A Platform for Building Integrated Telecommunications Network Management Applications

Telecommunications companies today are faced with rapid technological change, large heterogeneous environments, and a greater need to provide customers with products that ensure reliable, cost-effective network service. This means that these companies need a platform that has a visionary strategy that enables them to develop standards-compliant network management solutions for a continually changing environment.

by Prabha G. Chadayammuri

The telecommunications industry is going through phenomenal growth and change. This growth has made telecommunications networks essential to the daily activities of the enterprise and individuals. It has also given rise to the need for better ways to manage and maintain heterogeneous and multivendor networks.

Network management includes the operations, administration, maintenance, and provisioning (OAM&P) functions required to monitor, interpret, and control a network and the services it provides. When networks started to be used beyond the academic community and before deregulation and privatization of the telephone industry, there were fewer vendors, thus fewer multivendor management issues. Also, the rate of introduction of new network technologies was much slower. These conditions meant that network management could be ad hoc and vendor-specific. Today, issues such as multivendor networks and equipment, the need to automate certain network management tasks, and the rapid integration of new technologies have driven the need to standardize telecommunications network management.

Since the early 1980s, the standardization bodies have been developing and specifying a collection of standards for managing telecommunications networks. A portion of these standards, dealing with open systems, is contained in the X.7xx series of standards defined by the ITU-T (International Telecommunications Union—Telecommunications). Another series of standards, the M.3xxx series from ITU-T, defines a model known as the Telecommunications Management Network (TMN).[1]

TMN is based on the Open Systems Interconnection (OSI) systems management model, which is set of standards that define the rules for processing and transferring data over networks.[2] Such systems are called *open systems*. Although not intrinsically part of TMN, OSI systems management standards were developed jointly by the ISO and ITU standards bodies.

All of these standards, no matter how worthy, are simply collections of well-written guidelines without a platform and tools to build network management solutions. Choosing a network management platform is a critical strategic decision that has long-term implications. The development of large-scale telecommunications management systems requires a significant investment of resources. Solutions, once deployed, will be supported for many years.

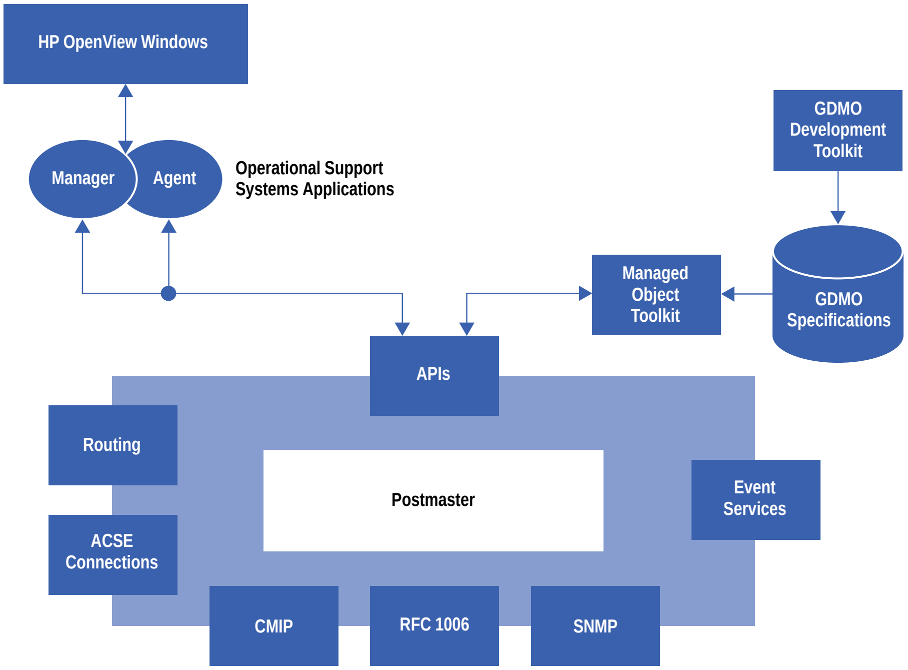
For equipment manufacturers and systems integrators, the network management foundation must enable rapid development of applications that can differentiate and add value to their products. For telecommunications service providers, the network management foundation must enable rapid deployment of new services that improve competitiveness and new operations that increase efficiency.

HP OpenView products provide the platform and enabling technologies required for network management solutions for today's telecommunications environment.

HP OpenView DM

The HP OpenView Distributed Management (DM) platform is a software platform for designing portable, standards-based systems for telecommunications management (see Fig. 1). HP OpenView DM products are focused on meeting the reliability, performance, distribution, and standards needs of telecommunications equipment manufacturers, service providers, and system integrators. The HP OpenView DM platform offers the following features for developing TMN applications.

Standards Support. The HP OpenView DM products support protocol, object, and service specifications defined by ITU, OSI, X/Open®, the Internet Engineering Task Force (IETF) for SNMP (Simple Network Management Protocol), and the Network Management Forum (NMF).[3]



CMIP = Common Management Interface Protocol
GDMO = Guidlines for the Definition of Managed Objects
SNMP = Simple Network Management Protocol

Fig. 1. *The main components of the HP OpenView Distributed Management Platform.*

There is also full support for network management protocols CMIP (Common Management Information Protocol), RFC 1006 (TCP/IP), and SNMP.[4,5]

Open Systems. The HP OpenView DM platform is built on an open systems architecture, enabling solutions to run on a variety of hardware platforms. Native support is implemented for HP 9000 workstations and servers running the HP-UX* operating system and Sun SPARC workstations running the Solaris and SunOS operating systems. Support for HP OpenView is also provided on other hardware and software platforms.

Postmaster. The postmaster serves as the integration point for management protocol stacks such as CMIP and SNMP, management APIs, and related facilities (e.g., routing, events, and association control). The postmaster provides distributed message routing and access to applications and services through standard management protocols. Finally, the postmaster reliably creates and manages associations (connections), maps objects to network addresses and protocol stacks, and routes requests from manager systems and responses from managed systems (agents).

Event Services. HP OpenView DM provides a set of services that management applications can use to control event and alarm messages from diverse network elements and systems. It includes a mediation service that collects, stores, filters, and extracts messages and an alarm management service that displays and correlates alarm messages and invokes external applications based on alarm data. Alarm management and event correlation services are described in ***Articles 3*** and ***4***, respectively.

HP Distributed Processing Environment (DPE). The HP DPE provides an Information Networking Architecture (INA) compliant platform for telecommunications services and operations systems. Trader services and an API framework simplify the development and deployment of distributed telecommunications applications. HP DPE is described in ***Article 2***.

Graphical User Interface. The HP OpenView windows graphical user interface (GUI) provides network operators and administrators with a consistent view of the managed environment and seamless integration of management functions, regardless of vendor or managed object type. HP OpenView windows provides a common interface that simplifies the development and use of management applications. Finally, the HP OpenView windows GUI is the key integration point for HP OpenView applications.

Modeling Toolset. The HP OpenView GDMO (Guidelines for the Definition of Managed Objects)[6] Modeling Toolset is an integrated suite of software tools for designing and analyzing objects used in network management applications. GDMO is an ISO standard that describes a consistent methodology for specifying managed objects in TMN applications.

The HP OpenView GDMO Modeling Toolset has a forms-based GUI that enables developers to create GDMO specifications and export them as ASCII files for use by the next application in the tool chain, the Managed Object Toolkit. The HP OpenView GDMO Modeling Toolset is described in ***Article 5***.

Managed Object Toolkit. The HP OpenView Managed Object Toolkit is a C++ code generator that accelerates the development of GDMO-based manager and agent applications (described below). The managed object toolkit includes an infrastructure that provides a collection of reusable objects that handle CMIS operations such as GET, SET, and ACTION.

Agent application development is improved because the Managed Object Toolkit takes the GDMO ASCII file and automatically converts the GDMO specification into an OSI-conformant, executable agent. The Managed Object Toolkit is described in ***Article 6***.

TMN Applications and HP OpenView

HP OpenView products have been adopted by many prominent equipment manufacturers and telecommunications service providers to implement a variety of TMN solutions. Some of the areas in which TMN applications can be built upon the HP OpenView foundation include:

- Services management for broadband networks including Synchronous Optical Network (SONET), Synchronous Digital Hierarchy (SDH), Asynchronous Transfer Mode (ATM), and residential services such as video-on-demand
- Provisioning and monitoring applications for broadband networks
- Network monitoring for outsourced customer networks managed by telecommunications service providers
- Customer gateways into public networks for real-time monitoring and data management
- Integration with other management platforms for TMN compatibility and a single view from a multivendor environment
- Element management systems for new equipment and new data communications services.

The HP OpenView DM platform has traditionally supported the OSI systems management model to provide TMN solutions. However, in recent years the Common Object Request Broker Architecture (CORBA)[7] from the Object Management Group (OMG) has attracted interest as a general model for distributed application development.

The combination of the CORBA and OSI models is an extremely powerful solution for TMN application development. Thus, HP OpenView DM platform development is moving in that direction.

The rest of this article will discuss various aspects of the TMN architecture and the OSI model and their relationship to the existing OSI-based HP OpenView DM platform and the evolving CORBA-based platform.

TMN Architecture

Fig. 2 shows the business, service, network, and element management layers of the TMN model and the interaction between applications in these different management layers. The functionality of applications in each of these layers is defined in ITU-T Recommendation M.3010.[1]

Network Element Layer. Functionality at this layer is provided by the network elements (e.g., switches, multiplexers, repeaters, hubs, terminals, etc.). These functions include operations such as performance data collection, alarm collection, protocol conversion, and so on. Applications at this level are responsible for managing network elements.

Element Management Layer. Functions at this layer are responsible for managing a subset of network elements, performing as a gateway to network elements in the upper layers, and keeping statistical and historical information about network elements.

Network Management Layer. Network management functions are used to support TMN applications that require a global view of the network. Data for this global view is collected from data summarized by the network element management layer. This layer is also responsible for the technical provision of services requested by the service management layer.

Service Management Layer. This layer is responsible for managing the services provided to customers. It provides the point of contact with customers for all service transactions, including billing, quality-of-service (QoS) data, service contracts, and so on.

Business Management Layer. This layer contains functions that are responsible for the whole enterprise. These functions include goal setting and budgeting, product planning and definitions, and agreements between jurisdictions.

Operation Systems and the Manager/Agent Model. The operations systems shown in Fig. 2 are integrated telecommunication management applications that implement the network management functions in the TMN layers. The operations systems are based on an agent/manager model. This model resembles the client/server paradigm in which applications in the manager role are clients and applications acting as agents would be servers. The agent/ manager model is also called a managed system (agent) and managing system (manager) architecture in TMN terminology. The agent/manager model is based on using objects to model the system being managed. Each object can have attributes that represent its state or relationship with other objects, its specialized behaviors (called actions), and notifications it issues to signal some event. Thus, an object encompasses a device’s behavior as well as its physical characteristics. An agent resides in an object and reports the object’s status to the manager. The manager, equipped with the capability to have a global view of the network, manages the agents and handles the notifications from agents.

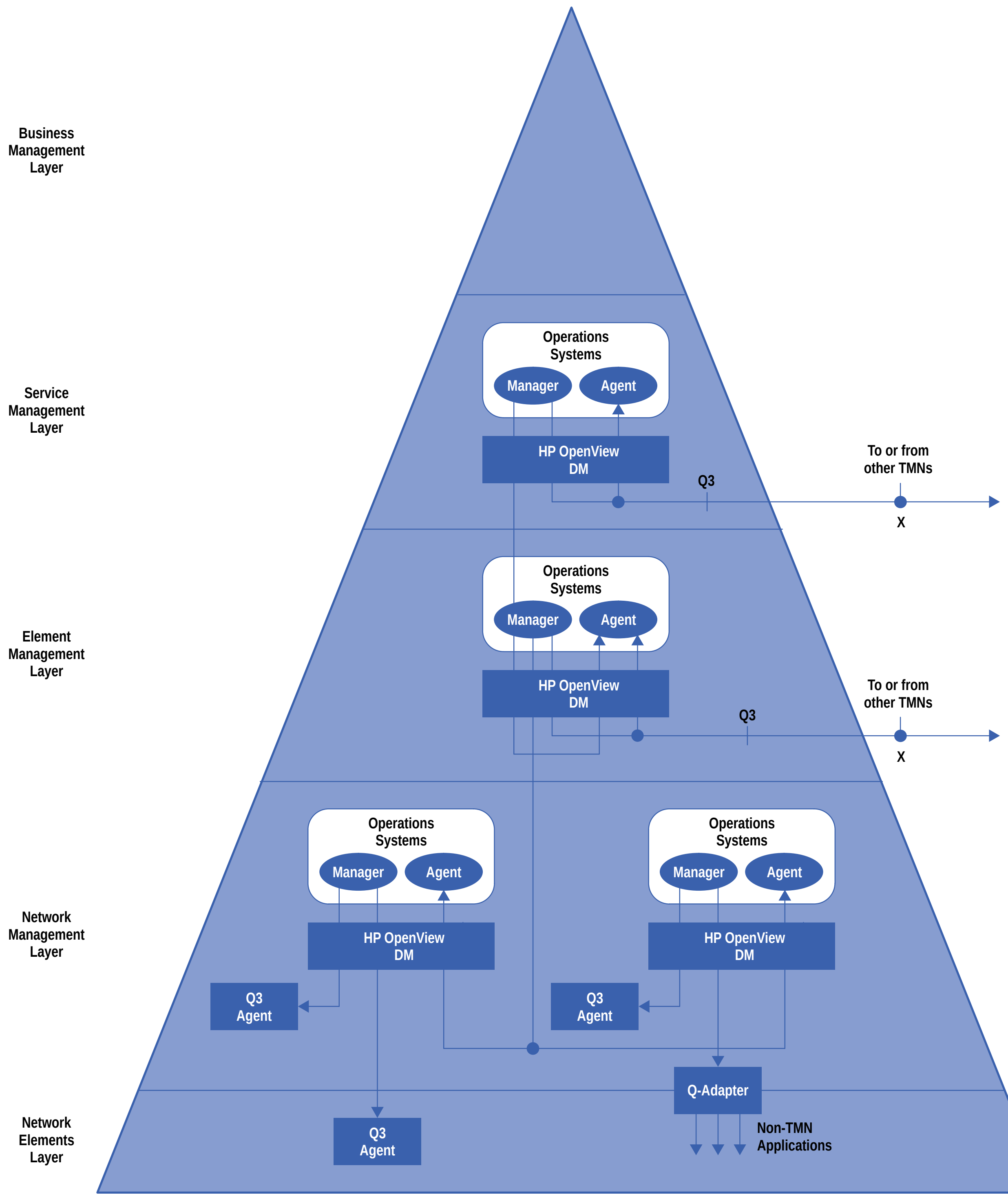


Fig. 2. *TMN architecture showing the network management layers and various TMN elements in each layer.*

Q3 Interfaces. Operations systems within and between TMN layers are required to use a set of standard interfaces called *Q3 interfaces* for the exchange of management information.[8,9] Q3 interfaces are responsible for connecting an operations system to a network element, an operations system to a Q adapter, an operations system to a mediation device, or two operations systems in the same TMN. Q3 specifications use the Common Management Information Service Element (CMISE) protocol[10] for management and the file transfer access and management (ftam) protocol for bulk transfer.

The standard way to convert a non-TMN function into a TMN function is called a Q adapter. Loosely stated, Q adaption is a translation between Q3 and the non-Q3 models at run time. Translation to a level less than Q3 requires a mediation device to raise the adaption to Q3 levels. The X reference points in Fig. 2 also perform an interface function. They provide an interface for communications with operations systems belonging to other TMNs or between TMN operations systems and non-TMN operations systems on other TMNs that support TMN-like interfaces. Q3 interfaces are generally regarded as appropriate for the X reference point.

The HP OpenView DM platform supports the APIs and protocols necessary for TMN applications. The HP OpenView DM platform provides the Q3 interfaces via the X/Open management XOM/XMP APIs and the C++ classes generated by the Managed Object Toolkit described in ***Article 6***. Faster APIs like the BER (Basic Encoding Rules) Management Protocol (BMP) and the generic data type dictionary APIs are available on the platform.[11] Application developers can build OSI applications using the APIs or the Managed Object Toolkit. The Managed Object Toolkit generates a complete application skeleton that can be customized by adding user-defined behaviors.

The OSI Model

As mentioned earlier, TMN is based on the OSI model and the HP OpenView DM platform supports the OSI model. In OSI system management, managed object classes are defined using GDMO (Guidelines for the Definition of Managed Objects). A managed object class has its state and relationships with other objects represented in its attributes, which can be accessed by GET and SET methods. The managed object class definition can have complex interfaces called actions and can specify notifications, which are emitted signal events associated with the object.

Abstract Syntax Notation One (ASN.1),[12] a data definition language, is used to describe the syntax of management data exchanged between objects. Behavior templates are used to define the semantics of operations on attributes and objects and are commonly expressed in natural languages. As a result, there is no standard way of parsing the behavior templates. The agent developer is allowed to implement the behaviors appropriately.

A managed object can be created or deleted by external commands if allowed by the object's GDMO specification. GDMO allows multiple inheritance, in which a given object can inherit all the operations, notifications, and behaviors of other objects. References 13, 14, and 15 provide many of the widely used objects, attributes, and notifications used in network management.

When defining new objects, these standard definitions are expected to be reused whenever possible. This is one of the challenging aspects of OSI object modeling. The GDMO Modeling Toolset, available on the HP OpenView DM platform, makes this task much easier. ***Article 5*** describes the GDMO Modeling Toolset.

Management Interactions

Fig. 3 shows the seven-layer OSI reference model.[2] Each layer has a clearly defined role in the transfer of information over a network. For systems management, the application layer is of the greatest interest. Applications interoperate with each other using application service elements (ASEs), which are defined by the application layer. The Common Management Information Service Element (CMISE), the Remote Operations Service Element (ROSE), and the Association Control Service Element (ACSE) are the most important ASEs used for systems management. The protocols used to implement these service elements are also defined as part of the ISO standard specifications.

OSI systems management operates like the agent/manager model described above. An application issuing management operations and receiving notifications is said to be acting in the manager role, and an application performing management operations and emitting notifications on behalf of managed objects is said to be acting in the agent role. An open system is made up of managed objects and the various processes involved in processing and transferring information.

A manager is expected to establish an association with an agent using the ACSE before attempting any management interaction. If the association goes down, both parties can detect it. When the association is set up, the manager and the agent exchange management information about their respective capabilities, including authentication schemes, encoding schemes, maximum data sizes, multiple object selection capabilities, and so on. These capabilities are called functional units. The HP OpenView DM platform supports both direct user control over association management and the automatic connection management mode in which the user does not have to be directly involved in the association management.

Once the association is set up between a manager and an agent, management information can be exchanged. The manager is allowed to perform CREATE, DELETE, and ACTION operations on the managed objects and GET and SET operations on their attributes as defined in their GDMO specification. The agent performs the operations on the managed objects on behalf of the manager and sends replies back to the manager.

The managed objects emit notifications (events) specified in their GDMO specifications. Notifications usually signify something of interest happening at the object, like its creation, deletion, or attribute value change. The agents deliver the notifications either directly to the manager or indirectly through event forwarding discriminators, which are managed objects that filter events coming from agents. This filtering ensures that only events of interest are received by the manager. The OSI-based HP OpenView DM platform supports the most generic form of event discrimination available today.

Another important aspect of OSI system management is that it is based on an asynchronous message passing model, as are most other network management protocols in use today. All operations can be classified into four primitives (or types): requests, replies, confirms, and indicates. These primitives are used in the following way:

1. To perform an operation, a manager sends a request message.

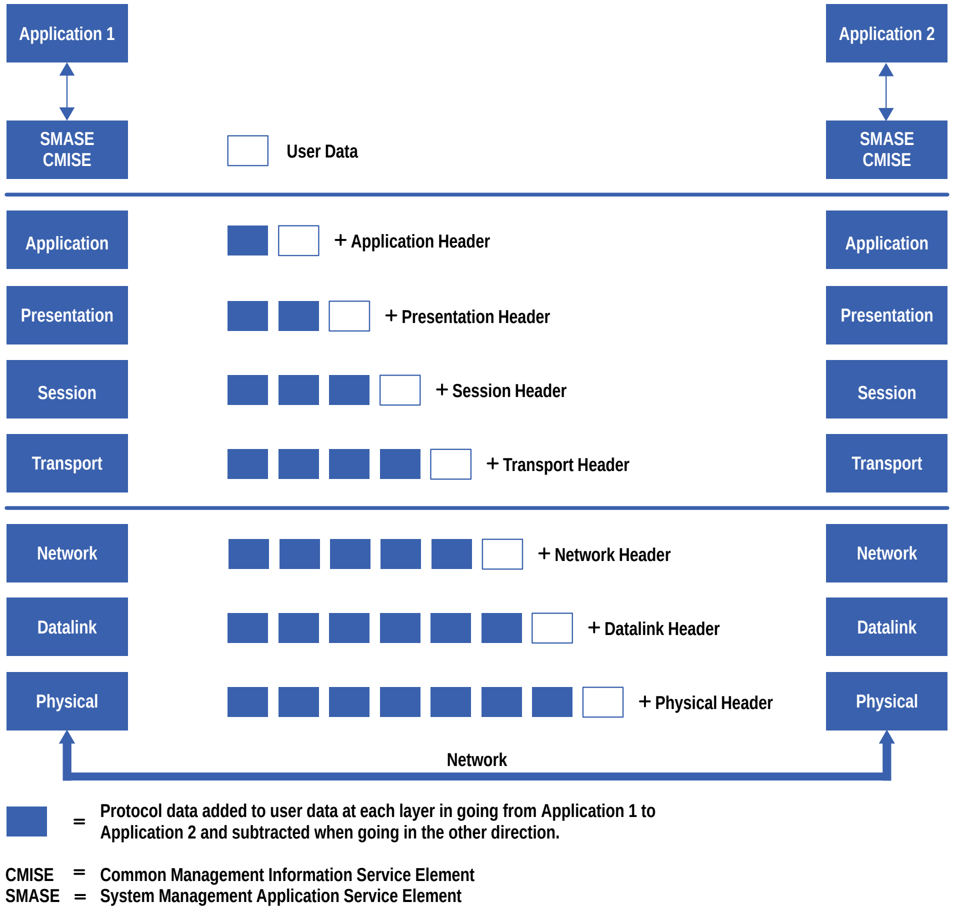


Fig. 3. *OSI stacks showing the significance of the OSI system management standards.*

2. When the message shows up at the agent, it is received as an indicate message.
3. Later, the agent may send a reply message.
4. The reply message is received at the manager as a confirm message.

The agent sends a reply message if the original request required a confirmation. The CMISE GET, CANCEL-GET, CREATE, and DELETE operations are always confirmed, whereas the SET, EVENT-REPORT, and ACTION operations can either be confirmed or unconfirmed. Request and reply messages are always directed outward from the application and indicate and confirm messages are always directed inward.

The Open Protocol Interface Architecture

Fig. 4 shows the HP OpenView DM postmaster with the attached API stacks, protocol stacks,[4,5] and intermediate stacks.[16,17,18] The postmaster is at the heart of the OSI-based HP OpenView DM platform. Applications bind to the postmaster processes running on different nodes. Postmasters on the different nodes coordinate interactions between applications bound to them.

The HP OpenView DM postmaster is built on an architecture known as the Open Protocol Interface, which is based on the OSI messaging model described above.

Messages flow into the postmaster either through the API stacks or through the protocol stacks. The processed messages that are sent out and need confirmations are kept on a sent queue awaiting confirmations. When the confirm messages come in they are matched with the corresponding request messages. This store-and-forward mechanism allows greater reliability in the message delivery. Flow control mechanisms are implemented to address congestion problems.

The X/Open management APIs (XOM/XMP) and the BER Management Protocol (BMP) are the API stacks. These and the CMIP, SNMP (RFC 1157), and RFC 1006 protocols are all supported by the OSI-based HP OpenView DM platform. This support, along with the requirements of association control and routing, provide the full complement of OSI conformance.

User-defined API stacks and protocol stacks can be added easily to the HP OpenView DM platform using the Open Protocol Interface architecture. New API and protocol stacks continue to be added by HP OpenView DM users. This flexibility allows easy integration of existing legacy applications into the management framework.

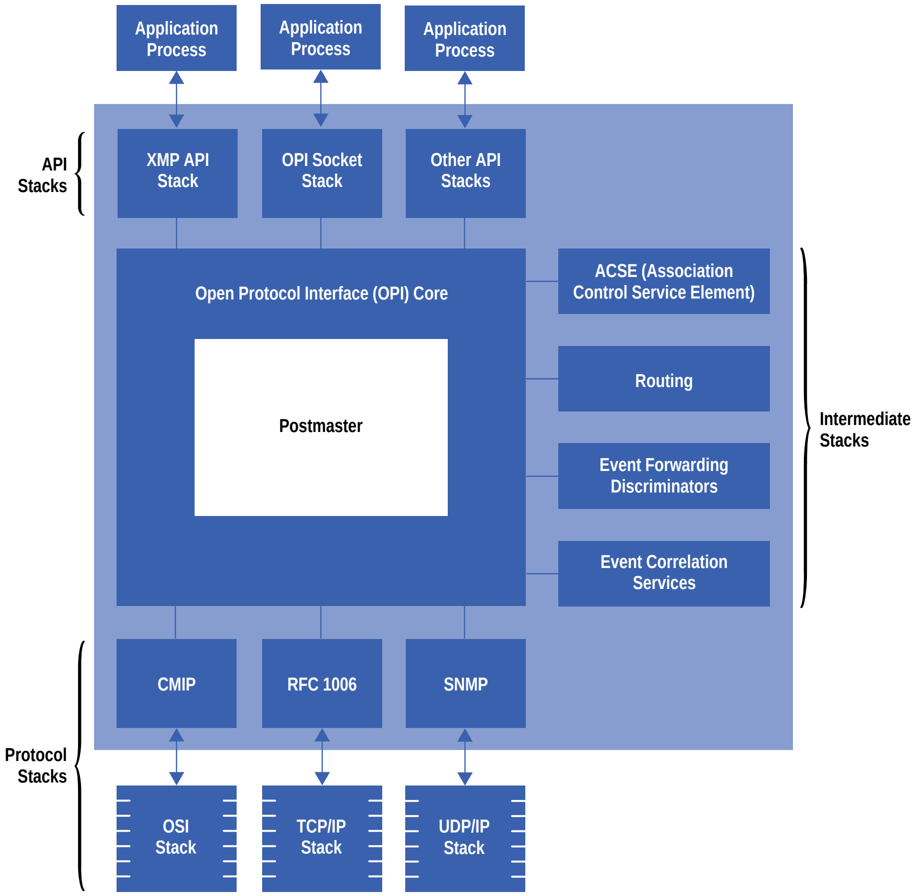


Fig. 4. *The HP OpenView DM postmaster showing the open protocol interface core and the attached API, protocol, and intermediate stacks.*

The intermediate stacks on the OSI-based HP OpenView DM platform are used to set up associations, determine routes, perform event forwarding discrimination, and so on. Each message is passed through its configured set of intermediate stacks for processing.

The intermediate stacks can also be used for data concentration or other similar purposes. This makes the Open Protocol Interface architecture ideal for building TMN mediation devices. For instance, the Event Correlation Service stack on the postmaster performs event correlation for events that pass through the stack.

Adding intermediate stacks is relatively trivial. This allows extreme flexibility in customizing the platform for specific needs. Consider, for instance, how user-defined security might be added to the platform. The Open Protocol Interface architecture presently allows security information (authentication token and authorization data) to be specified in each message. Today such information is regarded as opaque and is not interpreted by the stacks. If a user-defined intermediate security stack were added to the platform, the security information in the messages could be intercepted. The user stack could interpret the information and accept or reject the message, implementing user-specific behaviors.

The Open Protocol Interface development kit is separately available as an HP product.

Naming and Containment

To perform the operations and actions described above, there has to be a way of addressing the object instances. In OSI system management, each object instance has to have a unique name, known as the object's *distinguished name*. The uniqueness of the name is guaranteed by naming all objects with respect to a containing object or its parent instance. The only (virtual) object not contained in another is called the root. The relationship between the parent (superior) object and the child (subordinate) object is called containment.

Since every object instance (except root) is contained in its parent instance, an acyclic, hierarchical tree of object instances can be constructed. This is known as a *containment tree*. The idea of collecting objects based on containment is particularly useful in defining operations that apply to multiple objects. Such operations are called scoped operations in OSI systems management terminology. The CMISE GET, SET, DELETE, and ACTION operations can be scoped and result in the operations being applied to all objects that fall within the specified scope. Multiple object selection and actions make the OSI system management model far more powerful (and complex) than simpler models like SNMP.

The object instance name is required in all CMISE transactions. When automatic connection management is used, the HP OpenView DM postmaster uses a service called the *object registration service* to identify the target application for each request from its instance name. The object registration service allows users to configure the object location externally, enabling one to build highly scalable systems that provide complete location transparency.

Naming and containment are described in more detail in ***Article 6***.

CORBA-Based Application Development

So far, we have gone over the standards support and other features of the OSI-based HP OpenView DM Platform. Since most telecommunication resources today are modeled using a GDMO specified object, these applications tend to be in the element management layer of the Telecommunications Management Network.

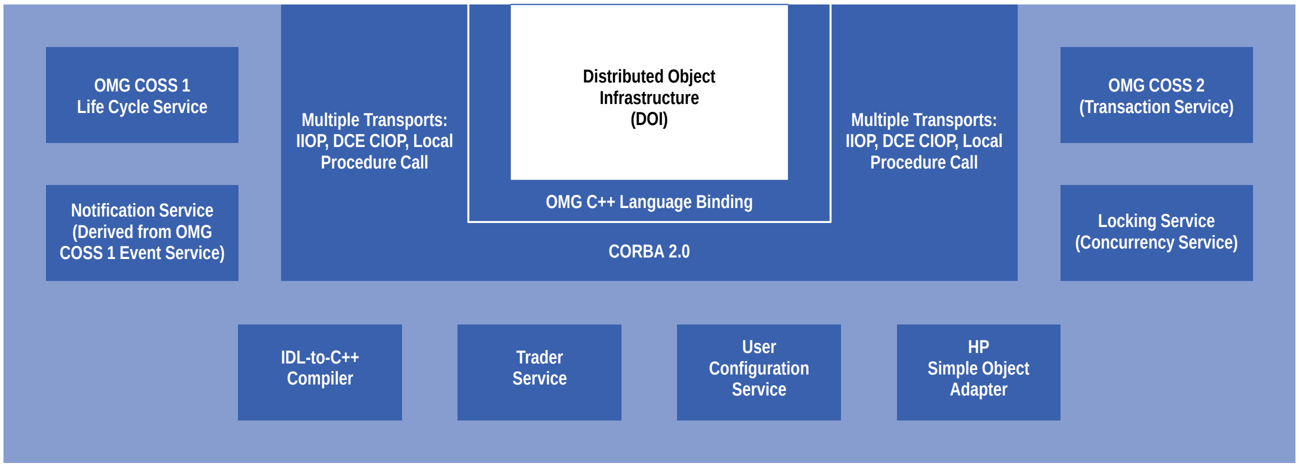
As we move up the TMN hierarchy (Fig. 1), the need for greater distribution, reliability, database access, and user interface access become obvious. TMN standards do not constrain the internal structure of applications. As a result, several nonstandard models are in use that need to be integrated into a single model to reduce costs.

The new HP OpenView telecom management platform addresses these specific issues with the use of the Common Object Request Broker Architecture (CORBA) from the Object Management Group (OMG). CORBA provides a highly scalable distributed object model. The OMG has a large industry participation and addresses all aspects of object modeling.

The new HP OpenView telecommunication management platform uses the HP ORBPlus distribution backplane for application interactions. HP ORBPlus supports the standard IIOP and DCE CIOP transports as well as a highly optimized local procedure call mechanism.

The OMG Common Object Service Specifications (COSS)[19] define a basic event service. Even though this service is implemented on the HP OpenView platform, it is not sufficiently robust for telecommunications management applications. HP OpenView, therefore, has developed a CORBA-based notification service,[20] which allows users to register with a notification manager for events filterable on multiple attributes.

The CORBA-based HP OpenView telecom platform also comes with the OMG naming and life cycle services, the OMG standard transaction service, and a location service, called the *trader service*. The collection of CORBA components and services, known as the HP OpenView distributed object infrastructure, is shown in Fig. 5.


OMG COSS 1
Life Cycle Service
Notification Service (Derived from OMG COSS 1 Event Service)
Multiple Transports: IIOP, DCE CIOP, Local Procedure Call
Distributed Object Infrastructure (DOI)
OMG C++ Language Binding
CORBA 2.0
Multiple Transports: IIOP, DCE CIOP, Local Procedure Call
OMG COSS 2 (Transaction Service)
Locking Service (Concurrency Service)
IDL-to-C++ Compiler
Trader Service
User Configuration Service
HP Simple Object Adapter

OMG = Object Management Group
CORBA = Common Object Request Broker Architecture
COSS = Common Operational Support Services
IDL = Interface Definition Language

Fig. 5. *The HP OpenView distributed object infrastructure showing the various standard services supported.*

The notification service for the distributed object infrastructure provides the same value in the CORBA-based platform as the OSI event-forwarding discriminator does in the OSI-based platform. The event-forwarding discriminators implemented on the HP OpenView DM platform are more suitable for Q3 notifications.

The CORBA-based OpenView platform also provides a more scalable version of the relationship service known as the topology service and a database strategy based on the industry standard ODBC (Open Database Connectivity) interfaces. The topology service enables the developer to define relationships between topological entities, which are the abstract objects corresponding to the elements in a network. The ODBC interface is a transparency layer that the X/Open Consortium developed to allow access to relational databases. This API allows a great degree of independence from specific databases, with a trivial performance loss.

With the availability of a CORBA-based platform, application development is made considerably easier. Object modeling is done in IDL, the OMG's Interface Definition Language. IDL has the same capabilities as GDMO and ASN.1 combined.[21] Also, the CORBA-based platform allows operations on remote objects to be just as easy as operations on local objects, although local object access would be faster.

In the model shown in Fig. 6, CORBA-based applications access Q3 and other objects using adapters, or HP OpenView DM APIs. Q3 adapters can be generic adapters that provide a CMIP interface in IDL,[22] adapters that follow the mappings from the X/Open Joint Interdomain Management (JIDM) task force,[23] or class-specific adapters that expose modified JIDM interfaces.[24]

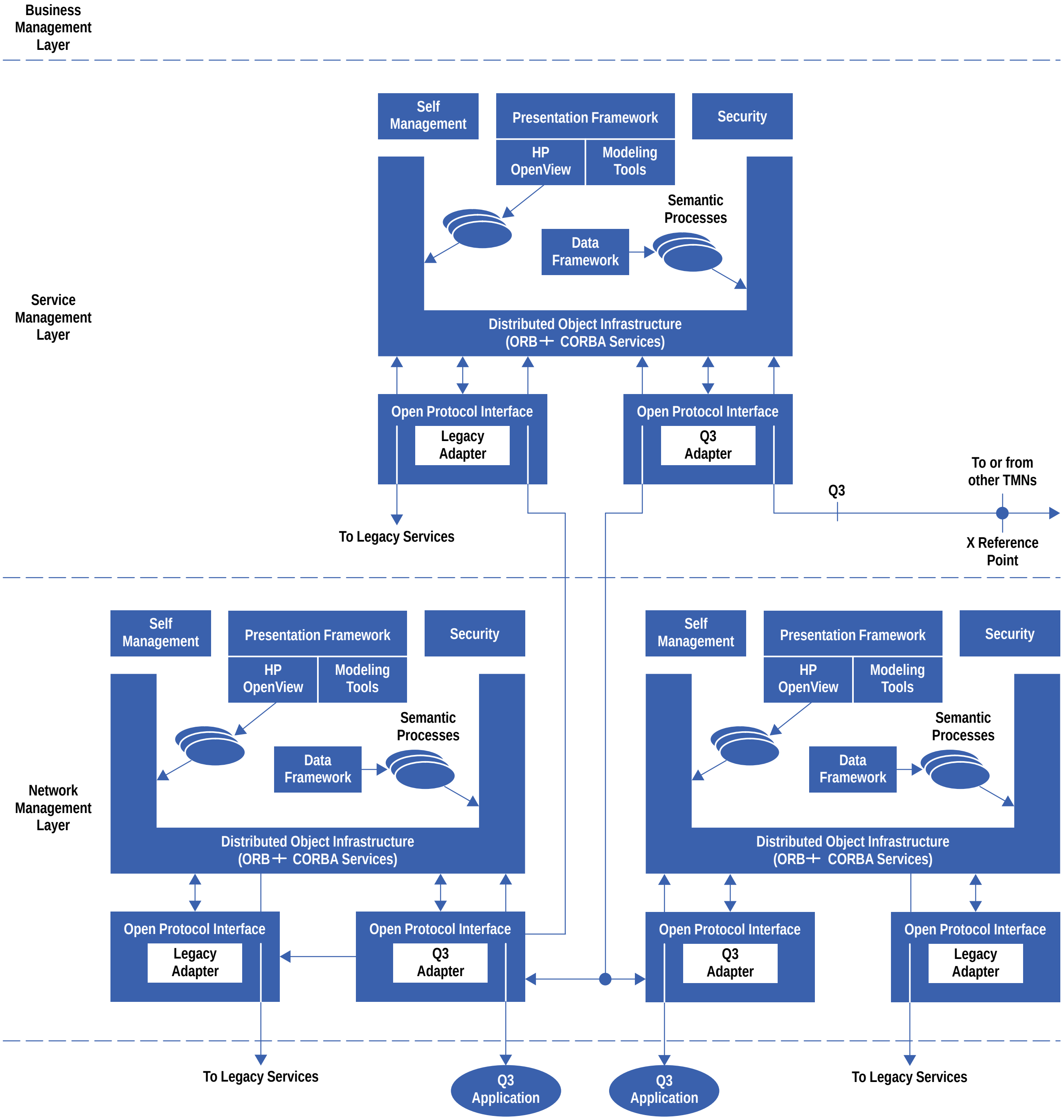


***Fig. 6.** TMN applications built with CORBA accessing Q3 and other object models.*

The JIDM adapters can be static or dynamic. The static adapters are built for specific GDMO Management Information Bases (MIBs) and are expected to offer better performance. The dynamic adapters use the generic facilities of the CORBA architecture (DII and DSI), which are more flexible than the static interfaces. The JIDM activity produces mappings for CORBA-Q3 interaction and CORBA-SNMP interaction. The CORBA-based HP OpenView platform will supply adapters after the standards in this area have stabilized. The Open Protocol Interface architecture discussed before is ideally suited for building Q3 and SNMP adapters to CORBA.

With the use of adapters, all other object models appear to be CORBA objects to the application developer. Applications use CORBA to gain distribution, standard language mappings, and common object services for portability and the topology services for data integration. The ODBC layer supports transparent access to multiple databases. In addition, a suite of enterprise management tools are available from other HP organizations that greatly enhance CORBA-based application development.

Summary

The new HP OpenView telecom platform combines the power of the CORBA model with the support for OSI management standards. As SNMP-based management gains acceptance in the telecom industry, the HP OpenView SNMP-based management platform will be integrated into the above model.

For pure Q3 access, developers today are encouraged to use the OSI-based HP OpenView DM platform. Q adapters, mediation devices, Q3 manager applications, and Q3 agents usually found in the TMN element management and network element layers fall into this group.

For highly scalable distributed applications requiring transaction processing, user interfaces, database access, and greater control over the quality of service usually found in the TMN network and service layers, developers should use the CORBA-based HP OpenView telecom platform.

See the ***Glossary*** for definitions of telecommunications terminology.

References

1. *Principles for a Telecommunications Management Network*, ITU-T Recommendation M.3010 (see also Recommendations M.3200 and M.3400), 1992.
2. *Open Systems Interconnection* (*Basic Reference Model*), ITU-T Recommendation X.200, 1994.
3. *Network Management Forum*, Forum 04, 1990.
4. *Simple Network Management Protocol*—RFC 1157, May 1990.
5. *Common Management Information Protocol Specification*, ITU-T Recommendation X.711, 1991.
6. *Guidelines for the Definition of Managed Objects*, ITU-T Recommendation X.722, 1992.
7. *The Common Object Request Broker—Architecture and Specification*, OMG Document 95-03-04, July 1995.
8. *Lower-Layer Protocol Profiles for the Q3 Interface*, ITU Recommendation Q.811, March 1993.
9. *Upper-layer Protocol Profiles for the Q3 Interface*, ITU Recommendation Q.812, March 1994.
10. *Common Management Information Service Definition*, ITU-T Recommendation X.710, 1991.
11. *BER Management Protocol*, HP OpenView 4.1 Users Manual, 1996.
12. *Specification of Abstract Syntax Notation One*, ITU-T Recommendation X.208, 1993.
13. *Definition of Management Information* , ITU-T Recommendation X.721, 1992.
14. *Generic Management Information*, ITU-T Recommendation X.723, 1993.
15. *Generic Information Model*, ITU-T Recommendation M.3100, 1992.
16. *Service Definition for Association Control Service Element*, ITU-T Recommendation X.217, 1992.
17. *Event Report Management Function*, ITU-T Recommendation X.734, 1993.
18. K.A. Harrison, *A Novel Approach to Event Correlation*, HP Laboratories, Bristol, England.
19. *Common Object Services Specification*, OMG Document 94-01-02, 1994.
20. *Event Notification Service*, COSS-1, OMG Document 94-01-02, 1994.
21. *Comparison of Object Models*, OMG Document 94-03-07, 1994.
22. E. Shen, M. Shan, and M. Robinson, *CMIP Interface—TC'95 Model*, HP Laboratories Research Report, 1995.
23. *Joint Interdomain Management Specification Translation*, X/Open Company, 1996.
24. *Class-adapter—Blanca Architecture Paper*, preliminary version, 1996.

▶ Go to Article 2

▶ Go to Table of Contents

▶ Go to HP Journal Home Page