The OSF Distributed Computing Environment: Building on International Standards

Every savvy manager knows that information is critical to business success. For that reason, people are constantly looking to improve the way it is stored, retrieved, and distributed. Computers have simplified all three tasks; nevertheless, distributing information across an entire enterprise is still difficult to do. To solve this problem, companies are developing enterprise-wide systems that allow computers in different departments to share information easily.

Enterprise-wide information systems must meet some important criteria. For example, they must allow users to locate the information they need. Furthermore, to protect sensitive information from unauthorized access, they must provide reliable security mechanisms. In addition, they must be manageable.

Organizations attempting to create such systems encounter several obstacles. Typically information is distributed among a large number of computers, spread throughout an organization. Further complicating the situation, these computers may have been manufactured by different suppliers and run incompatible communications protocols. An enterprise-wide information system must transcend those differences and appear as an integrated system that brings the information scattered throughout the enterprise to every user.

THE ROLE OF STANDARDS

Several international organizations have been working to ease these problems by establishing standards for the distribution of information. As a result, there are established protocols that enable computers to communicate, as well as standards for high-level services such as directory services, electronic mail, and file-data transfer.

International standards define the protocols and services that form the foundation on which systems that distribute information can be built. Standard protocols, for example, allow computers from different manufacturers to communicate and exchange data. Standard services, such as the ISO directory service, make it easy for computers to locate information stored throughout the enterprise.

IMPLEMENTING STANDARDS

The Open Software Foundation Distributed Computing Environment (OSF™ DCE) allows programmers to build distributed applications that put these standard protocols and services to work. DCE is a collection of tools and services that ease the development of distributed applications, which, in turn, make information available throughout a network. This technology

allows users to retrieve information easily — regardless of where it is stored — and distribute it to wherever it is needed. DCE implements all the relevant international standards and will support relevant future standards as they evolve.

This paper looks at DCE in the context of open systems and standards. It begins with an overview of DCE and the RPC model of programming. Next it discusses how DCE uses distributed information standards from the International Organization of Standards (ISO) and International Telephone and Telegraph Consultation Committee (CCITT). Finally it summarizes how DCE uses standard programming interfaces defined by IEEE POSIX and X/Open.

Overview of DCE

DCE PROVIDES SECURITY AND RELIABILITY

Distributing applications across networked computers eases the sharing of information; however, it also introduces new complications to programmers and users. Computer networks are inherently insecure, susceptible to unauthorized eavesdropping and data tampering. In addition, large enterprise-wide networks make it difficult to locate the resources scattered throughout a vast computing network.

Distributed applications also are vulnerable. They are prone to network failure as well as the failure of critical resources on the network. The network itself also can introduce new inefficiencies to distributed applications. For example, a poorly designed distributed computing environment can overwhelm the network and distributed resources, creating data bottlenecks that reduce the performance of the applications.

DCE is a consistent, integrated set of technologies that solves these problems by easing the development, use, and management of distributed applications. It is designed to be independent of hardware, operating systems, and networks, allowing DCE distributed applications to run on a wide variety of computers simultaneously. DCE provides security and directory services to solve the problems of network security and locating distributed resources. It makes extensive use of sophisticated data caching and replication techniques to increase performance and reliability. In addition, DCE presents programmers with a threaded process model of concurrency, which allows them to write applications that run on several computers concurrently.

DCE hides the complexity of the network by presenting the programmer with a procedural programming model that automatically extends the application into the distributed environment. This programming model, the remote procedure call, or RPC, uses DCE security and directory services, ensuring that the application runs securely and can locate distributed resources.

RPC PROGRAMMING SERVICES

In the early 1980s, researchers in both academe and industry began to look for a new model for building distributed applications. They sought a model that would allow programmers to build distributed applications easily, using well-developed procedural programming skills. The RPC model these researchers developed is an extension of the procedural programming model.

RPC allows an application running on a local computer to call procedures that are available on a remote computer. It introduces only two new concepts to the programmer. One is the concept of an interface, the set of procedures that are available on the remote computer. The other is the concept of an exception, an asynchronous event that occurs when a remote procedure call fails because of an unforeseen network problem.

INTERFACE DEFINITION NOTATION

An interface is declared using a language called the interface definition notation (IDN), stored in a separate interface definition file. The IDN allows a programmer to declare functional procedures, variable data types and RPC-specific attributes that are used in declaring data types and procedures. The key feature of the IDN is that it closely resembles declarative programming languages such as C and therefore is easy for programmers to use.

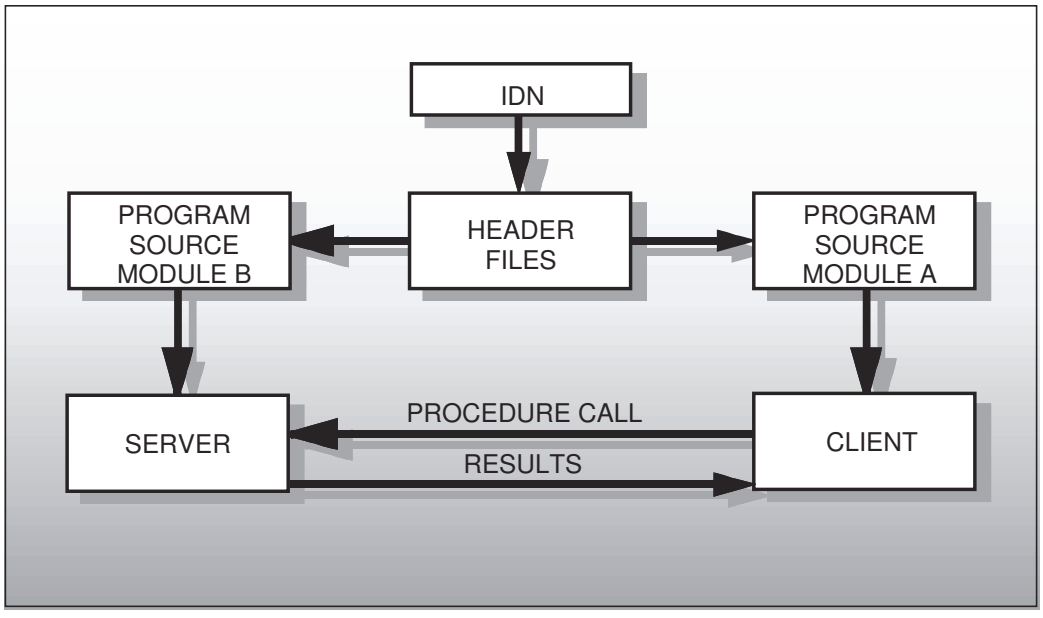

Figure 1

The IDN file is used by both the local program and the remote program to ensure that the data types and procedures are the same (see Figure 1). The IDN data type specification allows the RPC mechanism to translate data when the local and remote computers store the data in a different format. For instance, it allows a local computer using most-significant-byte first integers to send data to a remote computer using least-significant-byte first integers.

The procedure declarations in the IDN allow the programmer to qualify how data is used by the remote procedure. For instance, the IDN lets a programmer qualify whether a remote procedure parameter is passing data into the procedure, or passing data back to the caller. Because the act of passing a parameter causes data to be transferred across the network, this qualification tells the RPC mechanism to transfer data only when necessary, increasing the efficiency of the remote procedure call.

EXCEPTION MECHANISM

RPC provides an exception mechanism that allows a programmer to determine when an unforeseen network problem has caused the application to fail. The programmer uses the exceptions by specifying that a local handler procedure be called when an exception occurs.

ISO OSI Standards

In 1977 ISO embarked on a mission to establish a model for distributed computer communications services. This model, the Open Systems Interconnect (OSI) model, solves two problems. First, it defines standard protocols that could be supported by all computer and software vendors to promote industry-wide interoperability. Second, it establishes standard application services that could be used for building a distributed computing environment.

OSI COMMUNICATIONS MODEL

In the OSI model, computer communication protocols are separated into seven layered components, numbered 1 through 7. The lower layers describe the method of transferring data through network wires, and the higher layers describe how that data can be used by application programs. Each layer builds on previous layers and presents the functions of all the lower layers to the applications programmer. This model is very flexible because it allows new services to be built on top of existing OSI service components.

Starting with layer one, the layers of the OSI communications model are the physical, data link, network, transport, session, presentation, and application layers. The first four layers are often referred to as the end-to-end services. These services are usually provided as part of the operating system. The upper three layers are usually provided as run-time libraries that can be loaded and used by distributed applications and services.

ISO has defined a few distributed application services that can be used for supporting distributed applications. One is the ISO directory service, which is used to locate information and computer resources distributed through the computing environment. Another is the ISO service for handling mail messages, which is used for transferring and delivering electronic mail messages. The last is the file transfer and access method, which is used for transferring file data from one computer to another.

The communication services available under the OSI layered communications model do not hide the complexity of the network. As a result, programmers must understand all layers of the communications model in order to write distributed applications. These services were specifically developed to be used by the ISO distributed application services and not as a general programming model for building distributed applications. A distributed programming environment such as the DCE allows programmers to use the standard ISO services, while shielding them from the complexity of the services.

ISO OSI AND THE RPC PROGRAMMING MODEL

The RPC programming model fits in well with the OSI model (see Figure 2). It can be developed using the entire suite of ISO communication service elements. RPC applications can be used with ISO application services such as the ISO directory services. The RPC interface isolates the applications programmer from the underlying distributed application services. As new ISO services are defined, they can be added to the distributed programming environment without any change to the RPC application. DCE sits at OSI layers 5, 6, and 7 offering application, presentation and session services. DCE uses layers 1 through 4 and is transport independent.

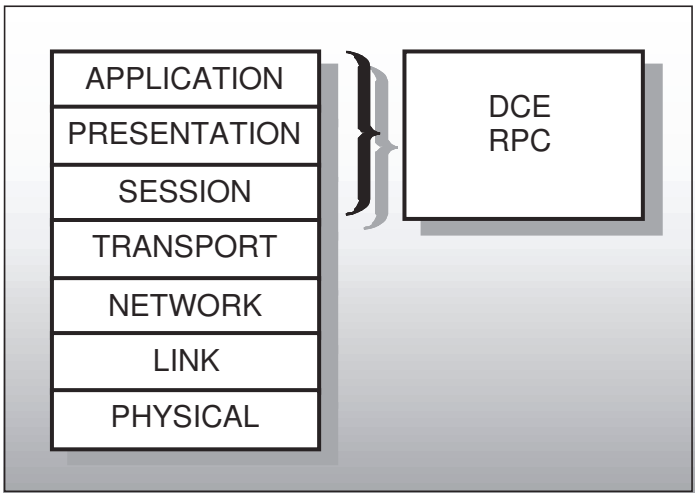

Figure 2

DCE uses this model to isolate the programmer from both the directory and security services. DCE RPC applications can use one of three directory services transparently including the ISO directory service (CCITT X.500) , U.S. Department of Defense Domain Naming System (DNS), and the RPC-based DCE Cell Directory Services (CDS).

ISO DIRECTORY SERVICE (CCITT X.500)

DCE provides an extended version of the ISO directory service (CCITT X.500) specified in the 1988 standard. This allows DCE clients to share and access directory information with organizations supporting implementations of this standard. DCE uses the X.500 global directory service to locate individual DCE installations, as shown in Figure 3. The DCE Global Directory Service (GDS) is based on Siemens' DIR-X™ product, a well-tested version of the ISO directory service. It has an extension that allows it to protect and secure data stored in the directory, increase its reliability through the use of directory replication, and enhance its performance through the use of sophisticated caching techniques.

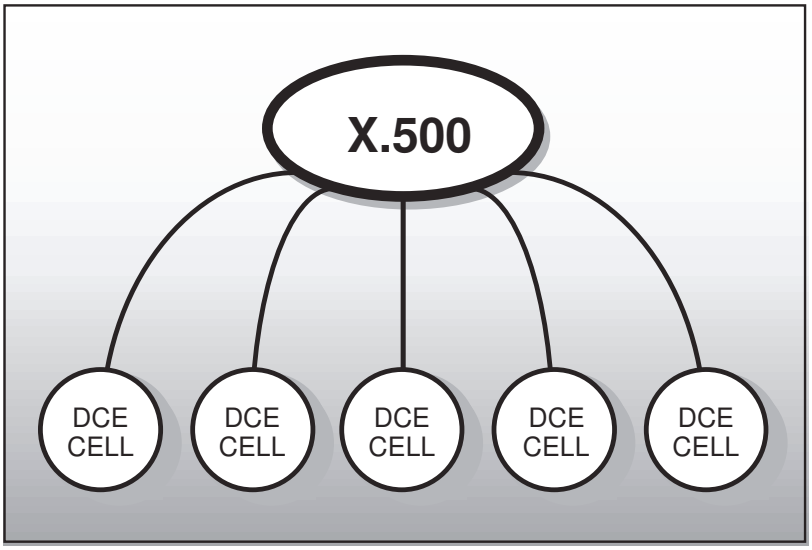

Figure 3

ISO MESSAGE SYSTEM (CCITT X.400) AND ISO FTAM

X.400 and FTAM (File Transfer and Access Method) are defined as application services at level 7 of the OSI communications stack, and therefore use lower-layer OSI services. X.400 is a protocol and service specification for the encapsulation and delivery of mail messages. DCE complements this technology by giving developers tools for building distributed mail applications. These applications can provide a secure and transparent mail system throughout the enterprise and can use X.400 as a delivery mechanism to other organizations.

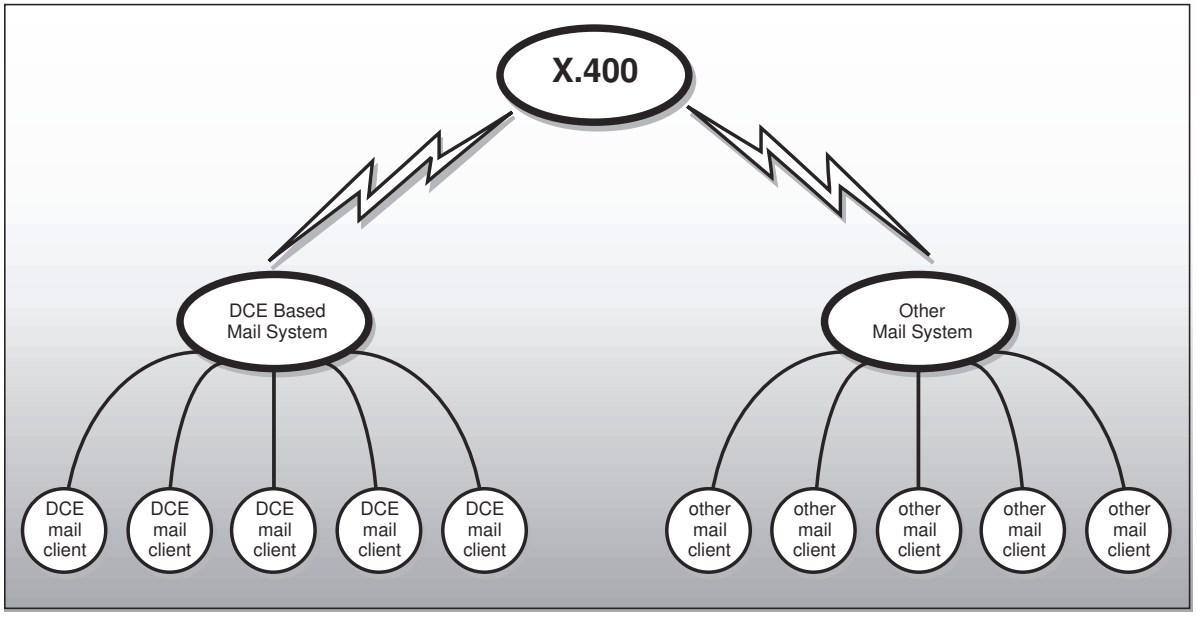

Figure 4

FTAM is a protocol specification for transferring files between computers. The DCE Distributed File System (DFS) can use FTAM to access different file systems. DFS allows DCE users to access remote files in the same way they access local files. By allowing users to take advantage of sophisticated caching techniques, it meets efficiency requirements. Availability is provided by file system replication, security by the DCE RPC, and scalability by the DCE security and directory services.

IEEE POSIX Standards

The Institute of Electrical and Electronic Engineers has been developing standards in the area of software portability. Its POSIX standards are a set of programming interfaces that define a portable operating system. The DCE supports all relevant POSIX standards — both available and in draft form. OSF will continue to track the POSIX working groups to ensure compatibility with future versions.

POSIX THREADS MODEL OF PROGRAMMING

The DCE provides a threads model of programming that enables applications to run simultaneously on a number of computers. The DCE Threads service allows a programmer to make a number of RPC calls to different computers concurrently and synchronize the results of their execution. This service presents programmers with the popular fork-and-wait process programming model within a single address space.

The DCE Threads Service provides a superset of draft 4 of the POSIX 1003.4a threads model of programming. It is based on the well-tested implementation of CMA (Concert Multithread™ Architecture) threads from Digital Equipment Corporation.

POSIX FILE SYSTEM SEMANTICS

The DCE Distributed File System enables computer users and programmers to share files transparently throughout the distributed computing environment. The DFS programming and user interface supports the standard semantics of the POSIX 1003.1a file system specification. DCE will support the POSIX transparent file access standard when it is available.

POSIX ACCESS CONTROL LISTS

The DCE Security Service provides a means of controlling access to resources throughout the distributed environment. It allows a list of users or groups of users to be assigned individual rights to each resource using access control lists (ACLs). The DCE supports a superset of the POSIX ACLs specified in the 1003.6 draft standard.

X/Open Application Programming Interfaces

X/OPEN DIRECTORY SERVICE (XDS)

The DCE uses the XDS programming interface for accessing both the DCE Cell Directory Service (CDS) and the Global Directory Service (GDS). GDS is the DCE implementation of the ISO directory service (CCITT X.500). XDS is a standard interface available on a large number of computer platforms. The use of XDS by the DCE ensures that applications are portable as well as interoperable.

X/OPEN OBJECT MANAGEMENT (XOM)

DCE also presents programmers with the X/Open object management (XOM) programming interface for accessing ISO application services. XOM is used to develop applications that take advantage of services such as the ISO message-handling service (CCITT X.400). The combination of X/Open XDS and XOM provide DCE programmers with a flexible and portable ISO application programming interface.

X/OPEN TRANSPORT INTERFACE (XTI)

DCE supports the XTI portable networking interface, which enables DCE to run on a variety of transport protocols. These transport protocols are provided by a computer operating system through XTI, which is available on a number of operating systems including OSF/1™ and UNIX® System V.4.

Conclusion

The ability to distribute information easily is a prime concern for information-intensive organizations. Many companies are using distributed computing applications to build enterprise-wide information systems. However, it is difficult for information system developers to build distributed applications that run on the wide variety of computers. To do so, they need standardized communication services coupled with an easy-to-use programming environment for building distributed applications.

The OSF DCE solves theses problems by providing integrated tools and services. It incorporates all the relevant communication standards and will use emerging distributed computing standards as they become available.

Finally, the DCE offering presents application developers with the standards-based RPC programming model, which hides the complexity of the distributed environment. This provides an infrastructure for tying together the distributed communications services specified by international standards organizations.

©1992 Open Software Foundation. All Rights Reserved. Printed in U.S.A.
OSF and OSF/1 are trademarks of Open Software Foundation, Inc. Concert Multithread is a trademark of Digital Equipment Corporation. DIR-X is a trademark of Siemens AG. UNIX is a registered trademark of UNIX System Laboratories, Inc. in the United States and other countries.

Open Software Foundation / North America
11 Cambridge Center
Cambridge, MA 02142
United States
617 621 8700

4 Main Street
Garden Level #7
Los Altos, CA 94022
415 948 1171

620 Herndon Parkway
Suite 200
Herndon, VA 22070
703 481 9819

Open Software Foundation / Europe
Avenue des Pleiades-laan 15
1200 Brussels, Belgium
32 2 772 88 88

Kronstadter Strasse 9
D-8000 Munchen 80
Germany
49 89 93 20 10

Research Institute — Grenoble
2 Avenue de Vignate
38610 Gieres
France
33 76 63 48 63

Open Software Foundation / Asia
11-10, Kita-Aoyama 2-chome
Minato-ku, Tokyo 107
Japan
81 3 3479 4740

O-DCE-WP19-1

OPEN SOFTWARE FOUNDATION

The OSF Distributed Computing Environment: Building on International Standards

A White Paper
April 1992