EDP ANALYZER
Editor and Publisher: RICHARD G. CANNING

EDP ANALYZER
© 1981 by Canning Publications, Inc.

JULY, 1981
VOL. 19, NO. 7

A NEW VIEW OF DATA DICTIONARIES

As we have been pointing out in recent issues, it appears to us as though the new data management systems (DMS) will have a very big impact on the computer field. We have discussed how these systems will encourage the use of database technology, promote end user programming, and allow faster development of new application systems. In this report, we discuss what is potentially an equally important role for them—support for the 'management of data.' A DMS data dictionary may meet most needs of a small organization. In a larger company, it can aid data administration by supplementing the full data dictionaries. Data dictionaries are here to stay.

Yuba College, with its main campus in Marysville, California, about 45 miles north of Sacramento, is a community college that has about 10,000 students enrolled. In addition to its main campus, the college has three mini-campuses located in three other counties, the furthest being some 90 miles away from Marysville.

Administrative data processing at Yuba College, until mid-1978, was on punched card equipment. The college had been reluctant to switch over to computers because of the disappointments experienced by other California community colleges with their computerized administrative applications. But by 1977, it was apparent that something had to be done; the punched card system just was not keeping up. So the college hired a director of data processing in early 1977.

The new director studied available hardware and software during the remainder of 1977. He came across a hardware/software combination that looked attractive, offered by North County Computer Service, of Escondido, California. The computer was a DEC PDP-11 (an 11/70 in this case), along with NCCS's data management system called USER-11—plus a student registration and records system that NCCS had developed to run under USER-11, covering student records, classes, instructors, classrooms, transcripts, and more. He recommended this combination, and it was ordered in December 1977.

ISSN 0012-7523. Multiple copy prices listed on last page. Photocopying this report for personal use is permitted, providing payment of $2.00 fee *per copy of report* is made to Copyright Clearance Center, Inc., 21 Congress Street, Salem, MA 01970; please include identifying fee code 0012-7523/81/070001-12$2.00.

During the next several months, the director worked with each of the departments that would be using the new system, such as registrar, personnel, office of instruction, and so on. As is usually the case, he found that the different departments viewed the data they used somewhat differently, based on their individual needs. By getting representatives of the departments together, he usually was able to get them to agree on common data definitions and data entry procedures. Occassionally, however, it made more economic sense to allow redundant data with slightly different definitions.

With these data definitions in hand, and working with the NCCS people, the director found it very easy to define the data to be used by the package so that it would meet the college's own needs. This was quite different from the typical situation, where the user is forced to use the package's data definitions.

Thus, in this instance, he was able to add or delete data fields from records, or shorten or lengthen data fields. He determined which record types were to be inter-related and specified those inter-relationships—such as relating students' records to the course records for the courses they were taking. He and the NCCS people then entered these data definitions into the USER-11 data dictionary. These definitions became the ones that the users could access and examine, and were the same ones that the programs would use.

The characteristics of a data field include (1) a brief name (up to 8 characters), (2) field length, (3) an explanation of the field (up to 32 characters), (4) a prompt name to be displayed during input (up to 24 characters), (5) a report column heading version of the name, (6) an edit mask (for, say, showing location of the decimal point), and (7) field specification characters that indicate if the field is updatable, is to be supplied during initial entry of a record, is numeric only, etc.

An important point that the director made to us is that it is essentially not possible for programmers to change the data definitions that the programs use without at the same time changing the definitions that the users see. In theory these definitions could be different, but the programmer would have to go to extra effort (and violate policies) to accomplish it.

The new system arrived in June 1978 and was tested until mid-July. The student registration was to begin in early August. The college originally had intended to run both the old and the new registration systems, but budget limitations prevented this. So the college just started in using the new system as the students began registering—and it worked.

A few months later, a dean of one of the divisions asked about USER-11. He was given a user's manual plus about 30 minutes of instruction. Within one week, he had written an accounting package for his division. Shortly thereafter, another division became interested and asked to use the package—so the dean modified the data definitions and program logic slightly to include the needs of the other division. Somewhat later he was asked to do it again for a third division—so he has ended up as the package's enhancer and maintainer (not a particularly demanding task in this instance, we gather). The package probably needs to be completely redone now, we were told, and made into a standard package with common data definitions, so as to relieve the dean of this maintenance. But the package is working and no one has yet had the time to redesign it.

The college is presently considering getting a second PDP-11/70 with USER-11 for instructional use.

So Yuba College, with an integrated data system, has found that USER-11, including its data dictionary feature, has given them a very powerful tool. They were able to make the change—in one relatively easy step—from punched card processing to a useful, usable system that was tailored to their needs and that employs the latest computer technology.

Mobil Corporation

Mobil Corporation, with headquarters in New York City, is best known as a leading energy company. *Forbes* magazine lists Mobil as the second largest U.S. industrial company, with annual sales of almost $63 billion and over 213,000 employees.

We talked with the manager of data base administration (DBA), in Mobil's systems and computer services department at corporate headquarters. We were interested in learning how Mobil is approaching the question of data administration and the use of data dictionaries.

A large part of Mobil's domestic data processing is performed at two major computer centers. They are located in Princeton, New Jersey and Dallas, Texas. Both centers employ multiple IBM mainframes and the company makes extensive use of the IMS database management system.

Mobil's data base administration department has been evolving since late 1976. Since then, the DBA function has gained acceptance for three key objectives which lie at the heart of sound data base planning, they feel. These objectives are: (1) data is an important shared resource that should be managed and controlled just like other major corporate resources, (2) knowledge on how data and information are generated and used should become widely disseminated, and (3) control should be exercised over the quality of the data resource, to increase its effective utilization.

The DBA function exerts control over the data resources through management control of the application development, test, and production environments. Mobil's DBA function sees its proper role as a full-time participant in the planning, design, and operation of data base systems, with a view to insuring that adequate features and appropriate data safeguards are provided.

In all of these efforts, the manager of data base administration sees the data dictionary as the primary tool for his function. The data dictionary is being used to organize the collection, storage, and retrieval of information about data. This meta-data (data about data) includes: the data's forms, characteristics, and inter-relationships; descriptions of what data is held and the sources of the data; how data is defined and structured; the different forms in which it may appear for different purposes; the different usage contexts, and how it relates to other data, systems, and user reports.

Further, the facilities of the Mobil Data Dictionary (MDD) are useful to all system development projects, non-database as well as data base. The main thrust is to capture information describing data items and their attributes, highlighting their inter-relationships. Functional analysis is also performed through the dictionary, based on its ability to define business activities in a machine readable form. Data dictionary usage is of particular value to applications using data whose characteristics are already contained within the MDD.

The MDD is a tool, we were told, which enables the DBA function to: clarify and design data structures, avoid unwanted data redundancies, generate accurate and dependable data definitions, assess the impact of proposed computer system changes, and enforce standards related to data.

Because of the interest in management uses of data, the DBA function looks to see what management-type data can be provided by each new application system. The DBA introduces appropriate integrated data base designs that minimize the need for special processing to make that data suitable for current and future management uses.

As one example of his role, Mobil's DBA cited his firm's HURIS system—*hu*man *r*esources *i*nformation *s*ystem. Now under development, this system will establish a single source of 'people' data within Mobil, servicing such application functions as payroll, employee relations, benefit plans, and so on. In addition to their on-going review function and assistance in data base design and installation, the DBA function developed a control monitor that supports an unusually flexible security apparatus, as well as an on-line report request and distribution system for end user use. This latter handles both pre-defined and ad hoc reports, using the MARK IV package in batch mode and Answer/DB for quick query capability.

The first step in Mobil's approach to using the data dictionary was to acquire the basic dictionary capabilities and train the DBA staff members in their use. Then corporate policies were established that required dictionary use for all data base projects. The DBA is now in the process of developing additional tools, procedures, and educational materials to enhance the usefulness of the dictionary. This activity is expected to take several years for its full completion.

Ultimately, the DBA function must impose a discipline on data content and not just its form, the manager told us. His staff must co-ordinate the definition of all data that crosses departmental boundaries; for data that is used only by one department, there is less need to impose standards. The DBA function needs to clean up the existing data definitions and then monitor all additions, deletions, and changes to these definitions. A DBA function that just records existing unstructured data definitions is not sufficient, he added.

In the long run, monitoring and editing of data definitions are essential, if data resources are to become truly sharable. The editing of key data definitions found in all application systems is a very large undertaking, so Mobil is approaching it cautiously, building experience at each step.

An area of current interest is the mechanism by which the DBA function can support business planning. Beginning with the business objectives, the business processes needed to support those objectives are identified and recorded in the dictionary. Next, the information needs of those business processes are identified and recorded in the dictionary, and support the locating of data-sharing opportunities. All of this data—data about business opportunities, processes, information, and sharing potential—can be made more manageable in the context of a data dictionary.

These planning processes are not yet fully realized at Mobil. They depend on developing the DBA's own understanding of these methods and educating the corporate community in the advantages of the formal definitions of information entities.

Mobil has thus embarked on a broad program for the management and control of the firm's data resources.

Some problems with data

A *data management system* (DMS), as we have been discussing in recent issues, is a system that includes (1) a data dictionary for defining new files, records, and fields, plus indexes for accessing the records, plus a means for allocating disk space for the files, (2) a means for creating screen formats for inputting and validating data, (3) a means for entering data and updating the database, (4) a record selection and sorting capability, (4) a query capability, (5) a report formatting and column totalling capability, and (6) a means whereby specific application logic can be expressed.

As this list of functions indicates, a DMS can perform most of the routine aspects of a data processing application. To create a customized application system, generally all that is needed is to write the specific application logic and/or output formatting programs (for those cases where outputs must be prepared on pre-printed forms).

In recent issues, we have described DMS that run on mainframes and/or commercial time-sharing networks (such as RAMIS, FOCUS, and NOMAD), as well as some that run on mini-computers (such as INFO and Information on Prime, Vision on Four Phase, and USER-11 on DEC PDP-11). Also, as we will discuss next month, we have come across *program generators* that generate COBOL or PL/1 code to perform data management functions, for use on mainframes or minis.

(As mentioned in our last two issues, we have prepared a list of DMS that we have come across. For a free copy, write us.)

One of the key points of a DMS is its data dictionary facility. The user is asked to define a new file, including both record and field definitions. This is a quite limited data dictionary facility, when compared with the full data dictionaries that are on the market (and that we discussed in our January 1978 issue). Only a relatively few attributes of the data are defined, such as a short field name, field length, mode (alphabetic, numeric, etc.), brief explanation of field name, and the name to be used in input screen prompts and for output column headings.

All of these attributes can be expressed by the user in a manner that can be easily read and understood; when the user calls up from storage the definition of a file, the data fields and their characteristics are immediately apparent. In addition, it is these *same* definitions that are used by the programs for processing the file. This characteristic makes it essentially impossible for the definitions seen by the user to differ from the definitions used by the programs.

Further, these definitions are stored in the system library and are associated with the *files*, not with the programs. When a program calls for the use of a file, the system retrieves the appropriate definitions. And by using the 'directory' facility that most operating systems provide, a list of all files that are resident on the system can be quickly obtained, after which the data definitions for each of the files can be retrieved. Thus a person who is performing the data administrator function can have a very rapid access to all of the current data definitions.

We have made these points about the data dictionary facility of a DMS in order to contrast it with today's 'conventional' way of handling data definitions.

Conventional data definitions

The manner in which data has been defined in most of today's application systems has been a function of (1) the programming language used and (2) the habits of the individual programmers who have written the programs.

Some of today's programming languages have the characteristic that data definitions ('declarations') can be embedded in the programs. If these embedded definitions are used only locally in the program, no data administration problem is raised. But if the data associated with these definitions is stored or appears on reports, this characteristic can make it very difficult for anyone, including a person performing data administration, to obtain these data definitions in order to compare the definitions used in a number of programs.

One of the major advances toward disciplined programming that was made by the COBOL language came from COBOL's data division. In this division, all of the data definitions used by a *program* are stated explicitly. This made the task of retrieving all data definitions much easier than when those definitions are embedded and scattered throughout the programs.

But COBOL did not fully solve the problem of controlled data definitions. As just indicated, COBOL itself calls for the storing of the data definitions *in each program*. If the same data file is used by (say) five programs, each program would include the data definitions for that file. If changes are made in the data definitions for that file, it is necessary that *all* programs that use the file be identified and their data divisions changed.

To make maintenance of data definitions easier, many users of COBOL have adopted the idea of a data definition library, from which each program 'calls' its appropriate data definitions. In this case, if the programmers have consistently followed company policy, a change in data definitions may need to be made in only one place—in the data definition library.

But even this approach has had its difficulties. COBOL does not *require* that the data definition library be used; it only allows it. If a programmer decides to use his/her own data definitions for a program, for what seems to the programmer to be good reasons, this event can escape detection.

So what has been the end result? Many user organizations have ended up with multiple data files in which the same data item has been defined differently—different name, different field length, different updating cycle, etc. Also, different data items have ended up with the same name. And sometimes the same data item has been defined 'well' in certain instances and 'poorly' in others, to the point where the (supposedly) same data item in two files cannot be compared.

Other difficulties from conflicting data definitions have included (1) the need for special programs to convert 'division' data to conform with the 'corporate' data definitions, (2) loss of data integrity, due to mistakes because of the multiple definitions of the same data item, (3) confusion in interpretation, with consequent mistakes in decision making, due to the multiple definitions, (4) loss of opportunity for using the data for additional purposes, because the data items were poorly defined in the first place (too application-specific), and (5) loss of opportunity for sharing the data across organizational boundaries, again because the definitions were too application-specific.

In short, as computer users have studied their existing files, to see how well their data has been defined, all too often they have concluded that the situation is a 'mess.' Many of the data definitions have been embedded in programs (and often implicitly rather than explicitly), and there has been so much variation in the definitions, so that trying to clean up the mess would be a long and very costly effort, they have often decided.

With DBMS, problems still exist

Database technology has offered a solution to the problem of conflicting data definitions—but the promise it has offered has not been fulfilled to anywhere near its potential.

Database management systems (DBMS) offer the opportunity to store data in non-redundant form—that is, a data item would be stored only once and would be made available to all application programs that require it. Moreover, the data definitions would be *explicit* and would be stored with the database, not with the individual programs. In addition, the logical definitions of the data (the ones used by the programs) would be reasonably independent of the physical definitions (the way the data is physically stored).

In theory, if an organization starts using database technology, from that point on the data definitions would be under control.

In practice, that is not what has happened, in all too many instances. What has happened is that user organizations have used the database management system as a glorified data access method, in order to get a number of 'peripheral' benefits that the technology has offered. These benefits have included multiple access paths to the data, the built-in restart and recovery methods, query and reporting packages that have been mated to the DBMS, and so on.

Also, because of the high cost of cleaning up the existing data definition mess, database technology has been used mostly for new applications. A DBMS has frequently been acquired to manage the database for one new application. Subsequent applications have been set up on an application-by-application basis, with each application having its own database. There has been relatively little sharing of data among applications. The result has been that programmers have still had the opportunity to define the data for each application, if they were so inclined.

In addition, data administration (the planning and control of data and data definitions) often has been looked at as an expensive overhead function. To reduce the cost of data administration, compliance with standards and policies has been on a voluntary basis, with not a lot of monitoring.

However, some user organizations adopted the opposite view—that effective data administration *is* important—but with equally poor results. These were the organizations that decided to clean up all data definitions once and for all. Just identifying the definition problems proved to be such a huge, costly effort that the projects we are familiar with were soon disbanded. Management decided to 'limp along' with the current poor definitions rather than pay the price to clean them up.

In a good number of organizations, we have been told, they have acquired data dictionaries to help achieve an effective data administration. But these data dictionaries have been loaded only with after-the-fact data definitions, so that discrepancies still occur—the user definitions and the program definitions do not agree. However, in those cases where a management policy was set up *and enforced* that all *new* data definitions, and definition changes, must flow through the dictionary to the DBMS, then a step forward was made toward stopping the mess from getting worse. Just making this one step often is no simple matter, but it still falls far short of achieving effective data administration.

One can conclude, then, that database technology has allowed progress toward better data definitions, but that there is still a long way to go.

How about data sharing?

As mentioned, one of the main promised benefits of database technology was the opportunity to share data among many applications. But, as was also mentioned, in general this benefit has not materialized to any great extent.

Why not?

We have come across a number of explanations of why data sharing is not widely practiced. The real reasons are often either not admitted or are hard to pin down, so we suspect that it would be difficult to conduct any sort of statistical survey on this. The explanations tend, therefore, to be opinions.

The explanations that we have encountered fall into three categories: political, local needs, and security.

'Political'. Executives and managers tend to want to control 'their' data. They do not want reports on their operations to be released until they have had a chance to review the data, to see if data errors have caused performance to look bad (or to let them try to cover up bad performance data).

Managers often see themselves as competing with other peer managers for promotion to higher positions, and thus want to guard their data from being seen by these 'competitors.' They may feel that their departmental budgets are vulnerable, so as Peter Keen has said (in a panel session at the 1980 National Computer Conference), "They protect their budgets by hiding their data, or by disagreeing on what the data means." They perhaps fear that other managers will 'browse' through their data files, looking for embarrassing data. Or they may have purchased the data, with a consequent expense to their departments, and they do not want other managers to have free access to that which they have had to pay for.

Some managers see a strong data administration function as just another political ploy by the data processing department to get more control—so they resist anything like a central database to serve all applications.

Local needs. Many large organizations are, in fact, engaged in a variety of quite dissimilar businesses or activities. These businesses and activities have often been obtained by mergers and acquisitions (including governmental agencies getting new responsibilities by acquisition).

The managers of the diverse units may make a good case that, because their needs are so specific, it is not possible to come up with a workable set of common data definitions. Since the data definitions for the diverse units differ, it is not feasible to have a corporate database that serves all applications—hence data sharing among the units makes no sense. So say these managers.

Security. Another argument against the corporate database is the risk it involves. The data represents a very valuable resource. To the extent that it is centralized, the risk of loss, damage, or undesired disclosure increases—so any efforts to make data more sharable will increase these risks, claim the proponents of this argument.

The net result of these attitudes is that data sharing has not been achieved to as great an extent as proponents of database technology originally expected.

Of course, it would be incorrect to say that standard data definitions and the sharing of data have not occurred. Some data, such as financial data, *must* flow across organizational boundaries, from the operating units to central accounting. Standard data definitions clearly are desirable in such cases, and most organizations have imposed these standards.

Also, computer technology is allowing the various levels of management to probe into data files, to look for (say) explanations of performance variations. Where top management sees the need for such probing, it is likely to mandate standard data definitions.

Further, computer technology is allowing a more disciplined approach to planning and budgeting than some organizations have found to be feasible before. In such cases, management will expect planning and budgeting data to move up and down the management hierarchy efficiently, using computer methods, and this will require standard data definitions.

There is another use of data that argues for better data definitions, without bringing in the idea of sharing data among organizational units. This is the concept of *multiple uses* of the data, within the organizational unit that controls it. A data administration function can develop a set of 'good practices for defining data' that can help provide more uses for the data.

One can say, therefore, that data sharing has not reached the expected magnitudes—and probably will not, at least in the foreseeable future. But management's desire to probe data files, as well as the possibility of making more use of existing data files, are likely to increase the pressure for better data definitions.

What is needed?

We see this need for better data definitions, as well as better control of them, as applying to a wide spectrum of organization sizes. In the past, much of the new computer technology was designed only for medium and large mainframe use. But today's new technology is usable not only on mainframes but also on minis—and, in some cases, on micro-computers.

What this means, of course, is that organizations which have their own computer for the first time—departments of some large organizations, as well as many medium-size and small organizations—now have the opportunity to repeat the mistakes of the past. We have seen some evidence that this repetition of mistakes is, in fact, occurring in the area of data definitions.

So, whether an organization is large or small, and whether it has been a long-time user of computers or not, there is a need to 'stop creating a mess' in data definitions.

To bring this area under control, two things are needed:

- Effective data administration
- Effective use of data dictionaries

Let us now look at what is involved for each of these activities, and the role that the data dictionary part of a DMS can play in them.

The data administration function

As described earlier, there have been numerous problems in the use of data that have been caused by poorly controlled data definitions. User organizations have used several approaches to help reduce these problems.

One of these approaches has been the centralization of data processing. Many companies have merged outlying, smaller data centers into one or a few large centers; over the years, we have discussed a good number of these cases. One of the main incentives for this centralization, of course, has been economy of scale—to reduce costs in both the computing equipment and in the operating staff. But in addition, this step has brought the data files, and hence the data definitions, under more central control.

Another approach has been to issue corporate data standards, for data that must flow across organizational boundaries. As the discussion above has indicated, this approach has run into difficulties when the different divisions of the company have been in quite different businesses. Standardization has been resisted because of the different needs.

Still another approach has been the use of common application systems. This approach requires that the different units of the organization have very similar data processing needs, so that it is practical to get them to use common systems—and, hence, common data definitions.

But these approaches have been only partly successful in controlling data definitions. Even with centralized processing, divisions can still retain their own data definitions. Data definition standards are difficult to develop and to enforce. Common systems generally apply to only a part of an organization's data processing activities, leaving the door open for the proliferation of data definitions in the other applications.

So even if an organization is using one or more of the above approaches, the need still exists for effective data administration.

Goals of data administration. The control of data definitions should allow users to merge and compare similar data that flows across organizational boundaries. As the use of data communications grows, common data definitions will aid the flow of data among the various units of an organization. User departments will be in a better position to make multiple uses of their data, if they have followed good data definition practices when setting up the definitions. And where management considers the sharing of data among organizational units to be desirable, good data definitions will help accomplish this.

Data that can be used for management decision making should be controlled, so that its use for this purpose can be fully exploited. Standard data definitions would probably be mandated for such data.

Also, higher levels of management may want the capability of probing lower level data files, looking for explanations of unexpected performance. Staff members may want to probe files, or to get subsets of files, for performing planning activities. In both of these areas, common data definitions will help.

The use of common data definitions will make application development easier. Users will be discouraged from making personal-preference variations in the definitions—hence the definitions will, in many cases, already be available, and perhaps so will the data files. If a user does, in fact, have a legitimate need for a variation in a data definition, this variation should be subject to a review and approval process.

The use of good, controlled data definitions will make program maintenance easier. It will not be so likely that a change to a controlled definition will affect different programs differently, as is often the case when there are several definitions in use for that data item. By providing independence between logical and physical data definitions, there is less need to modify programs when physical definitions are changed.

And where data definitions have been standardized, there is a greater chance that organizational units can exchange programs for handling 'their' data. The duplication of programming effort will be reduced.

Desired characteristics of data administration. These goals may seem worthy—but what price might an organization have to pay to achieve them? Or, stated another way, what should be the characteristics of data administration in order to make the cost of it bearable?

Not overly restrictive. In theory, the control of data definitions should apply only to data that flows across organizational boundaries. Data that is used only locally should be open to local definition. For instance, departments that have their own computers and that set up data files that are used only within the department should be free to define those data files as they desire.

The need here, it seems to us, is to make it as easy as possible for users to get the benefits of new computer technology—but without creating future problems.

It will not be an easy task to achieve this desired result. When a department sets up a new data file, it may view that file as local. But other departments may soon see the usefulness of the application and want to do the same thing—and even, perhaps, to use the same programs. Later, top management may wish to be able to probe some of these local files. The question then becomes: are these data files still 'local' or have they become 'common'?

Also, even for local files, departments probably should follow good practices (that may be developed by data administration) for defining data, so that they will not run into some problems later on that will limit their use of the files.

So it will be difficult to draw clean boundaries on which data definitions should be under the control of data administration and which should not. Conceivably, data administration will at least be asked to advise on all local files, in addition to its control of all common files.

Fast service. As new applications are being programmed, to run either on central data processing computers or on departmental computers, the developers will object to any lengthy waiting time to get new data definitions approved. The problem becomes particularly troublesome when the organization has many far-flung computers. So data administration must provide fast service for issuing new and changed data definitions.

Not-high cost. Data administration is an overhead function. Problems have arisen when this function has tried to do too much too soon. Potentially, there is a tremendous amount of *meta-data* (data about data, such as data definitions); for instance, we have come across cases where 25 to 30 attributes of a data item are carried in a data dictionary. If the data administration function undertakes a project to collect anything like this amount of meta-data about the data items in all of the existing application systems, chances are that costs will skyrocket. The project may well be cancelled at the first budget-cutting time.

Note the conflicting requirements here. Data administration might have to control most data definitions within an organization, and give fast, responsive service on requests for new or changed data definitions. But at the same time, the costs of data administration must be carefully controlled. It will have to be a lean, efficient function.

As we discussed in our January 1978 issue, a full, on-line data dictionary facility would seem to be an essential tool for effective data administration in larger organizations. And, as discussed in that same issue, top management must provide the basic policies (the 'edicts') and continued support, if effective results are to be achieved. In a small organization, such as a small company or a department of a large one, some degree of control must be exercised over the data definitions created by users, performed by the person most responsible for the computer. This person must set up policies and procedures that will insure his being informed when new data definitions are created or existing ones changed.

A point to note. In a large, widespread organization with multiple data centers and many application systems, it is physically impossible for one person to comprehend and understand all of the data definitions. A full data dictionary will be an essential tool; even with it, it will be hard to meet the requirements of data administration described above.

In a small organization, however, it is not unreasonable to expect one person to comprehend and understand all of the data definitions. The complexity generally is nowhere near as great as in a large company.

With these comments in mind about an effective data administration function, let us now see how the data dictionary function in a DMS fits in.

What a DMS dictionary offers

We see the following characteristics of the data dictionary feature of a DMS as supporting effective data administration.

Limited attributes. The data dictionary of a DMS provides for storing only a few attributes of data items. These are the attributes that are essential for entering, storing, and printing out data; the application programs cannot run without them. Moreover, the dictionary is used only for the files being processed under the DMS. So the dictionary is very pragmatic and practical. There is little chance for the data administration function to run up high costs in collecting huge amounts of meta-data when this dictionary is used.

Mandatory use. The data dictionary provides the only realistic way for defining the data that the programs running under the DMS will use. Programmers cannot 'not use it.' So the dictionary will always reflect all of the current data definitions.

Consistency. Both users and programs see the same definitions. There is no chance for the programs to use definitions that differ from what the human sees, when he/she asks that the definitions be displayed.

Responsiveness. All data definitions currently in use are stored in one place—in the DMS dictionary—and are immediately available upon demand. The data definitions are related to the data files which are stored under the DMS; a list of those files typically can be obtained quickly via the 'directory' feature of the DMS or operating system. So whoever is performing the data administrator function can easily locate all current data definitions, in support of fast, responsive service.

Supports prototyping. A DMS is a very handy tool for developing application systems by prototyping (building a system quickly, trying it out, changing it as necessary, and repeating the process). We can attest to that by personal experience. The data dictionary feature of the DMS is a necessary part of this capability. So it is likely that any new data definitions that are created, as a new application system is being developed, will be 'right'—that is, they are what the user wants them to be. With conventional development methods, sometimes almost-right definitions have been tolerated because they were detected after the programs had been written and it became too expensive to correct them.

So it seems to us that the data dictionary feature of a DMS can be a very powerful tool for data administration. It does have some shortcomings in this regard, however.

Some shortcomings. The data dictionary feature of a DMS is *not* the equivalent of a full data dictionary. We see it more as a complement to a full data dictionary in the data administration function of a large organization. For a small company, it may come close to meeting the data administrator's needs, however.

A DMS data dictionary, as mentioned, holds only a few attributes for each data item—short name, length, short explanation, etc. A full data dictionary can store many other attributes. If the data administration function is collecting existing data definitions from a number of current programs, probably more attributes will be wanted than the DMS data dictionary can handle.

In fact, it is not clear that one would even want to use this dictionary for storing data definitions of systems that are not being run under the DMS. It probably would be difficult to use it as an 'information resource dictionary,' a concept which will be discussed shortly.

We see the need for a 'bridge' between the data administrator's full data dictionary and all DMS dictionaries in a large organization. Probably this bridge should allow two-way communication. Whether or not the DMS dictionaries can be entered *only* by way of the full dictionary is something to be decided by each user organization.

Also, a DMS dictionary does not have a 'test' portion that is separate from its 'production' portion (although somewhat similar results can be achieved by using different names for the test files and production files). The data definitions in the dictionary are 'production' definitions, used by the active programs—but, at the same time, they are susceptible to being changed by anyone at any time, the way 'test' definitions are. This is a potential vulnerability.

Further, the DMS data dictionary does not have a design check facility, as do some development dictionaries (as discussed in our January 1978 issue). With this facility, after system analysts have entered what they think are all of the data definitions, the dictionary checks for inconsistencies and errors—data fields in files that have no input, fields that have been defined but not used, output fields that come from nowhere, etc. However, as we indicated earlier, the DMS supports prototyping, which probably is at least the equivalent of this design check facility.

Finally, the DMS data dictionary does not provide the extensive documentation that a full data dictionary does. For instance, it does not provide a cross-reference listing of all files in which a given data item is used. However, it typically does have a global search capability, with which all sets of data definitions can be searched for all occurrences of a data item name. This capability is equivalent to a cross-reference listing and has the advantage of always being current, which a listing may not be.

Even with its shortcomings, we see the data dictionary of a DMS as being a very useful tool for data administration, in both large and small organizations.

Information resource dictionaries

In this 'new view' of data dictionaries, we actually are looking in two directions. One direction is toward the rather austere, but still very useful, data dictionaries in today's data management systems. The other view, which we will mention only briefly here, is toward the much broader use of dictionaries in the future—the so-called 'information resource dictionaries' (IRD). The IRDs are not here yet, but they seem to be on the horizon.

To give a better picture of the possible future role of dictionaries in information systems, here are a few highlights of a working conference held last October and jointly sponsored by the U.S. National Bureau of Standards and the Association for Computing Machinery. This was the third such working conference sponsored by these organizations on the general theme of 'Data Base Directions.' The specific title of this working conference was "Information Resource Management—Strategies and Tools."

The working conference was attended by 68 invited participants from business, government, and universities. In general, each was invited because he/she is working at the forefront of the subject area; all are experts in their field. These participants were organized in four working groups, each with an assigned subject within the overall topic. The overall topic, in turn, had to do with the likely role of data dictionaries in information systems of the not-distant future.

The goal of each group was to discuss the assigned subject and then develop consensus thinking on what is likely to happen technologically in the next few years.

The message that came through to us was a surprising agreement on how information will be managed in the future. In general, the four groups felt that the term 'data dictionary' was too narrow; something like 'information resource dictionary' would be more appropriate.

Also, they typically rejected the role of a dictionary as only a support tool for a database management system.

Further, two of the groups independently discussed the idea of 'enterprise information' that begins in the business planning activity of the enterprise. This business planning information would be passed down through the levels of the organization, with the information at each level defined in the IRD for that level. Each level of management would add necessary details and then pass the information either down or up, as appropriate.

Why up? The planning process normally runs into snags; ideas that looked satisfactory at first turn out to be not feasible during the detailed study. When such difficulties are uncovered, this information must be passed upwards, so that the plans can be re-considered. Thus, the whole planning process is iterative.

Another point receiving attention was that, as the business plans are developed, the plans for information systems to support those business plans should also be developed. So the business information system plans, too, would flow down through the various levels of the enterprise, with more details added at each level. Again, snags can be encountered which call for revision of the information system plan or even the business plan.

This, in brief, is our interpretation of how this group of experts foresaw computerized information flowing through organizations in the future. To do this, an 'information administration function' will be needed, to develop, manage, and control the information definitions to make such information flows possible. This function would probably be an enlargement of today's data administration function.

If this group of people was seeing the not-distant future realistically (and we think they were), then dictionaries will be taking on a much more important role in the months and years ahead.

And the data dictionaries of today's data management systems provide a step in the right direction, we think, for getting better control of data definitions.

SUBJECTS COVERED BY EDP ANALYZER IN PRIOR YEARS

1978 (Volume 16)

Number

1. Installing a Data Dictionary
2. Progress in Software Engineering: Part 1
3. Progress in Software Engineering: Part 2
4. The Debate on Trans-border Data Flows
5. Planning for DBMS Conversions
6. "Personal" Computers in Business
7. Planning to Use Public Packet Networks
8. The Challenges of Distributed Systems
9. The Automated Office: Part 1
10. The Automated Office: Part 2
11. Get Ready for Major Changes
12. Data Encryption: Is It for You?

1979 (Volume 17)

Number

1. The Analysis of User Needs
2. The Production of Better Software
3. Program Design Techniques
4. How to Prepare for the Coming Changes
5. Computer Support for Managers
6. What Information Do Managers Need?
7. The Security of Managers' Information
8. Tools for Building an EIS
9. How to Use Advanced Technology
10. Programming Work-Stations
11. Stand-alone Programming Work-Stations
12. Progress Toward System Integrity

1980 (Volume 18)

Number

1. Managing the Computer Workload
2. How Companies are Preparing for Change
3. Introducing Advanced Technology
4. Risk Assessment for Distributed Systems
5. An Update on Corporate EFT
6. In Your Future: Local Computer Networks
7. Quantitative Methods for Capacity Planning
8. Finding Qualified EDP Personnel
9. Various Paths to Electronic Mail
10. Tools for Building Distributed Systems
11. Educating Executives on New Technology
12. Get Ready for Managerial Work-stations

1981 (Volume 19)

Number

1. The Coming Impact of New Technology
2. Energy Management Systems
3. DBMS for Mini-computers
4. The Challenge of "Increased Productivity"
5. "Programming" by End Users
6. Supporting End User Programming
7. A New View of Data Dictionaries

(List of subjects prior to 1978 sent upon request)

PRICE SCHEDULE (all prices in U.S. dollars)

	U.S., Canada, Mexico (surface delivery)	Other countries (via air mail)
Subscriptions (see notes 1,2,4,5)		
1 year	$48	$60
2 years	88	112
3 years	120	156
Back issues (see notes 1,2,3,5,)		
First copy	$6	$7
Additional copies	5	6

NOTES

1. Reduced prices are in effect for multiple copy subscriptions and for larger quantities of a back issue. Write for details.
2. Subscription agency orders are limited to single copy subscriptions for one-, two-, and three-years only.
3. Because of the continuing demand for back issues, all previous reports are available. All back issues, at above prices, are sent air mail.
4. Optional air mail delivery is available for Canada and Mexico.
5. We strongly recommend AIR MAIL delivery to "other countries" of the world, and have included the added cost in these prices.

Send your order and check to:
EDP ANALYZER
Subscription Office
925 Anza Avenue
Vista, California 92083
Phone: (714) 724-3233

Send editorial correspondence to:
EDP ANALYZER
Editorial Office
925 Anza Avenue
Vista, California 92083
Phone: (714) 724-5900

Name ______________________________

Company ______________________________

Address ______________________________

City, State, ZIP Code ______________________________

____

IN YOUR FUTURE: INTEGRATED IRM?

At a recent conference sponsored by *Business Week* magazine (as reported in the 2/9/81 issue of *Computerworld*), Arthur H. Schneyman of Mobil Oil Corporation gave his ideas on how companies may organize their information service activities in the future. Schneyman is manager of planning and analysis in Mobil's systems and computer services department.

These information service activities include not only computer services and tele-communications but also typing, mail room, text processing, records management, copying, office design—and strategic planning for these services.

Today, these information services are being handled by information resource units tacked on to the various line functions within a company, said Schneyman. But costs of these services are rising rapidly, and their effectiveness has a definite impact on company operations.

So, he said, why not combine all of these scattered information resource units into one line organization, under an information resources manager (IRM)? Cost savings and improved effectiveness should result.

The idea seems feasible, he said, for companies with substantial experience in the use of computers, and particularly the ones which have already combined computer services and tele-communications in one organization unit. But it still may take five years to implement this new concept, Schneyman feels.

It is interesting to relate Schneyman's views to those of the participants in the NBS/ACM working conference reported in this issue. It is hard to say just how much integration of information service activities will occur in most organizations, and how soon. But that is the direction of things, in the view of a number of forward thinkers.

Editorial: Richard G. Canning, Editor and Publisher; Barbara McNurlin, Associate Editor. While the contents of this report are based on the best information available to us, we cannot guarantee them.

Missing Issues: Please report the non-receipt of an issue within one month of normal receiving date; missing issues requested after this time will be supplied at the regular back-issue price.

Microform: EDP ANALYZER is available in microform, from University Microfilms International, Dept. P.R., (1) 300 North Zeeb Road, Ann Arbor, Mich. 48106, or (2) 30-32 Mortimer Street, London W1N 7RA, U.K.

Prices: Prices of subscriptions and back issues on page 13.

Declaration of Principles: This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional service. If legal advice or other expert assistance is required, the services of a competent professional person should be sought.—*From a Declaration of Principles jointly adopted by a Committee of the American Bar Association and a Committee of Publishers.*