Introduction to the Open Blueprint: A Guide to Distributed Computing

G326-0395-02

Introduction to the Open Blueprint: A Guide to Distributed Computing

G326-0395-02

Third Edition (December 1996)

International Business Machines Corporation provides this manual "as is," without warranty of any kind either express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. IBM may make improvements and/or changes in the product(s) and/or program(s) described in this manual at any time.

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 CGMD
P.O. Box 12195
Research Triangle Park, NC 27709-2195
United States of America

FAX: (United States and Canada) 1-800-227-5088

Internet: USIB2HPD@vnet.ibm.com
IBMLink (United States customers only): CIBMORCF at RALVM13
IBM Mail Exchange: USIB2HPD at IBMMAIL

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. Subject to IBM's intellectual property or other legally protected rights, any functionally equivalent product, program, or service 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
500 Columbus Avenue
Thornwood, NY 10594
USA

Trademarks

The following terms are trademarks of the IBM Corporation in the United States and/or other countries:

Advanced Peer-to-Peer Networking
AnyNet
AFP
AIX
AIX/6000
APPN
Common User Access
CICS
CUA
Distributed Relational Database Architecture
DB2
DRDA
IBM
IBMLink
IMS
Information Warehouse
MQSeries
Open Blueprint
OS/2
OS/390
OS/400
System Object Model
VSE/ESA
Workplace Shell

The following terms are trademarks of other companies:

Apple	Apple Computer, Incorporated
AT&T	American Telephone & Telegraph Company (SM)
Bell	AT&T Bell Laboratories Incorporated
Borland	Borland International, Incorporated

C++	American Telephone & Telegraph Company, Incorporated
CORBA	Object Management Group, Incorporated
CompuServe	CompuServe Incorporated and H&R Block, Incorporated
DCE	The Open Software Foundation
DEC	Digital Equipment Corporation
Digital	Digital Equipment Corporation
Encina	Transarc Corporation
Finder	Apple Computer, Incorporated
Hewlett-Packard	Hewlett-Packard Company
HP	Hewlett-Packard Company
Intel	Intel Corporation
IPX	Novell, Incorporated
Java	Sun Microsystems, Incorporated
Lotus	Lotus Development Corporation
Lotus Notes	Lotus Development Corporation
Macintosh	Apple Computer, Incorporated
MCI	MCI Corporation
Microsoft	Microsoft Corporation
Microsoft Windows	Microsoft Corporation
Motif	Open Software Foundation, Incorporated
NetWare	Novell, Incorporated
NFS	Sun Microsystems, Incorporated
Northern Telecom	Northern Telecom Limited
Novell	Novell, Incorporated
ObjectStore	Object Design, Incorporated
OpenDoc	Apple Computer, Incorporated
Open Software Foundation	The Open Software Foundation, Incorporated
OSF	Open Software Foundation, Incorporated
OSF/DCE	Open Software Foundation, Incorporated
POSIX	Institute of Electrical and Electronic Engineers
PostScript	Adobe Systems, Incorporated
SCO	The Santa Cruz Operation, Incorporated
Solaris	Sun Microsystems, Incorporated
Sun	Sun Microsystems, Incorporated
Sun Microsystems	Sun Microsystems, Incorporated
SunOS	Sun Microsystems, Incorporated
Sybase	Sybase Corporation
Taligent	Taligent, Incorporated
Tandem	Tandem Computers Incorporated
Tivoli	Tivoli Systems Inc., an IBM Company
TME 10	Tivoli Systems Inc., an IBM Company
Transarc	Transarc Corporation
UniForum	/usr/group
Unisys	Unisys Corporation
UNIX	UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.
UnixWare	Novell, Incorporated
VAX	Digital Equipment Corporation
VMS	Digital Equipment Corporation
Windows	Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.
Windows NT	Microsoft Corporation
Windows 95	(logo) Microsoft Corporation
WordPerfect	WordPerfect Corporation
Xerox	Xerox Corporation
XTI	X/Open Company Limited
X/Open	X/Open Company Limited

About This Book

This book reviews IBM's technical approach to distributed, networked computing in a multivendor, heterogeneous environment. It introduces the Open Blueprint structure and its components with a special focus on Internet technology.

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

The Open Blueprint structure 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 structure 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.

Spurred by the rise of the Internet and World Wide Web, network computing is the cornerstone of current and future information technology. Any strategy for this rapidly changing environment must be flexible and dynamic, accommodating technological advances while at the same time protecting significant current investments. So, even though this document reflects a specific point in time, the Open Blueprint structure and the details behind it will evolve as new technologies take hold. It has already evolved to encompass object-oriented concepts, Internet technologies, intelligent agents, and mobile computing; it will continue to evolve to include new and emerging technologies.

The Open Blueprint structure 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 distributed or networked 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 integrate with and can interoperate with other installed products, regardless of supplier.

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 or other providers who support the open standards contained in the Open Blueprint structure.
- Consultants and service providers who offer integration services to customers

Contents of This Book

This book is organized as follows:

- **"Introduction"** discusses today's dynamic business environment, key milestones in the evolution of information technology (IT), the challenges faced by IT organizations today, and provides a high level overview of the Open Blueprint structure, its development, and its use.
- **"Description of the Open Blueprint Structure"** 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 structure 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 structure.
- **Appendix, "Standards"** summarizes the standards and protocols included in the Open Blueprint structure.
- **Appendix B, "Bibliography"** provides a Bibliography.

This 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 component. A widely known or recently developed product might be used to characterize the function, but that product might 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 comprehensive view of the digital revolution and how it is impacting daily life, see *The Digital Economy* by Don Tapscott. In his previous book, *Paradigm Shift*, co-author Tapscott makes the case for complete planning and for adopting a standards based architecture like the Open Blueprint structure. This is reinforced in Tapscott's latest book *The Digital Economy*.

These and other references are listed in Appendix B, "Bibliography" on page 55.

What's New In This Edition

This edition is focused on the Open Blueprint structure and the Internet. It describes the way in which the Open Blueprint provides a structure and context for the Internet and the World Wide Web in a network computing environment. The following items are described in this enhanced version of the Open Blueprint structure:

- Network computing support:
  - Web Browser resource manager added
  - HTTP resource manager added
  - Virtual Machine resource manager supporting Java added
  - TCP/IP function enhanced
  - Security functions expanded to include public key support

  - Directory resource manager expanded to include LDAP support and multiple, federated directory services
  - Internet voice telephony function added
  - Network access to functions of Application/Workgroup and Data Access Services added
- Collaboration resource manager now includes both synchronous and asynchronous collaboration
- Intelligent Agent Management resource manager added
- Internationalization resource manager added
- Systems Management support now based on Tivoli TME 10
- Mobile computing considerations included
- Multimedia support restructured
- User Interface resource manager renamed Human Computer Interaction resource manager to reflect industry terminology and enhanced function
- OSI transport services references removed
- URLs provided for referencing standards

Open Blueprint Structure Information on the World Wide Web

In 1996 and into 1997, the Open Blueprint documentation will increasingly be online. The Open Blueprint Web edition enables timely updates of the component description papers, addition of new components as technology demands, and the ability to stay current with trends and directions in the industry. Also included in the online edition are other papers that address current issues related to the Open Blueprint structure. The intent is to provide dynamic, focused, and relevant information about the Open Blueprint structure and its use. To visit the Open Blueprint home page, use URL http://www.software.ibm.com/openblue.

Introduction

"Limits on sharing among systems puts limits on the business. New opportunities often require new 'slices' and 'dices' of resource existing across diverse computer systems. *The inability to integrate resources restricts the ability of an enterprise to respond to opportunity*."

This insight from Hal Lorin in his book *Doing IT Right* is shared by many industry pundits and CIOs.

Furthermore, in today's dynamic environment, opportunities do not last very long—maybe days, weeks, or months, but certainly not years. Add to this environment the worldwide impact of the Internet on business and communications and you increase competitive pressures exponentially. Those businesses that are not ready to capitalize on these opportunities will not survive.

One way to be ready is to develop an enterprise information technology (IT) architecture that reflects business goals and allows the fast redirection of integrated IT resources to take advantage of the opportunities.

This section:

- Describes today's dynamic environment and how the Internet revolution impacts business
- Examines how information technology has evolved over the past several decades
- Describes the demands that the current dynamic environment places on information technology
- Provides an overview of the Open Blueprint structure and how it can help meet those demands

Dynamic Business Environment

The explosive growth of the World Wide Web and the Internet has surprised us all. One minute it is an arcane network of networks used mainly by scientists in universities and government to share research, and the next minute it is helping people send messages to each other, do their banking, attend a course through distance learning, track packages, buy and pay for just about anything, and even consult with a physician in another state. "The Internet is the first medium where you can conduct the entire sales cycle in one place," says Mark Kvamme, President and CEO of CKS/Group, an interactive ad agency in Cupertino, California.[1]

No one knows for sure, but most estimates say that there are between 20 and 40 million Internet users over 40,000 interconnected networks that connect 3.9 million computers. The estimates really get wild when addressing what those numbers might be at the turn of the century: 400 to 800 million users over 1 million interconnected networks that connect 100 million computers.

What are we really seeing? Myriad folk are *getting used to computing and computers*. This is true despite the fact that computers are not yet as easy to use as the telephone (although the industry has really progressed in computer ease of use). The Internet phenomenon has even challenged the conventional wisdom that technology only becomes useful and accepted when it becomes invisible.

For companies to survive, let alone compete, they must be able to move with lightning speed in getting their products and services to the market. The window of opportunity has decreased from years to months to weeks and sometimes days or hours! The term "Web year" was coined to describe the accelerated cycle required for Web products and is generally agreed to equal about three months.

[1] As quoted in *InformationWeek*, June 24, 1996.

But it is not just Web products that feel the crunch. Just as client/server technology brought more power to end users, the Internet technology is shifting power from companies to their customers. Indeed, marketing and advertising executives attending a trade show, AdTech '96, were told that their business is being fundamentally changed by online or network computing. “We're witnessing the biggest economic transformation since the invention of interchangeable parts,” claimed George Gilder, one of the keynote speakers. With the incredible amount of product information and online transactions available to online consumers, they now have virtually unlimited choice of products.

The judicious use of information technology is key to competitiveness. A company's IT infrastructure must be flexible enough to enable it to respond quickly. For Northern Telecom CIO Dave Cox, “the goal is to be able to build our systems instantly.”[2] In addition, the IT infrastructure and goals must be totally aligned with the business and its goals. This is not simple to do and, in fact, has been the objective of most recent reengineering efforts. In *The Digital Economy* Don Tapscott points out that one reason that reengineering has failed is that it focuses only on business processes, only one piece of the puzzle. He advocates a focus on the business model—“The new economy demands that companies change their business model, *and the new technology enables it.*”

There is a potential pit here, however, into which even the most progressive and responsive companies can plunge if they do not plan carefully. That pit is one made up of incompatible parts and technologies across different areas of an enterprise and network. Rapidly changing technology often resembles a moving target and can easily get out of focus, causing rework, repair, and rethinking, not to mention real frustration and blown plans and budgets. The chief protection against this very real danger is to operate within a structure that is flexible enough to accommodate new technologies as well as current ones (which run the business today) and that identifies functional boundaries in a way that makes sense.

IT Trends

It is easy to see why aligning business goals and IT goals is difficult right now—companies are struggling to figure out exactly what their business goals are and which business models will help achieve them. Even when that is done (if it is ever really done), it is still a challenge to align IT with the business goals and models. To see why, it is helpful to take a brief look at some of the IT trends that have shaped the current environment.

Historical Perspective

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

[2] As quoted in Don Tapscott's *The Digital Economy*

to LANs, and work groups could share information more easily. The hegemony of the IT department was vanishing; there 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 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 a powerful environment for general purpose, non-scientific computing as well. In the networking computing world, Unix has emerged as a powerful server platform.

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 both in application design and 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.

The Need for Collaboration

Functionally based hierarchies gave 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 was upgraded to video conferencing which is now moving to the user's desktop where a combination of the telephone and the PC allows people to converse and draw on their PCs as if they were in the same conference room drawing on a shared white board. Groupware, such as Lotus Notes, has enabled collaboration and resource sharing among teams. Users are looking to computer technology to eliminate the limitations of location and distance to increase their productivity.

Impact on the Future

The thread that weaves all of these trends and advances together and provides the most visible benefit to the end user is the Internet. The availability of the Internet to the average user was achieved only with the development of the World Wide Web and a graphical user interface that simplified access. There was now an easy interface based on a well understood concept—a document—and a user could move comfortably from one document to the next with one click of a mouse. This was accomplished through a document browser, or Web browser, which became the average user's on ramp to the information highway. It unleashed the power of the Internet for everyone.

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, bringing new meaning (and challenges such as security) to the old New York Telephone slogan: “We're all connected!”

The impact of the explosion in the Internet can be felt at all levels of information technology. From desktop applications to server and mainframe topologies to emerging software and hardware technologies, the demand for Internet access is driving IT solutions. Business processes and policies are also driven to change as employees discover information on the Internet that leads to greater efficiency and effectiveness in their work. Desktop access from work is becoming the norm, and individual employees bring their consumer expectations of content and availability not only to the Internet itself but also to the applications on IT systems that provide the infrastructure to facilitate access. The ramifications of these expectations are that at a system and network level, the advent of the Internet is having a ripple effect which touches many components. To effectively use and manage this infant giant, systems and networks must be designed and structured to integrate the growing technology base focused on Internet access and use. They must provide:

- Easy access
- Effective network support
- Easy use and integration of Internet information
- Security for Internet interactions
- Growth path for future function

Add to this dynamic mix the need to get real work done, run the business, and meet next quarter's profit objectives, and you have enough contradictions and potential conflicts to bring IT systems to their knees. Without a context or structure within which to place these opposing demands, the probability of costly mistakes rises considerably. The idea behind the Open Blueprint structure is to provide that context, to propose a structure which can help make sense of conflicting needs and technologies while keeping existing systems and code useful. The current focus on the Internet is an excellent test for a system or network structure—so many new capabilities are being added at various levels that their ability to integrate inside such a structure demonstrates both the flexibility and inclusiveness of the Open Blueprint structure.

This is not just a structure for IBM systems and networks, but for any of today's heterogeneous, multiplatform environments common in business throughout the world.

Legacy

The interesting thing about the evolution of IT is that, so far, nothing has gone away. That is, even though new technology and paradigms dominate, the older technologies and paradigms do not die. In many cases they do not even fade away! This, of course, is one of the things that makes the environment so complicated. There is an old joke that characterizes the problem. Don Tapscott alludes to it in his book, *The Digital Economy*: "Yes, God is alleged to have created the world in [only] six days, but he didn't have an installed base to work with."

The difference between today and a few years ago is that it takes a lot less time for a system to become a legacy!

Heterogeneity and Standards

Post centralized era computing is characterized by computers and software from multiple vendors. Even if two software products can talk together, they might not really work together because of:

- Different data organizations
- Different interfaces to invoke services such as accessing data
- Different methods for locating resources in a network
- Different methods for controlling access to sensitive resources
- Different "look and feel" making it difficult for a user to move from one piece of software to another
- Different management schemes for each environment

There are also some benefits to our multivendor environment:

- A business is not locked into one vendor's set of products and can decide its own rate of technology adoption.
- Competition between vendors tends to lower prices of products and services.
- Competition also tends to cause faster delivery of key hardware and software functions.

One way to mitigate the interoperability problems of heterogeneity is standards—whether legislated through standards bodies, informally agreed to by consortia, or merely used by most providers because of market acceptance or presence. Standards in themselves do not guarantee interoperability, but they are a big first step.

In fact, one reason that the World Wide Web portion of the Internet is so attractive is that it provides standard ways of doing things that are fairly simple and easily adapted to different computing environments. The "Gold Rush" has begun; suppliers are enabling their products to be accessible through the World Wide Web at an incredible rate.

IT Challenge

Information technology providers have a big challenge: how to harness the power of new and emerging technologies to create an IT infrastructure flexible enough to allow their organizations to capitalize on fleeting business opportunities while protecting past investments and maintaining the integrity of company data and other assets.

IT Requirements

The IT infrastructure must address the following requirements:

Transparent Global Access

Users need access to data and applications wherever they reside, whether inside or outside of the enterprise boundaries. This access must be transparent, not requiring people to know where the resources are or to do anything overt based on the location of the resources.

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 users' desktops are on a stationary, personal computer or a mobile hand held device, they should be able to perceive the network as a single system. For example, users want to log on once, provide one password, and have access to all 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 the functions they need without being restricted to a particular 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 applications that deploy and operate over the network. 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 enable response 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, including public networks populated by non-secure PCs and unknown servers. 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 OS/390, OS/400, and OS/2 must be protected, while still enabling the exploitation of new and emerging technologies such as objects and compound documents, mobile computing, intelligent agents, and especially the Internet and the World Wide Web.

IT Architecture

The above requirements, taken together with the challenges of the dynamic business environment and world wide connectivity, lead to the conclusion that a plan, or architecture, is crucial for survival and growth. In their book *Paradigm Shift: The New Promise of Information Technology*, Don Tapscott and Art Caston assert that "an enterprise architecture is required to realize the shift to network-centric computing." But this architecture is not the same as past architectures. According to Tapscott in *The Digital Economy*:

> In the past, an architecture was really the design of a system that had been created to meet specific application needs. In the new business environment, organizations have little idea what their application needs will be in two, let alone five or ten years. Consequently, we need architectures that can enable the exploitation of unforeseen opportunities and meet unpredictable needs.

Often, past architectures were really "marketectures" or "productectures" that were simply lists of products and that did not provide any insight into the nature of the environment, nor allow for instant response to new opportunities.

The IT infrastructure is about as interesting to an end user as the utility infrastructure is to a person using a toaster or a hair dryer. It is only noticeable if it fails—that is, if the toaster or hair dryer does not work after it has been plugged into the wall socket. However, from the provider's point of view, the infrastructure, IT or other, is absolutely crucial. According to Northern Telecom's CIO Dave Cox, "infrastructure is a mind set issue. From the customer's standpoint it means service on demand and paying for what you use. From the provider view it means deploying and investing in anticipation of demand."[3] Many items that Northern Telecom used to manage separately are now managed as infrastructure.

A continuing debate in the distributed architecture arena is that of how many tiers one should have. Classical client/server systems are two tiered—a client and a server. Most of the business logic is located with the user interface logic on the client. It was soon discovered that these two tiered applications did not scale upward very well when organizations tried to extend them broadly across networks. According to Thomas Kaneshige writing in *Client/Server Computing*, "the solution is multi-tiered architectures. This involves three vertical tiers—interface, server, and database—and horizontal server to server tiers as well." As with many debates, there is no right answer. But, where is it written that applications can use only one model of distribution? Why not use the model that best fits the requirements of the situation? The real issue is the development of an IT architecture that allows one to tier any way that makes sense. That is, it is important to have the ability to "slice and dice" in any way that best addresses the opportunity.

An excellent definition of an architecture for today's environment 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.

IBM needed an architecture to ease its transition from host centered, through client/server, and into network computing and to help it make sense of the many emerging technologies in our dynamic industry.

---

3 As quoted in *The Digital Economy*

The Open Blueprint structure was developed for that purpose. As it turns out, the Open Blueprint structure has been used by many organizations as a reference model from which to build their own architectures and infrastructures for network computing.

Open Blueprint Overview

The Open Blueprint structure is a functional view of the distributed, networked environment. It serves as a reference framework that defines the services required by applications in a multivendor (that is, heterogeneous), network environment. As in Meyer's city planning view of architectures cited earlier, the Open Blueprint describes a set of building codes or guidelines for the “planned” software community.

Three Points of View

The Open Blueprint structure can be viewed on three levels: as an architecture, as a set of technologies, and as a basis for specific products.

Architecture: An architecture reveals an organization's *Weltanschauung*, or perspective, its “world view” of its products and strategies. All of the functions considered crucial would be contained in that organization's architecture. Incorporating long held values such as security, reliability, and availability, the Open Blueprint structure takes a comprehensive view of the network computing environment. It is a view that spans the enterprise and beyond, including clients, servers, and the network. An organization that just focused on the network, for example, would have a very different *Weltanshauung*, and consequently a different, less comprehensive architecture.

The pictorial representation of an architecture is often drawn as a set of related blocks, each containing functions that are important to that view. The Open Blueprint graphic is made up of blocks representing functions that provide the services required by applications in a heterogeneous, distributed, networked environment. The arrangement of the blocks is important as are the boundaries between them. Sometimes, the blocks appear to form layers. In those cases, more explanation than just the picture is necessary to determine the relationships among 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 structure is functionally layered.

Technologies: Once the perspective is known, and the functions for each block are specified, the next level involves the choice of the technologies for each function or component in the architecture. These choices are key, because they will indicate the degree of openness and interoperability inherent in the architecture and the architecture's flexibility in accommodating new technologies. The technologies included in the Open Blueprint structure 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 CORBAservices: Common Object Services Specification.

Products: 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 a one-to-one relationship between components in the Open Blueprint structure and products because there are considerations other than architectural for the contents of products. For example, if certain functions are always performed together, it might make sense to package them together in one product.

Single System Image

The Open Blueprint structure 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.

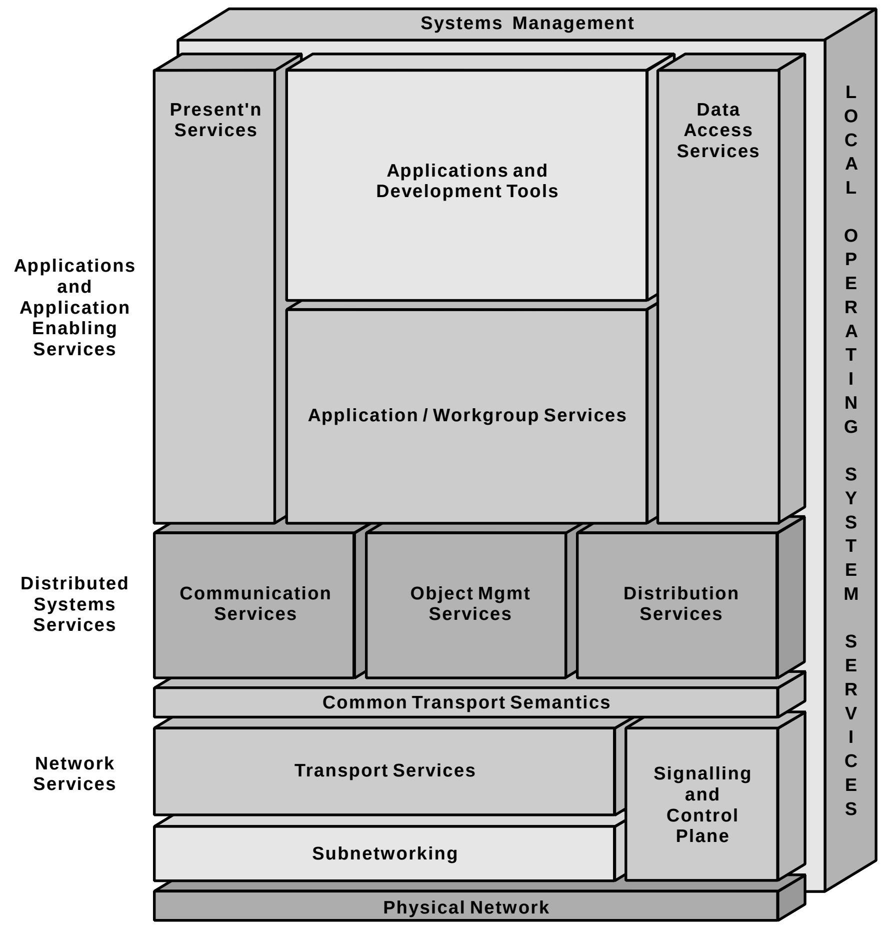

Figure 1. The Open Blueprint structure

The Open Blueprint structure enables a network of systems to function as a unit, as a *distributed operating system*. A distributed operating system is composed of multiple systems separated from each other and connected by a communications network. Services on each system manage resources cooperatively 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 structure. 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, client/server, or networked applications.

Figure 1 represents one instance of a platform in the network. Figure 2 on page 10 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 structure.

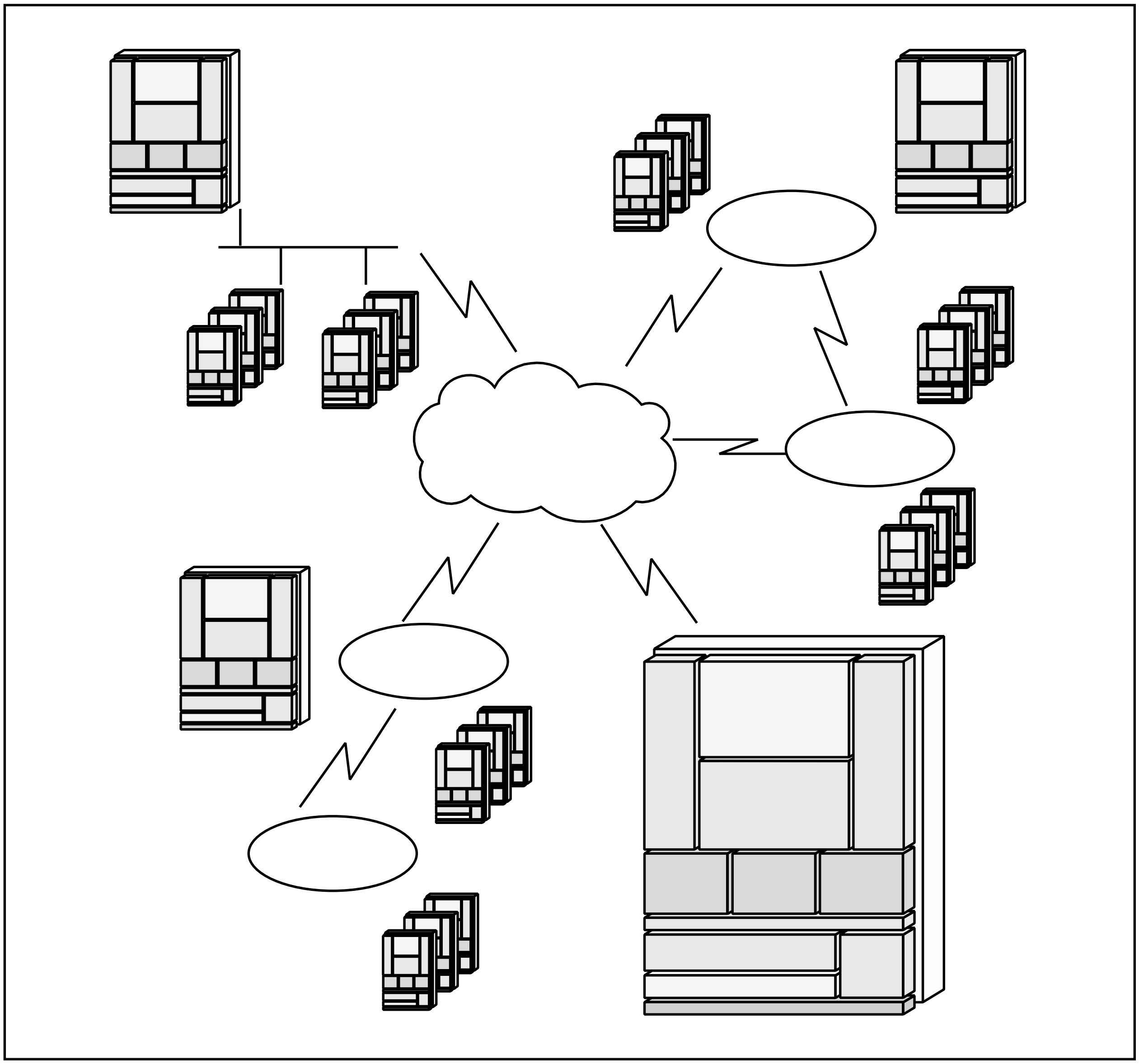

Figure 2. Distributed Operating System

Open Blueprint Components

As shown in Figure 1 on page 9, there are several sets of component services in the Open Blueprint:

Network Services

- **Common Transport Semantics** supports protocol independent communication in the distributed network.
- **Transport Services** provides the protocols for transporting information from one system to another, such as TCP/IP, SNA/APPN, 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 component 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 components by providing common functions such as directory and security.

Application Enabling Services

- **Presentation Services** define the mechanisms for interaction between applications and users, including Web browser, GUI, and multimedia support.
- **Application/Workgroup Services** are common functions, such as collaboration, 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 that enable either system administrators or automated procedures to manage the distributed operating system.

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

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

Building Blocks

The blocks in the Open Blueprint diagram (Figure 1 on page 9) contain multiple components and do not correspond to specific products. The Open Blueprint structure is implemented by different products on different system platforms; the Open Blueprint structure does not describe how the implementing software is packaged into offerings. Rather, it provides software principles and guidelines, defines the desired functions, and specifies important boundaries, interfaces, and technical attributes.

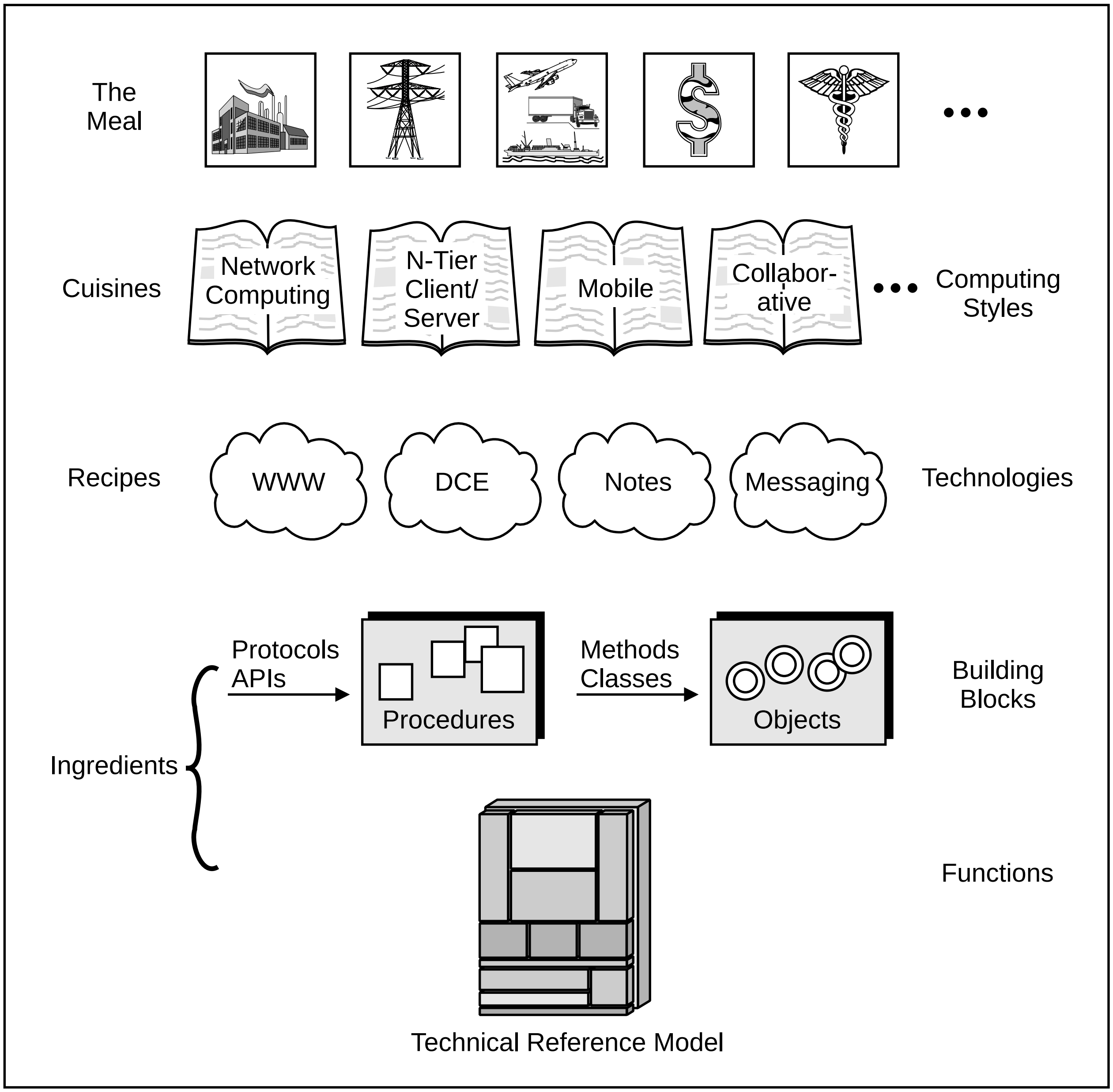

Figure 3. Open Blueprint Building Blocks

The Open Blueprint components can be combined in different ways to address varying requirements or environmental constraints. At the most generic level (at the bottom of Figure 3) the Open Blueprint is a reference model that defines the functions necessary for heterogeneous, networked computing. It defines the lowest, most granular level of "ingredients" and certain basic rules and relationships among these ingredients.

These ingredients can provide either procedural or object-oriented interfaces and can coexist and work with each other. They are the building blocks that can be combined in different ways to form "recipes."

These recipes represent various subsets of the Open Blueprint structure that narrow the focus from the generic to the specific and help organizations plan for and choose technologies and products.

There are various building blocks that, when combined, form higher level building blocks that provide infrastructure. Examples are the Web browser and HTTP support based upon generally accepted Internet standards; collaborative infrastructure based upon pervasive Lotus Notes technology, or the messaging and queueing model.

Just as a group of related recipes form a particular “cuisine,” the infrastructure building blocks can be combined into various computing “styles” (for example, n-tier client/server or network computing). The computing style “cuisines” are implemented by products that are combined to form the base for various industry solutions.

The Open Blueprint structure does not dictate or limit the internal structure of applications, nor of the components themselves. Two-tier, three tier, N-tier, network computing—all are possible with combinations of Open Blueprint components.

In its most generic role, the Open Blueprint structure provides a reference model for creating specific IT architectures, or for performing general planning and analysis. Various subsets of the structure can be chosen to focus on a specific implementation. Products can be built according to or evaluated against the Open Blueprint structure and the standard APIs and protocols specified in it.

Integration

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 structure provide designated interfaces and protocols. Products that use components defined within the Open Blueprint structure, rather than their own unique mechanisms, are truly integrated with the Open Blueprint structure. For example, 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 structure, 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.

Open Blueprint Usage

IBM developed the Open Blueprint structure for its own use as:

- A **communications vehicle** for software directions
- A **reference model** to:
  - assist in product integration
  - provide a common development vocabulary and context
  - provide a common technical structure
- A **tool** to help manage software development

The Open Blueprint structure has developed into **intellectual capital** that is considered the most comprehensive in the industry and that is used by IBM consulting and services organizations to assist their customers in the development of IT architectures.

As IBM communicated its directions using the Open Blueprint structure as an umbrella, many customers saw that it had value for themselves as well:

- It gave them a **context** in which to understand both IBM's directions as well as the nature of the distributed environment.

- As a **reference model**, it helped many customers create their own unique IT architecture.

- It provided an **analytical tool** for prioritizing and planning for distributed, network computing.
- It was used as a **checklist** when developing or acquiring software products.

Exciting uses of the Open Blueprint structure are highlighted on the Open Blueprint home page at http://www.software.ibm.com/openblue.

Open Blueprint Evolution

The pace of technology introduction has shown no sign of slowing down. New advances in software and hardware are constantly being brought to market with varying rates of acceptance by users. The importance of each of these new technologies varies by business, and by function within a business, but the net result is that new capabilities and functions must be accommodated within the context of existing systems and networks. The smaller the impact on the existing structure, the more easily new technologies can be incorporated. Systems that are responsive to change provide very high value to their users and to the businesses they support. But how do you evaluate whether or not a business's IT architecture will accommodate a new technology? How can you minimize the disruption of next year's perfect solution if you cannot even describe it at this moment? How can a business avoid making costly investments in a technology that cannot interact with other elements of the system or network?

The answer lies in the structured planning of systems and networks. As technologies proliferate in the market, the careful consideration of how to leverage them inside an existing environment assumes more and more importance. The Open Blueprint structure provides exactly that: a blueprint or road map for structuring new technology or function within an existing framework. Open Blueprint components are functionally layered. New technologies can be either contained within a single functional component, such as HTTP, or provided by multiple components, as in the case of mobile computing. For functionally complex technologies like those that support the Internet, the key is that the functional components are flexibly defined so that the relationships among components retain their logic as new technologies are added. New technologies that offer alternatives or substitutes for existing implementations of a function can even coexist with antique functions or services in the system or network. If entirely new functions are added, the same concepts apply. Existing relationships among functional components are not impacted if a new component is added. That new function can develop its own relationships to other functional components without effecting the overall integrity of the Open Blueprint structure. If changes are required because of a new addition, those components can be clearly identified and the changes anticipated as part of a sizing exercise. The impacts can be anticipated and absorbed more easily than if each addition proceeds on a “put it in and see what happens” approach.

Other Architectures

Other software and hardware suppliers have architectures and reference models. For more details and a comparison of several of the structures, see the report titled *Enterprise Architecture: A Comparison of Vendor Initiatives*, by independent consultant John Tibbetts of Kinexis. It can be accessed through the Open Blueprint home page, http://www.software.ibm.com/openblue.

Industry consortia have also put forward their approaches to a reference model for network computing. The most notable is the X/Open Architectural Framework (XAF), which was developed at the request of the X/Open User Council.[4] XAF is based on the U.S. Department of Defense architecture, the Technical Architecture Framework For Information Management (TAFIM), which was developed from the IEEE POSIX Guide to Open Systems.

[4] The X/Open organization and the Open Software Foundation (OSF) have merged to form The Open Group.

XAF provides a generic technical reference model from which a specific architecture or sets of architectures can be defined. It is a functional definition, and in that way is similar to the Open Blueprint structure. However, a significant difference is that at the XAF level, no relationships between the functions have been defined. In the Open Blueprint structure, relationships have been defined, and technologies have been chosen, which makes the Open Blueprint more specific than XAF. In fact, the XAF document has an entire chapter devoted to the process of using the generic XAF structure to make various choices to tailor an architecture to the needs of a specific environment. The Open Blueprint structure could be thought of as the result of IBM's following that process.

Summary

Because of the Internet and its World Wide Web, the window of opportunity for moving products to market is shrinking rapidly. Companies must be able to exploit their IT infrastructures to access and, as necessary, remix their IT resources.

Many customers have turned to IBM and its Open Blueprint structure for help in creating their own infrastructure and transitioning to network computing. The Open Blueprint is a structure that identifies the necessary building blocks for open, distributed, network computing and that helps 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 structure defines a consistent way to manage the network to hide the complexities from application developers and end users.

Description of the Open Blueprint Structure

The Open Blueprint structure provides support for the execution, development, and management of distributed, network applications. The resource manager is the principal structuring element of the Open Blueprint structure. 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 structure 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 structure can be considered part of the application solutions they support. These services are reused by enterprise business processes.

Distributed, network 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 part and one or more server parts:

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

Applications request multiple system services to accomplish their functions. Applications can take advantage of the Open Blueprint structure 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 that are supported by both procedural and object-oriented interfaces. 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 parts 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.

Applications can be designed to support communications, collaboration, and cooperation among groups of users. This may range from situations where information is distributed in an ad hoc fashion to more highly structured processes where the steps are predefined and deterministic. The resultant applications may be passive, in that they leave control in the hands of the user or workgroup, or active, in that they play a more proactive or directive role by controlling the flow of group work.

Many industrial strength programs provided by users, vendors and IBM that are often termed applications are, structurally, 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 4 on page 18, 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.

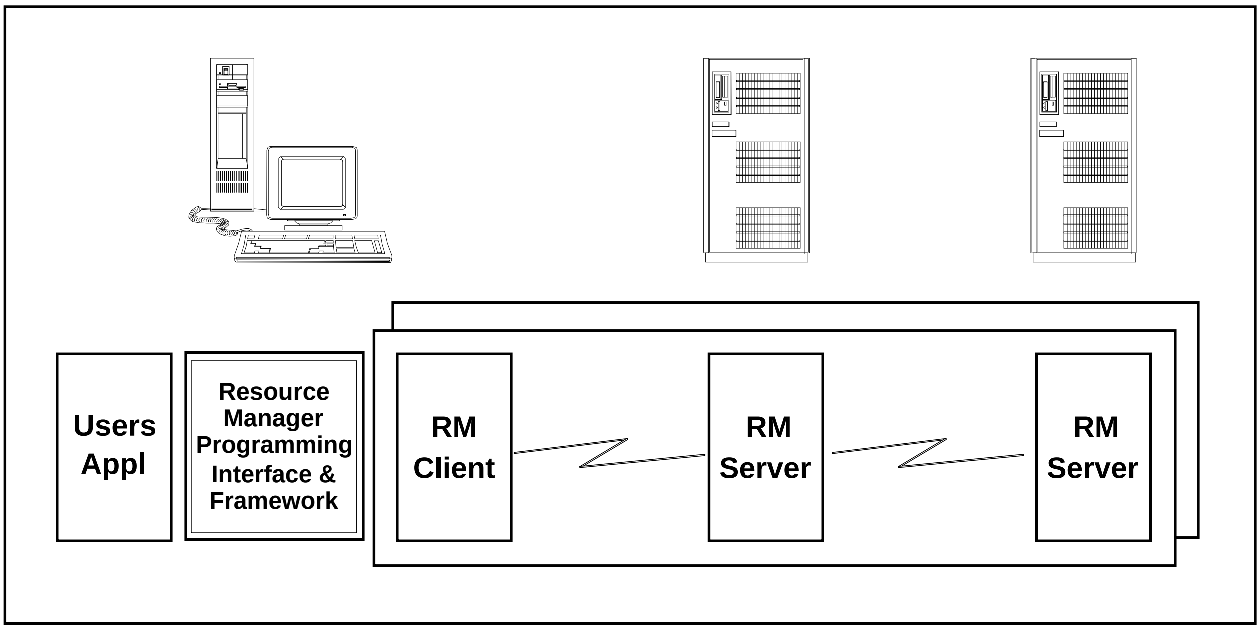

Figure 4. The Distributed Resource Manager Structure

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 Open Blueprint services.

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 structure, 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 CORBAservices: Common Object Services Specification.

A client system can be fixed or mobile. Depending on application support requirements, systems can be configured to contain primarily the client parts of distributed resource managers. This would be a typical configuration for an end user workstation. As an alternative, a client system could contain only the Web

Browser and Virtual Machine resource managers and their required communication support. Even smaller systems, such as personal digital assistants (PDAs), could be configured to 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, Open Blueprint components 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 (interoperate)

The interfaces and protocols specified in the Open Blueprint structure represent an attempt to strike a 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 structure:

- 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 structure 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 structure. 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, “Standards” on page 45 for a list of key documented standards for interfaces and protocols.

Open Blueprint Components

This section describes the resource managers in each of the component services of the Open Blueprint structure, as shown in Figure 1 on page 9:

- 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 1970s and early 1980s, 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 23) 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 structure.

As shown in Figure 5, Network Services consists of Common Transport Semantics, Transport Services, Subnetworking, and the Signalling and Control Plane.

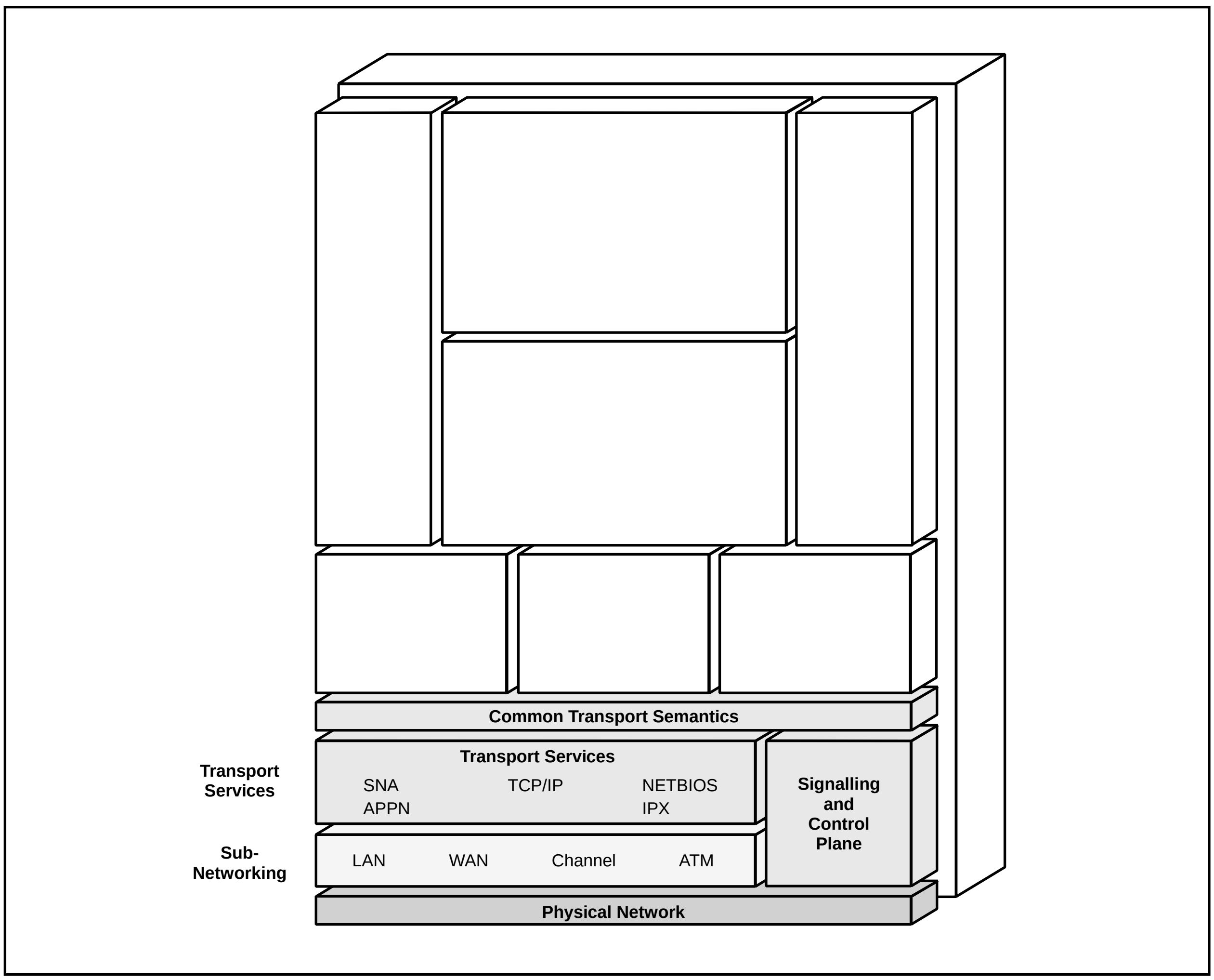

Figure 5. Network Services

Common Transport Semantics: Common Transport Semantics (CTS) insulates applications and Open Blueprint higher level services from 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 structure, 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 adopted specifications for MPTN based upon IBM's architecture.

With AnyNet family 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.

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

- Transmission Control Protocol/Internet Protocol (TCP/IP)
- Systems Network Architecture/Advanced Peer-to-Peer Networking (SNA/APPN)
- 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
- Asynchronous Transfer Mode (ATM)

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

Asynchronous Transfer Mode 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.

Subnetworking 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), cell relay technologies, 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 6 on page 23).

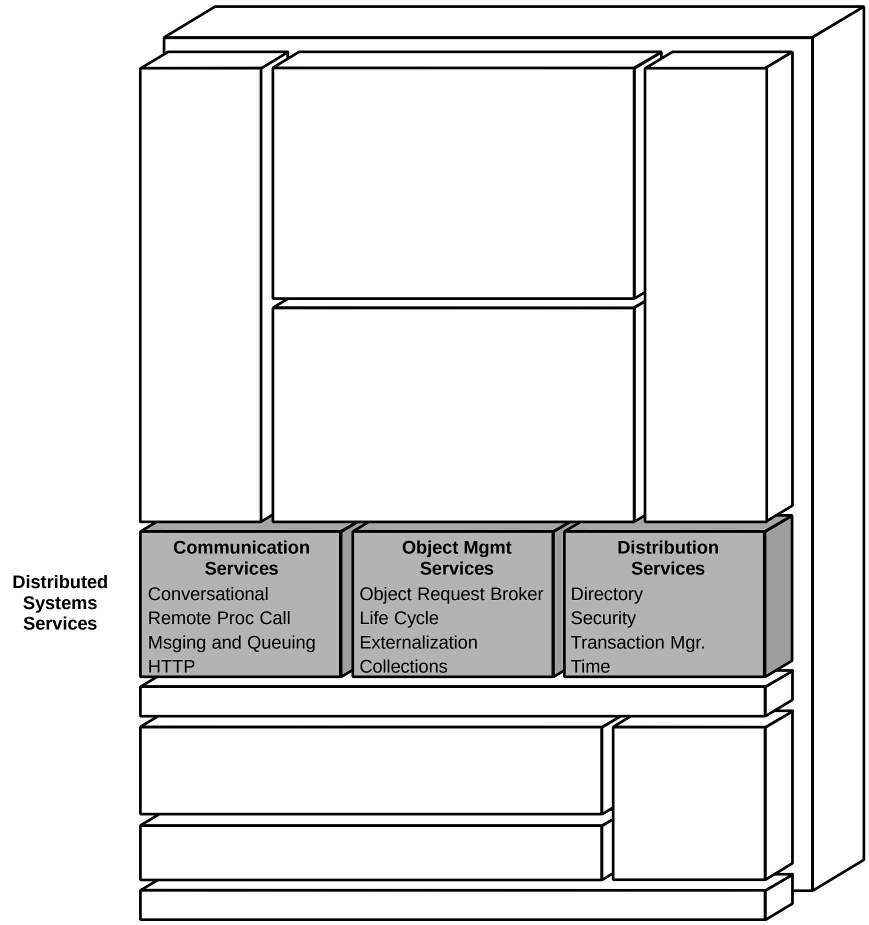

Figure 6. Distributed Systems Services

Communication Services: The Open Blueprint structure supports four 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 X/Open's Common Programming Interface for Communications (CPI-C). 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, OS/390, 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 application to handle the message. The assured delivery allows work to be committed by the sender without waiting for the receiving application to complete, knowing that the message will be delivered despite system or network failures.

The Message Queue Interface (MQI) is in the MQSeries of products on the following platforms: OS/2, Windows, Windows NT, OS/400, OS/390, VSE/ESA, Digital VMS VAX, AIX, Tandem NonStop Kernel, HP-UX, SunOS, Sun Solaris, SINIX, UnixWare, AT&T GIS UNIX, Pyramid DC/OSx, and SCO UNIX.

Messaging and Queuing is particularly suited to:

- High volume, networked transaction applications
- Intermittently connected networks such as mobile computers
- Heterogeneous networks with multiple protocols or machine architectures
- Networks spanning time zones in which network applications are available at different times
- Access to enterprise application from the World Wide Web

HTTP: In this model, one part sends a request for service to another through a structured message defined by the Hypertext Transfer Protocol (HTTP) specification. The HTTP resource manager supports the Common Gateway Interface (CGI) which provides access to a wide variety of function and content in the distributed system.

HTTP-based communication is fundamental to the growth of the Internet and World Wide Web. IBM has implemented HTTP servers on OS/2, AIX/6000, OS/400, and OS/390.

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 6 on page 23, Object Management Services include the Object Request Broker and the Life Cycle, Externalization, and Collections basic object services. Communications between objects can take place within a process (address space), between two processes in the same system, or between two processes running in separate systems. Objects in different processes use the Object Request Broker (ORB) to communicate. Objects written in different languages that are executing within a process use the IBM System Object Model (SOM).

Object Request Broker: The Open Blueprint Object Request Broker (also known as Distributed SOM) supports inter-process object communication. The ORB enables object distribution by brokering method invocations to remote objects across processes or machine boundaries. The ORB supports a language neutral object model. The ORB enables object method invocations across different language environments. The ORB enables release-to-release binary compatibility so that elements of an application can be updated independently. Using the ORB, objects can be accessed independent of their location, across processes or across distributed systems.

The Object Request Broker resource manager is an implementation of the Object Management Group (OMG) Common Object Request Broker Architecture (CORBA). CORBA is a standard model defined by OMG and published by X/Open. CORBA defines the ORB, Common Object Services, and the Common Facilities required for supporting objects in a distributed environment. The ORB utilizes many services in the Open Blueprint structure, including security, directory, and communication services, to accomplish its function.

System Object Model (SOM): When communication between objects developed using different source languages is required to take place within a process, SOM provides the object request broker function. SOM supports a language neutral object model. It enables in process object method invocations across different language environments and enables release-to-release binary compatibility within a process. SOM supports C++, Java, and Smalltalk. Other languages are to be added in the future.

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: 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. This service provides a means through which objects can be packaged and transported.

Collections provides basic capabilities for bundling a collection of objects such as sets, queues, and lists.

Distribution Services: Distribution services enable a single system view of the network. See Figure 6 on page 23.

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

DCE functions are supported on AIX/6000, OS/2, OS/390, OS/400, and other platforms. DCE is also supported by DEC, HP, Sun, Transarc, and many others.

Naming and Directory: Naming provides the facilities required to identify and locate 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 structure includes a federated namespace based on X/Open Federated Naming concepts and the Internet Engineering Task Force (IETF) Lightweight Directory Access Protocol (LDAP). 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 Open Blueprint Directory resource manager supports this federated namespace. Multiple namespaces can be implemented by different directory services. The Open Blueprint Directory resource manager gives distributed system users a central place to store information, which can be spread across many systems and retrieved from anywhere.

The Open Blueprint Directory resource manager provides two classes of general directory services, the global directory service, which may be either an X.500 or an Internet Domain Name System (DNS), and an enterprise directory service, such as the OSF DCE Cell Directory Service (CDS), the Lotus Notes Name and Address Book, and the LDAP directory service. The LDAP is used to form the federation across all supported directory service implementations.

Security: In earlier centralized systems, the local operating system authenticated the users' identities and authorized access to resources. In a distributed environment, individual workstations in a network are not necessarily secure. Therefore, 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 services let administrators register users and resource managers, provide for the mutual authentication of clients and servers, and enable resource managers to provide access to resources only to authorized users. The Open Blueprint security services also defines services for auditing user activity. These services include both secret key (symmetric key) and public key (asymmetric key) technologies.

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, maintains knowledge of the time of day and synchronizes system clocks in a network to provide time services with a limited, but known degree of accuracy.

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 structure 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 CORBAservices Object Transaction Service interfaces.

Application Enabling Services

As shown in Figure 7 on page 28, Application Enabling Services consist of Presentation Services, Application/Workgroup Services and Data Access Services.

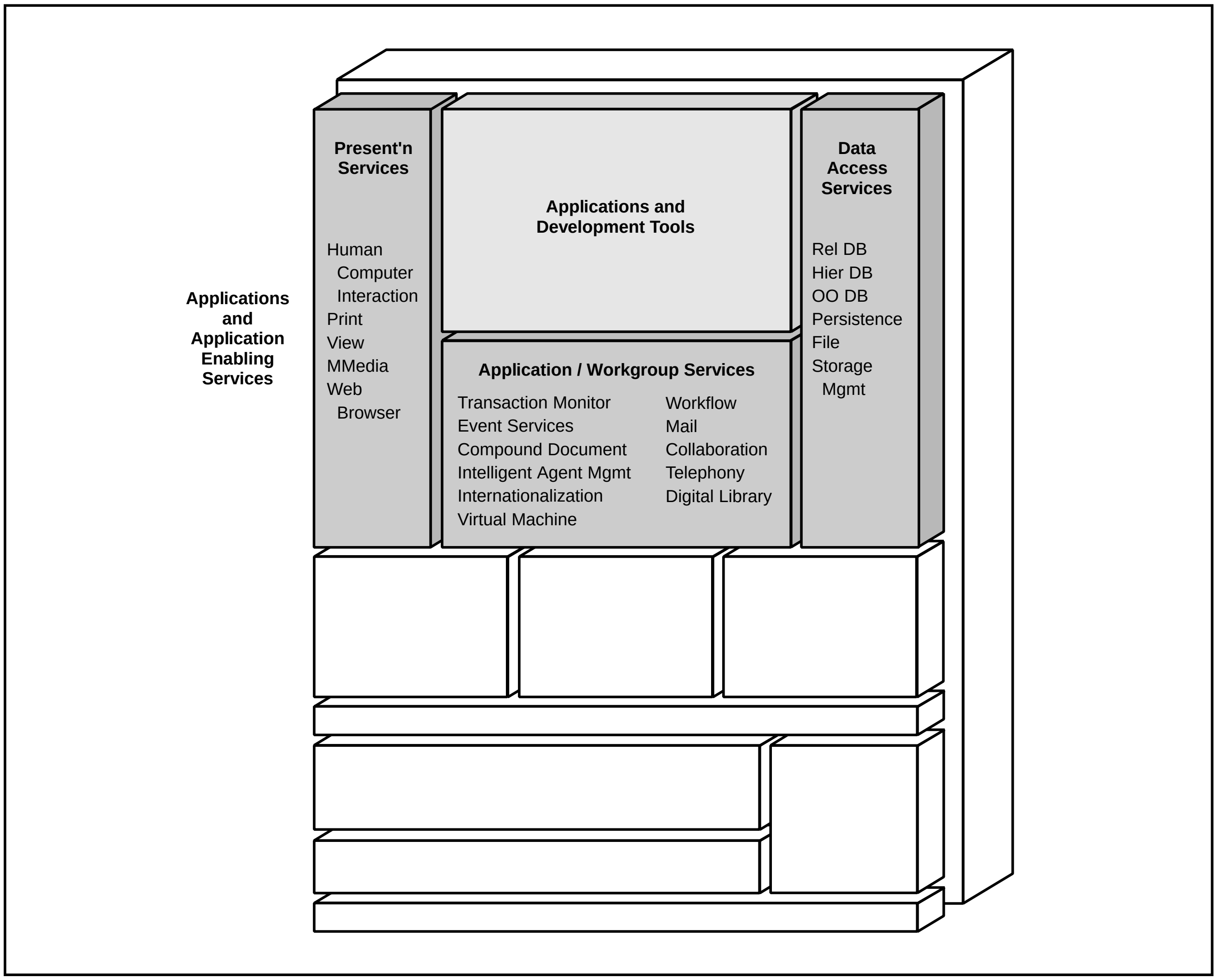

Figure 7. Application Enabling Services

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

Human Computer Interaction: The Open Blueprint Human Computer Interaction resource manager, with its associated technologies, 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 **human computer interaction** describes the front of screen appearance and function of an application or system, along with the mechanisms for the interaction between the human and the computer 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 visual space is a desktop. The user's 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 95 Desktop, 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 enables this application software cooperation that appears seamless to the end user.

The key compound document technology in the Open Blueprint structure 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 natural computing technologies, such as pen, speech, agents, handwriting recognition, and virtual reality, make it possible to evolve from the traditional desktop into something even more powerful and intuitive. The messy desk metaphor is being superseded by specialized user environments, called places, targeted for performing specific tasks in the office, the factory, the home, the retail store, and “on the road” when 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. Multiple users can occupy a place at the same time. Actually, places are compound document containers and contain things that are compound document parts.

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. It 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. The end user can 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 such as queues, jobs, and printers. It can modify job priorities, monitor the print network, and add, delete or reallocate print resources.

View: The Open Blueprint View resource manager provides the ability on the workstation to view printable documents based on 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 retrieve and present, or capture and store, distributed graphics, audio, video, and animation data. Because multimedia data is digitized, it can also be searched and manipulated. Multimedia applications can capitalize on the interactivity of a computer and the ability to emulate human modes of communication to make the computer easier to use. In the Open Blueprint structure 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 guaranteed quality of service.

Multimedia tools and applications can include capture and creation, composition and editing, and the interactive presentation of live and stored multimedia data. Multimedia content may be stored in a digital library to facilitate sharing and reuse. Multimedia services can be used to extend line of business applications or in new applications such as distance learning.

Web Browser: Web browsers have become the client interface of choice for Internet computing. As the user's gateway into the World Wide Web, Web browsers provide a client based end user environment for information delivery and interaction with Web servers. The Open Blueprint Web Browser resource manager provides the user with a point and click means to browse and navigate Web content without overt knowledge of the underlying servers accessed. Because the Web Browser resource manager can provide distribution presentation support for a wide variety of server based applications, it provides the capabilities of a “thin client.”

Application/Workgroup Services: Application/Workgroup Services provide high level application or workgroup oriented functions. See Figure 7 on page 28.

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 to simplify the 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 including HP-UX, Sun Solaris, and Digital Unix. The IMS transaction monitor API has been implemented on a variety of platforms supporting applications associated with OS/390 systems. The Encina transaction monitor API has been implemented across a range of UNIX platforms and Windows NT.

Event Services: Event Services allow an application to send information representing the occurrence of an event to another program. This service separates the communication between the program generating the event and the program reacting to the event. These programs can be on the same node or on different nodes 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 “Human Computer Interaction” on page 28, which describes the impact of compound documents on the interaction between people and systems. 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, data interchange, presentation, and interaction. 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 CommonPoint initiative.

Intelligent Agent: Intelligent agents are software entities that carry out some set of operations on behalf of a user or another program with some degree of autonomy and, in so doing, employ some knowledge or representation of the user's goals and desires. Intelligent agents can help cope with the flood of information available in the network computing environment and can help simplify user interfaces.

The Intelligent Agent resource manager provides functions and facilities that an intelligent agent uses to do its job. Object-oriented technology is the base for the following sub-components of the resource manager:

- Adapters, the “eyes, ears, and hands” of the intelligent agent
- Engines, the “brains” of an intelligent agent
- Knowledge, used by the engines when they operate
- Libraries, knowledge persistent across agent activations
- Views, tools for the user to interact with the intelligent agent

Specific intelligent agents are composed of instances of these resource manager sub-components.

Internationalization: For an application to meet the specific needs and preferences of a variety of users in different countries, it needs to be internationalized at build time for general worldwide use, and it needs to be localized for a particular language, culture, and character data encoding at run time. This allows any user, at any location, to use that application and have it provide culturally correct processing results.

The description of the language, cultural, and character data encoding needs of users in a particular environment is contained in a data structure called a locale. The Open Blueprint Internationalization resource manager supports the definition, management, and usage of the information contained in locale data structures.

The programming model and interfaces of the Internationalization resource manager have been defined through the Institute of Electrical Engineers (IEEE) Portable Operating System Interfaces (POSIX), X/Open Joint Internationalization Group (XoJIG) and UniForum development efforts, and the ANSI/ISO C programming language community. In addition to this work, IBM extensions are provided to the standard base support.

Virtual Machine: The Open Blueprint Virtual Machine resource manager implements the industry standard, object-oriented Java execution environment defined by the JavaSoft division of Sun Microsystems, Inc. It provides a platform independent execution environment for Java programs such that the same binary programs can execute on all Java-compliant hardware and software configurations. Developers can create applications and content with the same tools, in the same language, and using a consistent programming model on both client and server; the resulting binary programs can execute on any Java-compliant virtual machine. The implementation is provided primarily by a mapping onto the other resource managers and services of an Open Blueprint distributed system.

The Virtual Machine resource manager emphasizes integrated support for networking. It supports World Wide Web technology through the use of the HTTP and Web Browser resource managers and HTML.

The Virtual Machine resource manager implements a distribution paradigm where a program to be executed is generally fetched on-demand from a server elsewhere on the network.

Workflow: Workflow management provides a set of functions that help customers define, execute, manage, and reengineer 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 by separating the process definitions from the application performing the process. 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.

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 among workflow services.

Mail: Electronic mail is supported as a mechanism for users to exchange electronic mail and as a high level service that other applications can use to send all types of materials to end users or other application 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 Mail resource manager 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
- Interorganization business communication and transactions, including EDI
- Group scheduling and calendar systems
- Electronic publishing and distribution
- Electronic forms routing and workflow applications

There are several open industry standard APIs that have become popular for application access to electronic mail services. Those included in the Open Blueprint structure are 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 multiplatform 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 output of the X.400 API Association (XAPIA). It provides access to 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. TCP/IP provides Simple Mail Transfer Protocol (SMTP), which supports the TCP/IP SENDMAIL application. Open Systems Interconnect (OSI) standards define the X.400 Message Handling Service protocol. IETF defines the Post Office Protocol Version 3 (POP3). 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 resource manager supports standard SMTP, X.400, POP3, and SNADS protocols. Access to other important proprietary message transfer protocols is through gateways.

Collaboration: Collaboration is the foundation of groupware, which is software that supports teamwork and cooperation among people.

The Open Blueprint Collaboration resource manager provides the basic functions that support collaborative activities. Collaboration relies on a:

- Shared space, which can be a virtual room, a white board, or shared on line space
- Shared information, which is the input to and an important work product of a collaborative process

Collaboration can be between two people, or it can take the form of many-to-many information sharing.

The Collaboration resource manager support organizes shared information using a paper document metaphor and places the shared information in a collaboration document store.

Both synchronous and asynchronous collaboration are supported.

Synchronous collaboration is the most common type of collaboration, where activity occurs during a single period of time although the participants can be in different locations. Examples of synchronous collaboration are audio/video conferencing and face to face meetings.

Asynchronous collaboration removes all constraints on time and location. Many copies of the collaboration document store can be spread over many locations, and the only time limit is the completion date. Asynchronous collaboration enables, for example, a document produced in the United Kingdom to be reviewed in Australia after the United Kingdom users have left work for the day. The United Kingdom users can pick up the completed reviews after they arrive at work the next morning.

The Collaboration resource manager supports sharing information that is contained in collaboration document stores. Participants can browse the information in a document store and search for a particular document or set of documents. Collaboration enables participants to edit the documents and make the updates available to other participants. Alternatively additional information or reviewers' comments can be attached to the original so the original document is preserved.

Replication makes all copies of a collaboration document store essentially identical over time. If a user makes changes in one copy of the collaboration document store, replication ensures that those changes are added to all replicas. Replication allows collaboration document stores to be distributed to more than one location and kept synchronized.

The Collaboration resource manager uses the replication infrastructure to support mobile clients. Both the mail and collaboration document stores are supported. Replication enables mobile caching and reconciliation of data, that is the ability to make a replica copy of the collaboration document store on a workstation and enable the mobile user to work off line on the replica. If changes are made to the local collaboration document stores, when the user connects again to the network and carries out the reconciliation function, changes are replicated back to the server (and updates are sent to the workstation).

Audio/video conferencing is a form of synchronous collaboration. In the Collaboration resource manager, audio/video conferencing is built on the Intel ProShare architecture base, which is rapidly becoming a standard architecture for audio/video conferencing activities. The ProShare architecture defines how conference connections are established among desktop systems. Once the connections are established, individuals can participate in an audio/video conference and can share data such as files and images within a shared document store.

The Collaboration resource manager's audio/video conferencing capabilities enable users to take advantage of the collaboration document store, information access, management, and security functions to enhance audio/video conferencing. Specific conferencing facilities include maintaining conference address lists, the journaling of conference sessions, and the logging of conference activity. In addition application programming interfaces (APIs) and conference enabled messaging are supported.

The Collaboration resource manager also supports calendaring and scheduling. This support includes setting up appointments, scheduling and notification of meetings, confirmation handling, and room reservations. Additional functions include free time searches, repeating appointments, and conflict warnings. In the collaborative process, this feature enables all users to participate in organizing meetings and related collaborative activities. Users can view the schedules of other users, enabling them to set their own schedules accordingly.

The Collaboration resource manager works with the Relational Database and HTTP resource managers to allow access to relational and World Wide Web data through the document metaphor.

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).

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

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 prerecorded message 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 or place outgoing calls. Through the use of voice related technologies such as speech recognition and synthesis, it can transform voice to text and text to voice.

Internet voice allows communication over the Internet using the voice and sound features of the workstation. New enhancements to the Internet protocols allow the voice packets to be prioritized, to travel expediently over the network, and to arrive at their destination in a timely manner that preserves the fidelity of the voice. Internet voice applications provide user interfaces and access to directory services to allow access to the desired party.

Digital Library: The Digital Library resource manager supports the storage and distribution of information objects. It provides functions 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[5] 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.

The Digital Library resource manager includes support for intellectual property rights management that is built on the Open Blueprint security functions for identification and authentication. Technology that protects intellectual property from misuse or misappropriation, such as watermarking, is also provided.

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 Services
- File
- Storage Management

IBM's Information Warehouse solutions provide 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.

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 in a relational database. The ISO standard for embedded SQL is SQL92 and the level supported by the Open Blueprint for interoperability is SQL92E (entry level). The functions of SQL are available through the DB2 family of products on all major IBM platforms and on HP-UX, Solaris, Windows NT, Windows 95, SINIX, and SCO Openserver.

In addition to the embedded form of SQL defined by ISO SQL, the Open Blueprint Relational Database resource manager also 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 key database and connectivity vendors, several of whom have DRDA implementations available today on IBM and non-IBM platforms.

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.

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

Object-Oriented Database: Object-oriented database facilities represent an emerging technology with no 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 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 object-oriented database or a traditional data store.

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

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

File APIs are supported by a file system 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 structure 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. DFS supports a wide variety of semantics including byte stream requests, client caching, server to server caching, and file system administration.

Storage Management: The Open Blueprint 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 Open Blueprint 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 systems

- Long term file or named binary object preservation through data archiving
- Cost effective storage 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 either 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 structure includes a backplane element that allows for resources to be managed across open, multivendor, enterprise-wide, distributed systems (see Figure 8 on page 38). That structure is based on Tivoli Systems TME 10. TME 10 defines the necessary management applications, management services, and structure 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. TME 10 and the Open Blueprint structure address complementary aspects of capabilities needed to address a multivendor enterprise environment.

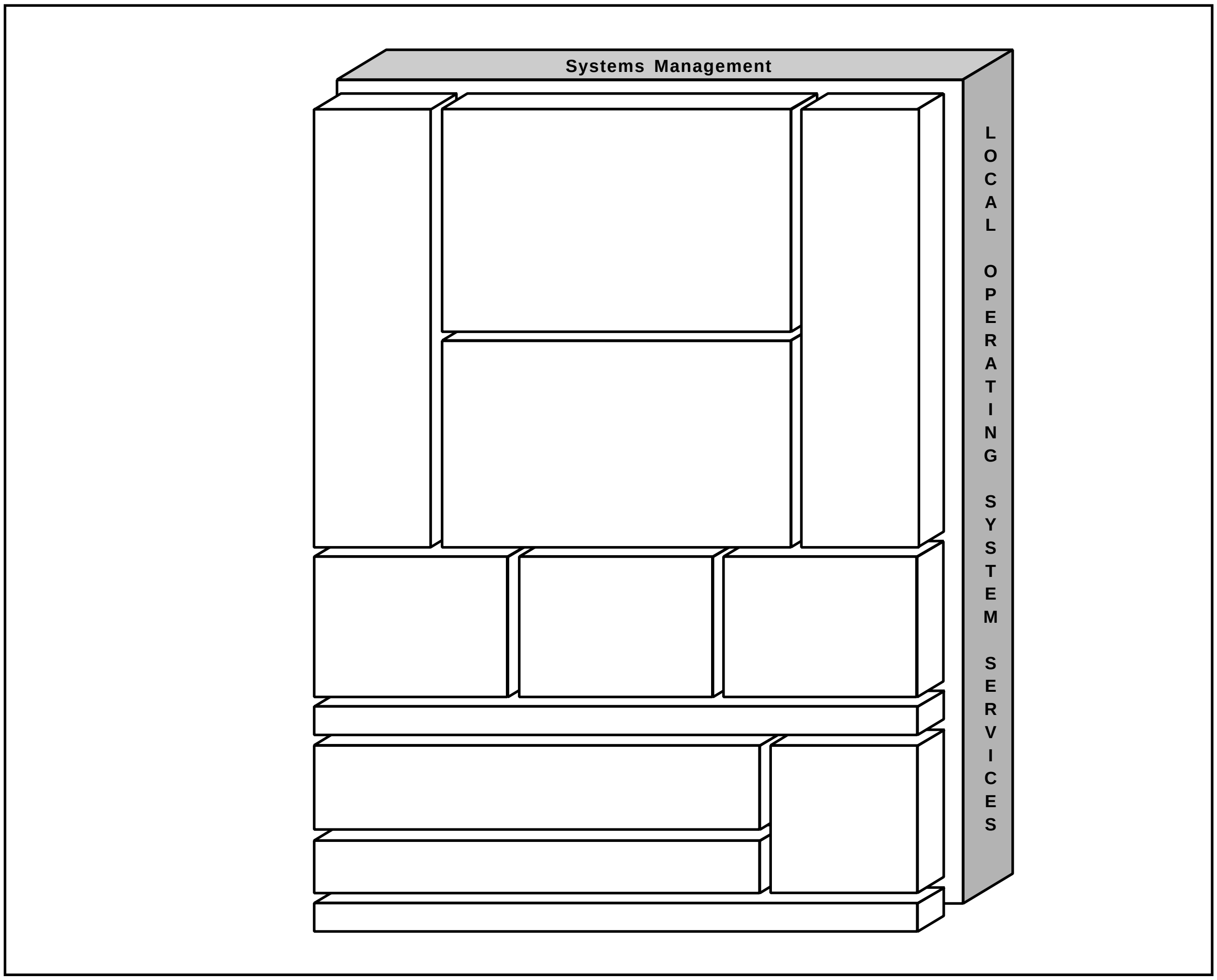

Figure 8. Systems Management and Local Operating System Services

Four major elements are included in the systems management backplane.

Systems Management Views provide common access to systems management end user data. Views provide a common way to see and handle sets of systems management data.

Systems Management Applications provide functions to interact directly with systems administrators and automate management tasks. Management applications are categorized into four disciplines:

- Deployment Management supports tasks associated with the configuration and change management activities for all of the frequently changing components of the enterprise.
- Availability Management gathers, collects, and routes information regarding the state of all aspects of the enterprise computing environment for proactive management action.
- Security Management ensures that users have access to the applications and data they need to do their jobs. Security management applications provide administrative support for Open Blueprint Security resource managers.
- Operations and Administration facilitates the automation of activities that provide the integrity and reliability of the entire computing environment. Operations and administration applications also support the administration of resource managers and their resources.

Systems Management Services provide common services for resource managers and System Management Applications unique to the needs of systems management. Through the use of these services, Systems Management Applications can have an integrated appearance, share data among applications, and be shielded from the underlying technology used to access management information. The Systems Management Services can be placed in three categories:

- Framework Services provide basic object specification, creation, interaction, and method invocation. Framework Services are provided in the implementation of the CORBA-compliant Object Request Broker and Basic Object Adapter (BOA).
- Application Services provide the fundamental services for modeling or storing objects for management purposes. These services define the set of intrinsic operations that all policy driven objects can inherit and implement.
- User Interface Services offer a set of platform and presentation independent user interface services that allow simplification of application development.

Managed Resources are resources owned by resource managers. They are available for interrogation and control by Systems Management Applications.

Agent Services: Agent services are used by the resource managers to access and set the state of their managed resources. Agent services insulate resource managers from awareness of the location of the managing system. They include the ability to:

- Send notifications
- Collect vital product data
- Respond to operational commands and requests for state information
- Collect performance information
- Enforce software license policies
- Capture error and problem determination information
- Request distribution of software

Systems Management Enablement of Applications and resource managers: The TME 10 Application Management Specification (AMS) is the foundation for systems management enablement of resource managers and distributed applications. To effectively manage an application, the management application needs information on how it is structured and on how to control and monitor it. The AMS provides a format and content for Application Description Files (ADFs) which allow the application developer to describe standard management information.

Systems Management Standards: The systems management component structure supports the following current industry standards:

- IETF

  TME 10 supports the Simple Network Management Protocol (SNMP).

- OMG

  The Common Object Request Broker Architecture (CORBA) allows interoperability among independently developed object-oriented applications across heterogeneous computer networks.

- X/Open and OMG

  The systems management services in TME 10 incorporate the Common Management Facilities (XCMF) standardized by X/Open and OMG. These facilities include a naming service, event service, and life cycle service.

- Desktop Management Task Force (DMTF)

  The Application Management Specification is based on the software MIF as defined by DMTF.

- POSIX

  User administration is compliant with the POSIX 1387.3 definition for user groups.

In addition to the current standards, there are several emerging standards being incorporated into the systems management component structure:

- IETF SNMP V2
- Internet standards

Local Operating System Services

Local operating system services are local resource managers that support Open Blueprint distributed resource managers. They manage local system resources such as CPUs and devices. Local operating system services include such functions as work management, environment state support, event handling, local system logon, security context management, multimedia system services, locking, accounting, tracing, journaling, and program management.

Standard interfaces to local operating system services maximize portability of code across platforms, including standards from POSIX and X/Open's Portability Guide.

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 structure is to make it easier for developers to build on Open Blueprint services 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 structure supports programming development by providing:

- Resource manager definition
- Standards adherence
- Integration objectives and criteria (through the use of other resource managers and common services)

Many products today implement the standards defined by the Open Blueprint structure; future products will also implement them. A major Open Blueprint goal 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 structure promotes the integration of a broad range of client/server and network computing products under a single, network-wide signon process, a federated directory for locating distributed resources, and common services and graphical user interface for administration. Through the use of a broad set of industry accepted technologies for the underlying directory and security services, the Open Blueprint structure also promotes interoperability with other vendors' products.

Summary

Figure 9 shows all of the components of the Open Blueprint structure.

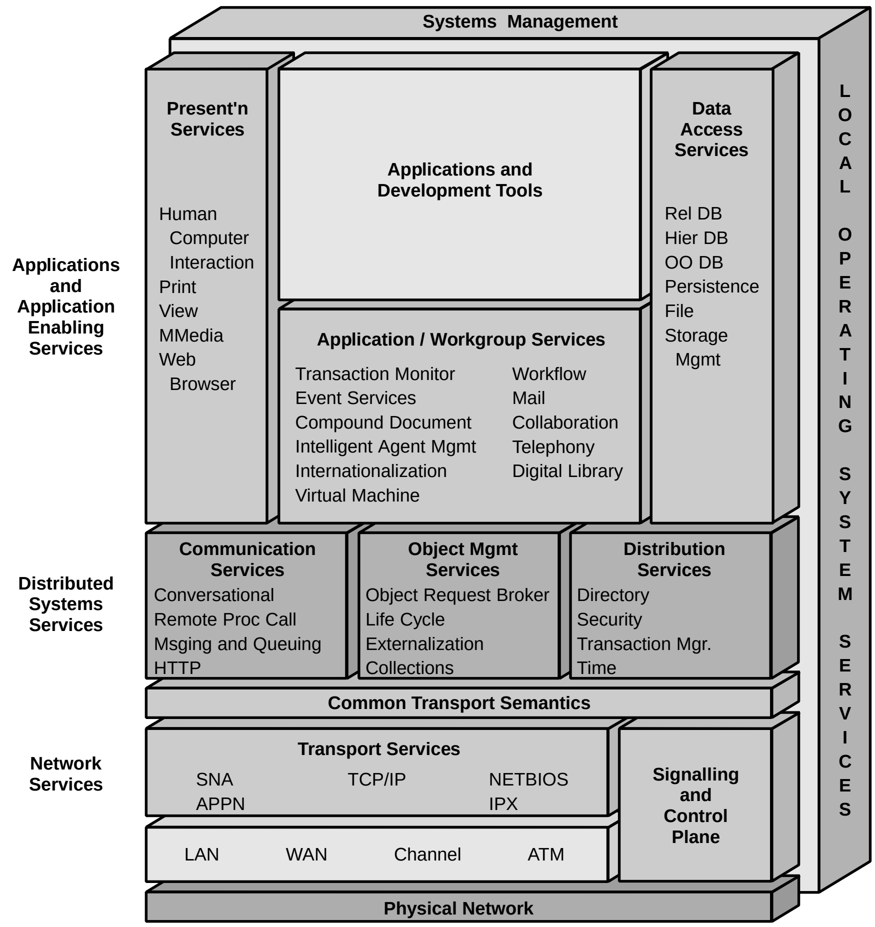

Figure 9. Open Blueprint Components

The Open Blueprint structure is IBM's view of a standards based, modular enterprise architecture. While the Open Blueprint structure 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*)
- “One of the things that IBM I think had discovered is that openness only works if you structure it.”(Kevin Kelly, Manager of Technology Architecture, Royal Bank of Canada)

Even though IBM created the Open Blueprint structure originally for its own use, many other organizations are using it as a reference model to create their own internal IT architectures, as an analytical tool for prioritizing and planning, or as a checklist for developing or acquiring software.

Examples of how the Open Blueprint structure is being used are available on the Open Blueprint home page, URL http://www.software.ibm.com/openblue.

The Open Blueprint structure 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. In turn, it will evolve as new technologies are incorporated.

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

- 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 structure 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 structure is measured by the products and solutions it enables and their contribution to customers' efforts to compete and grow in the dynamic, network computing environment of today and tomorrow. And, according to the Datapro report titled *IBM Open Blueprint Architecture*, if the Open Blueprint "delivers on its promise,... it will provide a 'single system image' for developers, administrators, and end users for the next 30 years."

Appendix. Standards

The following table lists some principal industry interfaces and protocols in the Open Blueprint structure. The Open Blueprint structure is based on formal standards that have achieved widespread industry acceptance. The Open Blueprint structure also includes several *de facto* standards, either in the absence of a formal standard, or to achieve interoperability with existing installations.

The references for ISO/IEC, IETF, OMG, and ITU-T standards are to standards number. The references for X/Open are to an International Standard Book Number (ISBN). The references for company documented interfaces and protocols are denoted by an asterisk (*). You can get information on these interfaces and protocols from the contributing company. The IBM documents are listed below. For convenience, URLs are provided to enable access to the various organizations' Web pages.

Although X/Open and OSF have combined to form The Open Group, the X/Open and OSF references are used.

IBM Interfaces	Document Order Number	IBM Protocols	Document Order Number
CICS API	SC33-1007	APPC	SC31-6808
DL/I	SC26-8015	APPN	SC30-3422
IMS TM API	SC26-8017	DDM	SC21-9526
MQI	SC33-0850	DRDA	SC26-4651
OS/2 PM API	G25H-7103/4	IPDS	S544-3417
		MPTN	GC31-7073/4
		NetBIOS	SC30-3587
		SNA Sync Point	SC31-8134
		SNA Formats	GA27-3136

The referenced standards documents can be located and ordered at the following web sites:

Organization	URL
The ATM Forum	http://www.atmforum.com
Component Integration Laboratories (CILabs)	http://www.cil.org
The Desktop Management Task Force (DMTF)	http://www.dmtf.org
ECMA	http://www.ecma.ch
International Electrotechnical Commission (IEC)	http://www.iec.ch
The Institute of Electrical and Electronic Engineers, Inc. (IEEE)	http://www.ieee.org
International Organization for Standardization (ISO)	http://www.iso.ch
International Telecommunication Union - Telecommunication (ITU-T)	http://www.itu.ch
Internet Engineering Task Force (IETF)	http://www.ietf.org
Open Software Foundation (OSF)	http://www.osf.org
Object Management Group (OMG)	http://www.omg.org
National Center for Supercomputing Applications (NCSA)	http://www.ncsa.uiuc.edu
Worldwide Web Consortium (W3C)	http://www.w3.org
X/Open Company Ltd (X/Open)	http://www.xopen.org

Network Services

Common Transport Semantics

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

Transport Services

Interfaces	Protocols	Organization	Reference
XTI		X/Open	ISBN 1-85912-049-0
Berkeley Sockets		Berkeley	4.3 BSD Document
Winsock		Microsoft	*
	TCP/IP	IETF	791, 793, 768, 1889
	IPSec	IETF	1825-1829
	SSL	IETF	draft-freier-ssl-version3-01
	RSVP	IETF	draft-ietf-rsvp-spec-13
	APPN	APPN Implementors Workshop	IBM - *
	NetBIOS	IBM	*
	IPX	Novell	*
	Appletalk	Apple	*

Signalling and Control Plane

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 4.0

Distributed Systems Services

Conversational

Interfaces	Protocols	Organization	Reference
CPI-C		X/Open	ISBN 1-85912-135-7
	APPC	IBM	*
	OSI TP	ISO/IEC	10026

RPC

Interfaces	Protocols	Organization	Reference
DCE RPC	DCE RPC	X/Open	ISBN 1-85912-041-5

Messaging and Queuing

Interfaces	Protocols	Organization	Reference
MQI		IBM	*

HTTP

Interfaces	Protocols	Organization	Reference
CGI 1.1		NCSA	http://hoohoo.ncsa.uiuc.edu/cgi/interface.html
	HTTP 1.0	IETF	1945
	HTTP 1.1	IETF	draft-ietf-http-v11-spec-07
	HTTPS	Netscape	*
	SHTTP	IETF	draft-ietf-wts-shttp-03

Object Management

Interfaces	Protocols	Organization	Reference
CORBA 1.1		OMG	CORBA 1.1 (12/91)
CORBA 2.0		OMG	CORBA 2.0 (07/95)
CORBAservices		OMG	CORBAservices
Collections			
Externalization			
Identity			
Life Cycle			
	IIOP, DCE-CIOP	OMG	CORBA 2.0 (07/95)

Directory

Interfaces	Protocols	Organization	Reference
XDS		X/Open	ISBN 1-85912-007-5
XOM		X/Open	ISBN 1-85912-008-3
Federated Naming		X/Open	ISBN 1-85912-045-8
LDAP		IETF	1823
Naming Service		OMG	CORBAservices
	Cell Directory Service	X/Open	ISBN 1-85912-078-4
	X.500	ISO/IEC	9594 (88 Standard)
	Domain Name Service	IETF	1034,1035
	LDAP	IETF	1777
	URL	IETF	1738

Security - Identification and Authentication

Interfaces	Protocols	Organization	Reference
DCE SEC_RGY	DCE Registry	OSF	DCE 1.2
GSSAPI		IETF	1508, 1509
Security Service		OMG	CORBAservices
	Kerberos	MIT/OSF	Kerberos 5 Beta 7

Security - Access Control

Interfaces	Protocols	Organization	Reference
DCE SEC_ACL	DCE ACLs	OSF	DCE 1.2 and POSIX 1003.1e, 1003.2c

Security - Certificate Management

Interfaces	Protocols	Organization	Reference
	Certificate Definition	ITU-T	X.509v3

Time

Interfaces	Protocols	Organization	Reference
DCE DTS	DCE DTS	X/Open	ISBN 1-85912-067-9
Time Service		OMG	CORBAservices

Transaction Manager

Interfaces	Protocols	Organization	Reference
Transactional C	Encina	Transarc	*
TX		X/Open	ISBN 1-85912-094-6
XA		X/Open	ISBN 1-872630-24-3
CICS API		IBM	*
IMS TM API		IBM	*
	SNA Sync Point	IBM	*
Object Transaction Service		OMG	CORBAservices

Application Enabling Services

Human Computer Interaction

Interfaces	Protocols	Organization	Reference
OS/2 PM		IBM	*
Motif		OSF	MOTIF Specification
Windows		Microsoft	*
OpenGL		Silicon Graphics	*
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

Print

Interfaces	Protocols	Organization	Reference
Distributed Print API		IEEE	POSIX 1387.4/D8
	DPA	ISO/IEC	10175-2
	lpr/lpd	IETF	1179
	NCP	Novell	*

Print Data Streams

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

Web Browser

Interfaces	Protocols	Organization	Reference
	HTML 2.0	IETF	1866
	HTML 3.2	W3C	http://www.w3.org/pub/WWW/TR/WD-html32

Relational Database

Interfaces	Protocols	Organization	Reference
SQL (1992E)		ISO/IEC	9075
SQL CLI		X/Open	ISBN 1-85912-081-4
ODBC		Microsoft	*
	DRDA	IBM	*

Hierarchical Database

Interfaces	Protocols	Organization	Reference
DL/I		IBM	*

Persistence Services

Interfaces	Protocols	Organization	Reference
Persistent Object Service		OMG	CORBAservices

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	*

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 Notification Service		OMG	CORBAservices

Internationalization

Interfaces	Protocols	Organization	Reference
Locale Services		X/Open	ISBN 1-85912-002-4

Compound Document

Interfaces	Protocols	Organization	Reference
OpenDoc		CILabs	*

Virtual Machine

Interfaces	Protocols	Organization	Reference
Virtual Machine Invocation		JavaSoft	*
Native Method Invocation		JavaSoft	*
Remote Method Invocation		JavaSoft	*
Debugging Tool Interfaces		JavaSoft	*
JIT Compilation Hooks		JavaSoft	*

Mail

Interfaces	Protocols	Organization	Reference
Vendor-Independent Messaging (VIM)		VIM Consortium	*
Messaging API (MAPI)		Microsoft	*
Common Messaging Call API		XAPIA	*
	SMTP	IETF	821,822
	X.400 Messaging Handling Services	ISO/IEC	10021
	POP3	IETF	1938
	SNADS	IBM	*
	MIME	IETF	1521,1522

Telephony

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

Systems Management

Interfaces	Protocols	Organization	Reference
XMP		X/Open	ISBN 1-872630-32-4
SNMP	SNMP	IETF	1157
SNMPv2	SNMPv2	IETF	1907
XCMF		X/Open	ISBN 1-85912-047-4
User Admin.		IEEE	POSIX 1387.3
	MIF	DMTF	*
	AMS	Tivoli	*

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 45.

Appendix B. Bibliography

Articles

"Riding the Real Trends in Technology," D. Kirkpatrick, *Fortune*, February 19, 1996.

"Client/Server: Can It Be Saved?," B. Caldwell, *InformationWeek*, April 8, 1996.

"Cyber Maxims of Esther Dyson," C. Dreifus, *New York Times Magazine*, July 6, 1996.

"Inside Microsoft," K. Rebello, *Business Week*, July 15, 1996.

"Customer Power," C. Wilder, *InformationWeek*, June 24, 1996.

"Migrating to Multitiering," T. Kaneshige, *Client/Server Computing*, June 1996.

"The Information Appliance," *Business Week*, June 24, 1996.

"Plan Your Enterprise Architecture," P. Cottey and R. Chang, *InformationWeek*, June 24, 1996.

"Architecture With The Works," *InformationWeek*, July 1, 1996.

"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

The Digital Economy: Promise and Peril in the Age of Networked Intelligence, D. Tapscott, McGraw-Hill, 1996.

Doing IT Right, H. Lorin, Manning, 1996.

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

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

The Essential Distributed Objects Survival Guide, Robert Orfali, Dan Harkey, and Jeri Edwards, John Wiley & Sons, Inc., 1996.

Other IBM Open Blueprint Materials

Open Blueprint Technical Overview, GC23-3808.

Open Blueprint Technical Reference Library, SBOF-8702 (Hard Copy), SK2T-2478 (CDROM). This publication includes this *Introduction* and the *Open Blueprint Technical Overview* listed above as well as Component Description Papers for each Open Blueprint component. Each of the Component Description Papers is also individually orderable in hard copy. The table below provides ordering information.

Application Development in the Open Blueprint	GC23-3909
Asynchronous Transfer Mode and the Signalling and Control Plane in the Open Blueprint	GC23-3910
Open Blueprint Collaboration Resource Manager	GC23-3912
Open Blueprint Compound Document Resource Manager	GC23-3913
Open Blueprint Conversational Resource Manager	GC23-3914
Open Blueprint Digital Library Resource Manager	GN62-0039
Open Blueprint Directory Resource Manager	GC23-3915
Open Blueprint Event Services Resource Manager	GC23-3916
Open Blueprint File Resource Manager	GC23-3917
Open Blueprint Hierarchical Database Resource Manager	GC23-3918
Open Blueprint HTTP Resource Manager	G325-6593
Open Blueprint Human Computer Interaction Resource Manager	GC23-3933
Open Blueprint Intelligent Agent Resource Manager	G325-6592
Open Blueprint Internationalization Resource Manager	G325-6583
Open Blueprint Mail Resource Manager	GC23-3919
Open Blueprint Messaging and Queuing Resource Manager	GC23-3920
Open Blueprint Multimedia Resource Manager	GC23-3921
Open Blueprint Network Services	GC23-3932
Open Blueprint Object Oriented Database Resource Manager	GC32-3922
Open Blueprint Object Broker Resource Manager	GC28-1213
Open Blueprint Persistence Services Resource Manager	GC23-3923
Open Blueprint Print Resource Manager	GC23-3924
Open Blueprint Relational Database Resource Manager	GC23-3925
Open Blueprint Remote Procedure Call Resource Manager	GC23-3926
Open Blueprint Storage Management Resource Manager	GC23-3927
Open Blueprint Telephony Resource Manager	GC23-3928
Open Blueprint Time Resource Manager	GC23-3929
Open Blueprint Transaction Manager Resource Manager	GC23-3930
Open Blueprint Transaction Monitor Resource Manager	GC23-3931
Open Blueprint View Resource Manager	GC23-3934
Open Blueprint Virtual Machine Resource Manager	GC25-6588
Open Blueprint Web Browser Resource Manager	G325-6589
Open Blueprint Workflow Resource Manager	GC23-3935
Security in the Open Blueprint	GC23-3936
Systems Management in the Open Blueprint	GC23-3937

Open Blueprint Support for Mobile Computing is available in softcopy at URL:
http://www.software.ibm.com/openblue/id1u2/cover.htm

The Open Blueprint home page, located at http://www.software.ibm.com/openblue, contains information related to the Open Blueprint. It is dynamic, with new information being added as it is available.

Other Publications

X/Open Architectural Framework, available through The Open Group at their Web site, http://www.xopen.org/public/arch.

IBM Open Blueprint Architecture, a Datapro Report, available from the Datapro Information Services Group of Delran, New Jersey. It is also available via the Open Blueprint home page, http://www.software.ibm.com/openblue.

Enterprise Architectures: A Comparison of Vendor Initiatives, J. Tibbetts, available via the Open Blueprint home page, http://www.software.ibm.com/openblue.

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

G326-0395-02