required by the compiler to satisfy references. Once the engineer had met testing requirements and felt certain their changes were stable, they would check in the files to the HP SRC environment and copy the files to the Ready Group so that others in the project would pick up those solid changes. The check in and copy was done with the CHKINCOPY command. For example, `CHKINCOPY FILEA` would check in FILEA to the HP SRC environment and then copy it out to the Ready Group, e.g. SOURCE. Two defaults were invoked in this example. First, the group SOURCE was set up as the default Ready Group through a prior customization of the CHKINCOPY UDC. Secondly, the revision number was left to be set by HP SRC. The product defaults to incrementing the value of the locked revision by one. Note that revision numbers are stored as numbered pairs (e.g. the default increment of revision 1.7 would be 1.8).

On roughly a weekly basis, we would perform a master build; that is, a build which included a full compilation with everyone's latest and greatest changes. The actual frequency of the master builds was determined by the team and was based on coordinating logical breaking points of each engineer's work. Everyone would check in their files as they completed their modules, and then the "build" engineer on the project, would execute a LISTREV to check for files with outstanding locks on them (see example under **Code Documentation**). If a file was found to be locked then the builder would ask the engineer to either check it in or simply declare that the file was not ready for the build (i.e. missed the deadline!) Either way, stable and minimally unit-tested software would be in the environment. Next, a COPYREV would be performed to the Ready Group. The COPYREV command would copy the latest revisions of the file set specified (wildcard characters are allowed) from the HP SRC environment to the Ready Group but without locking any of the files. For example, `COPYREV TST@,,TESTING` would copy the latest revision of all files starting with TST@ to the group TESTING. Note the default for the second parameter means to retrieve the latest revision. This operation guaranteed the inclusion of only the files which had been formally checked in according to procedure. This prevented inadvertent modification of files in the Ready Group and discouraged the introduction of untraceable changes (under the guise of "quick fixes") directly to the official, Ready Group. Typically the master build processes including the massive COPYREV would be done overnight in batch mode. This "off hours" scheduling effectively avoided the introduction of new files from engineers doing active development. After the master build, our regression tests were run and checked. Almost all of the master build process was done using HP SRC's batch mode processing capability. On the rare occasion when a serious bug was introduced, the file or files with the bad changes were checked out by the responsible engineer, the builder would COPYREV the previous stable revision to the Ready Group, and recompile. Once the master build and regression tests were verified to be successful, a symbolic name was assigned to the "release" through the ADDSYM command (e.g. `ADDSYM @,,RELEASE5`).

8. Symbolic Naming

Through the powerful feature of symbolic naming, a name can be assigned to the files and their corresponding revisions that make up a release. Most HP SRC commands that deal with files allow the option to reference them by a revision number (e.g. 1.2, 4.5, 1.2.2.2) or a symbolic name (e.g. RELEASE1, X.00.05, A.00.00, LASTBUILD, etc.) This allows for reconstructing entire releases through a single command. Consider the "mini-example" where you have a release named X.00.05 that is made up of the following files and corresponding revisions: S1 revision 2.3, S2 revision 2.1, and S3 revision 2.7. Then the following single command, **COPYREV S@, X.00.05** would copy the revisions of those files to your group.

This feature proved its "bootstrapping" worth during our alpha testing phase. A bug had been reported against our X.00.06 internal release of the software. When we went to duplicate it, we found that the program file was missing. An oversight had occurred and the program file for that release had not been checked in! (Note that checking in binary files is allowed with HP SRC.) Development had occurred since that release had been distributed precluding reconstruction of the release from our Ready Group. Our rescue came through a COPYREV of the source files for that release using the X.00.06 symbolic name to reference the correct revisions of the file set. After a fresh build, we were on the road again!

9. Parallel Development

A major problem that HP SRC helps solve is concurrent development on the same set of files by different project teams. An example with a programming team is one where a team is working on major enhancements to software that has been released while the other team is making bug fixes to that same software. The solution to this concurrent development problem should NOT involve the programmer changing a file and then immediately re-typing those same modifications to the other team's corresponding file; rather, the software should be modified independently with plans for merging the separate changes together. This provides a more stable, insulated environment for each team to work in and greater control in the introduction of the independent changes.

HP SRC addresses this problem through the use of branch software development which facilitates relatively independent development on the same file. This is done by creating a new revision of this file along a separate development path parallel to the main one. This is easily accomplished by checking a file out and then checking it back in as a branch by explicitly specifying a branch revision number. For example, if one were to check out a file, FILEA revision 1.3, and then check it in as revision 1.3.1.1 (e.g. **CHKIN FILEA,1.3.1.1**), a branch development path would have been created. Users checking out revision 1.3.1.1 would get those changes along that development path. A subsequent check out of revision 1.3.1.1 and check in would result in a new revision 1.3.1.2 being created along that branch development path. Meanwhile, mainline development in parallel could occur from the "trunk" revision 1.3 such that a check out of that revision followed by a subsequent check in would result in revision 1.4, then 1.5, and so on. Any changes made on the branch starting at 1.3.1.1 or along the mainline development path continuing at 1.4, would be independent of each other. Later, the two paths could be merged with the ADDDIFF command. The LISTTREE command depicts a more visual representation of what has been described and looks something like this:

```
File: FILEA

1.5
|
1.4
|
1.3
|---1.3.1.1
1.2 |
|    1.3.1.2
1.1
```

An elementary example and a portion of ADDDIFF output may be demonstrated by building on the example above. Suppose a block of code (non-COBOL code chosen to emphasize HP SRC's multilanguage use) for FILEA's revisions 1.3, 1.5, and 1.3.1.2 looked like:

```
    rev 1.3              rev 1.5              rev 1.3.1.2
       .                    .                      .
Begin                Begin                Begin
  If a=c then          If a=e then          If a=d then
    begin                begin                begin
     a:=1;                a:=1;                a:=1;
     bc:=d;               bc:=d;               bc:=d;
     b:=c;                b:=c;                b:=c;
    end;                 end;                 end;
end.                  end.                  d:=e;
       .              $include f1          end.
       .                    .                      .
```

In order to merge the changes from the branch, check out the latest revision 1.5, then issue the command: **ADDDIFF FILEA,1.3.1.1-1.3.1.2**. This will add in all of the differences created by the branch revisions 1.3.1.1 and 1.3.1.2 to the checked out file in your group. The resulting merged file's same block of code would look like:

```
      .
Begin
??????????????????? Beginning of Conflict 1 ??????????????????????????
  If a=d then
??????????????????? Middle of Conflict 1 ?????????????????????????????
  If a=e then
??????????????????? End of Conflict 1 ????????????????????????????????
    begin
     a:=1;
     bc:=d;
     b:=c;
    end;
  d:=e;
end.
$include f1
      .
```

Key changes involve the lines "**$include f1**" from revision 1.5 and "**d:=e**" from revision 1.3.1.2. See that they have automatically been merged in. The line with "**if a=c then**" from revision 1.3, "**if a=e then**" from revision 1.5, and "**if a=d then**" from revision 1.3.1.2 could not be automatically merged. HP SRC accessed that a change occurred to this line on both development paths and therefore could not determine which one to use. This kind of logic problem has to be manually resolved by a programmer. Since the file's conflicts are well-marked, it is easy for the programmer to find them, to determine the correct line(s), and to delete the incorrect lines and surrounding lines that "flag" the conflict. The user is now ready to check the file back in as revision 1.6.

HP SRC's branching and merging capability was an indispensable feature in bootstrapping the project. As previously noted, we leveraged existing code of another internal tool. This tool, which served as our bootstrapping prototype, was still under active development by another team in the company. The challenge was to keep the two geographically separate development efforts in synchronization such that the enhancements and fixes of the two teams were preserved. The geographic differences resulted in a different process than the more typical one described in the previous paragraph. But, through the aid of the ADDDIFF command for merging changes in from the other team, our goal was accomplished.

Generally merges were done on a monthly basis. The "other" team would deliver the set of files that they had modified since the last merge. They determined that set of files in their own HP SRC environment through LISTREV functionality that enabled them to select the names of only those files that had changed since the date of the last merge. Those files were imported and each one was checked in as a branch development revision from the revision of the previous merge. A symbolic name had been assigned (via ADDSYM) to mark that revision of the previous handoff. An ADDDIFF was then executed and a resultant new file was created with the merged changes included. The files were then examined to manually resolve any conflicts that could not be done automatically. One typical merge session for the two teams saw 17 files change. In a single afternoon, we had merged the files, resolved the conflicts, checked in the resultant merged file, and recompiled the entire product! Major changes were integrated in a task that might have taken a week to do without the merge facility.

10. Code Documentation

HP SRC automatically generates key information in the proper documentation of software code. Code documentation is one of the most important tasks in holding down the high cost of maintaining software. It also is one of the most drudgerous ones for programmers and is often skimped on or not done at all. HP SRC aids in the automatic and "semi-automatic" production of documentation. Further, human error is reduced in recording some of a file's most basic modification history. Through a single LISTREV command a user can find out:

- who authored a particular revision of a file,
- when they made their modification,
- what the description is for the entire file (Description text),
- what the description is for the particular revision (Log text).

The latter two characterizations are prompted for by HP SRC and are supplied by the user. Although the user can reply with basically a null response ("//") or customize their UDCs to ignore it; at least by default, the prompt does occur and reminds the programmer what they ought to be doing! Another alternative to being prompted for Log text includes specifying an actual string of Log text in the CHKIN command line. Log and Description text can be updated through the use of the COPYLOG, CHGLOG, and CHGDESC commands. Hence, HP SRC does promote good programming practices. Except for the Description text, the above information can be viewed in two different ways:

(1) The command `LISTREV SRCPAPER,FORM=LONG` could be output to the printer or terminal and its MPE XL output would look something like:

```
HP SRC stat file for the PUB.TOOLS environment

SRCPAPER

Owner:        ALAN.TOOLS
Head Rev:     1.2
Mask:         NOMASK
Prefix:       ''
Symbols:     -none-
Locks:       -none-

Description:
Bootstrapping HP SRC:  HP Software Revision Controller Paper
for Interex in 1989.
----------------------------------------------
```

```
Rev:    1.2
Date:   TUE, JAN 24, 1989,  3:06 PM
Author: ALAN.TOOLS
Log:
This has the section on LISTREV documentation added to it.
------------------------------------------
Rev:    1.1
Date:   MON, JAN 23, 1989,  7:52 AM
Author: ALAN.TOOLS
Log:
This contains the outline for the paper.
============================================================
```

(2) When a user lists the actual file that has been checked out with the keyword LOG embedded in the text of the file, the keyword gets expanded in the text file with information pertinent to the revision such as the revision number, date, time, author, and Log text. This is practically free documentation! In the example below, we see the result of the MPE XL PRINT command on the file called SRCPAPER after it has been checked in and out of HP SRC twice. Before the initial check in of the file, the file contained the keyword LOG.

```
:PRINT SRCPAPER

$LOG: SRCPAPER $

REVISION 1.2 TUE, JAN 24, 1989,  3:11 PM  ALAN.TOOLS
This has the section on LISTREV documentation added to it.

REVISION 1.1 MON, JAN 23, 1989,  7:52 AM  ALAN.TOOLS
This contains the outline for the paper.

<< Actual text of the file would continue on here >>
```

With the use of the ADDPREFIX command, the user can add in a prefix character string so that expansion of the LOG keyword will be preceded by that. For a language like COBOL, you might want to have your Log text expanded with an asterisk in the first column (after the line numbers of course!) For example, **ADDPREFIX COBSRC1,*** would make those lines comments when the Log text was expanded. Other available keywords that you may want expanded in your files are AUTHOR, DATE, FILE, HEADER, LOCKER, REVISION, and SYMBOL.

11. Finding Bugs

In using HP SRC to track down bugs introduced between two revisions, we used HP SRC's LISTREV command to report what files were changed from the last build of a release and who made the changes. Through subsequent use of the LISTDIFF command, a report was produced of what actual source lines were added, deleted, or changed between revisions of the suspected culprit file(s). These powerful features came in handy on more than one occasion during the project as our productivity in tracking down bugs was improved. All changes to a file are truly documented and tracked!

Another use of LISTDIFF in the project was borne out of the need to determine if an "unknown" file in one's group and account had been modified and if so, the completeness of said changes. On occasion, a engineer would find a file in their group resulting from a more general check out of the set of files that they felt would be needed to complete their programming task. The state of the file: whether it needed any modifications after all, whether it had already been modified since check out, whether it still needed modifications, etc; would be unknown to the engineer. If the engineer decided that the file did not need changes after all, they would just UNLOCK it from the HP SRC environment and purge it from their group. Otherwise, they would identify whatever changes had been made to the file (if any) with the LISTDIFF command and integrate any final ones necessary. Often, the changes in a file would be one line type changes, such as adding an error number to the constant declarations, or a global variable to the globals area, etc. Without HP SRC, these type of changes can be tedious to find!

12. Migration from MPE V to MPE XL

Originally, the prototype had been developed on MPE V and we initially followed suit and used an MPE V machine as our development platform. About halfway through the coding cycle, we migrated HP SRC, which is written in PASCAL, to an MPE XL system for continued development. This amounted to a recompile using the Native mode PASCAL compiler with a couple of OS dependent compiler switches set to deal with the larger pointer size on the XL machine. The record size of the HP SRC Stat file (basically a directory containing user and delta file information) was increased from 19 words to 26 words. This turned out to be a very simple and smooth migration and uncovered no new problems as we ran our development effort in production on the XL system. The conversion of an HP SRC environment from an MPE V one to an MPE XL one is:

- automatically done when the first HP SRC command is executed to the HP SRC environment on MPE XL
- a one time overhead to make the conversion,
- totally reversible (you can migrate back to MPE V in the same way).

The one-time, initial delay for the transparent conversion of an MPE V HP SRC environment to the MPE XL machine took in rough, *ballpark figures* about one second.

CONCLUSION

Version control systems are necessary to provide increased programmer productivity and help assure application integrity. Top systems will address a variety of software development problem areas such as file access conflict management, versioning, audit trails, parallel development, and security. HP SRC is such a system and provides features for check in/check out functionality, reporting of status information, "differencing" of revisions of files, merging of changes from separate but parallel development paths, and in addition, for providing support to non-programming activities such as documentation management.

HP SRC's actual use in the development of itself provided the product with:

- assured quality - as its source files were controlled by itself,

- high useability - as its intuitive command interface was designed primarily for programmers by progammers, and

- solid performance - as its developers were intolerant of productivity impediments to their own work.

Bootstrapping a product from the start of its lifecycle provided for the early detection of its problem areas and the identification of its strengths by first hand experience rather than esoteric thought. This insight allowed for the management of issues early on in the project lifecycle (e.g. the design stage) when they were easiest to solve and the least costly. Additional motivation to fix bugs and implement enhancements in the goal of "satisfying our customers" took on new meaning as those customers were not only HP's external ones, but our peers in the company, and **ourselves!**

ARCHIVE THEORY

Terry Floyd, Mehrdad Laghaeian
Blanket Resources, Inc.
12337 Jones Rd. Suite 408
Houston, Tx 77070
(713) 469-0869

Archiving, as a science, has a long history. The word archive has Greek and Latin roots: *archium* - the residence of a chief magistrate; *archivum* - the collection of official documents stored there. [1] One of the meanings for the word ark, as used in the Bible is: a place of protection or security; a refuge or asylum. [2] The word archive has always been used as a noun.

One of today's leading authorities on archiving and computers, Michael G. Cook, says of archives:

> Firstly, media which had been generated by an organization in the course of its business and which had turned out to be worth keeping; and secondly, that these archives had been selected by some means or other from a larger body of media produced by the same process which had not passed the selection test and much of which was not worth keeping in the long run. [3]

This implies that there is some middle ground between saving everything and saving nothing. The science of archiving is concerned with preserving and organizing; the art of archiving is knowing what to keep.

There are several definitions used by archivists which might be slightly paraphrased for this discussion. Accession is the physical transfer of the data. Appraisal is the process of determining whether to save certain data. Arrangement is organizing within archiving principles. Description means the establishment of intellectual control through the preparation of "finding aids" (i.e. indexing and choosing keys). Finally, review, to the archivist, means surveying to determine whether or not to allow open access (usually under the law). [4]

Archive, as a verb, means to save something that you want to throw away. In the case of databases, the production computer system runs slowly because of too many record entries, so systems practice is to purge inactive ones (perhaps putting them to paper or files). The decision about whether to archive to paper or digital format depends on how often someone wants to look at, copy, or analyze the information.

If the data is truly inactive forever (but, who can know for sure?), then archiving to paper (reporting) seems to be a feasible alternative. If one decides to report to paper, all information in the files should be properly formatted and printed. If to electronic media, either hard disk, optical CD, or tape is available. Frequent access rules out tape. High speed access indicates hard disc. If huge amounts of data are present, CD may be the right choice.

Data stored in a digital archive may be made available to on-line users for listing or recovery. The on-line archive can be available through the same commands which access the production databases, making the archival access to the information invisible to those using it.

There are two basic principles of archiving, as related by Sir Hilary Jenkinson (the father of modern archiving): 1) *provenance* and 2) *respect for original order.* Provenance is the principle that records from different record keeping units should not be intermixed. The intent of both of these rules is to preserve the independence and order set forth by the creators of the documents. [5]

The organization of the data may be meaningfully preserved or raw, flat files may be output. Providing the ability to dearchive (restore), or access the information makes a database like the one from which the records have been archived desirable.

Consider the case of flat file output. If reading from an Image database and writing to a file, records could consist of all of the items in each dataset, probably in the same order and of the same length as originally defined in the schema. In this case, for example, if there is an invoice header in one set and numerous invoice lines in an associated detail set, two flat files could be produced. The header file could be sorted by invoice number, and the detail file could be sorted by invoice number and line item number. As the records as deleted from the production Image database, they are written to the flat files. Dearchiving en masse from the flat files back to the production database would be fairly straightforward. But, dearchiving, or simply listing to screen or paper, any individual document would require a lengthy sequential read of both files (or a still time consuming binary search by the sorted key). This case is even worse if more than one dataset is required to store the original document.

Using an Image database of similar (even slightly improved) design as the original will make the entire process much simpler. As records are deleted from their production datasets, "clones" are written to the on-line archive's mirror datasets. Writing the records to the archive before deleting them can be very helpful during recovery from a system failure archiving. The archiving principles of respect for original order and provenance dictate that the duplicate database approach be followed.

One can archive every record or summarize. Summarized data takes less space, but some data has been lost; the changes to the database have been purged and, inevitably, some significant little piece of information is not available for retrieval or analysis.

There are some valid reasons to summarize data to a historical database, but archival is not one of them. Summarization certainly can be used to save disc space. It may also be used to dramatically increase performance in selected reporting situations. For example, take one small aspect of the information contained in invoice line items -- sales by product line by fiscal month. Reports can be written which process each line of each invoice each month and summarize the sales by product line. However, to compare twelve fiscal months of sales at the end of every month in this manner requires not only a lot of processing time, and not just redundancy (having been processed each month before), but that a whole year's worth of data be available. Therefore, it is desirable to summarize sales by product line by fiscal month and save this new information in a database. The summarization can be done on-line, during invoice processing (which will slow down the invoicing function) or in one pass at month-end.

The desirability of a summarized historical database does not negate the need for detailed archives, unless every conceivable aspect of the information in the documents is preserved. Since some data does not lend itself to summarization and excessive processing time for handling every item that can be summarized precludes doing so, detailed archiving is the only viable solution to guaranteeing future availability. In fact, considering the volume of information in her obvious item for summarization, product sales by fiscal period for five years (60 records per product number), an archive of the summarized historical database might be a reasonable approach for a company with thousands of product numbers.

An important point to make is that a document may never simultaneously exist in both the production database and the archive database. During archiving, as the records are added to the archive, they are deleted from the production database. The same is true of dearchiving: as records are moved back to production, they are deleted from the archive. The question then arises: what if the users create a new document in the production database with the same identifier (invoice number, for example) as an old one in the archive? This is where the slight redesign of the database mentioned above comes into play. If the invoice header was in a master dataset in the original design, with invoice lines in a detail dataset linked by invoice number, the archive design will use an automatic master for the invoice number and details for both the header and the line items. Appended to the cloned fields from the original dataset, the archive dataset for the header contains the date the invoice was archived. If a duplicate invoice number is archived at a later time, no collision occurs. The list commands and the dearchive commands can be made to recognize duplicates and display the date they were archived to help decide which one to use.

As information continues to accumulate, the on-line archive may itself be archived and date stamped to tape. If the off-line data ever needs to be accessed, it can be moved back to the on-line archive.

Care must be taken to upgrade the archive databases along with the production database. As new fields are added to the on-line live version, they must be maintained on the on-line archive files. This means converting the archive database with each new enhancement release. The off-line archive, however, does not get converted. As the records are written to off-line archive, a release version code and date stamp are attached to each record. If some of these records must be recovered after an upgrade of the on-line system, some new (or hopefully not, some missing) fields will have to be dealt with (through default or time-consuming creation of new data from old).

Data compression is an option for archiving/dearchiving either the live on-line or off-line archive databases. It is usually a CPU intensive task, so the on-line archive should not be compressed if speed and low impact on CPU performance are important. The off-line archive can usually be scrunched with little impact.

The easiest way to name the archive is to use the same field names (items) and set names (files), but to change the database name and/or the group name. Field and set names may be prefixed with some character, such as 'X', to make Data Dictionary access simpler.

The importance of keeping an archive is apparent to some and not to others. In the short run, saving documents for on-line retrieval is needed because the users actually do access them. But as they age, they are looked at less frequently, then, finally, never again as individual documents. But, analysis of masses of documents makes it desirable to have the exact details of the documents available so that trends may be established. One can never predict which fields will become relevant in the future.

There may be another reason to keep archives. In closing, here are some remarks made by Sir Hilary Jenkinson in a speech to the Society of Archivists in 1960:

> In trying to explain...why...archives should be preserved...I have used the similitude of trees: pointing out the comparative obviousness of the beauty or use of boughs, leaves, flowers, and fruit and their absolute dependence on the roots.[6]

BIBLIOGRAPHY

1. Johnson, Charles. The Care of Documents and Management of Archives, Society for Promoting Christian Knowledge, 1919.

2. The Random House Dictionary of the English Language, 2nd Edition, Random House, Toronto, 1987.

3. Cook, Michael G. The Management of Information from Archives, Gower Publishing, Aldershot, Hants, England, 1986.

4. Daniels, Maygene F. and Walch, Timothy. A Modern Archives Reader: Basic Readings on Archival Theory and Practice, National Archives and Records Service, U.S.G.S.A., 1987.

5. Pederson, Ann. Keeping Archives, Australian Society of Archives, Inc., 1987.

6. Jenkinson, Sir Hilary. Selected Writings, "Roots", Alan Sutton Publishing, 1980.