G326-0395-01

Introduction to the Open Blueprint: A Guide to Distributed Computing

Introduction to the Open Blueprint: A Guide to Distributed Computing

G326-0395-01

Second Edition (June 1995)

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. A form for reader's comments appears at the back of this publication. If the form has been removed, address your comments to:

International Business Machines Corporation
Department 55JA, Mail Station P384
522 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX: (International Access Code)+1+914+432+9405

IBMLink (United States customers only): KGNVMC(D58PUBS)
IBM Mail Exchange: USIB2NZL at IBMMAIL
Internet: d58pubs@vnet.ibm.com

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
208 Harbor Drive
Stamford, CT 06904-2501
USA

Trademarks and Service Marks

The following terms, denoted by an asterisk (*) at their first occurrence in this publication, are trademarks or service marks of the IBM Corporation in the United States and/or other countries:

Advanced Function Printing
Advanced Peer-to-Peer Networking
AnyNet
AFP
AIX
AIX/ESA
AIX/6000
APPN
Common User Access
CICS
CUA
Distributed Relational Database Architecture
DB2
DRDA
HUMAN-CENTERED
IBM
IBMLink
IMS
Intelligent Printer Data Stream
IPDS
MQSeries
OS/2
OS/400
Presentation Manager
SystemView
System/88
Workplace Shell

Open Blueprint is a trademark or registered trademark of the IBM Corporation.

IBM encourages companies to use the Open Blueprint as a model for the development of their information technology architectures. The content and concepts within the Open Blueprint also can be applied to many client/server or distributed systems related information technology activities.

The following terms, denoted by a double asterisk (**) at their first occurrence in this publication, are trademarks or service marks of other companies:

AT&T	American Telephone & Telegraph Company
CompuServe	CompuServe
Digital	Digital Equipment Corporation
DOS	Microsoft Corporation
Encina	Transarc Corporation
Ethernet	Xerox Corporation
Hewlett-Packard	Hewlett-Packard Company
HP	Hewlett-Packard Company
IPX	Novell, Inc.
MCI	MCI Communications Corporation
Motif	Open Software Foundation, Inc.
NetWare	Novell, Inc.
NFS	Sun Microsystems, Inc.
Novell	Novell, Inc.
ObjectStore	Object Design, Inc.
OpenDoc	Component Integration Laboratories
Open Software Foundation	Open Software Foundation, Inc.
OSF	Open Software Foundation, Inc.
Palladium	Massachusetts Institute of Technology
PCL	Hewlett-Packard Company
PostScript	Adobe Systems Incorporated
Taligent	Taligent
Tandem	Tandem
UNIX	UNIX Systems Laboratories, Inc.
Windows	Microsoft Corporation
WordPerfect	WordPerfect Corporation
Xerox	Xerox Corporation
XTI	X/Open Company Limited in the U.K. and other countries
X/Open	X/Open Company Limited in the U.K. and other countries
X/Open Transport Interface	X/Open Company Limited in the U.K. and other countries

About This Book

This book reviews IBM's technical approach to distributed computing in a multivendor, heterogeneous environment. It introduces the Open Blueprint and its components.

The Open Blueprint is a structure for a distributed systems environment and provides the base upon which to build, run, and manage distributed applications.

The Open Blueprint incorporates industry standards and enables IBM to respond to customer requirements with products and solutions that:

- Connect and work seamlessly
- Run on many industry hardware and software platforms
- Integrate with products and solutions from IBM and other industry providers

The Open Blueprint paves the way for a computing environment with a "single system image." The network appears to the user as a single system with all functions integrated and accessible.

Open, distributed computing or client/server computing is the cornerstone of current information technology. Any strategy for this rapidly changing environment must be flexible and dynamic to accommodate technological advances, while at the same time protect significant current investments. So, even though this document reflects a specific point in time, the Open Blueprint and the details behind it will evolve as new technologies take hold. A good example is object-oriented technology that is influencing all aspects of open, distributed computing—building, running, and managing applications. While work on the Open Blueprint began with the procedural model, the Open Blueprint has evolved to encompass object-oriented technologies. The Open Blueprint will also evolve to include emerging technologies such as mobile computing and intelligent agents.

The Open Blueprint serves several audiences:

- It helps *customers* develop their own architecture and organize products and applications in an open, distributed environment.
- It informs *customers*, *software vendors*, *consultants*, *system integrators*, and *service providers* about IBM's directions for products and solutions.
- It guides *developers* as they meet users' needs by supplying products and solutions that include functions written to appropriate standards and that can be integrated and can interoperate with other installed products.

Who Should Use This Book

This book is intended for:

- Customers who are planning technology or architecture investments
- Software vendors who are developing products to interoperate with products developed by IBM
- Consultants and service providers who offer integration services to customers

Contents of This Book

This book is organized as follows:

- **"Introduction"** provides a high-level overview of the Open Blueprint and why it was developed.
- **"Description of the Open Blueprint"** provides more information about the component sets of services, their relationship with applications, their use of standards, and how they integrate with each other. The description of each component set of services in the Open Blueprint is organized as follows:
  - Functions and characteristics of the component
  - Standards that apply to the component
  - Products that either exemplify the function or implement the standards.
- **Summary** provides summary information including reactions to the public introduction of the Open Blueprint.
- **Appendix A** summarizes the standards and protocols included in the Open Blueprint.
- **Appendix B** provides a Bibliography.

The document is *not* intended to contain an all-inclusive list of IBM or other vendor-supplied products relating to the Open Blueprint. In no case is a product used as a model for the functions that are included in a resource manager. A widely known or recently developed product may be used to characterize the function, but that product may not be the only delivery vehicle for that function.

This document is a high-level introduction. For more technical details, see the *Open Blueprint Technical Overview*, GC23-3808.

For a complete look at how you can re-invent, re-engineer, or re-architect your enterprise through the transition from host-centered, mainframe-based computing to user-centered, network-based computing, see *Paradigm Shift* by Don Tapscott and Art Caston. It makes the case for complete planning and for adopting an architecture like the Open Blueprint that is based upon standards. This book, along with other sources of more detailed information, is listed in Appendix B on page 40.

What's New In This Edition

This edition describes the newly-enhanced version of the IBM Open Blueprint. The enhancements include:

- Expanded description of the support and usage of object-oriented function within the Open Blueprint.
- Asynchronous Transfer Mode (ATM) as a subnetworking technology.
- The Signalling and Control Plane to support ATM and telephony functions.
- Additional Application and Workgroup Services resource managers: Event Services, Compound Document, Collaboration, Telephony, and Digital Library.
- Additional Data Access resource managers: Hierarchical Database, Object-Oriented Database, Persistence Service, and Storage Management.

Because object-oriented technology is the basis for so many of the updates to this document, it is assumed that the reader has a **basic** level of understanding of objects. A good source for this level of knowledge is David Taylor's book, *Object-Oriented Technology: A Manager's Guide* (see Appendix B, "Bibliography" on page 40).

Introduction

In search of higher profitability, better service to their customers, and reduced expenses, businesses are reviewing their fundamental operations, with special emphasis on their investments in information technology. This section:

- Examines the information technology evolution over the past several decades
- Describes the demands that the current business environment places on information technology
- Shows how the Open Blueprint meets those demands.

Current Business Environment

“How to do more with less, and do it better” is how Aetna Life & Casualty Company Chairman Ronald E. Compton describes the challenge that today's businesses face. Mr. Compton suggests that reengineering provides a solution.

Business process reengineering means “starting over” according to Michael Hammer and James Champy in their book, *Reengineering the Corporation*. Rather than revisiting existing structures and processes to improve them, reengineering is “the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical contemporary measures of performance, such as cost, quality, service, and speed.” In their book, *Paradigm Shift*, Don Tapscott and Art Caston suggest that creating a new enterprise is a process of “generative learning. Rather than fixing the old, we are on a course to create the new. This involves considerable change in virtually everything we have come to know in our work lives.”

Aetna Life & Casualty Co. provides a good example. The company has completely reengineered how it issues an insurance policy. In one year, Aetna went from 3000 employees working on issuing policies in 22 business centers to 700 employees in 4 business centers. Furthermore, customers now get their policies in 5 days rather than 15.

In the previous environment, the steps required to issue a policy were done serially by different people, with their own applications and data. The reengineered process consists of a single generalist at a workstation connected to a network, taking advantage of new technology to connect him or her to data and applications throughout the company.

United Healthcare Corporation (UHC) is another company reaping benefits of reeingineering and distributed computing. UHC saves more than $1 Million each month in labor costs by reducing the need for human intervention in over 90% of its claims processing transactions. This savings is reflected in UHC's 53% increase in operating income since 1993.

As often happens in our industry, the “hype factor” surrounding reengineering and client/server increased at an exponential rate. Reengineering became the mantra for businesses seeking competitiveness and profitability. For some businesses reengineering was viewed as necessary for survival. And, no matter what the situation needing reengineering, the answer was usually “client/server.”

Then came the backlash: reengineering did not work. Reengineering teams were fired. Reengineering was denounced as a failure. And, to make matters worse, studies proved that client/server computing was actually more expensive than traditional computing!

Reality lies somewhere between the two ends of the spectrum. According to Phillip Stanley, the director of business systems planning at the Bank of Boston (as quoted in the March 20, 1995 issue of

Informationweek), “Reengineering has passed through the discovery and enthusiasm stages to a point of maturity where its difficulties are understood.”

However, each reengineering or client/server bashing item is balanced by a success story, indicating that perhaps a more moderate and realistic set of expectations has developed. A recent Computer Sciences Corporation (CSC) study of top management issues still shows reengineering in the top five concerns. However, it dropped from first place in 1994 to fifth place in 1995. Taking its place is: aligning IT and corporate business goals—making IT the slave and enabler of business functions, rather than the other way around.

To put the alignment of IT and business goals in perspective, it is helpful to look at a brief history of some of the trends that have shaped the current environment.

In the 1970s, the information technology industry was characterized by large centralized computing installations owned and managed by highly skilled and specialized staffs. Users, connected to the system by display terminals, were at the mercy of programs running in the central complex. If a user wanted a new or modified application, it was defined and evaluated by an IT professional, and then often placed at the end of a long list awaiting development.

This process began to change in the 1980s. A series of technological and sociological developments led to today's efforts to redefine corporate and organizational processes to operate in an open, multivendor, heterogeneous world.

The Emergence of Personal Computers (PCs) and Local Area Networks (LANs)

While end users were being frustrated by enormous application backlogs, the personal computer era was beginning. Desktop computing systems appeared in every department. Off-the-shelf, shrink-wrapped software was put into production in hours rather than months or years. PCs were connected to LANs, and workgroups could share information more easily. The hegemony of the IT department was vanishing; it was a “declaration of independence” by end users.

The Client/Server Revolution

The term *client/server* describes a model for distributed computing that centers upon the user. A *client* application, as the representative of the user, makes a request for a service (such as printing a file, accessing data, or sending mail) from a *server* that satisfies the request. Distributed computing broadens the reach of client applications by enabling the enterprise network to be “the computer.” That is, the network appears to the user as a single system, where traditional, host-based applications and client/server applications are both accessible. Client/server computing puts the user at the center of the computing universe. All information appears locally available to the user, regardless of whether it is located nearby or far away. Users want to view their computing universe through the windows on their desktops.

The Growing Importance of UNIX

UNIX**, developed in the early 1970s at Bell Labs and nurtured on college campuses, was dismissed by many as an arcane systems environment totally inappropriate for mission-critical applications. But as more robust implementations of UNIX appeared, and as the masters of the UNIX world graduated from the universities, they brought UNIX with them into the business world. Coupled with RISC-based workstations, UNIX made its mark in scientific computing. UNIX is proving to be a powerful environment for non-scientific computing as well.

The Emphasis on Open Systems

As employees at various levels of the organization began making their own hardware and software choices, “single vendor” shops disappeared. Most installations used hardware and software from a variety of suppliers. Users brought in many flavors of UNIX, and a variety of brands of department LANs and PCs:

- Windows** and OS/2* applications shared desktops
- Novell** NetWare** managed LANs, along with LAN Manager, LAN Server, and Vines
- DEC, Hewlett-Packard**, and IBM logos appeared side by side in midrange and minicomputer implementations
- DEC, Hewlett-Packard, IBM, and Sun all provided their own brand of UNIX and UNIX workstations
- Users accessed data managed by DB2*, Oracle, Sybase, Informix, Ingres, IMS/DB and many other tools
- Mainframes from IBM, Unisys, Amdahl and others processed important data and supplied information to the enterprise

End users needed to communicate between unlike environments, that is, they needed their applications to work together regardless of supplier. Users demanded freedom of choice.

The Arrival of Object-Oriented Technology

Object technology, which has long promised productivity improvements through code reuse, is now becoming a reality. Object technology simplifies desktop and distributed computing by enabling applications to deal with the complexities of the rich variety of available functions, such as 3D graphics, multimedia, and compound documents. Objects allow applications to extend, customize, and transparently distribute functions.

However, even though the concept of an object maps to the way we normally see the world, it represents a new paradigm in IT, a new way of thinking, and a different way of developing applications. The procedural paradigm must coexist with and migrate smoothly to object-oriented technology.

Because objects do mirror real life, object-oriented thinking and analysis can also provide new insights into business process reengineering (as described in Ivar Jacobson's new book, *The Object Advantage: Business Process Reengineering with Object Technology*).

The Need for Collaboration

Functionally-based hierarchies are giving way to cross-functional teams, supporting the need to share data and information across functional lines. Even though users' PCs had been connected into LANs in order to share resources such as files and printers, it became clear to the users that they needed a higher level, more immediate way to collaborate. Teleconferencing gave way to videoconferencing, that is now being supplanted by a combination of the telephone and the PC that allows people to converse and draw on their PCs as if they were in the same conference room drawing on a white board. Users are looking to computer technology to eliminate the limitations of location and distance to increase their productivity.

The Paving of the Information Superhighway

The commercial world has discovered the Internet, the democratic and “self-governing” network of networks originally set up to enable the government and academic institutions to share research. The United States Government launched its National Information Infrastructure (NII) initiative and information has become the new currency, causing sociologists to worry about the “information have-nots.” Those that are “wired” spend many hours surfing the net or wandering the web bringing new meaning (and challenges such as security) to New York Telephone's slogan: We're all connected!

In the centralized era, IT budgets revolved mainly around CPUs and storage devices. PCs and LANs were hidden in departmental budgets. With the recession of the 1980s, companies cut back wherever they could. All budgets were examined and company heads strove to understand the return on their investments in information technology. In many cases, the return was not there.

Taken together, these developments meant new possibilities for IT to serve their businesses. Several companies began to reengineer and rethink their businesses, harnessing the power of new technologies and seeing great productivity gains. These companies were able to do more, in fact, with fewer resources.

The quest for productivity and return on IT investment has created a major challenges for providers of information technology:

How to understand and integrate PC, LAN, and emerging new technologies, and take advantage of the commercial opportunities of the “Information Superhighway” while protecting past investments and maintaining the integrity of company data and other IT assets?

And, how to do this while insuring that IT serves and is aligned with the company's business goals?

Requirements for Information Technology

These challenges lead to the following requirements on suppliers of information technology:

Transparent Global Access

Users need access to data and applications wherever they reside. This access must be transparent so that people are not forced to navigate over physical barriers or boundaries.

Interoperability

Market research shows that most users think *open* means *interoperability*. Products from all suppliers must be able to work together. Openness, achieved by the use of formal and informal standard interfaces, formats, and protocols in products, will facilitate interoperability in a multivendor, heterogeneous environment.

Desktop Usability

Whether a user's “desktop” is on a stationary, personal computer or a mobile hand-held device, he or she should be able to perceive the network as a single system. For example, users need to log on once, provide one password, and have access to all enterprise facilities for which they are authorized. Graphical user interfaces (GUIs) should create an easy-to-use personal computing environment, masking the complexities of the network.

Platform Flexibility

Users should have access to all functions they need without being restricted to a single hardware or software platform. Software should scale up or down to meet user needs and be portable between platforms.

Fast, Efficient Application Development

Application developers need tools to create distributed client/server applications. These tools must hide the complexity of the network environment and provide a highly productive application development environment across the development life cycle, so developers can:

- *Design* applications that allow for the distribution of function
- *Build* applications that can execute on multiple platforms
- *Test and maintain* applications in a distributed environment.

Developers need tools to create reusable component software, and to be able to respond to requirements for rapid prototyping and change. They need object-oriented development tools and a smooth migration path from procedural-based to object-oriented development.

Manageability

System administrators require tools and automation to simplify the complexities of the environment. Manageability must be built into the environment. Further, the flexibility and usability of management tools should continually be improved.

Manageability in the open distributed environment is an important challenge for information technology. The departmental LANs and PCs evolved without the management mechanisms that are gospel in the centralized glass house. Security, relatively easy to handle in a centralized system, presents new challenges in a distributed environment populated by non-secure PCs. Distributing and keeping track of software is another critical requirement.

Investment Protection

Many companies invested heavily in information technology. No company can afford to ignore those investments and start from scratch. Therefore, open, distributed implementations must include current applications and operating environments. Investments in such popular environments as Microsoft Windows, Novell Netware, UNIX, or IBM MVS and OS/2 must be protected, while still enabling the exploitation of new and emerging technologies such as objects and compound documents, ATM, mobile computing and intelligent agents.

All of these requirements must be met while allowing for the constant “re-invention” required to remain competitive in today's dynamic environment.

These customer requirements motivated the development of the Open Blueprint which enables the development, execution, and management of distributed, client/server applications that work together in an integrated fashion in today's heterogeneous, multivendor environment. And, according to Natalie Engler, writing in the February 5, 1995 issue of *Open Computing*:

> ...without an integrated approach to security, network management, resource transparency, and interprocess communications, distributing applications among heterogeneous platforms has proven to be prohibitively complex and expensive.

The Open Blueprint provides such an “integrated approach.”

Open Blueprint Overview

The Open Blueprint is a standards-based architecture that defines the services required by applications in a distributed multivendor environment.

The Open Blueprint can be viewed on three levels: as a “world view,” as a set of technologies, and as a basis for specific products.

An architecture reveals an organization's *Weltanschauung* or perspective, its “world view” of its products and strategies. For example, the Open Blueprint views the world from a transactional perspective: it includes all the business-critical, industrial-strength principles of this world including security, reliability, and availability. A supplier that views the world from the desktop would have a different *Weltanshauung*.

The picture representing an architecture is often drawn as a set of related blocks, each containing a function that is important to that view. The Open Blueprint is made up of blocks, each representing functions that provide the services required by applications in a heterogeneous distributed environment. The arrangement of the blocks is important as are the spaces between them. Sometimes, the blocks appear to form layers. In those cases, more explanation than just the picture is necessary to determine the relationships between the layers. An architecture is rigidly layered if the functions in the top layers are dependent on the lower layered functions *and must use each layer between them and a target function to access the target function.* An architecture is functionally-layered if the functional dependencies are not necessarily vertical, and there is no requirement to drill down through the layers to get at a function in a lower layer. The Open Blueprint is functionally-layered.

Once the perspective is known, and the functions—or the blocks in the diagram—are specified, the second level involves the nature of the technologies for each function or building block in the architecture. The choices are key, because they will indicate the degree of openness and interoperability inherent in the architecture. The technologies included in the Open Blueprint embrace open standards: for example, the Open Software Foundation's Distributed Computing Environment (DCE) and the Object Management Group's Common Object Request Broker Architecture (CORBA) and Common Object Services Specification (COS).

Finally, there are the products that implement and use the functions defined in the architecture. There are many IBM and non-IBM products that deliver Open Blueprint function today, and more will come. There is not necessarily a one-to-one relationship between a block in the Open Blueprint and a product. This is because there are other than architectural considerations for the contents or products. For example, if certain functions are always performed together, it might make sense to package them together in one product.

The Open Blueprint addresses the challenges of the open environment by viewing a system as part of a distributed network and viewing the network as if it were a single system. According to Natalie Engler again in *Open Computing*, "by making a distributed heterogeneous network of systems look and act like a single system, it [common middleware services] promises to raise productivity, cut costs, and enable organizations to reengineer their information systems according to changes in the business."

The Open Blueprint enables a network of operating systems to function as a unit, as a "network operating system." A network operating system is comprised of multiple systems separated from each other and connected by a communication network. Services on each system cooperatively manage resources across the network in the same way as a single operating system manages resources on one platform. Each node in the network can be thought of as structured according to the Open Blueprint. While all services need not be on all platforms, all services need to be accessible by applications on all platforms. The equivalent services on each platform work together to provide seamless, network-wide support for distributed and client/server applications.

Figure 1 on page 7 represents one instance of a platform in the network. Figure 2 on page 8 represents a network in which each individual system can be thought of as structured according to the Open Blueprint. It is the second view that illustrates the true distributed nature of the Open Blueprint.

As shown in Figure 1 on page 7, there are several sets of resource management services in the Open Blueprint:

Network Services

- **Common Transport Semantics** supports protocol-independent communication in distributed networks.
- **Transport Services** provides the protocols for transporting information from one system to another, such as SNA/APPN*, TCP/IP, OSI, NETBIOS, and IPX**.
- **Signalling and Control Plane** provides the ability to establish subnetwork-specific connections.
- **Subnetworking** provides functions dealing with specific transmission facilities, such as various kinds of LANs, WANs, channels, Asynchronous Transfer Mode (ATM) and emerging technologies such as wireless.

Distributed Systems Services

- **Communication Services** provide mechanisms for parts of a distributed application or resource manager to talk to each other.
- **Object Management Services** provide common object services including transparent access to local and remote objects.
- **Distribution Services** assist the communication between parts of distributed applications and resource managers by providing common functions such as a directory and security.

Application Enabling Services

- **Presentation Services** define the interaction between applications and the user.
- **Application/Workgroup Services** are common functions, such as mail, which are available for use by all applications.
- **Data Access Services** allow applications and resource managers to interact with various types of data.

Systems Management provides facilities for a system administrator or automated procedures to manage the network operating system.

Local Operating System Services operate within the confines of a single system in a network. Examples of local services are managing memory and dispatching work.

Development Tools and Open Blueprint services help the application developer implement distributed applications that use standard interfaces.

The Open Blueprint provides guidelines for the integration of multivendor systems and the simplification of the more cumbersome aspects of distributed computing, such as multiple logons, multiple passwords, and unique application directories for locating resources. Products that align with the Open Blueprint provide designated interfaces and protocols. Products that use resource managers defined within the Open Blueprint, rather than their own unique mechanisms, are truly integrated with the Open Blueprint. For example,

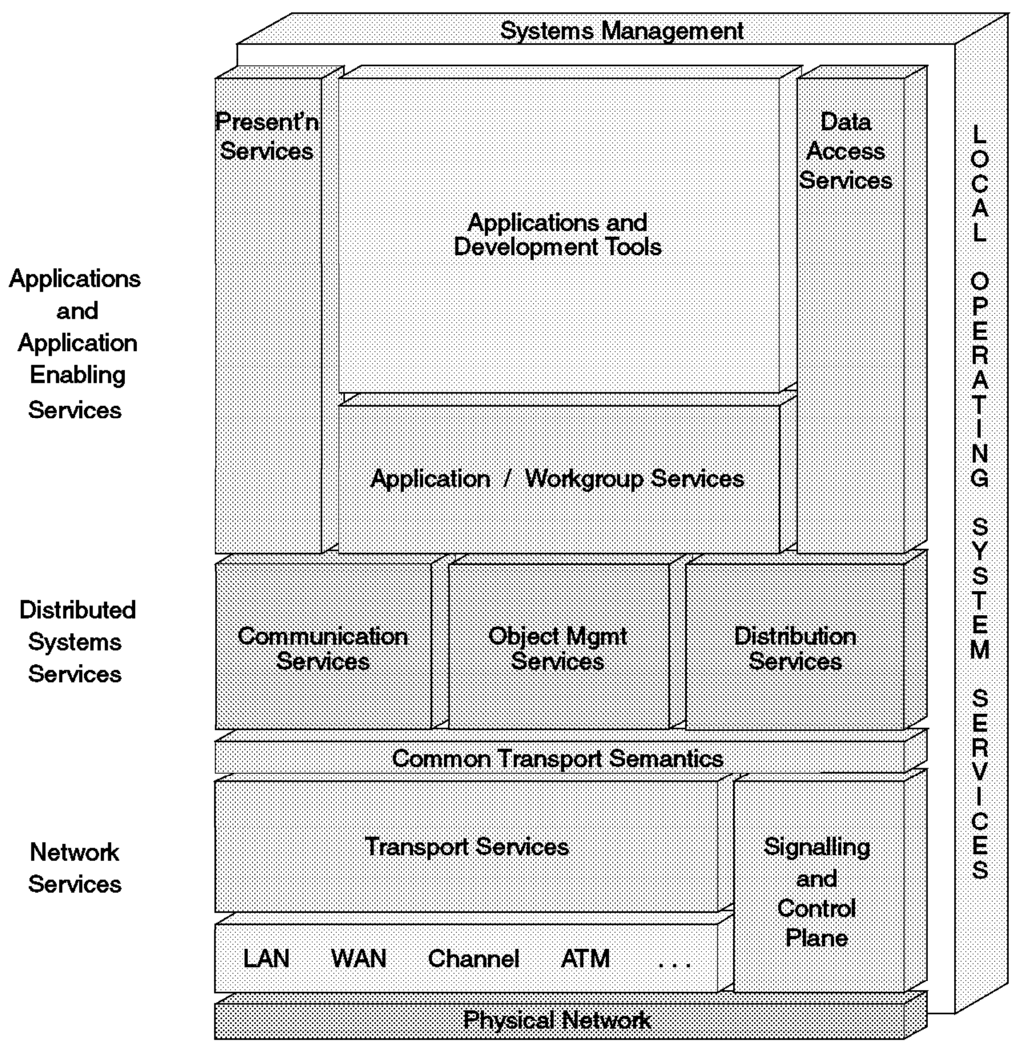

Figure 1. The Open Blueprint

if every application uses its own unique security facility, a user must present a unique password for each application. If applications and resource managers use the security services from the Open Blueprint, a single user password will suffice for the cooperating applications. This integration improves the single-system image of the distributed system as perceived by the end user and application developer.

When considered as an architecture or world view, the Open Blueprint and its component blocks represent abstract functions or services that create a robust distributed environment. At this level, the question of whether it represents a “procedural” or “object” view is not relevant. It is only at the next level, the technologies chosen to implement the functions, that the distinction is visible. The Open Blueprint includes both procedural and object-oriented interfaces and protocols from IBM and other industry sources. Those from industry sources are broadly-accepted standards. Many of these standards are the same ones included in X/Open's Distributed Computing Services (XDCS) Framework (please see the XDCS reference in the Appendix B, “Bibliography” on page 40) and OMG's Common Object Services Specifications (COS).

The blocks in the Open Blueprint diagram (Figure 1 on page 7) do not correspond to specific products. The Open Blueprint is implemented by different products on different system platforms. Also, the Open Blueprint does not describe how the implementing software is packaged into offerings. Rather, the Open Blueprint describes the technical attributes and characteristics of supporting software, reflects desirable functional modularity, provides software principles and guidelines, and specifies important boundaries and interfaces.

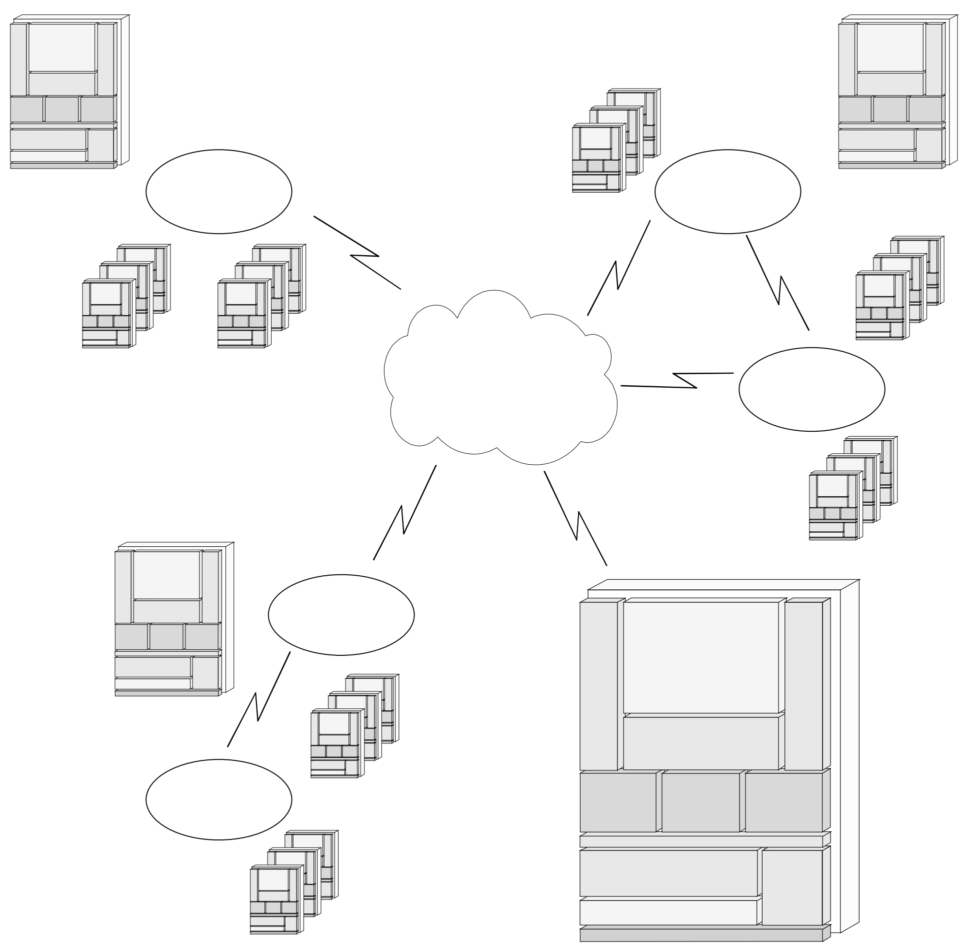

Figure 2. Network Operating System

The Open Blueprint describes techniques for building an open, heterogeneous, distributed system that is extensible by alternate component implementation and by support for evolving new technologies and functions. For example, the Open Blueprint currently supports three models for interprocess communication: Conversational, Remote Procedure Call (RPC), and Messaging and Queuing. If a fourth model were to be developed, it would be evaluated for inclusion. Because the Open Blueprint is modular, it is obvious that the fourth model would fit in next to the other three. However, if a new technology emerged in the area of one of the existing models, the Open Blueprint would not have to be modified. The new technology would potentially have a great impact on implementation and integration, but is handled architecturally within the existing structure.

The Open Blueprint is a structure for distributed computing that will help IBM and others deliver integrated, interoperable products and solutions:

- For *end users*, Open Blueprint integration hides the complexities of the network and makes it appear as a single system.
- For *application developers*, standard interfaces enable a single system view of the network and allow for the development of interoperable applications that can run on many platforms.
- For *system administrators*, the Open Blueprint defines a consistent way to manage the network to hide the complexities from application developers and end users.

Description of the Open Blueprint

The Open Blueprint provides support for the execution, development, and management of distributed applications. The resource manager is the principal structuring element of the Open Blueprint. This section describes resource managers in general, their relationship with applications, their use of standards, the specific resource managers that make up each of the component sets of services, and how they integrate with each other.

Application Focus

Systems and system software are developed and purchased to run *applications*. The Open Blueprint defines an application as the use of information technology to assist in the execution of a business process. The resource management services in the Open Blueprint can be considered part of the application solutions they support. These services are reused by enterprise business processes.

Distributed applications run in or exploit multiple systems to accomplish their functions. A client/server application is a distributed application that is split into a client portion and one or more server portions:

- The client portion drives the application, and typically supports end user interaction.
- The server portion or portions execute the function requested by the client portion and typically reside on different systems.

Applications request multiple system services to accomplish their functions. Applications can take advantage of the Open Blueprint by using its APIs to request services. These APIs simplify application development and maintenance because they mask the complexities of the infrastructure “plumbing” as well as the differences between platforms. Applications are protected from changes to the underlying structure and can use technologies as they are supported by the interfaces, both procedural and object-oriented. Object-oriented development enables productivity gains in application development resulting from large scale reuse and the ability of objects to directly model the business environment.

Some applications are divided into portions called *transactions*. A transaction may be more complex than is apparent to the end user. A transaction may involve many interactions with a variety of resource managers. In addition, while a transaction for one user is using a distributed resource, such as data, transactions for other users may concurrently use the same distributed resource.

Many “industrial-strength” programs provided by users, vendors and IBM that are often termed “applications” are resource managers. Elements of many resource managers, such as the part that provides the direct interface to humans, are sometimes called “application programs.”

Open Blueprint Concepts

Each resource type is managed by a **resource manager,** a set of programs that maintain the state of a resource and provide access to it. The resources that are managed may be distributed, and replicated, across many heterogeneous servers in the network.

A resource manager that operates in a single system is a **local resource manager**. A **distributed resource manager** operates across multiple systems. As shown in Figure 3 on page 11, distributed resource managers include **client** parts, that support the interfaces that requesters use, and **server** parts that perform functions on the resources requested by the client part.

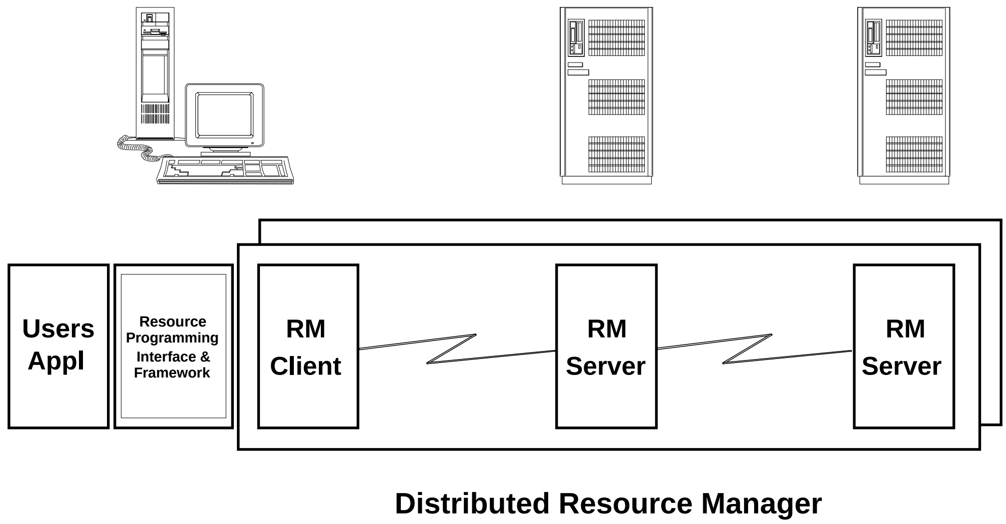

Figure 3. The Distributed Resource Manager Structure

The client part may do some of the processing (for example, validation), and is responsible for determining which server part should handle the request. The client part supports multiple protocols as needed to deal with a heterogeneous environment.

The client and server parts communicate through an agreed protocol, using services provided by the Open Blueprint.

The server part of a resource manager may process a request entirely by itself or it may transparently access another server part.

In the Open Blueprint, resource manager interfaces can be structured as a **framework**. Frameworks support heterogeneity by allowing different implementations of the resource manager. Resource manager implementations can be replaced without changes to the programs that use them.

These frameworks can either be procedural or object-oriented. Object-oriented frameworks consist of a group of object classes. Many object interfaces represented by the frameworks are those specified in the Object Management Group's (OMG) Common Object Services Specification (COS).

Depending on application support requirements, systems may contain primarily the client parts of distributed resource managers. This would be typical for an end user workstation. The frameworks for resource managers the applications use must be on the same platform as the applications. Smaller systems, like mobile computers or personal digital assistants (PDAs), could contain only a few resource manager frameworks and required communication support. The server portions of required distributed resource managers must be accessible somewhere in the distributed network.

Standards

To meet customer requirements for interoperability, portability, and integration, the components of the Open Blueprint must adhere to standards for:

- APIs, that provide a common way to invoke services
- Formats and protocols that allow independently produced software running on the same or different platforms to work together, or interoperate

The interfaces and protocols specified in the Open Blueprint represent an attempt to strike balance between “freedom of choice” and consistency. The choice of standards considers the need for customers and vendors to preserve their current software investments, while allowing them to adopt emerging and future technologies.

In the Open Blueprint:

- Standards are included from industry consortia, for example, X/Open**, Object Management Group (OMG), and Open Software Foundation** (OSF**) as well as from formal standards bodies, for example, ANSI, IEEE, and ISO
- “De-facto” and emerging standards are included when there are popular product implementations that provide investment protection and satisfy functional requirements (for example, IBM's CICS* and Microsoft's MAPI).

IBM will continue to play a leadership role in standards bodies across the world in promoting the adoption of existing standards, and in the development of new standards where none exist. IBM will also continue to encourage new and innovative ways to accelerate the development of standards.

Standards are included in the Open Blueprint because they provide the functionality and interoperability demanded by customers. IBM products will provide and use standard interfaces and protocols as described in the Open Blueprint. IBM products can then be mixed and matched with other vendors' products. In some cases, IBM products will add value by providing function beyond what is addressed by a standard. Further, as new technologies emerge and customers' needs evolve, some standards may be superseded. See Appendix A, “Standards” on page 33 for a list of standards for interfaces and protocols for each resource manager.

Open Blueprint Components

This section describes the resource managers in each of the component services of the Open Blueprint, as shown in Figure 1 on page 7:

- Network Services
- Distributed Systems Services
- Applications and Application Enabling Services
- Systems Management Services
- Local Operating System Services

Network Services

Communications and networking are at the heart of the infrastructure for a distributed system. In the more homogeneous world of the 1970's and early 1980's, the communication requirements drove the structure of applications and their enabling services. This typically resulted in tying the applications and subsystems to a particular communication model and type of transport network.

In the world of heterogeneous, distributed computing, applications must run on a variety of operating system platforms and networking environments. These requirements led to the separation of Communication Services (see “Communication Services” on page 15) from Network Services. This separation was reflected in IBM's Networking Blueprint that was introduced in 1992 as a roadmap for networking in open, distributed systems. It included the Common Transport Semantics which made the applications and resource managers independent from the network. The Networking Blueprint content has been merged into the Open Blueprint.

As shown in Figure 4 on page 13, Network Services consists of Common Transport Semantics, Transport Services, Subnetworking, and the Signalling and Control Plane.

Common Transport Semantics: Common Transport Semantics (CTS) insulates applications and the higher level services of the Open Blueprint from the the underlying network transport by providing a common view of transport protocols. This common view, coupled with compensations for the differences in protocols, enables applications and higher-level services to be transport-independent. Using Common Transport Semantics also enables the integration of networks with different protocols through transport gateways. In the Open Blueprint, Common Transport Semantics is provided through the Multiprotocol Transport Networking (MPTN) architecture.

Common Transport Semantics also provides access to the functions of the Signalling and Control Plane.

The X/Open Transport Interface** (XTI**) provides access to SNA/APPC, NetBIOS, OSI, and TCP/IP, but does not shield applications or services from network differences. IBM has developed the AnyNet* products (based on MPTN architecture) that shield requesters from network differences. X/Open has published preliminary specifications for MPTN based upon IBM's architecture.

With the AnyNet/MVS and AnyNet/2 products, businesses can run APPC applications over a TCP/IP network and TCP/IP Sockets applications over an SNA network. These functions give users flexibility in their network decisions.

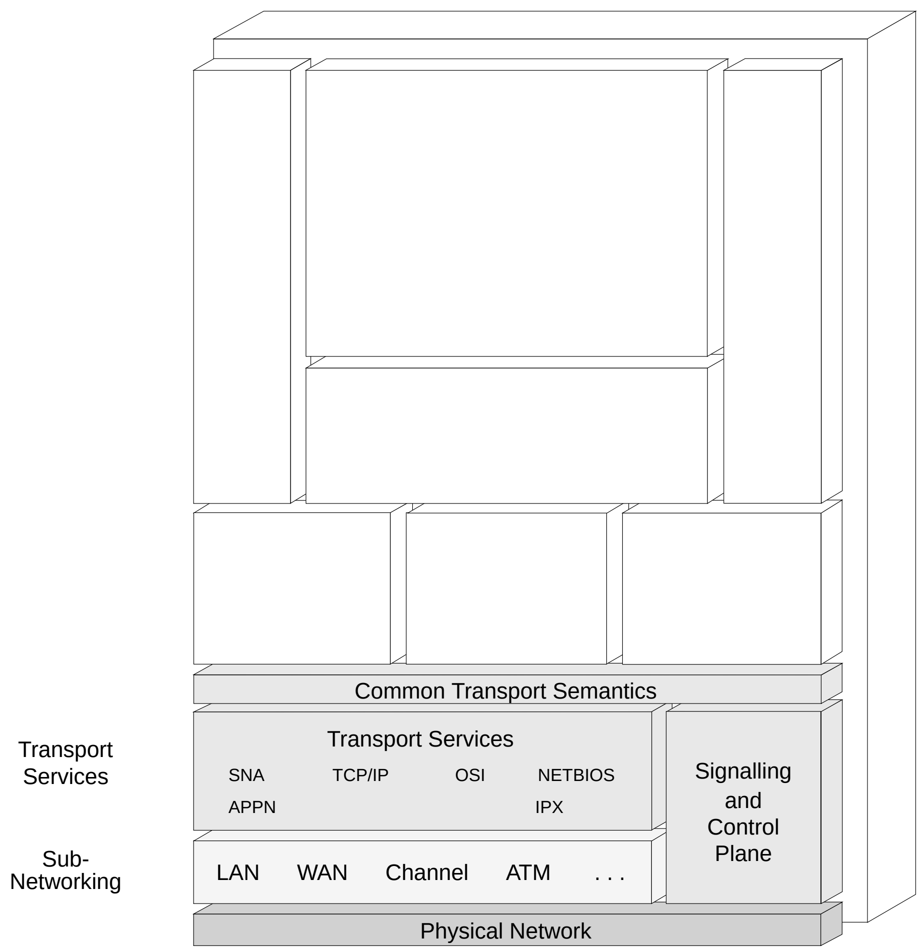

Figure 4. Network Services

Transport Network Services The Open Blueprint supports a variety of network protocols for transporting information over both wide area and local area networks. These include:

- Systems Network Architecture/Advanced Peer-to-Peer Networking* (SNA/APPN)
- Transmission Control Protocol/Internet Protocol (TCP/IP)
- Open Systems Interconnection (OSI)
- NetBIOS
- Internet Packet Exchange (IPX)

Each protocol supports interfaces used to access its services. Also included are various end-to-end network monitoring functions that safeguard data integrity and help to avoid congestion.

Signalling and Control Plane: In a communication network, **signalling** refers to the collection of procedures used to dynamically establish, maintain, and terminate connections. To do this, information is exchanged among the network users and the switching nodes. For each function performed, the corresponding signalling procedures define the sequence and format of the messages exchanged which are specific to the subnetwork (ATM, narrowband ISDN, Public Switched Telephone) across which the exchange takes place.

The Open Blueprint Signalling and Control Plane is based on the International Telecommunication Union - Telecommunciation (ITU-T) Integrated Services Digital Network (ISDN) Control plane. But it has been generalized to include the switch connection support for other network types and to include the control for the low level multiplexing of video, audio, and data needed by the ITU-T H.Series conferencing standards.

The Signalling and Control Plane is used by Open Blueprint Transport Services to enable operation over ATM subnetworking. It is also used by applications and higher level services that require the ability to directly establish subnetwork-specific connections.

Subnetworking: Subnetworking provides a structure to let networks evolve to accommodate and use new high-speed, highly-reliable transmission technologies without sacrificing business application and network investments.

Subnetworking includes four major types of network connectivity:

- Local Area Network: for example, Ethernet**, Token Ring, and FDDI
- Wide Area Network: for example, HDLC, SDLC, X.25, and ISDN
- Channel
- ATM

Each type offers a unique set of configurability, connectivity, and performance options at varying cost levels.

Asynchronous Transfer Mode (ATM) is a specific subnetworking technology that is key to addressing today's and tomorrow's networking challenges:

- Its characteristics allow integration of voice, data, and video
- It is a single common technology for local area and wide area networks
- It supports a wide range of speeds and does not require the same speed for each connection.

ATM is the transfer mode of choice for B-ISDN by the ITU-T, developed jointly with various national standards organizations such as the American National Standards Institute (ANSI) and the European Telecommunications Institute (ETSI).

ATM requires that the functions of the Signalling and Control Plane establish end-to-end connections before any messages are sent.

This is a rapidly changing field, in which major extensions to existing technologies are frequent. There are a number of emerging subnetworking technologies, such as wireless communication facilities, (which are key to mobile computing), and the very high-speed Synchronous Optical Network (SONET) technologies.

Distributed Systems Services

Distributed System Services consist of Communication Services, Object Management Services, and Distribution Services (see Figure 5).

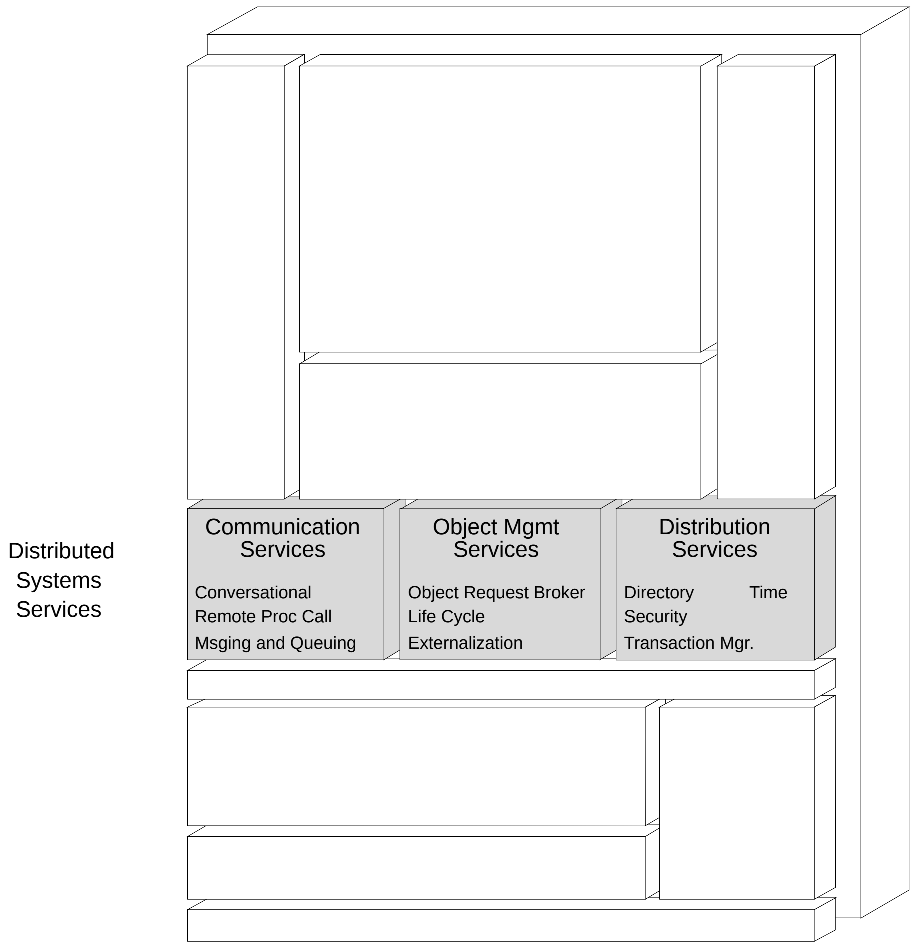

Figure 5. Distributed Systems Services

Communication Services: The Open Blueprint supports three models that describe how parts of distributed applications or services cooperate or communicate with one another.

Conversational: In this model, both parts are in “conversation” with one another and are synchronized, as are two speakers in a telephone conversation. They can each send messages back and forth.

The conversational model is based on program-to-program communication defined by SNA APPC. X/Open's Common Programming Interface for Communications (CPI-C) provides a common interface for the implementation of the conversational model on all major platforms.

Remote Procedure Call (RPC): In this model, one part requests a service from the other part and awaits a reply.

RPC is a widely accepted mechanism across many systems. Most programmers recognize it as a subroutine call. OSF has developed an RPC specification as part of its Distributed Computing Environment (DCE). IBM has implemented the DCE RPC on AIX/6000*, OS/2, MVS, and OS/400.

Messaging and Queuing: In this model, communication between processes is performed by the sender putting messages for the receiver onto a queue somewhere in the network. The communicating processes in the network do not have to be active simultaneously. Messaging and Queuing support provides time-independent communication, with the sender and receiver executing at their own pace.

Messaging and Queuing provides assured, once-only delivery of the message to the queue and can optionally start the target process or transaction to handle the message. The assured delivery allows work to be committed by the sender, so that the message will be delivered despite system or network failures during the delivery process.

The Message Queue Interface (MQI) is implemented on MVS and OS/400* in the Message Queue Manager products, which are part of the MQSeries*. MQI is also supported in the MQSeries of products on the following platforms: OS/2, DOS**, Windows, VSE, DEC's VAX, AIX/6000, Tandem's** Guardian, and System/88*. MQI will be supported on HP-UX, SunOS, UNIX, and UNIXWare.

Messaging and Queuing is particularly suited to:

- High volume, networked transaction applications
- Intermittently connected networks
- Heterogeneous networks with multiple protocols or machine architectures
- Networks spanning time zones in which network applications are available at different times.

Some types of applications are more suited to one model than another. An application can use more than one model. Each model uses the directory and security services to find the part (or service) requested and to verify that the access is authorized. The service can then be accessed across the appropriate network transport.

Object Management Services: Object technology is based on the principle that discrete, self-contained components, called **objects**, designed for reuse and developed to adhere to very specific rules of behavior, can greatly increase developer productivity. At the same time, these objects can absorb much of the complexity of the system and network by associating information with the programming required to access or manipulate that information. Software created with object components is more flexible and easier to change. In a distributed context, object components are useful for both portability and transparent distribution across platforms.

As shown in Figure 5 on page 15, Object Management Services include the Object Request Broker and the Life Cycle and Externalization basic object services.

Object Request Broker: The Open Blueprint Object Request Broker incorporates IBM's Systems Object Model (SOM) and related distributed SOM technologies that address the inherent interoperability problems of objects. Since each object-oriented language has a unique set of object characteristics or a unique object model, it has not been possible to write parts of object applications in different languages. The objects contained in a binary module compiled by one compiler cannot be used with objects created by another compiler.

SOM describes the way in which objects relate to each other providing a common ground for objects developed with many languages and systems.

It provides a rich, language-neutral object model that supports binary interfaces defined at the system level and independent of a particular compiler implementation. The SOM object model maps easily into the popular object-oriented programming languages and the 3GL base used in existing systems.

Distributed object support associated with SOM provides a complete implementation of the Object Management Group's industry-standard Common Object Request Broker Architecture (CORBA). Because of the seamless integration, CORBA's standards apply equally to both local and distributed objects. The Open Blueprint Object Manager provides CORBA-compliant support for the distribution of objects. The OpenDoc** technology, described in “Presentation Services” on page 19, uses SOM as its method for identifying and locating objects across language and location boundaries.

A key advantage of SOM and distributed SOM is location transparency; applications need not be aware of whether an object is local or remote. Access is provided through the same mechanism in either case.

A variety of system and language vendors are working with IBM to provide SOM implementations across a range of operating system platforms.

Basic Object Services - Life Cycle and Externalization: A set of basic object services has been defined by OMG. These services are essential in the SOM environment for application programmers.

Life Cycle supports the object creation, deletion, move, and copy functions:

- Object creation defines how objects come into being
- Object deletion causes an object to cease to exist
- Object copy causes a new instance of the same object to be created
- Object move changes the location of an object.

Externalization provides functions for the transformation of an object into a form suitable for storage or transmission and transformation back into an object, providing a means through which objects can be packaged and transported.

Distribution Services: Distribution services enable a single-system view of the network. See Figure 5 on page 15.

- The naming and directory service provides a consistent approach to naming and keeping track of network resources and their attributes.
- The security service protects network resources from unauthorized use by registering users (both system and human) and their authorization levels, by authenticating users, and by auditing access.
- The time service regulates the date and time across a network.
- The transaction manager coordinates resource recovery across the various systems in the network.

X/Open and IBM address the transaction manager. X/Open technology and the Distributed Computing Environment (DCE) specifications from the Open Software Foundation (OSF) address the other three services.

DCE functions are supported on the IBM AIX* Distributed Computing Environment Product Family, OS/2, MVS, OS/400, and other platforms. DCE is also supported by DEC, HP**, Novell's UNIX Systems Group, Sun, Transarc, and many others.

Naming and Directory: Naming provides the facilities required to refer to such network resources as servers, files, disks, applications, and disk queues. The use of a consistent naming model allows a resource to be accessed by name, even if a characteristic such as its location is changed. The directory service maintains information about the characteristics of a resource such as its name, network address, and creation date.

To simplify the tasks of using, administering, and writing applications in an open, heterogeneous, distributed environment, the Open Blueprint incorporates X/Open Federated Naming concepts and the directory structure specified in OSF's DCE technology. With federated naming, resources such as programs, hardware, data, and users in any location can be referred to by a name that follows a standard set of naming rules. The DCE directory model strikes a balance between the need for quick access to local resources, and the need for transparent access to remote resources in the network. The Directory Service consists of:

- The Cell Directory Service (CDS) from OSF/DCE is used for the local directory which keeps information about resources in the local *cell*. A cell is a set of machines that work together and are administered as a unit—a workgroup, for example.
- The Global Directory Service(GDS) keeps information about cells so that resources can be accessed across cells. X.500 (a standard developed jointly by ITU-T and ISO) or the Internet's Domain Name Service provides the protocol used in the global directory.

Security: In earlier centralized systems, the operating system authenticated the users' identities and authorized access to resources. Individual workstations in a network are not necessarily secure. Therefore, in a distributed environment, security operations must be performed by an independent set of services. Security in a distributed environment must support single signon and address such challenges as preventing eavesdropping, impersonation, and forgery.

The Open Blueprint security service specification lets administrators register users and resource managers, provides for the mutual authentication of clients and servers, and enables resource managers to provide access to resources only to authorized users. The Open Blueprint security service also defines services for auditing user activity. These services include DCE specifications and incorporate and expand on the Kerberos specification from MIT.

These security services meet relevant X/Open and POSIX specifications.

Time: The systems and applications that operate across a distributed environment need a consistent time reference to schedule activities (such as recovery) and determine the sequence and duration of events. Different components of a distributed application obtain time from clocks on different computers. The distributed time service, based upon DCE, synchronizes system clocks in a network to provide time services with a limited, but known degree of accuracy for distributed applications.

Transaction Manager: In transaction processing, application processing is divided into units of work called *transactions*. A transaction may involve only a limited number of interactions with a user at a workstation, but it may involve many interactions with many resource managers across the network. The Transaction Manager provides synchronization services so that multiple resource managers can act together to ensure that resources retain their integrity. The resources managed by each of them separately remain consistent according to relationships imposed externally, typically by the application.

The current use of the term Transaction Manager differs from earlier usage. This new terminology has been adopted to accurately reflect technical goals and the functional parts of the Open Blueprint and to correlate with standard industry terminology. Major IBM products such as Customer Information Control System (CICS), Encina**, and Information Management System (IMS*) are combinations of the Transaction Manager, the Transaction Monitor, and other functions.

A distinguishing feature of transaction processing is that all the resource changes associated with a transaction must be committed before the transaction is complete. If there is a failure during execution of the transaction, all of the resource changes must be removed. Resources managed in this manner are *recoverable*.

A typical example is a 2-part financial application that credits one account and debits another. If a failure occurs after the credit, but before the debit, the application would want to back out the credit. The Transaction Manager knows the entire transaction (or unit of work) involves the two activities and ensures that either both activities complete or that the accounts remain unaltered. If the two accounts are located in different systems, the Transaction Managers in each system cooperate to eliminate any effects of the failed transaction.

The Open Blueprint supports three procedural interfaces for transaction management:

TX Interface Used between the application program or its transaction monitor and the transaction manager to define the beginning and end of a unit of work.

XA Interface Used between the Transaction Manager and the resource managers to synchronize resource changes.

XA+ Interface Used between the Transaction Manager and the communication resource manager to inform a local transaction manager of the status of a distributed unit of work.

In addition, the Open Blueprint Transaction Manager supports the OMG COS Object Transaction Service interfaces.

The X/Open standard definition of the procedural interfaces is the basis of future work in transaction management. Current transaction management products will continue to support their existing interfaces, which are functionally equivalent to the X/Open interfaces. The functions of these current interfaces have served as a technology base for the X/Open transaction manager support.

Application Enabling Services

As shown in Figure 6 on page 20, Application Enabling Services consist of Presentation Services, Application/Workgroup Services and Data Access Services.

Presentation Services: Presentation Services provide services that support an interactive end user.

User Interface: The Open Blueprint User Interface resource manager supports the presentation of application and system information to end users. The overall objective is to do this in an intuitive fashion, allowing the end user to interact with the computer in a very natural, consistent manner.

The term **user interface** describes the front-of-screen appearance and function of an application or system. By itself it does not imply any uniformity between applications. Standards, such as Common User Access* (CUA*) on OS/2 and Motif** on AIX, have been adopted to ensure applications have a common look and feel.

The term **desktop** conveys a complete metaphor for user interaction: the user's space is a desktop and his/her tools and data are things on the desktop or are readily accessible from it. The desktop metaphor has been used in many operating systems throughout the last decade, such as Macintosh's Finder, Windows' Program Manager, OS/2's Workplace Shell*, and AIX's CDE. Users manipulate the desktop objects with “point-and-click” and “drag-and-drop” actions.

An important aspect of the user interface is the ability to integrate applications on the desktop. This includes, for example, enabling an action or a change of data in one application to be reflected in another application. Compound document technology is one way to accomplish this integration.

From a user's perspective, a compound document is a document with parts containing various types of media such as text, tables, movies, sound, and graphics. As the user “reads” the document, the underlying software changes because different software is required to handle different media types. From a systems perspective, a compound document is the manifestation of this special type of application software collaboration that appears seamless to the end user.

The key compound document technology in the Open Blueprint is OpenDoc. OpenDoc is an offering derived from technologies available through the Component Integration Laboratories (CIL). CIL is a consortium whose founding members include Apple, Borland, IBM, Novell, Oracle, Taligent**, WordPerfect**, and Xerox**. OpenDoc parts are key application building blocks.

Emerging HUMAN-CENTERED* technologies such as speech and handwriting recognition are enhancing the traditional desktop, causing its evolution into something even more powerful and intuitive. In addition, the “messy desk” metaphor is being superseded by specialized user environments, or multiple metaphors, targeted for specific task environments: the office, the factory, the home, the retail store, and “the road,” or travelling. These “places” contain people, things, and even other places necessary to accomplish a specific task or set of tasks. Places support sharing and collaboration between users, and multiple users can be “in” a place at the same time. Actually, places are compound document (OpenDoc) parts and contain things that are parts.

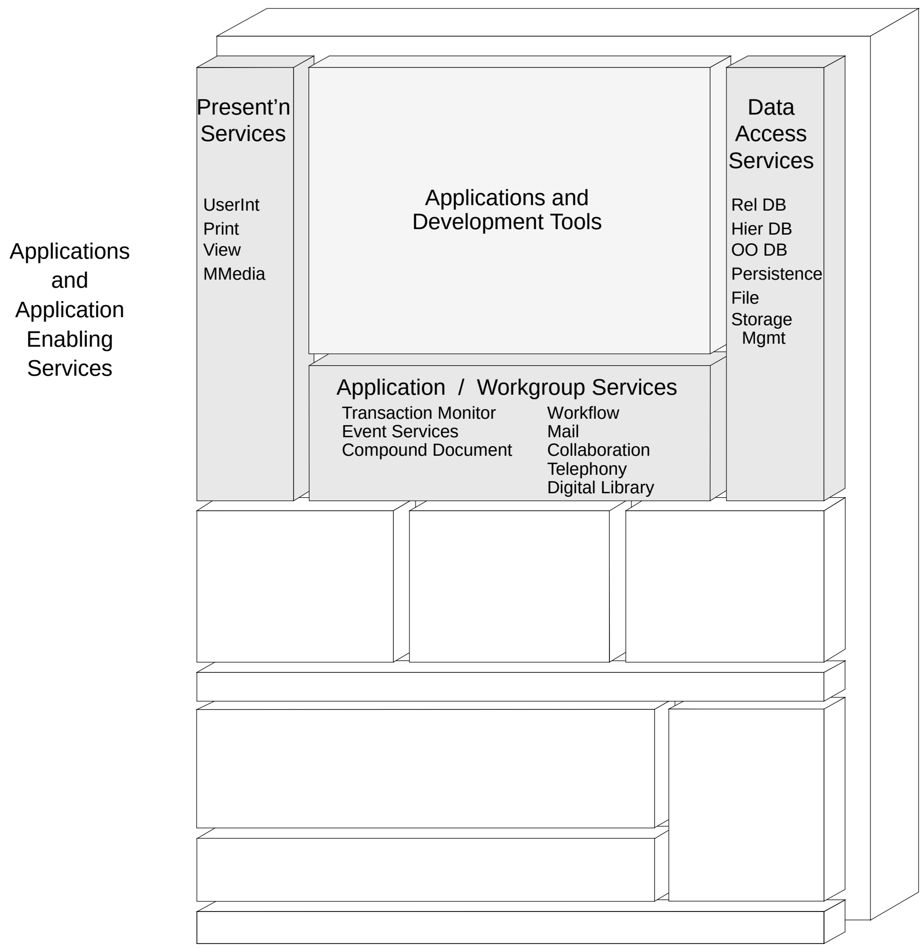

Figure 6. Application Enabling Services

These compound documents and parts provide a common, consistent way to create a rich, consistent user interface across platforms that today each have their own presentation API—for example, OS/2 has the Presentation Manager*, and Windows has the Windows API. Those platform-specific APIs will provide the underlying user interface support for the various compound document elements.

Print: The Open Blueprint Print resource manager facilitates print submission, print resource management, and operational tasks in a heterogeneous network environment.

Based on version 2 of the Palladium print management technology, and developed jointly at MIT with IBM, Digital, and Hewlett-Packard, the Open Blueprint Print resource manager conforms to the International Organization for Standardization (ISO) Document Printing Application (DPA) standard 10175, and tracks the emerging IEEE POSIX 1387.4 standard (formerly the POSIX 1003.7.1 standard), and the X/Open Printing System Interoperability Specification (PSIS) Extensions to ISO DPA 10175 and POSIX 1387.4.

The Print resource manager supports industry-standard print data streams such as PostScript, HPPCL, and AFP.

The Print resource manager supports an extensive set of end-user functions to submit and control print jobs. Through the client, an end-user or application can locate and query printers based on attribute values, view queues, track the progress of print jobs, and receive notification when jobs have completed or failed. Default job attributes can be specified, supporting the ability to produce consistent printed output, thus eliminating the need to repetitively list certain attributes at job submission.

The Print resource manager provides extensive centralized management and operational control over distributed print resources. Using Print services, an administrative application can manage print system resources (queues, jobs, printers, etc.), modify job priorities, monitor the print network, and add, delete or reallocate print resources.

View: The View resource manager provides the ability on the workstation to view printable documents based on six industry standard formats. A document can be viewed before, in lieu of, or after printing on paper, microfiche, or other media.

Multimedia: Multimedia support allows computers to display, manipulate, control, and retrieve distributed graphics, audio, video, and animation data. Because multimedia data is digitized, it can also be searched and manipulated. Multimedia combines the interactivity of a computer with human modes of communication that make the computer easy to use. In the Open Blueprint, multimedia facilities provide an environment in which a heterogeneous set of multimedia-supporting platforms and special-purpose equipment cooperate to support distributed interactive multimedia applications that process synchronized, time-based media. Time-based media must meet stringent timing constraints when played or recorded: for example, 24 frames per second for movies and 8000 samples per second for voice. In a distributed system, all services in the path that gets the specific data, such as voice, from its storage to the end user (in the case of voice, through a speaker) must cooperate together to agree to meet the timing or performance constraints. Otherwise, the data will be distorted, as in a movie that is playing too slowly. This agreement is referred to as assured "quality of service" which is discussed in more detail in "Collaboration" on page 23.

Multimedia tools and applications can include capture and creation, composition and editing, and the playing of live and stored multimedia data. A key multimedia application is audio-video conferencing in support of workgroup collaboration.

Application/Workgroup Services: Application services provide high-level application- or workgroup-oriented functions. See Figure 6 on page 20.

Transaction Monitor: *Transaction Monitor* is an industry term for functions that traditionally have been included in IBM's transaction processing systems.

The Transaction Monitor provides an environment for the development and execution of applications, the transaction programs. The monitor typically provides an application programming interface and support for efficient transaction execution. The Transaction Monitor supports a large number of users concurrently sharing access to the transaction programs and the resources they use. Transaction Monitors preallocate system and application resources such as address spaces, data, and other facilities. The preallocation allows transaction programs to be scheduled efficiently.

The Transaction Monitor uses the transaction manager directly, in many cases, to simplify the application programming implementation of the transaction. A transaction monitor also typically provides some application development and system management support for the transaction programs.

There are presently no formal standards for the transaction monitor application programming interfaces. The CICS transaction monitor API has been implemented on all major IBM platforms and on many non-IBM platforms. The IMS transaction monitor API has been implemented on a variety of platforms supporting applications associated with MVS systems. The Encina transaction monitor API has been implemented across a range of UNIX platforms.

Event Services: Event Services allow an application send information representing the occurrence of an event to another program. This service separates the communication between the program reporting the event, and the program reacting to the event (consumer). These programs may be on the same or different node in a network.

Event Services is an implementation of OMG's Event Services Specification.

Compound Document: The concept of a compound document was discussed in "User Interface" on page 19, which describes the impact of compound documents on the user interaction. Compound document component technology, though first applied to mixing data types and their editors within a single word processing application, is really generic in its ability to allow components to be integrated for data sharing and interchange, as well as user interaction and presentation. Components can update files, monitor physical equipment, perform computations, and so on. With the generalization of capabilities, what were once data types have evolved to be more generalized "parts" and what were once editors, have evolved to be generalized "part handlers."

Compound Document implementation is consistent with CIL's OpenDoc specification and Taligent's Common Point initiative as appropriate

Workflow: Workflow management provides a set of functions that help customers define, execute, manage, and re-engineer their business processes across a heterogeneous system environment. Business processes can be quite diverse, such as contracting for a purchase or processing a mortgage loan application. They can also include support activities, such as application development or installing software updates. The workflow manager is a driver of complex applications. It is a coordinating agent initiating the execution of work by people, and the execution of programs in multiple, distributed workflow-managed processes.

In many companies, valuable process definitions can be buried deep within the logic of application programs. Changing a process means changing application programs, which can be time consuming and expensive. Workflow management makes application programs serve processes, rather than the other way around. Workflow management helps organizations design processes, modify them more easily, and execute them more efficiently. An organization's processes are an important corporate asset, and companies in control of their processes have a definite competitive advantage in the 1990s.

IBM is a member of the Workflow Management Coalition whose goal is to foster interoperability among workflow products. The coalition is addressing application programming interfaces for access to workflow management services as well as formats and protocols for communication between workflow services.

Mail: Electronic mail is viewed not only as a mechanism for users to exchange text messages, but also as a high-level service that other applications can use to send all types of materials to end users or other programs. Applications in the distributed system can use electronic mail to transmit messages in many different forms. In addition to the interchange of unformatted text messages, the Open Blueprint electronic mail service supports transmission of:

- Formatted documents
- Unstructured data and files
- Programs
- Graphics
- Images
- Video, audio, and other multimedia data

Mail-enabled applications will use electronic mail service for many purposes. Some examples include:

- Electronic bulletin boards, conferences, and forums
- Inter-organization business communication and transactions, including EDI
- Group scheduling and calendar systems
- Electronic publishing and distribution
- Electronic forms routing and workflow applications

Presently, there are several open interface standard APIs that have become popular for application access to electronic mail services and are included in the Open Blueprint—Vendor Independent Messaging (VIM), Messaging API (MAPI), and Common Messaging Call (CMC):

- VIM: A common client API developed by a consortium of companies, including IBM. It is intended to be a multi-platform standard that can be implemented by a variety of applications running in different environments.
- MAPI: An additional client API for Windows. It was developed by Microsoft and has become a de facto standard due to its pervasive use in the industry.
- CMC: An additional API developed by the X.400 API Assocation (XAPIA). It provides access to additional services from applications supporting existing mail-enabled interfaces such as VIM and MAPI.

The pervasive and mission-critical nature of e-mail is driving increasing standardization of the mail APIs. Because CMC represents the most “open” approach, and protects investments made in existing applications, both the Electronic Messaging Association (EMA) and XAPIA have endorsed it. Most major e-mail vendors, including IBM, Microsoft, Lotus, and Novell, have announced support for CMC.

In addition to the interfaces, there is a set of electronic mail message transfer protocols that is widely used in today's networks. The Internet TCP/IP supports a SENDMAIL service and Simple Mail Transfer Protocol (SMTP). OSI systems support the X.400 messaging protocol. Value-added network providers like AT&T**, MCI**, and CompuServe** support their own protocols and supply gateways to support X.400 and SMTP. IBM SNA products support a store-and-forward distribution facility called SNA Distribution Services (SNADS). Novell NetWare provides a Message Handling Service (MHS) that is used widely in DOS systems. The Open Blueprint Mail facilities will use the Messaging and Queuing resource manager for server-to-server message transfer and will natively support standard X.400 and SNADS protocols. Access to other important proprietary message transfer standards is through gateways.

Collaboration: The Collaboration function supports applications that allow people who are physically distributed over a computer network to work together in real time. Examples of collaborative applications are desktop conferencing and distance learning. Typically, the applications themselves are also distributed over the network and can permit, for example, remote presentations and shared document editing among groups of people.

Collaboration functions include:

- The establishment of an environment, analogous to the call concept familiar in telephony, which sets the rules, or policies, for the interactions between the parties
- The support of dynamically changing configurations and networks, reflecting the arrivals and departures of participants and possible communication links.

In some cases, standards have been (or are being) developed to ensure interoperability over particular network types; such standards include the ITU-H.320 recommendations for data collaboration, defined with an emphasis on public switched data communication networks such as ISDN. Open Blueprint Collaboration enables the development of standards-compliant applications that are able to interoperate with other compliant implementations.

Collaboration also plays a role in supporting distributed multimedia—audio-video conferencing or distributed multimedia playback—by establishing logical communication pipes (or channels). These channels are connected to logical devices that support real devices such as cameras and speakers, as well as file systems containing multimedia objects. This allows data flow between devices without requiring application involvement.

Support for Quality of Service is a key function of the Collaboration resource manager. Quality of Service is the characteristic of a system that determines how successfully it can carry different kinds of information, and therefore its suitability for particular applications. As described in "Multimedia" on page 21, Quality of Service is particularly relevant to the support of multimedia data.

Telephony: The Telephony resource manager enables the merger of telephone and computer technologies, and provides the user with integrated voice and data services. This merger is commonly referred to as Computer Telephony Integration (CTI)—an essential ingredient in today's environment.

The two main functions of the Telephony resource manager are call control and voice processing.

Call Control encompasses basic telephone applications that control the simple telephone (that is, the standard 12 buttons), as well as more sophisticated call center applications that work closely with the telephone switch, providing multiple automated functions for call center personnel.

Voice processing works with the digitized voice itself. It allows the telephone to be an input device to access information. Voice processing can answer a call with a pre-recorded message and/or using tone recognition, can begin an audio dialog with the caller, allowing the caller to request a service such as reviewing an account balance. It can provide voice mail functions, place outgoing calls, and so on. Through the use of voice related technologies such as speech recognition and synthesis, it can transform voice to text and text to voice.

Digital Library: The Digital Library resource manager supports the storage and distribution of information objects. It provides the support to reproduce, emulate, and extend library services (such as information collection, organization, searching, analysis, synthesis and dissemination) to the information technology environment. It manages machine-readable information objects in libraries that scale from small digital libraries to distributed libraries with petabytes[1] of data and millions of objects, both personal and enterprise confidential, along with access to commercial copyrighted materials and public free-access libraries. Information managed includes text documents, and audio, video, graphics, and image data objects.

Data Access Services: The Open Blueprint includes access to both files and databases. The Data Access Services include:

- Relational Database
- Hierarchical Database
- Object-Oriented Database
- Persistence
- File
- Storage Management

IBM's Information Warehouse* solution provides extended capabilities for building data warehouses and enhancing the decision making process. SQL and DRDA are used to enable flexible and integrated access to distributed data in a heterogeneous environment. See "Other IBM Publications" on page 40 for sources of further information about the Information Warehouse solution.

Relational Database: Relational database facilities are made available through the Structured Query Language (SQL) by the Relational Database (RDB) resource manager. Originally developed by IBM, SQL has been accepted by ISO and X/Open as the standardized language for defining and manipulating data

[1] A petabyte is 10^{15}, or 1000000000000000.

in a relational database. The ISO standard for embedded SQL is SQL92 and the level supported by the Open Blueprint for interoperability is SQL92E. The functions of SQL are available through the DB2 family of products on all major IBM platforms except AIX/ESA*, and the announced direction is to make them available for that platform also.

In addition to the embedded form of SQL defined by ISO SQL, the Open Blueprint Relational Database resource manager supports the X/Open SQL Call Level Interface (CLI) and the ODBC interface defined by Microsoft.

To support distributed databases, IBM has published and implemented the Distributed Relational Database Architecture* (DRDA*), which defines the connectivity for client applications to access data stored at remote relational database management systems. DRDA uses the conversational model for communication between the client and server.

DRDA has been endorsed by all the key database and connectivity vendors, several of whom have DRDA implementations available today on IBM and non-IBM platforms.

Remote Database Access (RDA) has progressed to an international standard though ISO. A slightly different version is under development in X/Open. IBM continues to be heavily involved in these design activities and is monitoring the customer demand for RDA-based interoperation. Functionally, DRDA is a superset of RDA with increased design focus on scalability, performance, data integrity, and systems management.

Hierarchical Database: The Hierarchical Database resource manager provides hierarchical database facilities through Data Language/I (DL/I), the de facto standard hierarchical database manipulation language. It provides remote access to and interoperability among hierarchical databases using the Open Blueprint conversational model of communications. Support for clients and servers is provided for a wide variety of IBM and other environments.

Object-Oriented Database: Object-oriented database facilities represent an emerging technology with no well-accepted standards in place.

The Object-Oriented Database (OODB) resource manager stores data that is not “regular” in structure as is, for example, record or relational table data. Computer Aided Design (CAD) data used in many engineering applications is typical of this type of data, but a growing number of other applications use non-regular data storage and access. OODB enables transparent client access to objects, regardless of location or storage format.

IBM has negotiated a technology, development, and usage agreement with Object Design, Inc., to use its object-oriented database management system, ObjectStore**, in the development of software products.

Persistence Services: In object-oriented applications, some objects may be purely transient, that is, they are created, manipulated, and destroyed within the scope of the execution of the application. In other cases, objects may **persist** beyond the execution of an application. Still others may persist and be shared among different applications.

When application objects are to be permanently stored, they are referred to as **persistent objects**. Persistence Services are those services used to store and retrieve persistent objects; they provide the methods for moving and mapping between the in-memory representation to the disk-resident representation of an object. Object persistence can be automatic or under application control, and objects can be stored in either an OO database or a traditional data store.

Persistence Services also allow object-oriented applications to access existing data in traditional data stores.

File: The File resource manager includes access to multiple file systems throughout the distributed system, providing the image of a single file system.

File APIs can be classified as record and byte stream as follows:

Record Supports structured data (records, keys, and indexes)

Byte Stream Data structures are left to application conventions

Standard file APIs are supported by a file framework that routes requests to local or remote file system implementations. The framework insulates applications from multiple file system implementations. The framework includes support for commonly-supported standard procedural interfaces as well as emerging technology for object-oriented access to data.

File APIs are conveyed from file clients to file servers and among file servers through function protocols defined by specific file system implementations. While many file access protocols are used today (such as NCP, SMB, and NFS), two protocols are emphasized in the Open Blueprint because of the level of integration with other Open Blueprint services:

- Distributed Data Management (DDM) architecture for record file access.
- Distributed File System (DFS from OSF/DCE) protocols for byte stream file access. It supports a wide variety of semantics including byte stream requests, client caching, server-to-server caching, and file system administration.

Storage Management: The Storage Management resource manager provides for the protection and cost-effective management of storage resources. These facilities can be invoked directly by an API or transparently from another service. The transparent services integrate primarily with the File resource manager; applications using the base file systems will automatically receive storage management services such as

- File protection through state-of-the-art backup
- Long-term file or information object preservation through data archiving
- Use of cost-effective storage components through migrating infrequently-used data. On access, the data is transparently recalled to the file system for standard access.

These services operate transparently and provide data protection and management in a seamless manner.

The API is based upon the emerging X/Open Systems Management: Backup Services API: (XBSA).

Systems Management Services

Manageability of software and hardware resources is critical to users in the open, distributed environment. For example:

- Software must be able to be distributed electronically and installed locally or remotely.
- Configuration information for resources should be self-defining and should minimize user involvement.
- Basic error logging and fault isolation functions are required to support improved and automated problem determination.
- Standard management definitions and protocols should provide for the collection of error and performance data and the application of changes and fixes.
- Management facilities for system administrators are required for local and remote resources. Installations should be able to choose a centralized or distributed management scheme.
- Standard APIs for management are needed to facilitate interoperability and application portability.

The Open Blueprint includes a structure (shown as a backplane) that allows for resources to be managed across open, heterogeneous, enterprise-wide, distributed systems (see Figure 7). That structure is based on SystemView* which defines the necessary systems management functions and services in support of Open Blueprint resource managers and applications. Systems management is shown as a backplane to emphasize its importance across all the resource managers and to indicate that additional system management services are needed. SystemView and the Open Blueprint address complementary aspects of capabilities needed to address the multivendor enterprise environment.

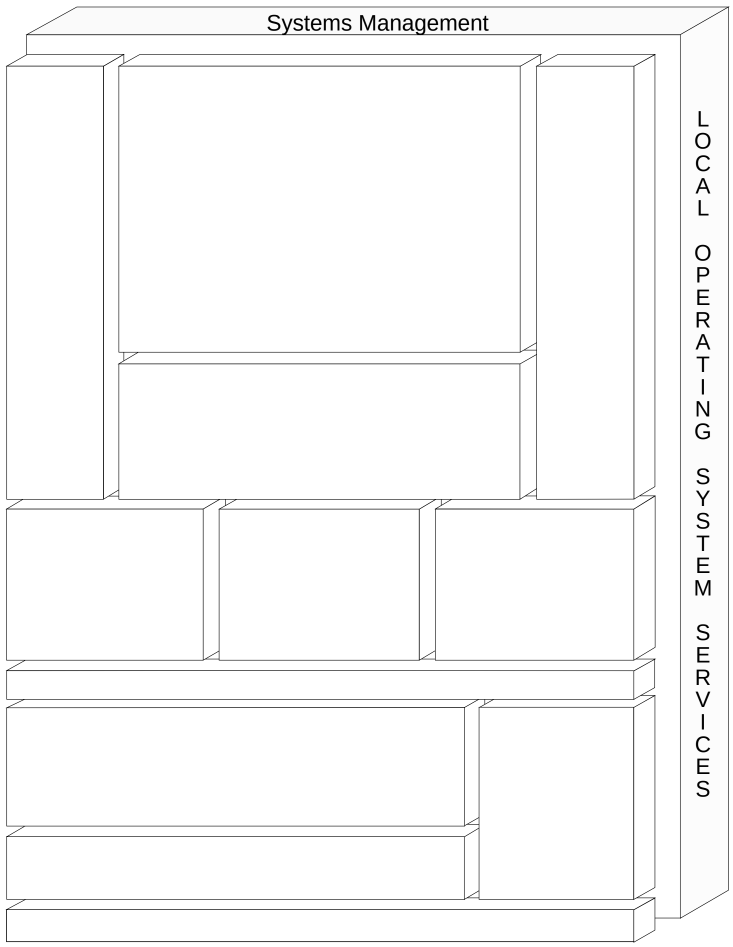

Figure 7. Systems Management and Local Operating System Services

Four major elements are included in the systems management backplane:

- **Systems management applications** provide functions to interact directly with system administrators, system programmers, and other personnel involved in the operation and administration of information systems. These applications also provide the ability to automate tasks and processes. Systems management applications are categorized into disciplines as defined by SystemView: business management, change management, configuration management, operations management, performance management, and problem management.
- **Systems management services** provide common services to resource managers and systems management applications unique to the needs of systems management. Through the use of these and other Open Blueprint services, systems management applications will have an integrated appearance to the user, share data, and be shielded from the underlying technology to access systems management information from remote systems.

- **Agents** provide services to resource managers to facilitate interaction with systems management applications.
- **Managed resources** are the resources themselves, which are owned by resource managers, and are available for interrogation and control by systems management applications.

In the Open Blueprint, system management supports *de jure* and *de facto* industry standards:

- The OSI X.700 Management Model.
- The Simple Network Management Protocol (SNMP), Common Management Information Protocol (CMIP), and Systems Network Architecture/Management Services (SNA/MS) protocols.
- The industry standard management APIs XMP and SNMP, that enable portability for management applications.

There are also several emerging standards, such as:

- POSIX System Administration services (1003.7 group)
- The Desktop Management Task Force (DTMF) Desktop Management Interface (DMI) APIs for Vital Product Data
- OSF/Distributed Management Environment (DME) NetLS API for License Management
- X/Open System Management Object Services APIs

The systems management structure will continue to evolve to support additional management protocols and APIs as they emerge. In addition, systems management software is evolving from procedural to object-based implementations.

Local Operating System Services

Local operating system services manage resources that are not distributed—CPUs, memory, and local devices, for example. Local operating system services include such functions as work management, environment state support, memory management, event handling, security context management, local system logon, locking, accounting, tracing, and journalling.

Standard interfaces to local operating system services maximize portability of code across platforms, including standards from POSIX and X/Open's Portability Guide. As the local operating systems evolve toward object-oriented structures, the services will be accessed through object services.

Application Development Tools

Application development (AD) tools are themselves applications and are increasingly based on client/server techniques. Development tools take advantage of the Open Blueprint services. For example, AD tools use object-oriented technology where appropriate. Tools also have graphical user interfaces built on IBM user-interface class libraries, which in turn, use Open Blueprint services. As another example, distributed debugging tools use the distributed system services, so the tools can use the network for distributed development and testing.

AD tools use the Open Blueprint services to make the task of development itself easier and more effective. But the primary importance of these tools in the context of the Open Blueprint is to make it easier for developers to build on the services of the Open Blueprint in creating and maintaining their own distributed applications. The tools accomplish this through the functions they offer to developers and the ease with which those functions can be accessed and incorporated into applications.

Integration

The Open Blueprint supports programming development by providing:

- Resource manager definition
- Standards adherence
- Integration objectives and criteria (by using other resource managers and common services)

Many products today implement the standards defined by the Open Blueprint; future products will also implement them. A major goal of the Open Blueprint is *integration* or *seamless interoperability.* Interoperability means that products can work together, usually because they have implemented the same set of standards; seamless interoperability means that the end user does not have to do anything unusual to get the products to work together and has a single-system view of the network.

The following scenarios are key focus areas for integration:

Single Signon
Lets the user have a single identification within the network. The "network" could refer to one business or physical network or to multiple networks. Single signon lets users log on with a single password and have access to all the network facilities for which they are authorized.

Network-wide security
Protects network resources and users. It encompasses three basic areas:

- Data encryption to protect data in the network
- Authentication of users and resource managers
- Resource access control to manage what a particular user can do

Authentication involves identifying the client and validating the server.

Network-wide directory
Provides information about resources in a network. It eliminates the need for product-unique ways to locate resources, and shields users from keeping track of where resources are located.

Global Transparent Access
Enables a user to access data or applications in a network or networks without concern for where they reside.

The Open Blueprint promotes the integration of a broad range of client/server products under a single, network-wide signon process, a single name-resolution directory for locating distributed resources, and common services and graphical user interface for administration. Through the use of DCE technology for the underlying directory and security services, the Open Blueprint also promotes interoperability with other vendor-provided products.

Summary

Figure 8 shows all of the components of the Open Blueprint.

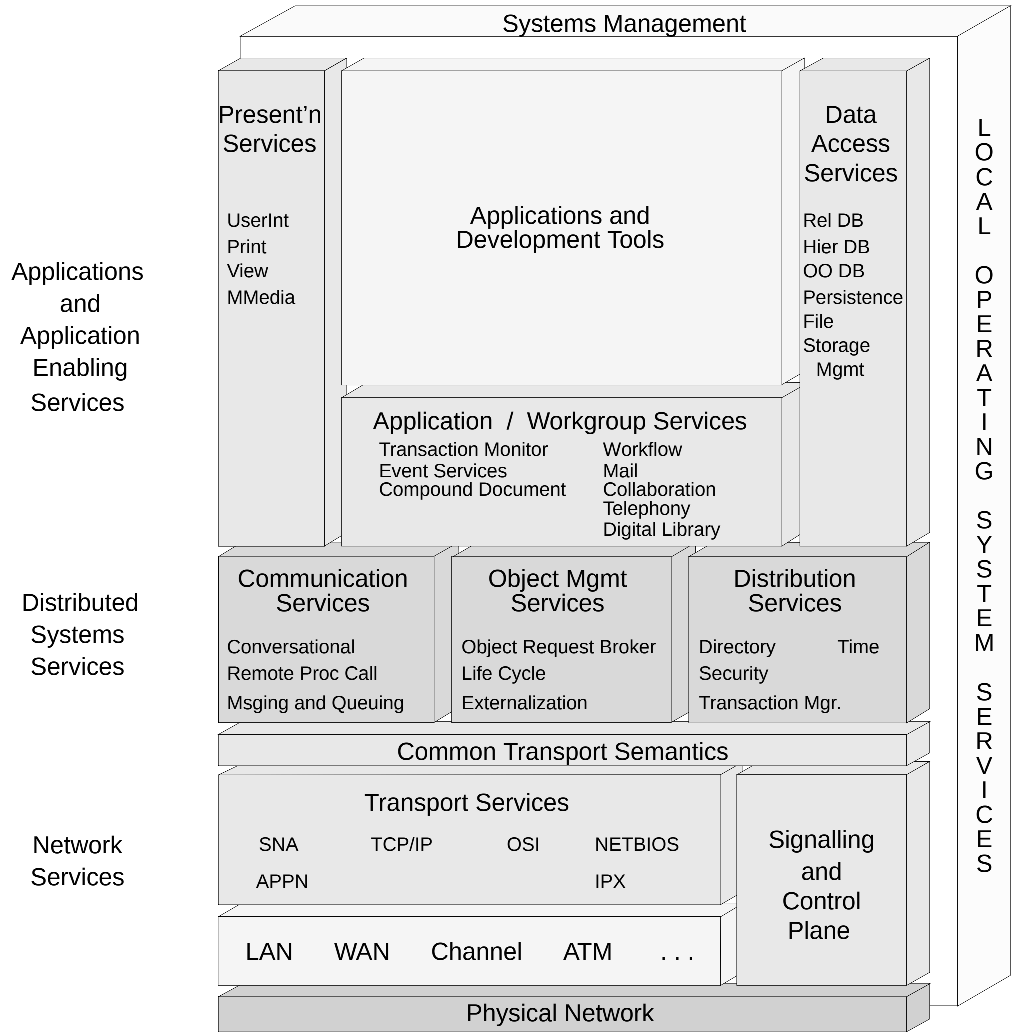

Figure 8. Components of the Open Blueprint

The Open Blueprint has been referred to as an "architecture," but just what does that mean? We are all familiar with the concept of architecture as it applies to the construction of buildings. In fact, a dictionary lookup will result in a generic meaning such as "profession of designing and constructing buildings," or "a style of design and construction." This definition applies to software, as an Information Technology (IT) architecture is used to plan and build software.

An interesting variation of the definition is offered by consultant N. Dean Meyer, president of consulting firm NDMA, as quoted in *Beyond Computing* magazine:

> Think of IT architecture in terms of planning a city rather than building a house. Architecture provides building codes that limit near-term design options for the sake of the community, but these codes do not tell individuals what kinds of buildings they need. Like building codes, an IT architecture should consist of a set of standards, guidelines, and statements of direction that permit step-by-step business-driven implementation without sacrificing integration.

The Open Blueprint does not dictate what to build but rather describes a set of codes or guidelines for the "planned" software community.

Don Tapscott and Art Caston are also proponents of standards-based IT architectures, as stated in their book, *Paradigm Shift: The New Promise of Information Technology*:

> An enterprise architecture is required to realize the shift to network-centric computing. The architecture defines the essential components (software, information, and technology) required, and inevitably leads to the adoption of industry standards.

The Open Blueprint is IBM's view of such a standards-based, modular enterprise architecture. While the Open Blueprint has been in use within IBM for a long while, it was publicly introduced in April of 1994. The reaction by the press, consultants, and customers has been extremely positive:

- "...Open Blueprint opens the door for considerable diversity...to bring a measure of order to existing free-for-all of products and services..." (*Midrange Systems*, August 26, 1994)
- "What's so notable...is the extent to which it breaks with IBM's past as a maker of proprietary systems." (*PC Week*, August 24, 1994)
- "The Open Blueprint will be a helpful, positive and much-needed framework for both IBM customers and IBM product planners to embrace the new paradigm in information technology." (Don Tapscott, author of *Paradigm Shift: The New Promise of Information Technology*)
- "Seer supports the concepts presented within the Open Blueprint and believes that their customers will benefit from the commonality and consistency of distributed services presented in this framework." (Charles Riegel, Vice President Worldwide Marketing, Seer Technologies)

Many IBM customers are using the Open Blueprint as a guide as they create their own internal architectures for deploying applications in the open, distributed environment. In addition to this role, the Open Blueprint also:

- Provides information on IBM's direction for products and solutions in the open distributed environment.
- Guides developers as they supply customers with products and solutions that meet their needs, both in function and ability to integrate and interoperate with other installed products.
- Provides a context for the incorporation of new technologies into a distributed environment. The Open Blueprint will, in turn, evolve as new technologies are incorporated. New and additional technologies—mobile computing for example, continue to be investigated.

Those products from IBM and other suppliers that align with and exploit the Open Blueprint:

- Support portability and transparency through common, industry-standard APIs
- Support interoperability via common, industry-standard formats and protocols
- Use infrastructure-provided resource managers instead of product-specific functions

The Open Blueprint is a means to an end, not an end in itself. As Robert Orfali, Dan Harkey, and Jeri Edwards remind us in their book, *Essential Client/Server Survival Guide*:

> Architectures help us identify structural elements that may be used as building blocks in the construction of ever more complex systems. Just like we buy homes, not plans, users in the computer industry buy solutions to business problems, not grand client/server architectures. But architecture determines the structure of the houses, high rises, office buildings, and cities where we live and work. In the computer analogy, architecture helps us determine the structure and shape of the client/server systems we can build to meet various needs.

The success of the Open Blueprint will be measured by the products and solutions it enables, and their contribution to customers' efforts at reengineering their business processes to achieve greater profitability and competitiveness.

Appendix A. Standards

The following table shows some principal industry interfaces and protocols in the Open Blueprint. It is based on formal standards that have achieved widespread industry acceptance.

The Open Blueprint also includes several *de facto* standards, either in the absence of a formal standard, or to achieve interoperability with existing installations.

References for ISO/IEC, IETF, and ITU-T are to Standards Number. References for X/Open are to International Standard Book Number (ISBN). You can get information on the company-contributed interfaces and protocols from the contributing company.

IBM Interfaces	Document Order Number
CICS API	SC33-1007
DL/I	SC26-8015
IMS TM API	SC26-8017
MQI	SC33-0850
OS/2 PM API	ISBN 1-56529-155-7

IBM Protocols	Document Order Number
APPC	SC31-6808
APPN	SC30-3422
DDM	SC21-9526
DL/I	SC26-8015
DRDA	SC26-4651
IPDS	S544-3417
MPTN	GC31-7073/4
SNA Sync Point	SC31-8134
SNA/MS	SC30-3346
SNA Formats	GA27-3136

Common Transport Semantics

Interfaces	Protocols	Organization	Reference
TLPB	XMPTN	X/Open	ISBN 1-859120-42-3
	OSI over TCP/IP	IETF	1006
	NetBIOS over TCP/IP	IETF	1001/1002

Transport Network

Interfaces	Protocols	Organization	Reference
XTI		X/Open	ISBN 1-872630-29-4
Berkeley Sockets		Berkeley	4.3 BSD Document
	APPN	APPN Implementors Workshop	IBM - *
	TCP/IP	IETF	791, 793, 768
	OSI	ISO/IEC	8072/3, 8602, 8473
	NetBIOS	IBM	*
	IPX	Novell	*
	Appletalk	Apple	*

Signalling and Control

Interfaces	Protocols	Organization	Reference
	ISDN	ITU-T	Q.931, Q.2931
	Conferencing	ITU-T	H.320

Subnetworking

Interfaces	Protocols	Organization	Reference
	LAN (Ethernet, Token Ring)	ISO/IEC	8802
	ISO DLC	ISO/IEC	3309, 4335, 7776
	Frame Relay	ITU-T	I.233
	ATM	ITU-T	I.150, I.327, I.361/2/3
		ATM Forum	UNI 3.1

* The contributing companies may be contacted for more details. The IBM document numbers are on page 33.

Conversational Communication

Interfaces	Protocols	Organization	Reference
CPIC		X/Open	ISBN 1-85912 057-1
	APPC	IBM	*
	OSI TP	ISO/IEC	10026

RPC Communication

Interfaces	Protocols	Organization	Reference
DCE RPC	DCE RPC	X/Open	ISBN 1-85912-0679

Messaging and Queuing

Interfaces	Protocols	Organization	Reference
MQI		IBM	*

Object Management

Interfaces	Protocols	Organization	Reference
CORBA		OMG	93-12-43
Common Object Services		OMG	94-1-1
Lifecycle Services		OMG	93-7-4
Externalization Services		OMG	94-9-15
Identity		OMG	94-5-5
	IIOP, DCE-CIOP	OMG	95-3-10

Directory and Naming

Interfaces	Protocols	Organization	Reference
XDS		X/Open	ISBN 1-872630-18-9
XOM		X/Open	ISBN 1-85912-0784
Directory Service	Cell Directory Service	X/Open	ISBN 1-872630-17-0
	X.500	ISO/IEC	9594 (88 Standard)
	Domain Name Service	IETF	1034,1035
Naming Service		OMG	93-5-2

* The contributing companies may be contacted for more details. The IBM document numbers are on page 33.

Security - User Registry

Interfaces	Protocols	Organization	Reference
DCE SEC_RGY	DCE Registry	OSF	DCE Security Specification

Security - Authentication

Interfaces	Protocols	Organization	Reference
GSSAPI		IETF/OSF	DCE Security Specification
	Kerberos	MIT/OSF	Kerberos draft

Security - Access Control

Interfaces	Protocols	Organization	Reference
DCE SEC_ACL	DCE ACLs	OSF	DCE Security Specification and POSIX 1003.1e,1003.2c

Time

Interfaces	Protocols	Organization	Reference
DCE DTS	DCE DTS	X/Open	ISBN 1-85912-0415

Transaction Manager

Interfaces	Protocols	Organization	Reference
Transactional C	Encina	Transarc	*
TX		X/Open	ISBN 1-872630-65-0
XA		X/Open	ISBN 1-872630-24-3
	OSI TP	ISO/IEC	10026
CICS API		IBM	*
IMS TM API		IBM	*
	SNA Sync Point	IBM	*
Transaction Service		OMG	94-8-4

* The contributing companies may be contacted for more details. The IBM document numbers are on page 33.

User Interface

Interfaces	Protocols	Organization	Reference
OS/2 PM		IBM	*
Motif		OSF	MOTIF Specification
Windows		Microsoft	*
X Window Xlib		X/Open	ISBN 1-872630-11-1
X Window Xt intrinsics		X/Open	ISBN 1-872630-14-6
	X Window System Protocols	X/Open	ISBN 1-872630-13-8

Distributed Print

Interfaces	Protocols	Organization	Reference
Distributed Print API		IEEE	POSIX 1387.4/d7
	DPA	ISO/IEC	10175-3
	lpr/lpd	IETF	1179
	SMB	X/Open	ISBN 1-872630-45-6
	NCP	Novell	*

Print Data Streams

Interfaces	Protocols	Organization	Reference
	IPDS	IBM	*
	PCL	HP	*
	PostScript	Adobe Systems	*

Distributed File

Interfaces	Protocols	Organization	Reference
	Distributed File System	OSF	DFS Specification
	SMB	X/Open	ISBN 1-872630-45-6
	NFS	SUN	*
	NCP	Novell	*
	DDM	IBM	*

Relational Database

Interfaces	Protocols	Organization	Reference
SQL (1992E)		ISO/IEC	ISBN 1-872630-63-4
SQL CLI		X/Open	9075
	DRDA	IBM	*
ODBC		Microsoft	*

* The contributing companies may be contacted for more details. The IBM document numbers are on page 33.

Hierarchical Database

Interfaces	Protocols	Organization	Reference
DL/I		IBM	*

Persistence Services

Interfaces	Protocols	Organization	Reference
POS		OMG	94-1-1 94-10-7

Storage Management

Interfaces	Protocols	Organization	Reference
Backup Services (XBSA)		X/Open	ISBN 1-85912-056-3

Transaction Monitor

Interfaces	Protocols	Organization	Reference
CICS API		IBM	*
IMS TM API		IBM	*

Event Services

Interfaces	Protocols	Organization	Reference
Event Services		OMG	93-7-3

Mail

Interfaces	Protocols	Organization	Reference
Vendor-Independent Messaging (VIM)		VIM Consortium	*
Messaging API (MAPI)		Microsoft	*
Common Messaging Call API		XAPIA	*
	X.400 Messaging Handling Services	ISO/IEC	10021
	SMTP	IETF	821,822

* The contributing companies may be contacted for more details. The IBM document numbers are on page 33.

Telephony

Interfaces	Protocols	Organization	Reference
Computer Telephony Integration (CTI)	CSTA	ECMA	179, 180

Systems Management

Interfaces	Protocols	Organization	Reference
	SNA/MS	IBM	*
XMP		X/Open	ISBN 1-872630-32-4
	CMIP	ITU-T/ISO/IEC	X.711
SNMP	SNMP	IETF	1157
DMI		DMTF	*

Local Operating System Services

Interfaces	Protocols	Organization	Reference
System calls, libraries		ISO/IEC	9945-1
Commands and utilities		ISO/IEC	9945-2
Threads		IEEE	POSIX 1003.1c

* The contributing companies may be contacted for more details. The IBM document numbers are on page 33.

Appendix B. Bibliography

Articles

“The Technology Payoff,” *Business Week*, June 14, 1993.

“New Priorities,” D. Bartholomew and B. Caldwell, *Informationweek*, March 20, 1995.

“Rearchitecting the Architecture Concept,” N. Dean Meyer, *Beyond Computing*, July/August 1994.

“Riding the Bleeding Edge of Distributed Computing,” N. Engler, *Open Computing*, February 1995.

“Ease On Down The Road to Client/Server: The Open Blueprint,” D. Baron, *IBM Software Quarterly*, Fall/Winter 1994.

Books

Paradigm Shift, Donald Tapscott and Art Caston, McGraw-Hill Inc., 1993.

Re-engineering the Corporation, Michael Hammer and James Champy, Harper Collins, 1993.

The Object Advantage: Business Process Reengineering With Object Technology, Ivar Jacobson, Maria Ericsson, Agneta Jacobson, Addison-Wesley, 1995.

Object-Oriented Technology: A Manager's Guide, Dr. David A. Taylor, Addison-Wesley, 1990.

Essential Client/Server Survival Guide, Robert Orfali, Dan Harkey, and Jeri Edwards, Van Nostrand Reinhold, 1994.

IBM Open Blueprint Materials

Introduction to the Open Blueprint: A Guide to Distributed Computing, G326-0395.

Open Blueprint Technical Overview, GC23-3808.

Open Blueprint Technical Reference Library, SBOF-8702 (Hard Copy), SK2T-2478 (CDROM). (**Note:** This publication includes the first two Open Blueprint publications listed above, as well as Component Description Papers for each Open Blueprint component.)

Other IBM Publications

Information Warehouse: An Introduction, GC26-4876.

Information Warehouse: Architecture I, SC26-3244.

IBM Security Architecture, SC28-8135.

Other Publications

Distributed Computing Services Framework, X/Open Company Ltd, ISBN 1-872630-64-2, November 1992.

Readers' Comments — We'd Like to Hear from You

Introduction to the
Open Blueprint:
A Guide to
Distributed Computing

Publication No. G326-0395-01

If you especially like or dislike anything about this book, please tell us what you think. Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness.

To request additional publications or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. Use this form for comments about the information in this book and how the book presents the information.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

To send us your comments:

- By mail, use this form.

  If you are mailing this form from a country other than the United States, you can give it to the local IBM branch office or IBM representative for postage-paid mailing.

- By FAX, use this number:

  FAX (United States & Canada): 914+432-9405
  FAX (Other Countries):
  Your International Access Code +1+914+432-9405

- Electronically, use any of the following network addresses:

  IBMLink (United States customers only): KGNVMC(D58PUBS)
  IBM Mail Exchange: USIB2NZL at IBMMAIL
  Internet: d58pubs@vnet.ibm.com

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

Name | Address

Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You
G326-0395-01

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL   PERMIT NO. 40   ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 55JA Building 708-2, Mail Station P384
522 South Road
Poughkeepsie, N.Y. 12601-5400

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.

G326-0395-01