Upper Layer OSI Protocols

In this report:

Datapro Summary

The OSI Reference Model and the basic set of protocols based on it have been under development for ten years and have reached maturity. Acceptance of OSI depends, in part, on the number of products that implement the protocols defined in the upper layers of the OSI model. The upper layers, the Presentation Layer and the Application Layer, provide the interface to the user's applications and present the external view of OSI to computing environments. The Presentation Layer provides the syntax for the information exchanged between applications. Both connection-oriented and connectionless protocols and services are defined under the Presentation Layer. The Application Layer is concerned only with the semantics of the information. The Application Layer consists of a number of protocols and services to meet the needs of users and designers. These are divided into three broad categories: Application Service Elements, Common Application Services, and Specific Application Services.

Introduction

Development of the OSI Reference Model and the protocols and services based on that architecture has been under way for ten years. A major barrier to the acceptance of OSI in corporate environments has been the unavailability of products for the protocols. Recognizing that OSI is reaching maturity, the federal government has mandated the use of OSI protocols through a Federal Information Processing Standard (FIPS) called the Government OSI Profile (GOSIP). GOSIP compliance became mandatory in August 1991.

An important factor affecting the acceptance of OSI is the availability of products that implement the protocols defined at the upper layers of the OSI Reference Model. The protocols at these two layers provide the interface to the user's applications. The uppermost layers of the OSI Reference Model present the external view of OSI to the computing environment. Applications are defined that incorporate the services provided by the lower layers. Without the Application Layer and the applications using the services provided, there would be no need for the OSI networks.

The upper layers of the OSI Reference Model consist of the Presentation Layer and the Application Layer. These two layers taken as a set provide both a syntactic and semantic framework for the application that performs the distributed processing function. The Presentation Layer provides a method for negotiating a syntactic structure for the exchange of information between applications. The Application Layer defines a set of protocols tailored to the demands of the specific applications.

The concept of the upper layers is to provide a consistent interface from the network environment to the applications that require distributed computing services. For applications that have common requirements, the Application Layer provides a single solution. For example, all applications that need a bulk data transfer can use the File Transfer Access and Management services.

—By James Moulton
President and Principal Consultant
Open Network Solutions, Inc.

© 1993 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

OSI Concepts

Service Definitions

The OSI Reference Model presents the architecture for the exchange of information by distributed applications. The model describes a layered approach where each layer has a defined scope and set of functions.

Since the model is purely an architecture, at each layer a service definition describes the abstract interface to a layer protocol. The service describes the interaction of the protocol user to the protocol implementation. Since it is abstract, however, it does not define a specific protocol interface used for implementation. Also, many protocols can support a single service definition.

Protocol Specifications

For each service definition, one or more protocols can be defined. Each protocol will follow the service interface while providing different mechanisms and functions. Conformance can only be tested against the protocol specification.

Modes of Communication

The OSI Reference Model describes two distinct types of communication: connection-oriented and connectionless data transmission (called connectionless). At each layer of the model, there is both a connection-oriented and a connectionless service. Additionally, protocols supporting each mode of operation are defined.

Connection-oriented communication was the original focus of the OSI Reference Model. In this mode of operation, there are three phases of communication: connection establishment, data transfer, and connection release. The data transfer phase is simplified by maintaining sufficient state information. A typical example of this mode of operation is the X.25 virtual circuit. (In the case of the Application Layer, connections are renamed *Associations*. An Association is used to describe the unique cooperation between the applications. The cooperation between the two applications manifests in the distinction that applications pass information that have semantic content. At the remaining layers, the data carries no semantic content.)

Connectionless Data Transmission was added to the model as a result of the emergence of datagram services at the Network and Data Link Layers. In a connectionless data transmission, there is only a data transfer phase. No state information is maintained, and each transmission is viewed as independent. Examples of connectionless mode transmissions are Logical Link Control (LLC) and the Connectionless Network Protocol (CLNP). For efficiency, the architecture requires that no segmentation or reassembly take place above the Network Layer. For that reason, the protocols at all of the higher layers are very simple. The majority of the functions involve address mapping from one layer to the next.

Presentation Layer

The Presentation Layer is responsible for the selection of a transfer syntax. The Presentation Layer deals with generic functions that are needed by many different types of applications, specifically, a common means of representing a data structure in transit from one system to another.

Connection-Oriented Presentation

Overview

The connection-oriented Presentation Layer Service and protocol are responsible for providing a representation (syntax) for the information exchanged between applications utilizing connections. By establishing a connection, negotiations of transfer syntax are possible.

The Presentation Layer encompasses two aspects of the information representation:

1. the representation of data to be transferred between applications; and
2. the representation of the data structure to which applications refer in their communication, along with the representations of the set of actions that may be performed on this data structure.

The Presentation Layer is only concerned about syntax and not with the semantics of the information. The semantics is only known by the applications. If the syntax is not understood by both systems, however, the semantics cannot be determined.

Document Information

The work on this standard is a joint effort between ISO/IEC JTC1 and CCITT. The standard is defined in the following documents:

- ISO 8882 Information Processing—Open Systems Interconnection—Presentation Service Definition
- ISO 8883 Information Processing—Open Systems Interconnection—Connection-Oriented Presentation Protocol

Purpose

The Presentation Layer is concerned with the syntax or representation of information in transit between two applications.

The Presentation Layer has two main functions:

- negotiation of transfer syntaxes; and
- transformation to and from transfer syntax.

The function of transfer syntax negotiation is supported by the Presentation protocol defined in ISO 8883. The protocols provide presentation context definition facilities. These facilities provide a means of determining the precise manner in which information is encoded for transfer to the other presentation entity.

A major feature of the protocol and services available at the Presentation Layer is the capability to pass Session Services through to the application unchanged. This "pass-through" approach allows the application to control Session services such as synchronization and checkpoints.

A major concept introduced at the Presentation Layer is that of a context. A context is the definition of data structures, the operations that are valid over the data structures, and the encoding of the information for transfer. The selection, maintenance, and switching of contexts are the major functions of the Presentation Layer and associated protocols.

The Presentation Layer protocol is important in that it allows the specification of information in a manner that is independent of the application. This allows systems that operate with disparate data structures and formats to exchange information in a meaningful way.

Functions

The Presentation Service Definition and Protocol Specification were completed in time for publication in 1988. Since that time, modifications based upon extensions to the Session protocol have been incorporated.

The Presentation protocol is divided into functional units. Functional units are logical groupings of procedures for the purpose of:

- negotiation during presentation-connection establishment for subsequent use on the presentation-connection; and
- specification of conformance requirements.

© 1993 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

The Presentation protocol consists of three functional units:

- kernel functional unit,
- context management functional unit, and
- context restoration functional unit.

The kernel functional unit is always available and supports the basic protocol elements of procedures. These procedures permit the establishment of a presentation-connection, transfer of data, and release of the presentation-connection.

The context management functional unit supports the context addition and deletion services. This functional unit is optional. The use of this functional unit is negotiated during connection establishment.

The context restoration functional unit provides additional functionality. The selection of this functional unit also requires the selection of the synchronization or activity services of the Session Layer.

Recent Changes

The Presentation Service and Protocol Specification remained stable throughout 1992. Recent work has focused on defining new transfer syntaxes and context defintions for use by the applications.

GOSIP Requirements

GOSIP Version 1 required minimal Presentation Layer functionality. The new GOSIP Version 2 permits the use of added functional units.

Most major manufacturers now have GOSIP-certified products that include the required Presentation Layer features. Most vendors have chosen to package the Presentation Layer Protocol with the application product.

Sponsoring Organization

The Presentation Layer standards are joint between ISO/IEC JTC1 (ISO) and the CCITT. The appropriate CCITT Recommendations are X.216 and X.226.

Connectionless Presentation Protocol

Overview

The connectionless presentation protocol is used in the connectionless protocol stack based on an application protocol through to the connectionless transport protocol. In the connectionless protocol stack, there are few functions available between the application and the Network Layer. The major functions available are address mapping and service mapping.

At the Presentation Layer, the protocol selects a context and transfer syntax based upon the needs of the application. There is no negotiation. If the destination presentation-entity is not capable of supporting that context, the communication fails.

Document Information

The connectionless presentation protocol is defined in the following documents:

- ISO 8882/1 Information Processing—Open Systems Interconnection—Presentation Service Definition, Addendum 1—Connectionless Mode Service Definition
- ISO 9596 Information Processing—Open Systems Interconnection—Connectionless Mode Presentation Protocol Specification

Purpose and Functions

The connectionless presentation protocol is based upon the needs of connectionless applications. The connectionless presentation protocol maps the address of the application to the appropriate session entity (SSAP). The presentation protocol does not perform segmentation or reassembly. It does not perform any error detection or error correction. The single function that the protocol performs is that of the selection of a context for the data transfer. The selected context is based on the requirements of the applications and the known requirements of the destination presentation-entity. The selected context may be selected through information supplied manually or gathered through system management.

Application Layer

The Application Layer is the highest layer in the OSI Reference Model. It is where the actual information and functions are generated for distributed processing.

Structure

The Application Layer contains a relatively large set of protocols and services to satisfy the needs of the various application users and designers. The services of the Application Layer can be divided into three broad categories:

1. Application Service Elements that are used by all applications and provide basic OSI services in support of communication.
2. Common Application Services that are used as required by other applications or users.
3. Specific Application Services that are used by a narrow class of users or other applications. These applications provide a specific service tailored to specific application needs.

The structure of the Application Layer is shown in Figure 1.

Virtual Terminal

Overview

The Virtual Terminal application is defined as a service (VTS, ISO 9040) and as a protocol (VTP, ISO 9041). The VTP is intended to provide a capability for exchanging information between terminal-oriented applications. With the VTP, it is possible to exchange information even when the terminal characteristics assumed by the applications are different. The VTP provides the mappings to ensure the successful exchange of information.

Document Information

Virtual Terminal application is defined in the following documents:

- ISO 9040:1990 Information Technology—Open Systems Interconnection—Virtual Terminal Basic Class Service
- ISO 9041-1:1990 Information Technology—Open Systems Interconnection—Virtual Terminal Basic Class Protocol

Sponsoring Organization

The sponsoring organization is ISO/IEC JTC1 SC 21 (International Organization for Standardization).

Purpose

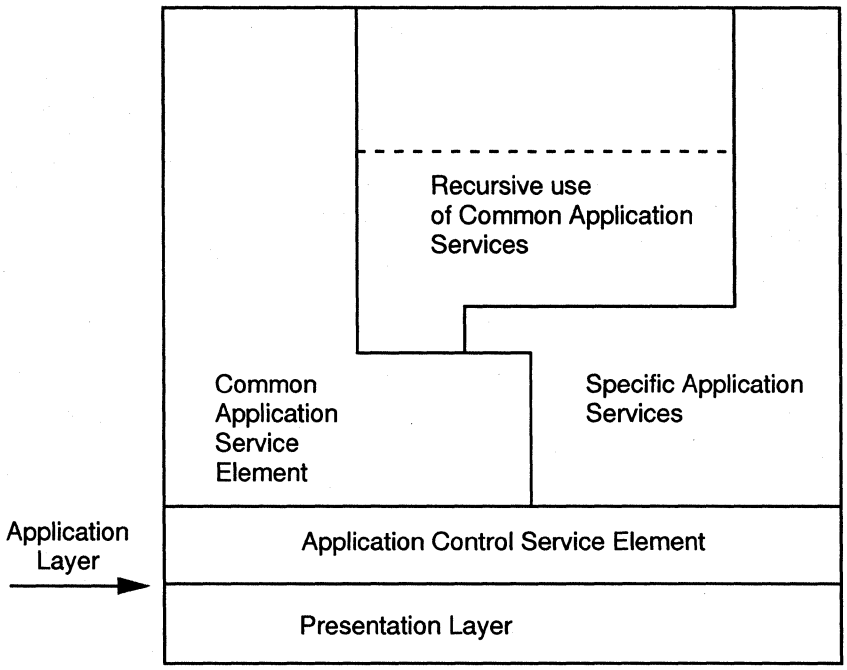
The VTS and VTP define a method for interactive applications requiring terminal-oriented communication expressed in terms of the transmission and manipulation of graphical images having the following characteristics:

a) the images are composed of character-box graphic elements organized into a one-, two-, or three-dimensional structure; and

b) attributes may be associated with any graphic element to qualify its mode of display.

© 1993 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

Figure 1.
Application Structure



The Virtual Terminal Basic Class Service offers the following services to the VT-user:

a) the means to establish a VT-association between two peer VT-users for the purpose of enabling Virtual Terminal information exchange;

b) the means to negotiate the VT functional units required;

c) the means to negotiate a consistent set of VTE-parameters;

d) the means to transfer and manipulate structured data in a way that is independent of the local representation of information used by each VT-user and that is independent of the way in which supporting communications media are used;

e) the means to control the integrity of the communication;

f) the means to terminate the VT-association either unilaterally or by mutual agreement;

g) the means to support either synchronous (S-mode) or asynchronous (A-mode) operation between the VT-users;

h) the means to exchange priority information to gain the immediate attention of the VT-user;

i) the means to terminate information transfer destructively and resynchronize the activity of the VT provider;

j) a facility for defining blocks in a display object (Blocks functional unit);

k) a facility for defining fields in a display object [Fields functional unit, also uses feature in n)];

l) additional optional access rules for control objects in S-mode (Enhanced Access rules functional unit);

m) means to control the asymmetry inherent in typical use of these features [uses the feature in l)];

n) a facility for defining control objects with content consisting of multiple data elements or a single partially updatable structured data element (Structured Control Objects function unit);

o) a facility for controlling data entry to fields using new standard types of control object [uses the feature in n)];

p) a facility for storing and using update information in Reference Information Objects (RIO functional unit); and

q) a facility for establishing a VT-association with the capability to switch between the modes of operation when the VTE is changed.

Recent Changes

The VTS and VTP are undergoing modification as additional classes of protocol are defined. However, the changes are being structured so that they do not impact the Basic Class. The new additional classes are meant to serve specific terminal types such as page devices.

GOSIP Considerations

The VTP was not included in GOSIP Version 1. It has been added as an application under GOSIP Version 2. Within the U. S. there has not been a demand for VTP. Vendors are just beginning to release products based on the GOSIP requirements. There are no certification tests available for VTP at this time; therefore, no certified products yet exist.

File Transfer Access and Management (FTAM)

Overview

FTAM provides the OSI file management and transfer capabilities. In order to provide the complete range of file services across a heterogeneous network, FTAM is based upon a Virtual Filestore model. The model provides a consistent view of all files available in the FTAM environment.

FTAM consists of four parts. These parts are:

- Part 1—General Introduction
- Part 2—Virtual Filestore Definition
- Part 3—File Service Definition
- Part 4—File Protocol Specification

The four parts define a complete application system. The system defines the filestore model, the services that operate over the filestore, and the protocol that exchanges the information.

Document Information

File Transfer Access and Management standards are defined in the following documents:

© 1993 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

- ISO 8571-1 Information Processing—Open Systems Interconnection—File Transfer, Access and Management—Part 1: General Introduction
- ISO 8571-2 Information Processing—Open Systems Interconnection—File Transfer, Access and Management—Part 2: Virtual Filestore Definition
- ISO 8571-3 Information Processing—Open Systems Interconnection—File Transfer, Access and Management—Part 3: File Service Definition
- ISO 8571-4 Information Processing—Open Systems Interconnection—File Protocol Specification

Purpose

The FTAM standard provides for the transmission, access, and management of a variety of different file types and formats across OSI networks media, without detailed knowledge of the particular characteristics of the remote machines.

FTAM allows different applications or different users of applications to transfer information without specific knowledge of the other system's file system characteristics. FTAM also allows users a degree of control over the file activity, as well as a set of capabilities and features. Other applications may use FTAM as a supporting service. In fact, FTAM can be used locally as a set of callable library routines.

Functions

The FTAM standard is composed of four parts, listed earlier. The General Introduction presents the basic terminology and broad FTAM concepts. The File Service Definition gives an overview of FTAM services provided to the user. The Virtual Filestore document gives information on the central model used by FTAM. Finally, the File Protocol Specification gives a detailed description of the protocol interactions necessary to accomplish the FTAM activity.

Standards are often modified and enhanced through addenda. There are three addenda currently under development for FTAM:

- overlapped access,
- filestore management, and
- protocol conformance.

Overlapped access deals with reading from and writing to different portions of a file simultaneously; filestore management involves an extensive set of directory commands, including search, list, and change directory.

The services of FTAM provided to the user are:

- the ability to communicate about files without specific knowledge of the other system,
- the facilities to express explicitly what the user requires,
- the ability to specify uniform file properties,
- the ability to specify record-level file access and positional file transfer, and
- detailed file management.

FTAM is a two-party file transfer protocol. A *controller* of the file activity (initiator) directs the action, and a *responder* responds to the initiator in a passive role. All file transfers and access operations occur between initiator and responder. An FTAM implementation may act as initiator, as responder, or as both.

FTAM is defined in terms of functional units and service classes. Service classes are described in terms of functional units; some of the functional units are mandatory within a service class and some are optional. The functional units in FTAM are:

- kernel,
- limited file management,
- enhanced file management,
- read,
- write,
- grouping,
- recovery, and
- restart.

For functional units, the kernel is the basic set of FTAM capabilities. Limited file management deals with the ability to create, delete, and interrogate properties of files. Enhanced file management deals with the ability to change file properties. Grouping allows concatenation of FTAM requests for efficiency.

Service classes are:

- the transfer class, which allows for the movement of files or parts of files between systems, placing emphasis on simple operation with a minimum of protocol overhead before and after the data transfer.
- the management class, which allows control of the Virtual Filestore by a series of independent confirmed service exchanges, but does not include file transfer mechanisms.
- the transfer and management class, which combines the features of the transfer class and the management class.
- the access class, which allows the initiating entity to perform a sequence of operations on the file access data units, providing for the manipulation of remote data.
- the constrained class, which leaves the selection of functional units to the designer of the distributed application, giving full flexibility of optimization, but no guarantee of a common functional kernel.

FTAM uses the concept of a Virtual Filestore. In the OSI environment, there is one conceptual representation of this Virtual Filestore model. In the actual systems, there may be multiple real filestore implementations based on the operating system. There must be a mapping between the real filestore and the Virtual Filestore. The nature of the mapping is a local issue.

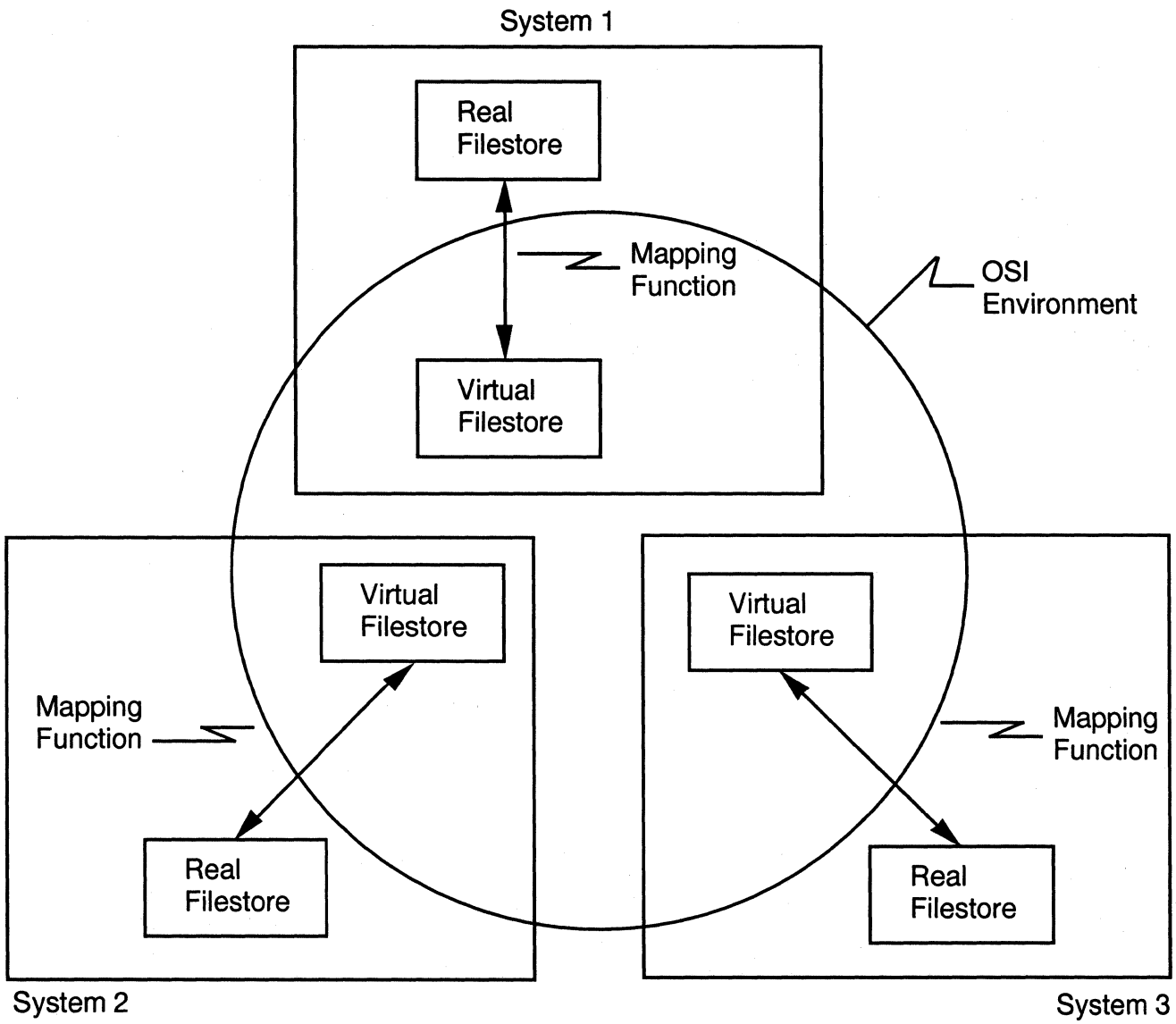
The generic FTAM model is applicable to most FTAM systems in use today. All of the characteristics of the virtual filestore can be recognized and interpreted by OSI file systems so that the communication is through this model. Figure 2 shows the relationship between virtual and real filestores.

The FTAM services and protocol are organized into *regimes*. A regime is a phase of operation in which certain well-defined operations and events may occur. The regimes accomplish the following functions:

- allow the initiator and the filestore to establish information about each other, including their respective identities;
- identify the file that is needed;
- establish the attributes describing the file and bulk data transfer which is to take place in this activity;
- engage in file management;
- locate the position in the file access structure of the data to be accessed; and

© 1993 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

Figure 2.
Virtual Filestore



- insert, replace, extend, or erase one or more complete FADUs (records).

The specific regimes are shown in Figure 3.

A virtual filestore schema is composed of the following:

- a file, which contains file attributes and file contents,
- a filestore, which may contain a number of files, and
- a connection, which involves active attributes and current attributes. There is a user attached to the connection. The schema is hierarchical with a tree-like structure. Specific parts of a file are defined using node identifiers. Many different access structures are possible.

FTAM has a set of diagnostics that communicates information about the status of an FTAM request. There is a provision for users to include additional explanatory material where appropriate. FTAM has four classes of errors, from minor errors to errors which destroy the FTAM activity.

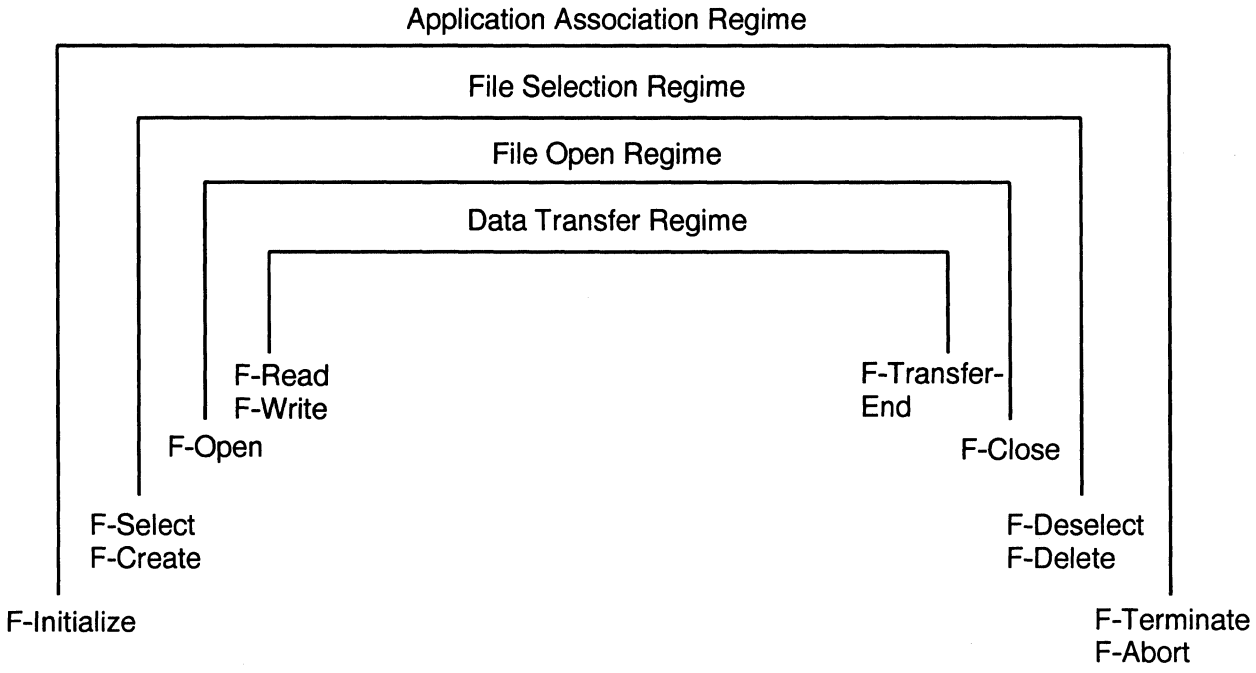
Two FTAM service types are defined. One type is called internal and supports the error recovery protocol. Errors are apparent to the file service user, and the user is allowed to directly control error recovery procedures. The other FTAM service type is called external; here the file service user has no awareness of error detection and recovery.

Recent Changes

FTAM just completed a complete review and reissue. There were no major changes as stability of the standard was a prime consideration. The major area of change was the definition of new file types. That is, new file types were added for the exchange of more complex files while maintaining the file structure. The majority of the work on file types is performed in cooperation with workshops such as the one defining GOSIP.

© 1993 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

Figure 3.
FTAM Regimes



GOSIP Considerations

FTAM was included in GOSIP Version 1, and most vendors have certified products. GOSIP Version 2 significantly extended the requirements on FTAM especially in the area of file types. Most of the products are certified to both GOSIP Version 1 and GOSIP Version 2.

Distributed Transaction Processing (DTP)

Overview

DTP is a complex transaction processing system that allows for the execution of transactions across multiple systems.

A major feature of DTP is the control of commitment so that all systems participating in the transaction are ensured of consistent results.

The DTP model is that of a single client requesting the execution of a transaction from one or more remote systems. Each remote system may parcel out the transaction to other systems, creating a tree structure. Transactions are phased operations. First, the systems participating in the transaction are "asked" if they can perform their assigned tasks. If any system responds with a "No," the transaction fails. After a consensus is reached about the systems' capability to perform the transaction, the client requests execution. After receiving the response from all systems, a commit phase is entered which ensures that all participants have successfully completed the task.

Document Information

The DTP is now an International Standard. It is now awaiting formal publication. There are few products available for DTP, and it should still be considered immature.

- IS 10026-1 Information Processing—Open Systems Interconnection—Distributed Transaction Processing—Model

Purpose/Function

Distributed Transaction Processing (DTP) provides a transaction service to its users. Specifically, DTP provides for the atomic execution of command/response pairs across multiple systems. DTP uses the features of other OSI protocols to ensure that each transaction is only executed when all component systems succeed. The use of commitment control allows the user to ignore the problem of coordination among multiple systems involved in the transaction.

The functions of DTP are:

- transaction definition,
- distribution of processing functions,
- coordination of execution responses,
- management of commitment and recovery, and
- reporting of results.

DTP is now complete. Further modifications will surely follow as implementation experience is gained and errors are uncovered in the specification.

Operation of DTP is as follows:

1. A DTP user submits a transaction,
2. DTP distributes the request to the systems participating in the transaction,
3. Participating systems report whether they can perform the function requested; recursively apply step 2; or report failure.
4. If all participating systems report that the transaction will succeed, the transaction succeeds; if any system reports failure, the transaction fails.
5. If the transaction succeeds, then a commit is issued whereby all participants "save" the results.

© 1993 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

6. If the transaction fails, then a rollback is issued whereby all participants return to the initial data.

GOSIP Considerations

DTP is too immature to be included in GOSIP Version 1 or Version 2. It is not included in the plans for GOSIP Version 3. Few products are available as yet.

Message Handling Systems (MHS)

Overview

The CCITT-defined MHS consists of a set of protocols that taken together provides a complete electronic mail system. The MHS, sometimes called X.400, provides for the submission of messages, the store-and-forward transfer of the message to the destination system, and the delivery of the message.

Document Information

MHS consists of the following series of documents:

- X.400—Message Handling Systems: System and Service Overview
- X.402—Message Handling Systems: Overall Architecture
- X.403—Message Handling Systems: Abstract Service Definition Conventions
- X.411—Message Handling Systems: Abstract Service Definition and Procedures
- X.413—Message Handling Systems: Message Store: Abstract-Service Definition
- X.419—Message Handling Systems: Protocol Specification
- X.420—Message Handling Systems: Interpersonal Messaging System

Purpose

The MHS is designed to provide electronic mail services to users of public data networks. Since the standards are generally applicable, MHS has become prevalent in all areas of networking including LANs.

Networks providing MHS can interconnect with any other network that provides MHS services corresponding to the standards. Providing a common platform for the exchange of electronic mail messages is the central importance of the MHS series.

The main feature of MHS is the specification of the InterPersonal Message Service (IPM). IPM provides the syntax and semantics for the exchange of messages between human users.

MHS is a store-and-forward service. The message is transferred from system to system, and each intermediate system takes responsibility for further protection of the message. In a store-and-forward system, messages may wait for relatively long periods of time in each node.

Functions

The MHS allows users to communicate by exchanging messages. There are three major MHS components:

- the Message Transfer System (MTS),
- the cooperating User Agents (UAs), and
- the Message Store (MS).

The MTS is composed of a set of Message Transfer Agents (MTAs) responsible for relaying the message from the originator's UA to the recipient's UA. The MTA servicing the recipient need not be active when the message leaves the originator's MTA; the message can be stored at an intermediate MTA or in the MS until the recipient's MTA becomes operational. Intermediate MTAs can also perform Application Layer routing based on address information contained in the message.

The MTAs can be managed by different organizations or administrations. An administration is either the PTT service in a country, or in the United States, a common carrier recognized by the CCITT. The collections of MTAs and UAs owned and operated by an Administration is called an Administration Management Domain (ADMD). The collection of MTAs and UAs owned and operated by a private organization is called a Private Management Domain (PRMD). All ADMDs must comply with the CCITT Recommendations. PRMDs that wish to use a message transfer system provided by an ADMD must comply with the CCITT Recommendations at the point of interconnection.

UAs have many functions that are outside the realm of standardization. The originator's UA assists in the creation and editing of a message; the recipient's UA stores the message until the recipient chooses to read it and can use certain message fields to determine the display order. The message submission and delivery interaction with the MTA, however, are standardized.

The originator's UA must supply to the MTA the message content, the address(es) of the message recipients, and the MTS services that are being requested. The message content is the information that the message originator wants transferred to the message recipient. The information about the address and service request is placed on the message envelope and is used by the MTS to deliver the message.

UAs can be implemented either in the same system as the MTA or remotely located from the MTA. A remote or standalone UA can be under the control of an ADMD, a PRMD vendor, or an organization that provides no message transfer services. Since the UA-MTA message submission and delivery interactions involve a transfer of responsibility for delivering a message, there must be a protocol between the remote UA and MTA to ensure that the transfer of responsibility occurs.

There can be many different types of User Agents. The MTS can be used to transfer data unrelated to IPM. As long as the recipient's UA can interpret the data sent by the originator's UA, meaningful communication can occur.

The MTS does not examine the message content unless the UA requests that the content be converted from one format to another before delivery. CCITT recognized that, although there were many potential UAs that could use the message transfer services, the most common use of the MTS would be to send a personal message from an originator to one or more recipients.

Message Transfer System

The MTS provides the following basic service to UAs:

- Message Identification,
- Submission and Delivery Time Stamp,
- Nondelivery Notification,
- Encoded Information Type Conversion, and
- Content Type Identification.

The following service elements can be selected by a UA on a per-message basis:

- Multidestination Delivery,
- Delivery Notification,
- Grade of Delivery,
- Deferred Delivery,
- Conversion Prohibition,
- Alternate Recipient Allowed, and
- Disclosure of Other Recipients.

© 1993 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

InterPersonal Message Service

The InterPersonal Message Service (IPM) is provided by a set of cooperating UAs called IPM UAs. This service allows a user to send an interpersonal message to one or more recipients and to have it received by those recipients. The IPM service is built upon—and uses—the services of the MTS.

The interpersonal message contains a header and a body. The interpersonal message header contains service elements that facilitate efficient processing of the message by the recipient's UA. The body is the information that the message originator wishes to convey to the message recipient.

Sponsoring Organization

The majority or the work on X.400 (MHS) is carried out under a collaborative committee of CCITT and ISO/IEC JTC1.

Recent Changes

As one of the most important application layer standards, MHS has undergone continued development. During the last year, however, there were no significant changes as the 1992 version of MHS was readied for publication.

GOSIP Considerations

GOSIP Version 1 mandated the MHS specified in the 1984 CCITT Recommendations. GOSIP Version 2 allows the use of the 1988 CCITT version. Certified products are available from many sources since this is the most popular and widely used OSI standard.

Directory Services

Overview

Document Information

The standardization of directory services began as an effort within CCITT. At that time it was focused on providing directory services for X.400 MHS systems. Before the completion of the X.500 series, the work became a collaborative effort with ISO. All text of the X.500 Recommendations is joint with ISO. The associated International Standard is ISO 9594 parts 1 through 7.

- X.500 The Directory—Overview of Concepts, Models, and Services
- X.501 The Directory—Models
- X.509 The Directory—Authentication Framework
- X.511 The Directory—Abstract Service Definition
- X.518 The Directory—Procedures for Distributed Operations
- X.520 The Directory—Selected Attribute Types
- X.521 The Directory—Select Object Classes

Purpose

The Directory Services provide directory capabilities required by OSI applications, OSI management processes, other OSI layer entities, and telecommunication services. Among the capabilities that it provides are "user friendly" naming, whereby objects can be referred to by names that are suitable for citing by human users, and name to address mappings that allow the binding between objects and their locations to be dynamic.

The Directory Service is not a general-purpose database system. It is assumed that as is typical with communications directories, there is a higher frequency of queries than of updates.

It is a characteristic of the Directory Service that, except as a consequence of differing access rights or unpropagated updates, the results of directory queries will not be dependent on the identity or location of the inquirer.

Functions

The directory is a collection of open systems that cooperate to hold a logical database of information about a set of objects in the real world. The users of the Directory Service, including people and computer programs, can read or modify the information, or parts of it, subject to having permission to do so. Each user is represented in accessing the Directory Service by a Directory User Agent (DUA) which is considered to be an application process.

The information held in the Directory Service is collectively known as the Directory Information Base (DIB).

The Directory Service provides users with a well-defined set of access capabilities, known as the abstract service of the Directory Service. This service provides a simple modification and retrieval capability. This can be built on with local DUA functions to provide the capabilities required by end users.

It is likely that the Directory Service will be distributed along both functional and organizational lines.

The Directory Service has been designed to support multiple applications. The nature of the application supported will govern which objects are listed in the Directory Service, which users will access the information, and which kind of access they will carry out. Applications may be very specific, such as the provision of distribution lists for electronic mail, or generic, such as the "interpersonal communications directory" application. The Directory Service provides the opportunity to exploit commonalities among the application:

- a single object may be relevant to more than one application; perhaps even the same piece of information about the same object may be so relevant.

To support this, a number of object classes and attribute types are defined that will be useful across a range of applications.

Directory Information Base

The DIB is a collection of objects. It is composed of entries that consist of a collection of information about one object. Each entry is made up of a set of attributes with one or more values. The types of attributes that are present in a particular entry are dependent on the class of the object that the entry describes.

The DIB entries are arranged in a tree structure called the Directory Information Tree (DIT). Entries reside at the vertices of the tree. Entries closer to the root of the tree represent more general objects such as organizations or countries. Entries lower in the tree may represent individual people or application processes. (In the DIT scheme, the root appears higher than the branches.)

Every entry has a distinguished name that uniquely and unambiguously identifies the entry. The distinguished name of an entry is made up of the distinguished name of its superior entry in the tree along with specially nominated attribute values from the entry.

Some of the entries at the leaves of the tree are alias entries, while other entries are object entries. Alias entries point to object entries and provide the basis for alternative names for the corresponding objects.

The Directory Service enforces a set of rules to ensure that the DIB remains well formed in the face of modifications over time. These rules, known as the Directory Schema, prevent entries having the wrong type of attributes for their object class, attribute values having the wrong form for the attribute type, and even entries having subordinate entries of the wrong class.

The growth and form of the DIT, the definition of the Directory Schema, and the selection of distinguished names for entries as they are added are the responsibilities of various authorities whose hierarchical relationship is reflected in the shape of the

© 1993 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

tree. The authorities must ensure, for example, that all of the entries in their jurisdiction have unambiguous distinguished names by carefully managing the attribute types and values that appear in those names. Responsibility is passed down the tree from superior to subordinate authorities, with control being exercised by means of a schema.

The Directory Service

All services provided by the Directory Service are in response to requests from DUAs. Requests from DUAs allow interrogation and modification of the Directory Service. In addition, requests for service can be qualified. The Directory Service always reports the outcome of each request that is made of it. The form of the normal outcome is specific to the request and is evident from the description of the request.

The Directory Service ensures that changes to the DIB, whether the result of a Directory Service request or by some other means, result in a DIB which continues to obey the rules of the Directory Service schema.

A User and the Directory Service are bound together for a period of time at an access point to the Directory Service. At the time of binding, the User and Directory Service optionally verify each other's identity.

Directory Service operations may be qualified through parameters, filters, and controls:

- Service controls—controls can be applied to the various service requests, primarily to allow the user to impose limits on the use of resources which the Directory Service must not surpass. Controls are provided on the amount of time, the size of the result, the scope of the search, the interaction modes, and the priority of the request.
- Security parameters—requests may be accompanied by information in support of security mechanisms for protecting the Directory Service information. Security parameters may include the user's request for various kinds of protections; a digital signature on the request; together with information to assist the correct party in verifying the signature.
- Filters—a way to express one or more conditions that an entry must satisfy in order to be returned as part of the outcome. This allows the set of entries returned to be reduced to only those relevant.

Directory Interrogation: The following services are provided to query the Directory Service:

- Read—a request aimed at a particular entry that causes the values of some or all of the attributes of that entry to be returned,
- Compare—a request aimed at a particular attribute of a particular entry and causes the Directory to check whether a supplied value matches a value of the attribute,
- List—a request that returns the list of immediate subordinates of a particular named entry in the DIT,
- Search—a request that causes the Directory to return information from all of the entries within a certain portion of the DIT that satisfy some filter, and
- Abandon—a request that cancels an ongoing request.

Directory Modification: The following services are provided to modify the DIB:

- Add entry—causes a new leaf entry to be added to the DIT.
- Remove entry—causes a leaf entry to be removed from the DIT.
- Modify entry—causes a sequence of changes to a particular entry. Either all of the changes are made, or none of them are made, and the DIB is always left in a state consistent with the schema.
- Modify relative distinguished name—changes the distinguished name.

Directory Protocols: There are two Directory protocols:

- Directory Access Protocol (DAP)—defines the exchange of requests and outcomes between DUAs and DSA.
- Directory System Protocol (DSP)—defines the exchange of requests and outcomes between two DSAs.

Each protocol is defined by an application context, each containing a set of protocol elements.

Directory Operation: The Directory Service is fully distributed. A Directory System Agent (DSA) is an application process that is part of the Directory Service whose role is to provide access to the DIB, to DUAs, and/or to other DSAs. A DSA may use information stored in its local database or interact with other DSAs to carry out requests. Alternatively, the DSA may direct a requester to another DSA that can help carry out the request.

The DUA interacts with the Directory Service by communicating with one or more DSAs. A DUA need not be bound to any particular DSA.

Sponsoring Organization

The majority of the work on Directory Service is focused in a collaborative group under both CCITT and ISO/IEC JTC1.

Recent Changes

Work on the Directory has continued at a frantic pace. The 1992 publication has been delayed due to the large amount of work required. The majority of changes have been to complete areas that were not done for the 1984 version.

GOSIP Consideration

Directory Services were not included in either GOSIP version and are optional in GOSIP Version 2. Vendors are rapidly providing implementations of Directory Services and are making them available in conjunction with MHS products. There are no conformance tests as yet. ■

This report was prepared exclusively for Datapro by James Moulton, president and principal consultant for Open Network Solutions, Inc. (ONS). Mr. Moulton has over 18 years' experience in data communication and telecommunication research and development. He has been involved in designing and standardizing the OSI protocols and has designed networks based on DOD protocols, TCP/IP, and GOSIP protocols. He is an active participant in international and national standards committees defining the network standards of the future. ONS, based in Sterling, VA, is a diversified company providing solutions to data communication and networking problems. It has designed major network systems, including WANs and LANs for both industry and government. ONS is heavily involved in researching the next generation of network protocols. It is experienced in both GOSIP and TCP/IP networks as well as the range of LAN operating systems.

© 1993 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

Upper Layer OSI Protocols

In this report:

Datapro Summary

The OSI Reference Model and the protocols based on it have been under development for ten years and are nearing maturity. Acceptance of OSI depends, in part, on the number of products that implement the protocols defined in the upper layers of the OSI model. The upper layers, the Presentation Layer and the Application Layer, provide the interface to the user's applications and present the external view of OSI to computing environments. The Presentation Layer provides the syntax for the information exchanged between applications. Both connection-oriented and connectionless protocols and services are defined under the Presentation Layer. The Application Layer is concerned only with the semantics of the information. The Application Layer consists of a number of protocols and services to meet the needs of users and designers. These are divided into three broad categories: Application Service Elements, Common Application Services, and Specific Application Services.

Introduction

The development of the OSI Reference Model and the protocols and services based on that architecture has been under way for ten years. A major barrier to the acceptance of OSI in corporate environments has been the lack of maturity in the protocols. It is only in the last few years that a sufficient level of maturity has been reached by a number of OSI protocols. Recognizing that OSI is reaching maturity, the federal government has mandated the use of OSI protocols through a Federal Information Processing Standard (FIPS) called the government OSI Profile (GOSIP). GOSIP compliance became mandatory in August 1991.

An important part of the acceptance of OSI is the availability of products implementing the protocols defined at the upper layers of the OSI Reference Model. The protocols at these two layers provide the interface to the users' applications. The uppermost layers of the OSI Reference Model present the external view of OSI to the computing environment. Applications are defined that incorporate the services provided by the lower layers. Without the Application Layer and the applications using the services provided, there would be no need for the OSI networks.

The upper layers of the OSI Reference Model consist of the Presentation Layer and the Application Layer. These two layers taken as a set provide both a syntactic and semantic framework for the application performing the distributed processing function. The Presentation Layer provides a method for negotiating a syntactic structure for the exchange of information between applications. The Application Layer defines a set of protocols tailored to the demands of the specific applications.

The concept of the upper layers is to provide a consistent interface from the network environment to the applications that require distributed computing services. For

—By James Moulton
President and Principal Consultant
Open Network Solutions, Inc.

© 1992 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

those applications that have common requirements, the Application Layer provides a single solution. For example, all applications that need a bulk data transfer can use the File Transfer Access and Management services.

OSI Concepts

Service Definitions

The OSI Reference Model presents the architecture for the exchange of information by distributed applications. The model describes a layered approach where each layer has a defined scope and set of functions.

Since the model is purely an architecture, at each layer a service definition describes the abstract interface to a layer protocol. The service describes the interaction of the protocol user to the protocol implementation. Since it is abstract, however, it does not define a specific protocol interface used for implementation. Also, many protocols can support a single service definition.

Protocol Specifications

For each service definition, one or more protocols can be defined. Each protocol will follow the service interface while providing different mechanisms and functions. Conformance can only be tested against the protocol specification.

Modes of Communication

The OSI Reference Model describes two distinct types of communication: connection-oriented and connectionless data transmission (called connectionless). At each layer of the model, there is both a connection-oriented and a connectionless service. Additionally, protocols supporting each mode of operation are defined.

Connection-oriented communication was the original focus of the OSI Reference Model. In this mode of operation, there are three phases of communication: connection establishment, data transfer, and connection release. The data transfer phase is simplified by maintaining sufficient state information. A typical example of this mode of operation is the X.25 virtual circuit. (In the case of the Application Layer, connections are renamed *Associations*. Association is used to describe the unique cooperation between the applications. The cooperation between the two applications manifests in the distinction that applications pass information that have semantic content. At the remaining layers, the data carries no semantic content.)

Connectionless Data Transmission was added to the model as a result of the emergence of datagram services at the Network and Data Link Layers. In a connectionless data transmission, there is only a data transfer phase. No state information is maintained, and each transmission is viewed as independent. Examples of connectionless mode transmissions are Logical Link Control (LLC) and the Connectionless Network Protocol (CLNP). For efficiency, the architecture requires that no segmentation or reassembly take place above the Network Layer. For that reason, the protocols at all of the higher layers are very simple. The majority of the functions involve address mapping from one layer to the next.

Presentation Layer

The Presentation Layer is responsible for the selection of a transfer syntax. The Presentation Layer deals with generic functions that are needed by many different types of applications, specifically, a common means of representing a data structure in transit from one system to another.

Connection-Oriented Presentation

Overview

The connection-oriented Presentation Layer Service and Protocol are responsible for providing a representation (syntax) for the information exchanged between applications utilizing connections. By establishing a connection, negotiations of transfer syntax are possible.

The Presentation Layer encompasses two aspects of the information representation:

1. the representation of data to be transferred between applications; and
2. the representation of the data structure to which applications refer in their communication, along with the representations of the set of actions that may be performed on this data structure.

The Presentation Layer is only concerned about syntax and not with the semantics of the information. The semantics is only known by the applications. If the syntax is not understood by both systems, however, the semantics cannot be determined.

Document Information

The work on this standard is a joint effort between ISO/IEC JTC1 and CCITT. The standard is defined in the following documents:

- ISO 8882 Information Processing—Open Systems Interconnection—Presentation Service Definition
- ISO 8883 Information Processing—Open Systems Interconnection—Connection-Oriented Presentation Protocol

Purpose

The Presentation Layer is concerned with the syntax or representation of information in transit between two applications.

The Presentation Layer has two main functions:

- negotiation of transfer syntaxes; and
- transformation to and from transfer syntax.

The function of transfer syntax negotiation is supported by the Presentation protocol defined in ISO 8883. The protocols provide presentation context definition facilities. These facilities provide a means of determining the precise manner in which information is encoded for transfer to the other presentation entity.

A major feature of the protocol and services available at the Presentation Layer is the ability to pass Session Services through to the application unchanged. This "pass-through" approach allows the application to control Session services such as synchronization and checkpoints.

A major concept introduced at the Presentation Layer is that of a context. A context is the definition of data structures, the operations that are valid over the data structures, and the encoding of the information for transfer. The selection, maintenance, and switching of contexts are the major functions of the Presentation Layer and associated protocols.

© 1992 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

The Presentation Layer protocol is important in that it allows the specification of information in a manner that is independent of the application. This allows systems that operate with disparate data structures and formats to exchange information in a meaningful way.

Functions

The Presentation Service Definition and Protocol Specification were completed in time for the publication in 1988. Since that time, modifications based upon extensions to the Session protocol have been incorporated.

The Presentation protocol is divided into functional units. Functional units are logical groupings of procedures for the purpose of:

- negotiation during presentation-connection establishment for subsequent use on the presentation-connection; and
- specification of conformance requirements.

The Presentation protocol consists of three functional units:

- kernel functional unit,
- context management functional unit, and
- context restoration functional unit.

The kernel functional unit is always available and supports the basic protocol elements of procedures. These procedures permit the establishment of a presentation-connection, transfer of data, and release of the presentation-connection.

The context management functional unit supports the context addition and deletion services. This functional unit is optional. The use of this functional unit is negotiated during connection establishment.

The context restoration functional unit provides additional functionality. The selection of this functional unit also requires the selection of the synchronization or activity services of the Session Layer.

Sponsoring Organization

The Presentation Layer standards are joint between ISO/IEC JTC1 (ISO) and the CCITT. The appropriate CCITT Recommendations are X.216 and X.226.

Connectionless Presentation Protocol

Overview

The connectionless presentation protocol is used in the connectionless protocol stack based on an application protocol through to the connectionless transport protocol. In the connectionless protocol stack, there are few functions available between the application and the Network Layer. The major functions available are address mapping and service mapping.

At the Presentation Layer, the protocol selects a context and transfer syntax based upon the needs of the application. There is no negotiation. If the destination presentation-entity is not capable of supporting that context, the communication fails.

Document Information

The connectionless presentation protocol is defined in the following documents:

- ISO 8882/1 Information Processing—Open Systems Interconnection—Presentation Service Definition, Addendum 1—Connectionless Mode Service Definition
- ISO 9596 Information Processing—Open Systems Interconnection—Connectionless Mode Presentation Protocol Specification

Purpose and Functions

The connectionless presentation protocol is based upon the needs of connectionless applications. The connectionless presentation protocol maps the address of the application to the appropriate session entity (SSAP). The presentation protocol does not perform segmentation or reassemble. It does not perform any error detection or error correction. The single function that the protocol performs is that of the selection of a context for the data transfer. The selected context is based on the requirements of the applications and the known requirements of the destination presentation-entity. The selected context may be selected through information supplied manually or gathered through system management.

Application Layer

The Application Layer is the highest layer in the OSI Reference Model. It is where the actual information and functions are generated for distributed processing.

Structure

The Application Layer contains a relatively large set of protocols and services to satisfy the needs of the various application users and designers. The services of the Application Layer can be divided into three broad categories:

1. Application Service Elements that are used by all applications and provide basic OSI services in support of communication.
2. Common Application Services that are used as required by other applications or users.
3. Specific Application Services that are used by a narrow class of users or other applications. These applications provide a specific service tailored to specific application needs.

The structure of the Application Layer is shown in Figure 1.

Virtual Terminal

Overview

The Virtual Terminal application is defined as a service (VTS, ISO 9040) and as a protocol (VTP, ISO 9041). The VTP is intended to provide a capability for exchanging information between terminal-oriented applications. With the VTP, it is possible to exchange information even when the terminal characteristics assumed by the applications are different. The VTP provides the mappings to ensure the successful exchange of information.

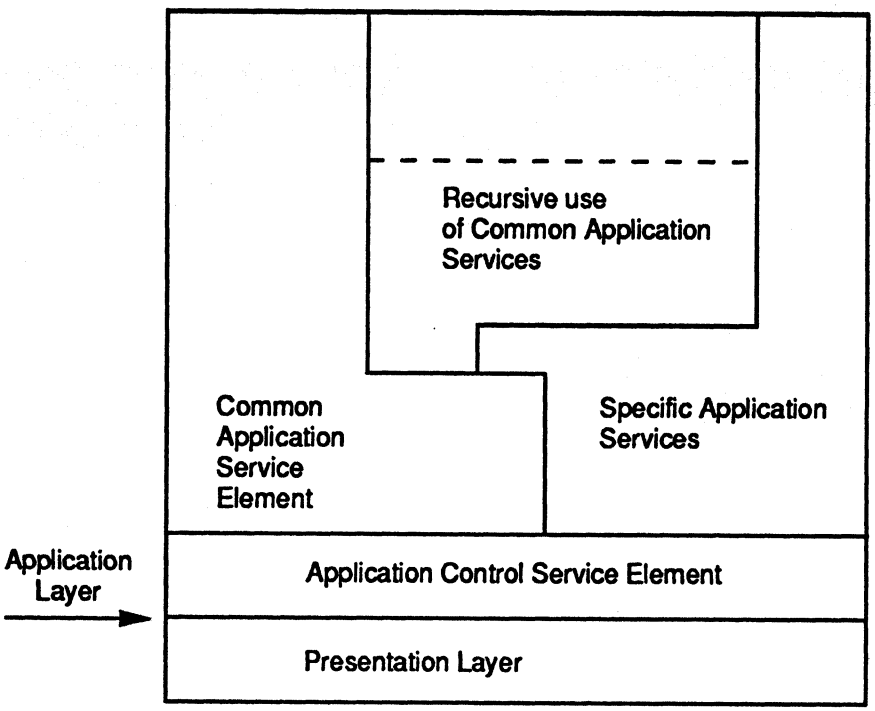
Document Information

Virtual Terminal application is defined in the following documents:

- ISO 9040:1990 Information Technology—Open Systems Interconnection—Virtual Terminal Basic Class Service

© 1992 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

Figure 1.
Application Structure



- ISO 9041-1:1990 Information Technology—Open Systems Interconnection—Virtual Terminal Basic Class Protocol

Sponsoring Organization

The sponsoring organization is ISO/IEC JTC1 SC 21 (International Organization for Standardization).

Purpose

The VTS and VTP define a method for interactive applications requiring terminal-oriented communication expressed in terms of the transmission and manipulation of graphical images having the following characteristics:

a) the images are composed of character-box graphic elements organized into a one-, two-, or three-dimensional structure; and

b) attributes may be associated with any graphic element to qualify its mode of display.

The Virtual Terminal Basic Class Service offers the following services to the VT-user:

a) the means to establish a VT-association between two peer VT-users for the purpose of enabling Virtual Terminal information exchange;

b) the means to negotiate the VT functional units required;

c) the means to negotiate a consistent set of VTE-parameters;

d) the means to transfer and manipulate structured data in a way that is independent of the local representation of information used by each VT-user and that is independent of the way in which supporting communications media are used;

e) the means to control the integrity of the communication;

f) the means to terminate the VT-association either unilaterally or by mutual agreement;

g) the means to support either synchronous (S-mode) or asynchronous (A-mode) operation between the VT-users;

h) the means to exchange priority information to gain the immediate attention of the VT-user;

i) the means to terminate information transfer destructively and resynchronize the activity of the VT provider;

j) a facility for defining blocks in a display object [Blocks functional unit];

k) a facility for defining fields in a display object [Fields functional unit, also uses feature in n)];

l) additional optional access-rules for control objects in S-mode [Enhanced Access-rules functional unit];

m) means to control the asymmetry inherent in typical use of these features [uses the feature in l)];

n) a facility for defining control objects with content consisting of multiple data elements or a single partially updatable structured data element [Structured Control Objects function unit];

o) a facility for controlling data entry to fields using new standard types of control object [uses the feature in n)];

p) a facility for storing and using update information in Reference Information Objects [RIO functional unit]; and

q) a facility for establishing a VT-association with the capability to switch between the modes of operation when the VTE is changed.

File Transfer Access and Management (FTAM)

Overview

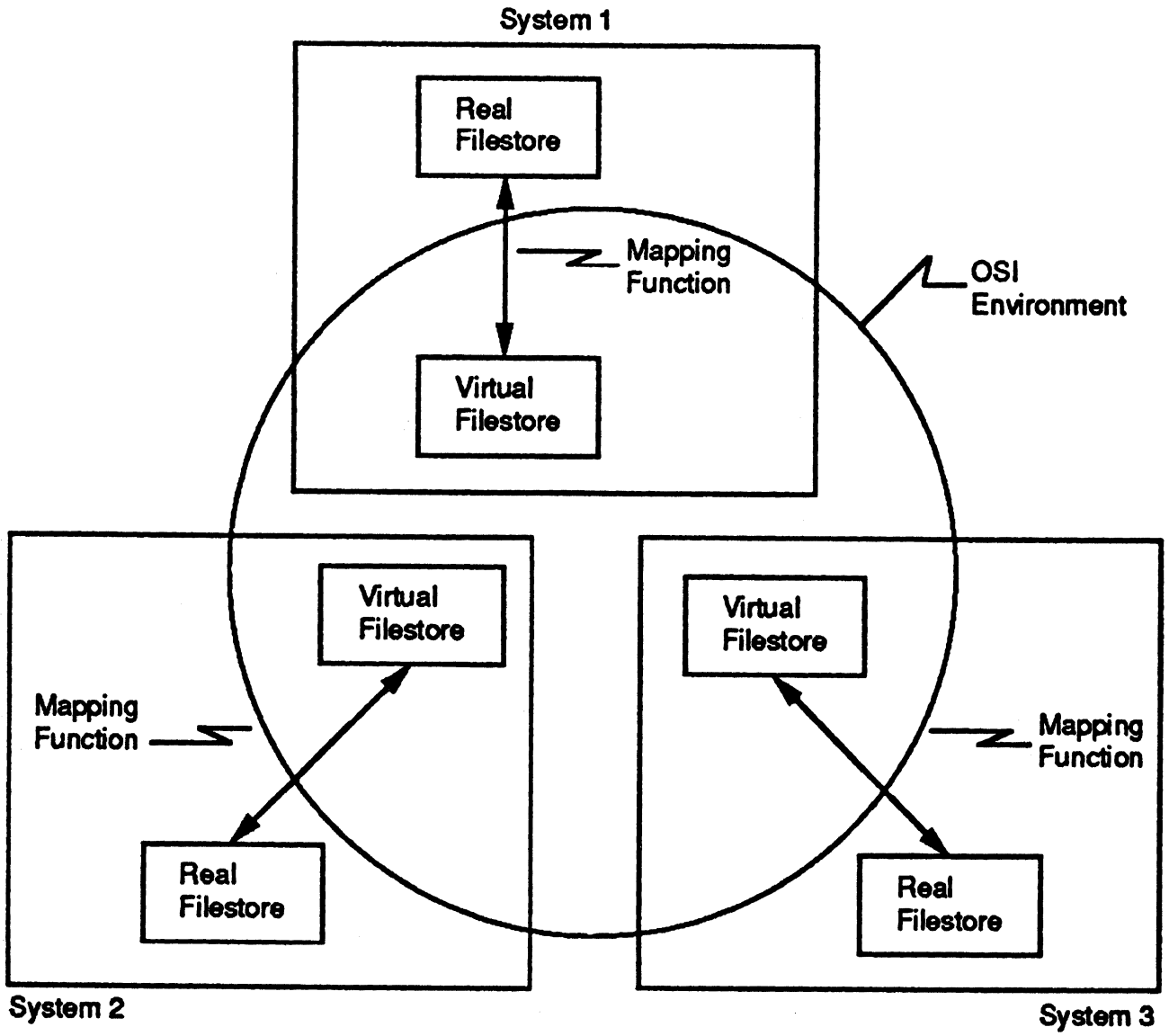
FTAM provides the OSI file management and transfer capabilities. In order to provide the complete range of file services across a heterogeneous network, FTAM is based upon a Virtual Filestore model. The model provides a consistent view of all files available in the FTAM environment.

FTAM consists of four parts. These parts are:

- Part 1—General Information

© 1992 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

Figure 2.
Virtual Filestore



- Part 2—Virtual Filestore Definition
- Part 3—File Service Definition
- Part 4—File Protocol Specification

The four parts define a complete application system. The system defines the filestore model, the services that operate over the filestore, and the protocol that exchanges the information.

Document Information

File Transfer Access and Management standards are defined in the following documents:

- ISO 8571-1 Information Processing—Open Systems Interconnection—Open Systems Interconnection—File Transfer, Access and Management—Part 1: General Introduction
- ISO 8571-2 Information Processing—Open Systems Interconnection—Open Systems Interconnection—File Transfer, Access and Management—Part 2: Virtual Filestore Definition
- ISO 8571-3 Information Processing—Open Systems Interconnection—Open Systems Interconnection—File Transfer, Access and Management—Part 3: File Service Definition
- ISO 8571-4 Information Processing—Open Systems Interconnection—Open Systems Interconnection—File Protocol Specification

Purpose

The FTAM standard provides for the transmission, access, and management of a variety of different file types and

© 1992 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

formats across OSI networks media, without detailed knowledge of the particular characteristics of the remote machines.

FTAM allows different applications or different users of applications to transfer information without specific knowledge of the other system's filesystem characteristics. FTAM also allows users a degree of control over the file activity, as well as a set of capabilities and features. Other applications may use FTAM as a supporting service. In fact, FTAM can be used locally as a set of callable library routines.

Functions

The FTAM standard is composed of four parts, listed earlier. The General Description presents the basic terminology and broad FTAM concepts. The File Service Definition gives an overview of FTAM services provided to the user. The Virtual Filestore document gives information on the central model used by FTAM. Finally, the File Protocol Specification gives a detailed description of the protocol interactions necessary to accomplish the FTAM activity.

Standards are often modified and enhanced through addenda. There are three addenda currently under development for FTAM:

- overlapped access,
- filestore management, and
- protocol conformance.

Overlapped access deals with read from and writing to different portions of a file simultaneously; filestore management involves an extensive set of directory commands, including search, list, and change directory.

The services of FTAM provided to the user are:

- the ability to communicate about files without specific knowledge of the other system,
- the facilities to express explicitly what the user requires,
- the ability to specify uniform file properties,
- the ability to specify record-level file access and positional file transfer, and
- detailed file management.

FTAM is a two-party file transfer protocol. A *controller* of the file activity (initiator) directs the action, and a *responder* responds to the initiator in a passive role. All file transfers and access operations occur between initiator and responder. An FTAM implementation may act as initiator, as responder, or as both.

FTAM is defined in terms of functional units and service classes. Service classes are described in terms of functional units; some of the functional units are mandatory within a service class and some are optional. The functional units in FTAM are:

- kernel,
- limited file management,
- enhanced file management,
- read,
- write,
- grouping,
- recovery, and
- restart.

For functional units, the kernel is the basic set of FTAM capabilities. Limited file management deals with the ability to create, delete, and interrogate properties of files. Enhanced file management deals with the ability to change file properties. Grouping allows concatenation of FTAM requests for efficiency.

Service classes are:

- the transfer class, which allows for the movement of files or parts of files between systems, placing emphasis on simple operation with a minimum of protocol overhead before and after the data transfer.
- the management class, which allows control of the Virtual Filestore by a series of independent confirmed service exchanges, but does not include file transfer mechanisms.
- the transfer and management class, which combines the features of the transfer class and the management class.
- the access class, which allows the initiating entity to perform a sequence of operations on the file access data units, providing for the manipulation of remote data.
- the constrained class, which leaves the selection of functional units to the designer of the distributed application, giving full flexibility of optimization, but no guarantee of a common functional kernel.

FTAM uses the concept of a Virtual Filestore. In the OSI environment, there is one conceptual representation of this Virtual Filestore model. In the actual systems, there may be multiple real filestore implementations based on the operating system. There must be a mapping between the real filestore and the Virtual Filestore. The nature of the mapping is a local issue.

The generic FTAM model is applicable to most FTAM systems in use today. All of the characteristics of the virtual filestore can be recognized and interpreted by OSI file systems so that the communication is through this model. Figure 2 shows the relationship between virtual and real filestores.

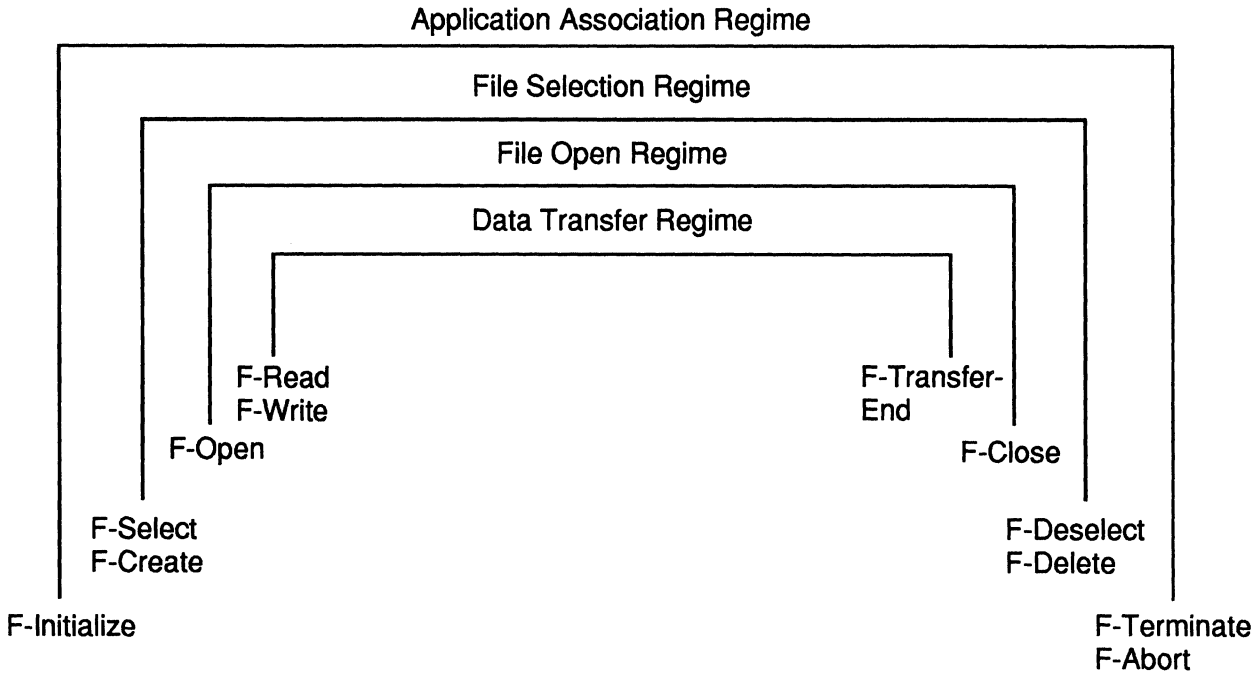
The FTAM services and protocol are organized into *regimes*. A regime is a phase of operation in which certain well-defined operations and events may occur. The regimes accomplish the following functions:

- allow the initiator and the filestore to establish information about each other, including their respective identities;
- identify the file that is needed;
- establish the attributes describing the file and bulk data transfer which is to take place in this activity;
- engage in file management;
- locate the position in the file access structure of the data to be accessed; and
- insert, replace, extend, or erase one or more complete FADUs (record).

The specific regimes are shown in Figure 3.

© 1992 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

Figure 3.
FTAM Regimes



A Virtual Filestore schema is composed of the following:

- a file, which contains file attributes and file contents,
- a filestore, which may contain a number of files, and
- a connection, which involves active attributes and current attributes. There is a user attached to the connection. The schema is hierarchical with a tree-like structure. Specific parts of a file are defined using node identifiers. Many different access structures are possible.

FTAM has a set of diagnostics that communicates information about the status of an FTAM request. There is a provision for users to include additional explanatory material where appropriate. FTAM has four classes of errors, from minor errors to errors which destroy the FTAM activity.

Two FTAM service types are defined. One type is called internal and supports the error recovery protocol. Errors are apparent to the file service user, and the user is allowed to directly control error recovery procedures. The other FTAM service type is called external; here the file service user has no awareness of error detection and recovery.

Distributed Transaction Processing (DTP)

Overview

DTP is a complex transaction processing system that allows for the execution of transactions across multiple systems.

A major feature of DTP is the control of commitment so that all systems participating in the transaction are ensured of consistent results.

The DTP model is that of a single client requesting the execution of a transaction from one or more remote systems. Each remote system may parcel out the transaction to other systems creating a tree structure. Transactions are phased operations. First, the systems participating in the transaction are "asked" if they can perform their assigned tasked. If any system responds with a "NO," the transaction fails. After a consensus is reached about the systems' capability to perform the transaction, the client requests execution. After receiving the response from all systems, a commit phase is entered which ensures that all participants have successfully completed the task.

Document Information

The DTP is at the Draft International Standard stage. It is now undergoing major editorial and technical revisions necessary to progress it to a standard. For this reason, DTP should be viewed as unstable. There should not be major functional changes in the services provided by DTP or in the operation of the protocol; however, detailed protocol mechanism changes can be expected.

- DIS 10026-1 Information Processing—Open Systems Interconnection—Distributed Transaction Processing—Model

Purpose/Function

Distributed Transaction Processing (DTP) provides a transaction service to its users. Specifically, DTP provides for the atomic execution of command/response pairs across multiple systems. DTP uses the features of other OSI protocols to ensure that each transaction is only executed when all component systems succeed. The use of commitment control allows the user to ignore the problem of coordination among multiple systems involved in the transaction.

© 1992 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

The functions of DTP are:

- transaction definition,
- distribution of processing functions,
- coordination of exuction responses,
- management of commitment and recovery, and
- reporting of results.

DTP is currently undergoing major reviews and changes. Details about the functions and procedures used may change substantially as an outcome of the meetings late in 1991.

Operation of DTP is as follows:

1. A DTP user submits a transaction,
2. DTP distributes the request to the systems participating in the transaction,
3. Participating systems report whether they can perform the function requested; recursively applies step 2; or reports failure.
4. If all participating systems report that the transaction will succeed, the transaction succeeds; if any system reports failure, the transaction fails.
5. If the transaction succeeds, then a commit is issued whereby all participants "save" the results.
6. If the transaction fails, then a rollback is issued whereby all participants return to the initial data.

Message Handling Systems (MHS)

Overview

The CCITT-defined MHS consists of a set of protocols that taken together provides a complete electronic mail system. The MHS, sometimes called X.400, provides for the submission of messages, the store-and-forward transfer of the message to the destination system, and the delivery of the message.

Document Information

MHS consists of the following series of documents:

- X.400—Message Handling Systems: System and Service Overview
- X.402—Message Handling Systems: Overall Architecture
- X.403—Message Handling Systems: Abstract Service Definition Conventions
- X.411—Message Handling Systems: Abstract Service Definition and Procedures
- X.413—Message Handling Systems: Message Store: Abstract-service Definition
- X.419—Message Handling Systems: Protocol Specification
- X.420—Message Handling Systems: Interpersonal Messaging System

Purpose

The MHS is designed to provide electronic mail services to users of public data networks. Since the standards are generally applicable, MHS has become prevalent in all areas of networking including LANs.

Networks providing MHS can interconnect with any other network that provides MHS services corresponding to the standards. Providing a common platform for the exchange of electronic mail messages is the central importance of the MHS series.

The main feature of MHS is the specification of Inter-Personal Messaging (IPM). IPM provides the syntax and semantics for the exchange of messages between human users.

MHS is a store-and-forward service. The message is transferred from system to system, and each intermediate system takes responsibility for further protection of the message. In a store-and-forward system, messages may wait for relatively long periods of time in each node.

Functions

The MHS allows users to communicate by exchanging messages. There are three major MHS components:

- the Message Transfer System (MTS),
- the cooperating User Agents (UAs), and
- the Message Store (MS).

The MTS is composed of a set of Message Transfer Agents (MTAs) responsible for relaying the message from the originator's UA to the recipient's UA. The MTA servicing the recipient need not be active when the message leaves the originator's MTA; the message can be stored at an intermediate MTA or in the MS until the recipient's MTA becomes operational. Intermediate MTAs can also perform Application Layer routing based on address information contained in the message.

The MTAs can be managed by different organizations or administrations. An administration is either the PTT service in a country, or in the United States, a common carrier recognized by the CCITT. The collections of MTAs and UAs owned and operated by an Administration is called an Administration Management Domain (ADMD). The collection of MTAs and UAs owned and operated by a private organization is called a Private Management Domain (PRMD). All ADMDs must comply with the CCITT Recommendations. PRMDs that wish to use a message transfer system provided by an ADMD must comply with the CCITT Recommendations at the point of interconnection.

UAs have many functions that are outside the realm of standardization. The originator's UA assists in the creation and editing of a message; the recipient's UA stores the message until the recipient chooses to read it and can use certain message fields to determine the display order. The message submission and delivery interaction with the MTA, however, are standardized.

The originator's UA must supply to the MTA the message content, the address(es) of the message recipients, and the MTS services that are being requested. The message content is the information that the message originator wants transferred to the message recipient. The information about the address and service request is placed on the message envelope and is used by the MTS to deliver the message.

UAs can be implemented either in the same system as the MTA or remotely located from the MTA. A remote or stand-alone UA can be under the control of an ADMD, a PRMD vendor, or an organization that provides no message transfer services. Since the UA-MTA message submission and delivery interactions involve a transfer of responsibility for delivering a message, there must be a protocol

© 1992 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

between the remote UA and MTA to ensure that the transfer of responsibility occurs.

There can be many different types of User Agents. The MTS can be used to transfer data unrelated to IPM. As long as the recipient's UA can interpret the data sent by the originator's UA, meaningful communication can occur.

The MTS does not examine the message content unless the UA requests that the content be converted from one format to another before delivery. CCITT recognized that, although there were many potential UAs that could use the message transfer services, the most common use of the MTS would be to send a personal message from an originator to one or more recipients.

Message Transfer System

The MTS provides the following basic service to UAs:

- Message Identification,
- Submission and Delivery Time Stamp,
- Non-Delivery Notification,
- Encoded Information Type Conversion, and
- Content Type Identification.

The following service elements can be selected by a UA on a per-message basis:

- Multi-Destination Delivery,
- Delivery Notification,
- Grade of Delivery,
- Deferred Delivery,
- Conversion Prohibition,
- Alternate Recipient Allowed, and
- Disclosure of other recipients.

InterPersonal Message Service

The InterPersonal Message Service (IPM) is provided by a set of cooperating UAs called IPM UAs. This service allows a user to send an interpersonal message to one or more recipients and to have it received by those recipients. The IPM service is built upon—and uses—the services of the MTS.

The interpersonal message contains a header and a body. The interpersonal message header contains service elements that facilitate efficient processing of the message by the recipient's UA. The body is the information that the message originator wishes to convey to the message recipient.

Sponsoring Organization

The majority of the work on X.400 (MHS) is carried out under a collaborative committee of CCITT and ISO/IEC JTC1.

Directory Services

Overview

Document Information

The standardization of directory services began as an effort within CCITT. At that time it was focused on providing directory services for X.400 MHS systems. Before the completion of the X.500 series, the work became a collaborative effort with ISO. All text of the X.500 Recommendations is joint with ISO. The associated International Standard is ISO 9594 parts 1 through 7.

- X.500 The Directory—Overview of Concepts, Models, and Services
- X.501 The Directory—Models
- X.509 The Directory—Authentication Framework
- X.511 The Directory—Abstract Service Definition
- X.518 The Directory—Procedures for Distributed Operations
- X.520 The Directory—Selected Attribute Types
- X.521 The Directory—Select Object Classes

The directory services work has progressed at a frantic pace since the acceptance of the Recommendation in 1988. Further refinements and extensions are expected to be completed by the close of the 1992 study period.

Purpose

The Directory Services provide directory capabilities required by OSI applications, OSI management processes, other OSI layer entities, and telecommunication services. Among the capabilities that it provides are "user friendly" naming whereby objects can be referred to by names that are suitable for citing by human users; and name to address mappings that allow the binding between objects and their locations to be dynamic.

The Directory Service is not a general-purpose database system. It is assumed that as is typical with communications directories, there is a higher frequency of queries than of updates.

It is a characteristic of the Directory Service that, except as a consequence of differing access rights or unpropagated updates, the results of directory queries will not be dependent on the identity or location of the inquirer.

Functions

The directory is a collection of open systems that cooperate to hold a logical database of information about a set of objects in the real world. The users of the Directory Service, including people and computer programs, can read or modify the information, or parts of it, subject to having permission to do so. Each user is represented in accessing the Directory Service by a Directory User Agent (DUA) which is considered to be an application-process.

The information held in the Directory Service is collectively known as the Directory Information Base (DIB).

The Directory Service provides users with a well-defined set of access capabilities, known as the abstract service of the Directory Service. This service provides a simple modification and retrieval capability. This can be built on with local DUA functions to provide the capabilities required by end users.

It is likely that the Directory Service will be distributed along both functional and organizational lines.

The Directory Service has been designed to support multiple applications. The nature of the application supported will govern which objects are listed in the Directory Service, which users will access the information, and which kind of access they will carry out. Applications may be very specific, such as the provision of distribution lists for electronic mail, or generic, such as the "interpersonal

© 1992 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

communications directory" application. The Directory Service provides the opportunity to exploit commonalties among the application:

- a single object may be relevant to more than one application; perhaps even the same piece of information about the same object may be so relevant.

To support this, a number of object classes and attribute types are defined that will be useful across a range of applications.

Directory Information Base

The DIB is a collection of objects. It is composed of entries that consist of a collection of information about one object. Each entry is made up of a set of attributes with one or more values. The types of attributes that are present in a particular entry are dependent on the class of the object that the entry describes.

The DIB entries are arranged in a tree structure called the Directory Information Tree (DIT). Entries reside at the vertices of the tree. Entries closer to the root of the tree represent more general objects such as organizations or countries. Entries lower in the tree may represent individual people or application processes. (In the DIT scheme, the root appears higher than the branches.)

Every entry has a distinguished name that uniquely and unambiguously identifies the entry. The distinguished name of an entry is made up of the distinguished name of its superior entry in the tree along with specially nominated attribute values from the entry.

Some of the entries at the leaves of the tree are alias entries, while other entries are object entries. Alias entries point to object entries and provide the basis for alternative names for the corresponding objects.

The Directory Service enforces a set of rules to ensure that the DIB remains well formed in the face of modifications over time. These rules, known as the Directory Schema, prevent entries having the wrong type of attributes for their object class, attribute values having the wrong form for the attribute type, and even entries having subordinate entries of the wrong class.

The growth and form of the DIT, the definition of the Directory Schema, and selection of distinguished names for entries as they are added, are the responsibilities of various authorities whose hierarchical relationship is reflected in the shape of the tree. The authorities must ensure, for example, that all of the entries in their jurisdiction have unambiguous distinguished names by carefully managing the attribute types and values that appear in those names. Responsibility is passed down the tree from superior to subordinate authorities, with control being exercised by means of a schema.

The Directory Service

All services provided by the Directory Service are in response to requests from DUAs. Requests from DUAs allow interrogation and modification of the Directory Service. In addition, requests for service can be qualified. The Directory Service always reports the outcome of each request that is made of it. The form of the normal outcome is specific to the request and is evident from the description of the request.

The Directory Service ensures that changes to the DIB, whether the result of a Directory Service request or by some other means, result in a DIB which continues to obey the rules of the Directory Service schema.

A User and the Directory Service are bound together for a period of time at an access point to the Directory Service. At the time of binding, the User and Directory Service optionally verify each other's identity.

Directory Service operations may be qualified through parameters, filters, and controls:

- Service controls—controls can be applied to the various service requests, primarily to allow the user to impose limits on the use of resources which the Directory Service must not surpass. Controls are provided on the amount of time, the size of the result, the scope of the search, the interaction modes, and on the priority of the request.
- Security parameters—request may be accompanied by information in support of security mechanisms for protecting the Directory Service information. Security parameters may include the user's request for various kinds of protections; a digital signature on the request; together with information to assist the correct party in verifying the signature.
- Filters—a way to express one or more conditions that an entry must satisfy in order to be returned as part of the outcome. This allows the set of entries returned to be reduced to only those relevant.

Directory Interrogation: The following services are provided to query the Directory Service:

- Read—a request aimed at a particular entry that causes the values of some or all of the attributes of that entry to be returned,
- Compare—a request aimed at a particular attribute of a particular entry and causes the Directory to check whether a supplied value matches a value of the attribute,
- List—a request that returns the list of immediate subordinates of a particular named entry in the DIT,
- Search—a request that causes the Directory to return information from all of the entries within a certain portion of the DIT that satisfy some filter, and
- Abandon—a request that cancels an ongoing request.

Directory Modification: The following services are provided to modify the DIB:

- Add entry—causes a new leaf entry to be added to the DIT.
- Remove entry—causes a leaf entry to be removed from the DIT.
- Modify entry—causes a sequence of changes to a particular entry. Either all of the changes are made, or none of them are made, and the DIB is always left in a state consistent with the schema.
- Modify relative distinguished name—changes the distinguished name.

Directory Protocols: There are two Directory protocols:

- Directory Access Protocol (DAP)—defines the exchange of requests and outcomes between DUAs and DSA.
- Directory System Protocol (DSP)—defines the exchange of requests and outcomes between two DSAs.

© 1992 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA

Each protocol is defined by an application context, each containing a set of protocol elements.

Directory Operation: The Directory Service is fully distributed. A Directory System Agent (DSA) is an application process that is part of the Directory service whose role is to provide access to the DIB, to DUAs, and/or to other DSAs. A DSA may use information stored in its local database or interact with other DSAs to carry out requests. Alternatively, the DSA may direct a requester to another DSA that can help carry out the request.

The DUA interacts with the Directory Service by communicating with one or more DSAs. A DUA need not be bound to any particular DSA.

Sponsoring Organization

The majority of the work on Directory Service is focused in a collaborative group under both CCITT and ISO/IEC JTC1. ■

This report was prepared exclusively for Datapro by James Moulton, president and principal consultant for Open Network Solutions, Inc. (ONS). Mr. Moulton has over 18 years in data communication and telecommunication research and development. He has been involved in designing and standardizing the OSI protocols and has designed networks based on DOD protocols, TCP/IP, and GOSIP protocols. He is an active participant in international and national standards committees defining the network standards of the future. ONS, based in Sterling, VA, is a diversified company providing solutions to data communication and networking problems. It has designed major network systems, including WANs and LANs for both industry and government. ONS is heavily involved in researching the next generation of network protocols. It is experienced in both GOSIP and TCP/IP networks as well as the range of LAN operating systems.

© 1992 McGraw-Hill, Incorporated. Reproduction Prohibited.
Datapro Information Services Group. Delran NJ 08075 USA