IBM Application Development Tools White Paper

Document Number ADWSTRAT

March 1996

Notices

References to IBM® products herein do not imply that IBM intends to make such products available in all countries around the world. This paper does not represent a commitment that IBM or any of the vendors mentioned herein will make generally available any specific products at a future date.

First Edition (May 1994)

Second Edition (October 1994)

Third Edition (November 1995)

Fourth Edition (March 1996)

IBM APPLICATION DEVELOPMENT

Corporate down sizing, changing regulations, increasing competition, expanding markets ... all put pressures on business and therefore more pressures on application developers than ever before. The role of information technology (IT) departments is evolving to enhance support for the distribution of IT resources and provide global access to information on heterogeneous systems and protocols. This must be accomplished while continuing to control complexity, reduce cost, and provide optimal business systems support for competitive advantage.

How can application developers play a key role in this environment? Most businesses need to develop and enable new types of applications for competitive advantage. Usually, this means developing new applications that fit new business processes and work flows that have been re-engineered to enhance customer service. On the other hand, developers benefit most by investment in skills in the fewest number of tools that produce applications that address these needs and build adaptive application systems designed to respond to change.

In order to rapidly build competitive business solutions, it is imperative to create a total team development approach that engages the business process experts with the software development staff who know how to build scalable, functionally robust applications. These teams must cooperate in building new competitive solutions. The team members need the right systems, tools, and technology and a development environment that promotes quality, productivity, sharing of information, and a rigorous development management process.

IBM believes that our leadership in the application development business will require a complete set of tools:

- business and information modeling tools to support business process re-engineering including workflow modeling and control, prototyping and management
- visual programming to improve productivity, support a multiplicity of application types, e.g. client, server, transaction, network, data-centric, document-centric, and scalable execution environments, e.g. workstation, server, client/server, using construction from parts capabilities to enhance the traditional approach of building applications using programming languages, i.e. a compiler or interpreter for source code programs
- an object repository to manage and store the assets of development, i.e. models, source code, visual objects, test cases, execution control, documentation, etc.
- and software configuration management and build facilities such as version control, configuration management, problem tracking and change control, distributed and parallel builds and distribution services

IBM′s strategy is to provide an integrated development environment for enterprise businesses and workgroups. The tools that deliver the strategy to design and build applications and manage development assets are depicted in Figure 1 and as follows:

- Support for business and information modeling and workflow management
- Visual construction from pre-built parts
- Programming languages of choice
- Support for platforms of choice
- Iterative, roles-based team development
- Evolutionary utilization of objects

IBM′s marketing approach to application development is to offer solutions for:

- modernizing legacy applications
- automating business processes
- developing object oriented applications

These solutions are offered as functional combinations of software tools with complementary service offerings that address the phases of application creation and maintenance:

- Modeling and design
- Build or construction
- Development and asset management

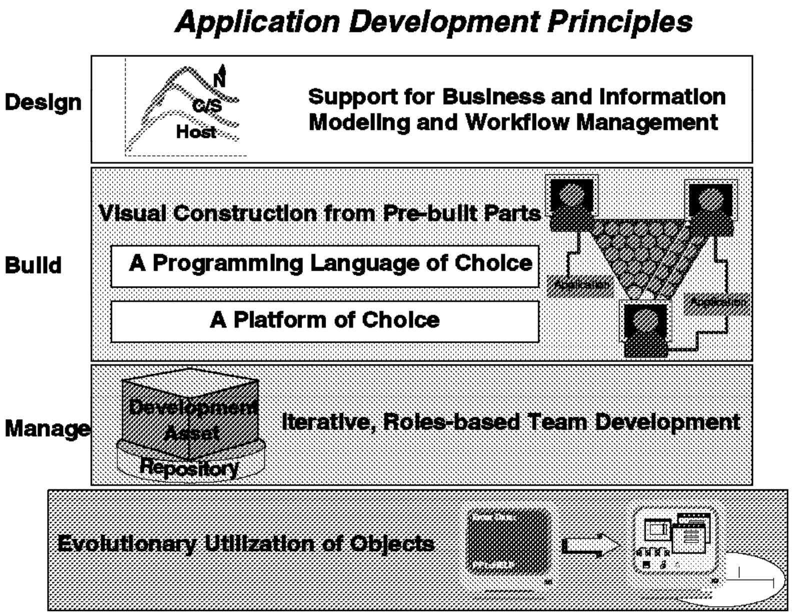


Figure 1.

Modeling and design tools

Modeling tools help describe a business problem and define an information technology solution. Design models must be a complete representation of the business process. Modeling tools must be able to update existing models to implement desired changes. Tools need to work together and share model information for different operations, i.e.:

- Business process re-engineering tools to assist with business process improvement and automation
- Information engineering tools to organize data and applications construction for more efficient development and maintenance of information systems
- Object modeling tools to develop new applications that use object technology to improve maintainability, flexibility, and implement applications directly from the modeling tool.

IBM Business Process Re-engineering tools

IBM Business Process Modeler (ProModeler), built on a time-tested IBM methodology, the Line of Visibility Engineering Methodology (LOVEM), graphically maps the way work moves through an organization. Process design teams can actually ″see″ a process while documenting, evaluating, and redesigning it. Processes are described in standard business terms, including process goals, critical success factors and quality measures. In this manner, teams highlight crucial points and identify bottlenecks.

ProModeler also identifies ways to improve team and customer interaction, identity new business opportunities and assess the impact of change on customers, vendors, and suppliers. These features enable teams to be more responsive to changes in their business environment.

Through integration with FlowMark™, professional developers can take the business model to the next step by creating, defining, documenting, simulating, and testing the supporting workflow processes. FlowMark is used to assign key company resources including people, programs and data to get the job done. FlowMark then manages and tracks execution of the process, i.e. through measurement and monitoring functions provided by FlowMark, companies can ensure that the processes fulfill their identified business goals, creating a closed loop for continuous process improvement.

Creating applications that integrate with FlowMark will be simplified with the addition of a library of visual construction parts that support workflow functionality in VisualAge™. These applications can be procedural or object oriented. FlowMark also supports the linking of existing applications with its run time management permitting businesses to use their existing application investments within the workflow application.

Lotus® Notes® provides a document and Notes centric means to informally control workflow. It can call FlowMark and FlowMark can call Notes. These offerings compliment one another in the execution environment, but ProModeler and FlowMark are especially useful in that they can model the effects of change prior to its implementation and bring discipline to controlling the flow of work to enforce business policies.

Reuse of existing and new application investments will also be enhanced through the business rule management function that will be integrated with FlowMark and the ProModeler. This tool checks rules that support the business and workflow processes against the rules that have already been implemented. The business rules management function also provides a link between the modeling teams and the application developers. It provides visual capabilities that support the transformation of business rules (described in business terms) to definitions and visual depictions of the relationships among applications (which implement the business functions) that are useful to application builders.

The combination of business modeling and workflow management results in a complete set of visual process design tools from IBM. They provide simulation, control, execution, and measurement for re-engineering a business process. IBM modeling tools also support developing applications from process models.

PACBASE/CS™ is also a complete set of design and implementation tools that come from Compagnie Generale d′Informatique (CGI), a wholly owned subsidiary of IBM. It is a model driven development environment for building enterprise server, batch and on-line applications. It has unique strengths in methodology support, including IBM′s Structured Design (SDDM), Merise, Yourdon, and, ultimately OO/AD. It supports applications in several OLTP, database, and communications environments that run on more than 40 IBM and non-IBM target systems. PACBASE/CS is especially appropriate when a single development environment is required to support IBM and proprietary systems from BULL™, DEC™, UNISYS™, and/or a variety of UNIX® environments. IBM and CGI will continue support for PACBASE/CS for its unique capabilities and through integration with VisualAge for client development and TeamConnection™.

IBM Information Engineering tools

Information engineering tools model database designs and the relationships of data elements and programs. An understanding of these relationships helps to more easily translate business changes into applications. Tools that keep track of the implementation of business rules and the use of information and data elements help reduce redundancy and promote reuse.

IBM′s business rules management function, mentioned earlier, visually helps translate business changes into application designs. It enables the description of business requirements to be recorded in a consistent fashion and with the discipline necessary for a complete implementation design by developers.

TeamConnection DataAtlas you to conceptually model the use of information by different applications. It keeps track of data definitions, suggests physical database designs, generates database designs that are optimized for performance, and includes data dictionary functions that are available to other application development tools.

DataAtlas, business rules management, and the build tools provide a complete development environment for many of the tasks associated with maintaining, modernizing, and extending most of today′s enterprise applications. They can be used in an integrated way with IBM′s business process re-engineering tools.

IBM Object Modeling tool

Object oriented technology is growing in popularity. Many organizations find that this technology fits their business needs. Critical to the successful use of object technology is a good design for the application. IBM is developing a new form of modeling that makes use of object oriented technology. The developer is able to model applications using business objects that represent business processes and messages and the tool produces an application based on the model.

Although this technology is still in development, ultimately it will produce application models and frameworks that can then be used in conjunction with visual build tools and with database design tools to produce an efficient and complete application. Frameworks produced in this way can be reused as templates for other applications. Together, these visual object modeling and build tools offer a complete object oriented application development environment for those wishing to pursue the advantages of the rapidly emerging object technology.

Build tools

Application developers have been using programming languages to build applications for years. This traditionally has involved understanding syntax, logic structure, and specifications for applications interacting with the environment where they run. This process is labor and skills intensive, frequently too slow to satisfy business needs and often produces applications that don't satisfy users' requirements. Many existing applications must be maintained and enhanced with new user functions and integrated with new applications.

In order to successfully implement and manage application solutions, developers need build tools and development management capabilities. IBM is providing a comprehensive set of build tools to improve developer productivity. IBM build tools support three key application development principles:

- Visual construction from pre-built parts for build time productivity
- A programming language of choice and a platform of choice to leverage programming skills for scalable client and server applications
- Evolutionary utilization of objects to mask the complexity of new technology and ease future application maintenance

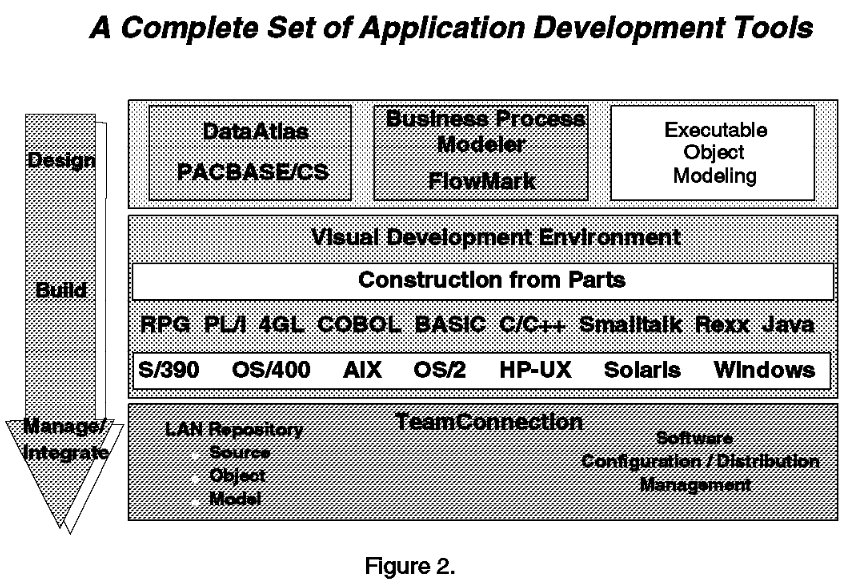


Figure 2.

Visual Construction from pre-built parts

IBM's strategy is to enable the developer to focus on business issues while lowering the barriers (depth of skills) to developing applications. Developers will be able to build applications from pre-built components -- construction from parts. This will dramatically improve development productivity while lowering the required entry skill for developers. IBM offers this visual development environment across a broad set of existing programming languages.

The visual construction approach provides benefits to those with existing transaction and database applications with visual tools to modernize these applications. It also provides for the addition of new client applications for existing host or workstation server functions. Since visual development using construction from parts reduces the need to know the underlying programming language and technology, it masks much of the complexity of developing client/server and/or networked applications and benefits those wishing to evolve to object application technology.

The VisualAge family of products, VisualAge for Smalltalk, VisualAge for C++, and VisualAge for COBOL, and VisualGen™ provides power programming for a distributed environment. These tools use visual construction from predefined parts on a developers palette to more easily build applications without using a programming language. Palettes combine easy client graphical user interface (GUI) construction from parts with powerful parts that represent application function such as business logic, data, transaction and network access, communication protocols, and multimedia to quickly build an application. A developer can use tools that come with VisualAge to build queries to databases without knowing the structured query language (SQL). The tools visually take the developer through the data access using information from the database and pre-built data access parts. VisualAge supports C++, COBOL, and Smalltalk applications for both responsive development and optimized object application execution.

VisualGen is a high productivity visual tool for generating three-tier (or one- or two-tier) transaction-based client/server applications. It is an object-based tool that uses the power of visual construction for both client and server applications. VisualGen has the unique ability to develop client and server parts as a single unit, then generate the tested applications to several different execution environments without changing specifications. This generator will be enhanced to support Smalltalk client functions such as partitioning and the use of SOM and Smalltalk parts.

IBM is extending its 3GLs with the inclusion of new technology and visual development environments. For example, VisualAge for COBOL for OS/2® and IBM COBOL for MVS enable the many developers familiar with this popular programming language to add object technology and visual construction of client/server applications to their skills without leaving the COBOL language. COBOL and PL/I are offered on MVS, OS/2 and AIX® with the same compiler technology. The visual tools for PC and workstation environments can be used to cooperatively develop applications for execution on other platforms such as MVS.

These 3GL integrated development environments include tools to help maintain and modernize existing applications. For example, visual tools will be provided to assist with program and application understanding and restructuring. Additional tools will be provided to more easily test 3GL distributed applications. Visual assistants are provided to enhance developer productivity for building transaction and database applications.

Ultimately, the visual builders of the VisualAge family will converge onto a single visual construction technology. This will enable parts libraries to be provided in a standard form, i.e. SOM/DSOM, that can be used with any programming language within the family.

Visual development environments from IBM are being delivered to extend several other 3GL programming languages such as RPG and BASIC. VRPG, which will run on OS/2 and 16- and 32-bit versions of the Windows™ family enables AS/400® programmers to extend their skills to client/server applications.

Lotus Notes development tools enable rapid development of applications that expand and extend Notes. Notes tools provide:

- Access to the Notes data store from VisualBasic™
- A high-level, object-based C language API for Lotus Notes
- LotusScript, a BASIC language tool for cross-application scripting in Lotus products

IBM is planning access to Lotus Notes from VisualAge and a converged BASIC scripting tool in 1996. IBM′s implementation of visual BASIC like functions in conjunction with libraries of BASIC controls and visual parts will support new applications including database, file and Notes access. It will run on the client platform of choice and will support the utilization of underlying technologies native to different environments, e.g. OLE™ (object linking and embedding from Microsoft™) on Windows and OpenDoc on OS/2 and AIX. Such a BASIC programming environment offered by IBM will provide client-centric development and is designed for scripting applications using OpenDoc parts. It can also build Basic OpenDoc components.

IBM′s scripting languages are designed to best serve the needs of a specific business requirement. For example, Script/X is designed for multimedia authoring. Object REXX that is available in many operating environments, and is imbedded in OS/2, is a general scripting tool best suited to write ′glue′ code to build new applications using pre-existing functions in the operating system or applications.

The construction from parts paradigm can extend skills to new technology based upon the use of visual tools. In addition to the access to Notes mentioned above, parts to integrate VisualAge with workflow and access the Web, new tools based on technologies such as Java will extend the power of visual building to those application types as well. Java is a technology for developing and executing logic attached to a Web page. It has been licensed by IBM from SUN™ and is being ported to OS/2 and AIX. A VisualAge for Java tool will enable such logic to be developed using visual construction from parts.

A programming languages of choice and a platform of choice

IBM′s strategy is to leverage existing skills, code and systems by providing the widest choice of languages across the widest choice of execution environments. IBM will provide its key compilers for IBM systems -- S/390®, OS/400®, AIX, and OS/2 and non-IBM environments -- Windows (3.1, 95, and NT™), HP™ and SUN. This will enable developer skills to apply to scalable client and server applications for a broad range of computing capacity.

For example, the languages of VisualAge are being offered on several platforms today and the list is growing. C++ is available on Windows, OS/2, AIX, MVS, OS/400, and SUN. Smalltalk for AIX, PL/I for AIX, and COBOL for OS/2 and AIX were all made available in 1995. This enables both client and server applications to be developed in the same language with the same development environment. C++ and Smalltalk will lead the trend to multiple platforms. They were introduced for Windows 95 and NT in March. Additional development and execution environments are planned for Smalltalk, C++, COBOL and VisualGen in 1996.

VisualGen includes a 4GL language for developer productivity. This allows it to generate the same application to different platforms without changing application specifications. VisualGen also generates character-based terminal applications, enabling developers′ skills to easily evolve to building both these host-based applications and client/server applications using the same tool. VisualGen supports data access across several data structures, a variety of middleware, clients on OS/2 and Windows and servers on S/390, OS/400, VSE, AIX and OS/2. It too will soon add 32-bit Windows platform support.

Traditional host 3GLs are being offered on additional platforms. Both COBOL and PL/I come with a visual development environment on OS/2 and AIX. COBOL for 32-bit Windows and UNIX will soon extend the range of COBOL skills to applications on those platforms as well.

IBM Open Class™, introduced with VisualAge for C++, and IBM Smalltalk class libraries enable a single development environment to provide portability to multiple execution platforms without changes to the application itself. They enable your organization or industry software suppliers to build upon these classes with across-industry and industry specific classes and frameworks that provide additional improvements in developer productivity and in application development responsiveness to business′ needs.

The use of class libraries masks the complexity of underlying systems infrastructures and provides pre-built and pre-tested functions for building applications and running them on multiple systems. For example, with the introduction of VisualAge for C++ for Windows, IBM integrated Taligent′s Compound Document Framework (CDF) into IBM Open Class. This simplifies OLE programming by enabling developers to create OLE applications with as few as two or three lines of code. Additional frameworks from Taligent, such as 2D graphics and print/view, will be integrated into IBM Open Class over time.

In addition to working with our industry on projects such as CDF, IBM continues to have close affinity with standards groups for all programming languages. For instance, IBM is working with the COBOL standards committees to ensure that IBM COBOL conforms to the emerging object standards for the COBOL language.

Conformance to ANSI and ISO programming language standards and offerings of the same IBM compiler technology across multiple platforms leads to a high degree of portability of applications developed using these tools. Portability is enhanced by using consistent middleware based on industry standards. IBM's Open Blueprint™ identifies categories of middleware and documents what resources are offered on several system platforms. IBM Software Servers provide a consistent implementation of several key software resource managers for scalable applications built to use server functions such as transaction management, database management, Notes, and message queuing.

Evolutionary utilization of objects

For those who recognize that object technology may be the best way to develop distributed applications, IBM offers visual construction from pre-built parts in C/C++ and COBOL that support both procedural and object application development. This enables the migration of skills to the best technology for solving business problems especially in conjunction with visual construction that can mask the underlying technology from the developer.

VisualAge for Smalltalk supports new object oriented applications using visual construction that can reuse many existing resources, including access to databases, transactions, and communication protocols, support distributed applications, and wrappers for VisualBasic™ controls (Windows only) and PC applications written in C or COBOL. This enables existing applications to be extended with new components that use object technology enabled on a wide choice of platforms.

The latter is part of our systems strategy to allow application components written in different programming languages to work together. This can be accomplished through the use of IBM's System Object Model (SOM). These applications consist of components that can be distributed across multiple servers and clients using Distributed SOM (DSOM). Our object strategy is to include a set of IBM pre-built parts with IBM tools, e.g. IBM Open Class and IBM Smalltalk class library. IBM is also working with industry suppliers through programs such as Object Connection™ to build parts in an open architecture that enable those parts to be used and reused in application solutions built with the VisualAge family.

Selecting A Development Tools Partner

IBM

- ☒ **Open, Standards Based**
- ☒ **Multi-Platform Support**
- ☒ **Re-use of Existing Applications / Skills**
- ☒ **Support for Distributed Client/Server**
- ☒ **Complete Object Structure**
- ☒ **Complete Set of Application Development Tools**

Figure 3.

VisualAge for C++ and VisualAge for COBOL and ultimately IBM BASIC support direct-to-SOM compilation to provide objects for management by each operating system that supports SOM/DSOM, i.e. OS/2, Windows, AIX/6000®, OS/400, and MVS as well as those that conform to the Object Management Group (OMG) Common Object Request Broker Architecture (CORBA). In addition to these direct-to-SOM compilers, build tools such as VisualAge for Smalltalk can reuse SOM objects as part of the application even though it uses a different object model (i.e. Smalltalk) for execution. VisualAge for Smalltalk includes SOM and DSOM Client support.

Today, class libraries are provided with tools like VisualAge for C++ and VisualAge for Smalltalk. Functions include collection classes of basic application building blocks, user interface classes, communications classes, data access classes, and multimedia classes to be reused as a part of the application being built.

IBM Open Class and IBM Smalltalk class libraries represent IBM's strategy to promote productivity from reuse of code and design. They contain a growing number of classes and frameworks that will simplify access to resources and the development of applications. This is done by abstraction of middleware interfaces and other functions (selecting parts is much simpler than coding procedural APIs) to protect the developer from the complexity of client/server environments and to enable rapid development of functionally rich client and server applications that access resources such as database, transaction, Web and Notes in a networked environment.

Development and Asset Management tools

All modeling, design, and build tools need a place to store the results of development. These roles-based assets need to be shared among teams of developers who apply different skills for designing and building applications and managing the development process. Assets must be managed for iterative change, different versions, and configured for installation with a build function to deliver the resulting application to its eventual execution environments and systems. Until recently, the integration of modeling information with application development tools has primarily been a manual process prone to error.

TeamConnection, groupware for developers, integrates software configuration management functions and object-oriented repository services. It provides a model for tool integration to support roles-based team development in a client/server environment. TeamConnection supports the development of host-based applications and client/server workstation and distributed applications. The developer has the choice of using either a procedural or an object development paradigm. Source, object and model assets will be stored, updated and managed for distribution to a variety of developers and for a variety of systems infrastructures. Other development resources such as JCL, README files, and documentation can be managed consistently with modeling and workflow information and the executable application assets.

TeamConnection integrates problem tracking and change control to ensure that application developers are more productive and project leaders can effectively manage and track the development process. TeamConnection automates and streamlines the application build process and integrates it with version and change management. The build function is tied to release management and extended to provide a framework for delivery to clients and servers using NetView™ Distribution Manager.

An open, extensible information model provides the vehicle for data sharing among the modeling and design tools and the build tools like VisualGen, VisualAge for C++, VisualAge for COBOL, and PL/I. The information model ensures continued support for new versions of existing tools and the integration of new tools with those that already exist. It has an extensible architecture and an object oriented information model that is an integral part of TeamConnection. Its repository services include constraint checking, version management, concurrent and distributed access, and data exchange for information model assets produced by several different modeling tools.

TeamConnection's LAN-based implementation allows developers to browse and reuse development assets with great speed. Host applications could be developed by checking out an entire program from a host-based library and little additional control was needed. The iterative object development technique using modular parts to assemble complete applications is significantly different from developing those traditional applications. A LAN-based implementation of asset management supports browsing different development objects and even different versions of the same object stored in a the repository. To enable easy migration to new techniques and coexistence of both object and procedural development, TeamConnection supports storage, retrieval and consistent management of all data elements, design, model, code parts, and objects in the same LAN-based client/server development environment.

TeamConnection provides persistent data store on OS/2 (server function), tool services at the client on OS/2, and distributed build functions for OS/2 and MVS. The platforms supported by TeamConnection will be quickly expanded. Server functions will be provided for Windows NT, AIX, and other UNIX platforms. Client functions will be available for the Windows family of products as well as for multiple UNIX systems. Distributed build capabilities will be extended to other IBM platforms such as VSE, OS/400, and AIX as well as to Windows platforms and other variants of UNIX. This will provide unprecedented platform range and support for the heterogeneous environments found in many enterprises.

Summary

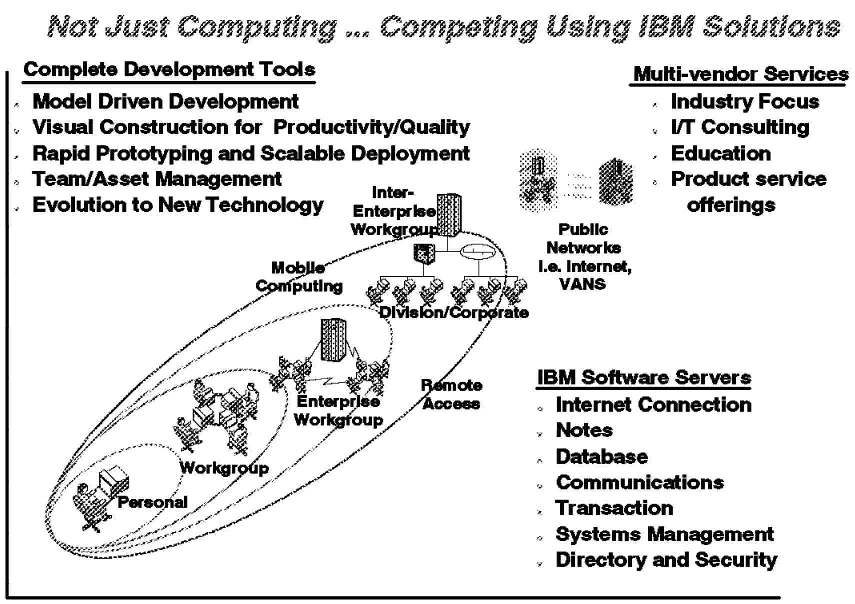
IBM's complete set of application development offerings provide significant value in these ways:

- Enable higher levels of developer productivity are enabled through intuitive and visual tools for automating business processes, modernizing existing applications, and leveraging the advantages of new technology
- Enhance productivity through support for visually building applications from pre-built parts by teams of developers with different roles and skill levels
- Provide parts to leverage middleware and infrastructure based on industry standards such as those identified in the Open Blueprint, and implemented across a broad spectrum of heterogeneous client and server systems
- Enable easier maintenance and application changes through modular construction
- Support object and workflow technologies

IBM's development and execution management assumes and manages the complexity of client-server applications that require support on many systems and configurations to ensure adaptability through flexibility and scalability.

IBM provides industry standard implementation of technology such as that offered by IBM Software Servers and brings leadership and experience to bear on your needs through education, consulting and services offerings. IBM can help determine the best course to follow and the best approach to getting started. IBM wants to satisfy your business requirements at a pace which is based on your priorities, your existing application assets and your skills and capabilities.

IBM's application development approach helps you get the most from people, leverage your application assets, minimize risk, waste, and failure, and allows you to use technology as a competitive weapon to deploy superior applications.



Trademarks

® AIX, AIX/6000, AS/400, IBM, OS/2, OS/400, S/390 are registered trademarks of International Business Machines Corporation.

Lotus and Lotus Notes are registered trademarks of Lotus Development Corp., a subsidiary of IBM.

™ FlowMark, NetView, Object Connection, Open Blueprint, Open Class, SOMobjects, TeamConnection, VisualAge, and VisualGen are trademarks of International Business Machines Corporation.

PACBASE/CS is a trademark of CGI, a subsidiary of IBM.

Bull is a trademark of Group Bull

DEC is a trademark of Digital Equipment Corp.

HP is a trademark of Hewlett-Packard Company

SUN is a trademark of Sun Microsystems, Inc.

UNISYS is a trademark of Unisys Corporation

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

OLE, VisualBasic, Windows, Windows NT, and Microsoft are trademarks or registered trademarks of Microsoft Corporation.

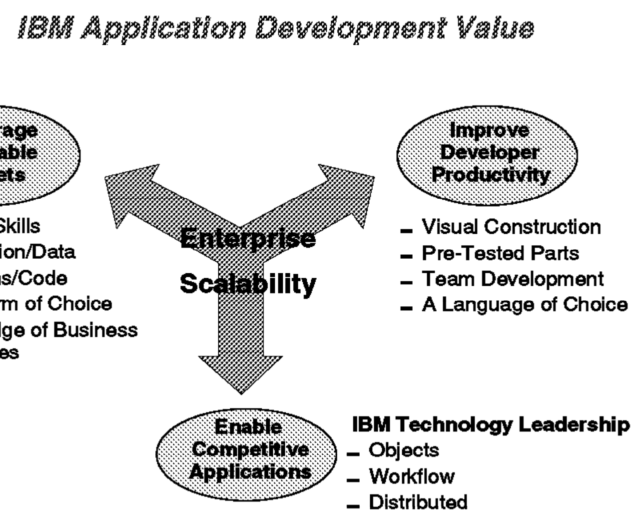


Figure 4.