OSI Systems and Network Management

Lakshmi Raman, ADC Telecommunications

ABSTRACT Data communications standards to allow exchange of information between two application processes in different heterogeneous computing environments have been developed by International Standards groups. With the development of these standards, the need for managing the communications protocols was realized as part of both the Internet and OSI standards suites. This article addresses the network management paradigm developed as part of OSI standards. The OSI network management application includes three different aspects: categories of network management, a protocol that specifies the structure for transferring network management information, and information models that define resource-specific management information for the specific management functions. These three aspects will be described in this article. Network management functions are grouped into five categories: configuration, fault, performance, security, and accounting. The resource is managed to accomplish these functions. These five categories have been used not only in OSI Network Management but also in specifying the management functions for Telecommunications network. These five categories are briefly discussed in the paper. The protocol structure for OSI network management is defined as an application service element known as CMISE. Regardless of the resource being managed, the protocol defines a basic set of operations applicable to network management. The article discusses the semantics of the various operations and the parameters associated with each operation. Using the structure defined by the protocol, for the various management functions, information is modeled to represent the managed resource. Object-oriented principles are used in defining information models. An introduction to these principles is provided. The management information exchanged is a combination of the three aspects. As part of OSI network management, information models to represent communication entites have been developed. An example is shown to illustrate the exchanged message for a management function. The article reiterates the three aspects and points out the advantages offered by this network management paradigm.

Data communications standards to allow exchange of information between two application processes in different heterogeneous computing environments have been developed by international standards groups. With the development of these standards, the need to manage the communications protocols was realized both as part of the Internet and open systems interconnection (OSI) suites of standards.

Historically, the suite of standards known as the X.700 series[1] were developed jointly by International Telecommunications Union Study Group 7 (ITU SG 7) and the International Organization for Standardization (ISO) to manage the OSI protocols in the end systems. Because the various components forming the management infrastructure are general, they have been applied not just to managing OSI protocols but to managing networks in general. The title of this article reflects this fact. However, this article describes these components in terms of managing an OSI protocol at a specific layer. An application of the principles explained in this article is found in a series of ITU Recommendations, commonly known as the telecommunications management network (TMN).

The OSI Reference Model defines protocol layers encompassing the functions required to enable successful communication between two systems for any application. Irrespective of the application, be it banking or directory white pages, it is possible to define and develop reusable communications infrastructure. Management of this infrastructure may be provided in one of three ways: protocol in a layer may include within itself management information; layer-specific management protocol may be defined; and management of the communicating entities and other applications may be considered as another application. The three cases are referred to as *layer operation*, *layer management*, and *systems management*, respectively. The focus of this article is on systems management, where application messages are exchanged in support of management functions.

The second section discusses the architectural principles or components that form the framework for OSI systems management. Several management functions have been defined and are grouped into areas or categories. These areas are presented in the third section. The Network Management Protocol that defines a common structure for the information exchanged between managing and managed systems is described in the fourth section. The OSI management framework has embraced the powerful object-oriented modeling approach that is gaining rapid acceptance in software development efforts. The fifth section provides an overview of these principles. Based on the several concepts introduced in these sections, it is often difficult for a casual reader to comprehend how all these play together to meet the final goal — ensuring that the communication infrastructure is adequate for network management. The sixth section brings these concepts together, building on the example used in the previous sections. While standards are complete to provide the necessary basic building blocks, work is in progress to extend the framework with concepts from the distributed processing environment. Some of these extensions are identified in the seventh section. A summary with the author's own thoughts and opinions on what lies ahead is provided in the last section.

ARCHITECTURE

CONCEPTS

The concepts described here are explained in detail in the "Systems Management Overview," ITU Recommendation X.701. An essential aspect of management refers to systems acting in manager and agent roles While it is common practice for a network management system (NMS) to be the manager

[1] *The equivalent ISO standards have numbers and form multiple series depending on whether they address architecture, protocol, rules for defining management information, or management functions.*

 0163-6804/98/$10.00 © 1998 IEEE

for the network elements being managed, the phrase "role" is used to denote that the manager-agent relationship applies to an instance of an information exchange. This facilitates the applicability of the various components described below to information exchanges not only between an NMS and an agent system but also between two NMSs. The communication component is based on this concept.

Management is an application that allows a system acting in the manager role to monitor and control the resources being managed. This is achieved by sending requests to the system acting in the agent role which has the responsibility for the resources. The management abstraction of the resources is known as a managed object, and the information component is based on this concept.

The concept of shared management knowledge is included to address interoperability between managing and managed systems. In order for the manager to perform the management functions, it is necessary that there be a common understanding of what the agent system supports and is capable of performing. Even though there are mechanisms to discover and learn the capabilities of the agent system, shared knowledge is essential to a successful exchange.

Three basic components that form the elements of the management architecture are described below. All these dimensions are essential to support a successful network management environment. The following sections provide a brief introduction of the three components and how they are supported by OSI systems management standards.

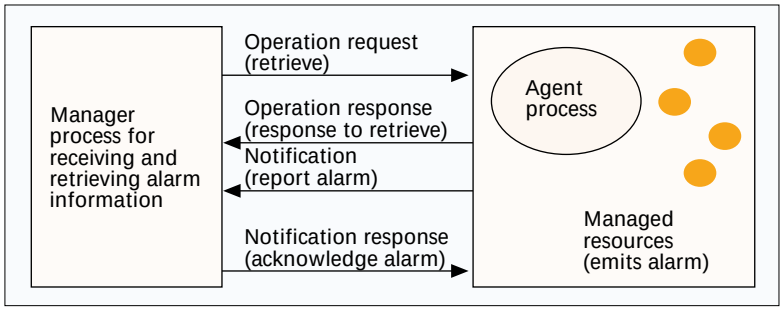

Figure 1. *OSI systems management overview.*

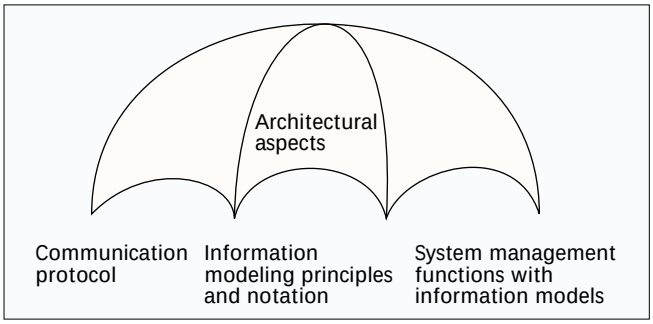

Figure 2. *OSI management document categories.*

The Functional Component

This component describes the various activities to be performed in support of management. X.700 has grouped the management functions into five areas. These are configuration, fault, performance, security, and accounting management. The reason for such a grouping was to facilitate rapid and consistent progress on each category in individual groups, and not to segregate NMSs for each area. Functions from one area will be influenced by others,[2] and a system may be implemented with *n* functions from different areas to meet the business objectives and market needs.

The Information Component

Management information exchanged between the managing and managed systems is dependent on both the function to be performed as well as the resources[3] to be managed. A major thrust of OSI systems management is to model the resources being managed. This implies that all the properties which can be monitored and/or controlled are defined in the model. The fifth section gives an overview of the modeling paradigm, using an object-oriented approach.

The Communication Component

The third dimension of facilitating a successful management interface between the roles of managing and managed systems is to have a well defined structure for the systems management protocol. The goal is to enable successful transfer and interpretation of management information. Communication requirements address support infrastructure capabilities such as reliable transfer and establishment of associations between application processes prior to management information exchange. The fourth section addresses this component.

OSI Systems Management Maze

An article on OSI systems management is not complete without the traditional diagram describing the manager and agent roles. Figure 1 presents a view that includes the three components mentioned above. The general figure is specialized for two alarm surveillance functions in the fault management area. The functional component is alarm surveillance. The communication component is addressed by requests and responses belonging to two categories: a manager role system issuing an operation request such as retrieve alarm information from a resource such as log or equipment, and receiving a response from the agent role system. The second category corresponds to different events emitted by the resources and are sent to the manager role system as notification. The manager system may acknowledge the receipt of the notification. The information component is represented by managed resources. For example, a resource such as a circuit pack may emit an alarm when it fails with a level of severity dependent on whether it is protected with another circuit pack or not. The information component defines these details.

These architectural components are expanded with detailed specifications so that interfaces between managing and managed systems can be implemented. A large volume of documents are available, and getting through this maze of documents may be daunting sometimes. However, these can be grouped in terms of four major areas, and understanding this structure may assist a newcomer to this topic. Figure 2 shows these categories.

The architecture documents in ITU Recommendation series X.700–X.703 (ISO 7498 Part 4 and 10040 Parts) provide the framework within which other details have been developed. The service and protocol specifications that define the structure for management messages are documented in X.710, 711, and 712 (ISO 9595 and ISO 9596 parts 1, 2). The principles and notation to describe the information are presented in X.720

[2] For example, performance degradation measured as a result of a performance monitoring function may result in reconfiguring a route to meet a defined quality of service.

[3] Resources include both physical aspects, such as the number of available ports, as well as logical aspects like the protocol entity.

series (ISO 10165 Parts). As one can expect, a large collection of documents exists to address the combined functional and informational aspects in the X.730, 740, and 750 series (ISO 10164 Parts). These include both requirements for individual functions and the information models (which essentially define the message exchanges indirectly) to support these requirements. This collection of documents form the foundation or infrastructure to build on when managing network elements that support different technologies and services.

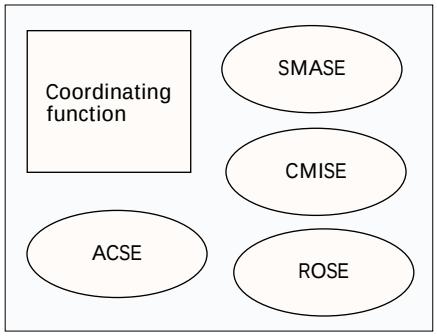

Figure 3. *System management application structure: ACSE, association control service element; CMISE, common managment information service element; ROSE, remote operations service element; SMASE, system management application service element.*

MANAGEMENT FUNCTIONS

As mentioned earlier, management functions are grouped into five areas: configuration, fault, performance, security, and accounting. In addition, this article also recognizes a sixth category because some of the functions available from standards cross the strict definition of these areas. Another way to view the common functions is "infrastructure support" for management. These areas are briefly discussed below.[4] Note that as part of TMN, operations functions have been identified and grouped into these five areas. The functions are identified in X.700 for OSI management and ITU Recommendation M.3400 for TMN.

CONFIGURATION

This functional area includes functions that allow a management system to provision resources and services, and monitor and control their state and status information. In addition, the agent may issue events autonomously when new resources are added/deleted or if the values of the properties change (as a result of internal operation or triggered by a managing system).

Examples of provisioning a resource include reserving bandwidth for a user by setting up a nailed-up connection. Service provisioning addresses assigning features requested by the user, such as call forwarding.

A generic state model (X.731) along with definition of events for transition between these states is defined and may be applied to different resources. Similarly, generic relationships are also defined.

FAULT

Fault management includes functions that address alarm surveillance, testing, and fault isolation. Alarm surveillance, as the name implies, allows reporting alarms with different levels of severity along with the possible cause of alarm. It also provides a summary of the alarms that are outstanding, and permits the manager to retrieve the alarm information. As part of OSI management, generic alarms that may be associated with various resources are defined.

A model for requesting tests, tracking the progress, and retrieving/reporting the results of the tests are defined. Different classes of tests, such as echo test and connectivity test, are defined and may be used based on the resource.

Fault detection and isolation functions permit the managing system to use techniques such as alarm correlation and diagnostics testing to determine the location and reason for the fault so that necessary corrective action may be taken.

[4] *The areas are defined using illustrative examples of functions. This is not meant to be exhaustive, only to provide a flavor of the type of functions in each area.*

PERFORMANCE

The performance management area includes functions to monitor performance parameters such as errored seconds, number of bad messages, collecting traffic statistics, and applying control to prevent traffic congestion. Monitoring and controlling the quality of service is another example of a performance management function. As part of this area, threshold values may be assigned for the parameters, and when the threshold is exceeded, events may be generated to inform the management system.

The performance parameters are dependent on the actual resource being monitored. For example, a transport technology using synchronous optical network (SONET) includes parameters such as loss of pointers, whereas for ATM the parameters monitored include the number of discarded packets. Even though the parameters may differ with the resource being monitored, generic mechanisms to gather data and to generate summary reports and statistics can be defined without being concerned with the actual resources. OSI management defines these mechanisms based on requirements (independent of resources).

SECURITY

Two aspects are to be considered as part of this functional area: management of security and security of management. It is essential to secure the exchange of management information. This is done by defining the security threats and services required to overcome these threats (e.g., access control, authentication, field level encryption). The communication protocol should have the ability to support securing the management exchange.

Securing the management exchange may be done at different levels using different mechanisms. Depending on the level of security, it is also essential to manage the security information. For example, one application may require that a specific parameter like a customer's charge card number should be encrypted using a public key method. In this case, the encrypted field addresses the topic of securing management information. In addition, it is necessary to define procedures for managing the key itself (what should be done if the private key is compromised). The latter is referred to as management of security.

ACCOUNTING

This functional area includes collecting usage data for the resources used in providing a service and then generating a bill, applying, for example, the tariff associated with the service. Here again, depending on the service, the usage information will vary. For example, a phone service often determines the length of time the connection was used versus a packet service which collects data on the number of packets sent. A general mechanism that can be specialized for a specific service is available as part of the OSI systems management standards. It is to be expected that while collecting the usage information and reporting on the values is subject to standardization, the generation of bills and application of tariffs in generating the bills are considered outside the scope of standardization.

COMMON

As mentioned earlier, even though only five areas are defined in the standard, this article introduces a common area to

address functions that cross the boundaries between these five areas. OSI management standards have developed several such functions. Examples include defining controls for emitted events to be forwarded to a management system; logging the events for later retrieval; and mechanisms to share and retrieve management knowledge.

A common function that is used extensively in all areas is controlling event reports irrespective of the event (alarm, threshold crossing alert, creation of a resource). The requirements of this function include setting up criteria for when an event is to be forwarded and what system(s) should be the recipient of the report. In addition, a schedule for forwarding the events and a list of backup system(s) (if communication with the primary system is lost) to receive the report may also be specified. Even though the actual criteria may differ with the type of event, the mechanism defined by the standard X.734 (ISO 10164-5) can be applied for all areas.

Name of service	Description
M-EVENT-REPORT	Reports an occurrence of an event from a managed resource. May be acknowledged.
M-CREATE	Requests to create the management view of a resource with specific values for the properties; response includes the result.
M-DELETE	Requests to delete the management view of resource(s); response(s) includes the result.
M-GET	Requests to retrieve values of the properties (attributes) of the managed object(s); response(s) include the results.
M-SET	Requests to modify values of the attributes of the managed object(s); responses may or may not be present.
M-ACTION	Requests that an action be performed by the managed object(s); the response if any is determined by the definition of the action.
M-CANCEL-GET	Requests to cancel an outstanding get request; response indicates the result of cancellation.

Table 1. *Services supported by CMISE.*

MANAGEMENT PROTOCOL

SYSTEM MANAGEMENT APPLICATION STRUCTURE

The application layer of the OSI Reference Model can be further structured based on the requirements of the application. OSI standards have been defined in a modular fashion so that maximum reuse is possible across different applications. The reusable unit is called an application service element (ASE).[5] The various ASEs are combined based on the needs of the application. For network management applications, the structure is shown in Fig. 3.

The building blocks required are: ACSE, to set up the association between peer application entities; and the combination of ROSE, CMISE, and SMASE, including the information models for management information data transfer. The coordinating function represents the logic required for these ASEs to cooperatively work together. The network management protocol defined by CMISE is discussed in the next section. The supporting mechanism used by the CMISE is the request reply paradigm defined by ROSE.

PROTOCOL INFRASTRUCTURE

CMISE defines a basic structure suitable for all network management areas mentioned in the third section. In accordance with the style of defining an application-layer standard, CMISE consists of a service definition and a protocol specification to support the services.

CMISE assumes a reliable transfer mechanism, and hence an association is established using a connection-oriented transport protocol. The reliability offered is not without cost. Before management data can be sent, an association must be established. Resources are also dedicated for the period of association even if there is no management traffic.

[5] *A more complex description of application layer using object-oriented concepts has become available recently. For simplicity this article does not include these extensions, and these are not necessary to gain a basic understanding of the overall structure.*

Let us now look at the common network management services. As shown in Fig. 1, the services belong to two classes: manager-driven requests for operations and corresponding responses, and autonomous reporting from the agent to inform the manager of the events. Table 1 summarizes these services. "M-XXX" indicates that these are management services.

As noted in the description, get, set, create, and delete services are the basic database operations. The action service is used in cases where set is not suitable. An action instead of a set is applicable when parameters of the action request are not modeled as attributes of the resource (a set operation is defined only on attributes of an object).

Without going into details, it should be noted that the power of CMISE stems from features, referred to as *scoping* and *filtering*, which can be applied to get, set, action, and delete operations. As indicated in the description column, these services can be requested to be performed on one or more managed resources. The scoping feature is used to select the candidates for performing a request. Irrespective of whether the operation is directed at a single or multiple objects, the filtering capability enables establishing a criteria as a logical expression. An operation request is performed if the managed resource meets the criteria.

These services and features are collected into groups of functionality referred to as *functional units*. It is possible to negotiate the use or otherwise of these features for each association between the managing and managed system.

The protocol structure supporting these services is composed of the following components: a sequence number to correlate requests and responses; the identity of the resource (by type or class and instance name); and information pertinent to the requested operation. In other words, the paradigm used here is to define a generic set of management operations suitable for all management functions and resources, and model the resources in terms of the properties that can be managed with these operations. As a result, the major emphasis in OSI management is on information modeling, which is discussed in the next section.

INFORMATION MODELING

The topic of information modeling (even within the limited context of network management) is too extensive to be covered adequately in this article. Therefore only the major concepts are discussed here. The approach used for information modeling stems from the concepts first developed as part of

object-oriented analysis and object-oriented programming.

MODELING RESOURCES

Resource Properties — A managed object presents a view of the resource to the management system with properties that are manageable. A resource such as a line interface card is used to provide telecommunication services. However, only some aspects of the line card are manageable by a management system. These properties are reflected in the managed object that represents the line card.

Several instances of line cards can be considered to belong to a class or type called the *line card class*. This class includes properties that are present in various instances. The terms *managed object class* and *managed object instance*[6] are used to denote the management abstractions of the type and instance of a type. Properties are grouped together into packages to facilitate reuse of the specification. When an instance of a class is created, the association between the package and the property is not visible. A given package may be mandatory or conditional. A mandatory package implies that all instances of the class are required to support the properties defined for that package. The conditional package may or may not be supported by an instance, depending on whether the condition evaluates to true or not.[7]

The properties are modeled using the following constructs: behavior that explains how the object behaves as a whole; attributes that reflect properties (e.g., state of a line card, type of services supported, required signaling protocol); actions that may be requested on the resource (perform diagnostic test on the line card); and notifications that may be emitted (failure of the card, loss of communication). In defining the attributes, the allowed operations are also included. For example, the state that denotes if the card is working or not will be read only. If the line card can support different services, it is possible to set the card to support a different service (this can imply more than just changing a value since there will be other systemwide impacts/effects that must be described). In addition to describing the semantics of the property, if it is exchanged on an interface, it is necessary to associate a syntax so that the management system can interpret the information unambiguously. The syntax of the information is specified using a programming-language-like notation known as Abstract Syntax Notation One (ASN.1). The information model defining the resources themselves are represented using a notation referred to as "Guidelines for the Definition of Managed Objects" (GDMO).

Two concepts of importance in defining information models using object-oriented methodology are inheritance and encapsulation. With inheritance, a generic managed object class may be defined first and specialized later with additional properties. The relationship between the generic and specialized classes is called super- and subclasses. A subclass inherits the properties of the superclass and adds to it.[8] A subclass may be defined to inherit from more than one superclass (multiple inheritance). This concept allows building models that reuse existing specifications, and facilitates software reuse. With encapsulation, an object is responsible for its integrity, and the internal operations are not visible. The interface to outside is that messages are sent to the object boundary, and if the requested operation will result in compromising the integrity of the object, it will not be performed. Behavior definitions should capture these constraints or conditions. In some cases additional constraints may be imposed based on the real resource being represented by an object defined in the standard.

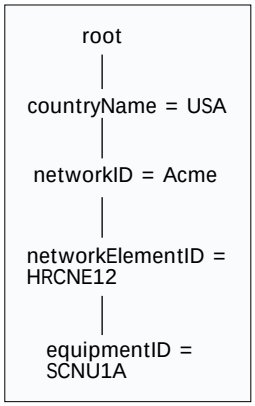

Figure 4. *Naming releationships.*

Detailed definitions of the principles and notation for representing them are described in X.720 and X.722, respectively.

Initial principles for modeling the management views of a resource included the relationship between resources implicitly. As an example an attribute may be used to point to the related object. Even though a relationship can be identified in terms of properties of a resource, all aspects of a relationship were not modeled. In other words, treating a relationship as a separate construct independent of how it is represented was lacking with the object-oriented principles. Additional modeling principles were added so that a relationship can be defined with all its properties, such as the type of resources that participate in the relationship, the cardinality among the participants, and rules for resources to enter and exit from relationships. These principles and representation notations are described in the General Relationship Model (GRM — X.726). GRM has not been used extensively as the recommendation was approved recently. However, new standards for the network-level model for transmission networks and functions, such as management domains are specified using these principles.

Managed Object Identification — Referencing a managed object in OSI management follows the scheme developed by the directory standard in X.500. The goal is to identify the managed object in an unambiguous manner. The unambiguity is achieved by uniquely referencing the managed object, referred to as the *subordinate object*, relative to its containing object (superior object). By recursion, the name of a managed object is globally unique. Consider a circuit pack in a network element called HFCNE. If this is deployed in a network called Acme in the United States, the globally unique name of the managed object is {countryName = "USA," networkID = "Acme," networkElementID = "HFCNE12," equipmentID = "SCNU1A"}. Figure 4 shows pictorially how a managed object is named uniquely.

In this example, the object country with the attribute country name is an X.500 directory defined object class and is used as the superior for a network which is defined as a managed object. The globally unique name is achieved by assuring that relative to the superior the name is unique. The term *relative distinguished name* (RDN) refers to the identification relative to the superior. For example, equipmentID = "SCNU1A" is an RDN relative to a specific network element.

OSI systems management specifies in the protocol three forms for referencing a managed object: global name (also known as distinguished name), which is the complete sequence of RDNs starting from root, and a local name, which is unique relative to a context. In the above example, once an association is established with HFCNE12, all references can be made relative to this starting point. The context is well defined; hence, using (equipmentID = "SCNU1A") is enough

[6] *It is not required to use instance. Managed object by definition refers to an instance of a class.*

[7] *The condition may be such that it translates to the support being optional for an implementation.*

[8] *Some OO methodologies allow modification of properties. This is not permitted here.*

7	0	Managed object class = circuitPack	Managed object instance = {equipment ID = "SCNU1A"	Event time = 1997030504 30	Event type = equipment alarm	Severity = major	Reason = board failure	Protecting unit = {equipment ID = SCNU1B}

Figure 5. *An example of NM information exchange.*

to refer to the resource unambiguously. Note that the local name does not include the name relative to which the context is set. The third form is any arbitrary string and does not use the scheme mentioned above. Even though the protocol supports this form, it is not used in the modeling principles.

Management Information Base and Tree — A collection of managed objects and the properties implemented within a system using the schema defined by the information model is referred to as a management information base (MIB). The naming scheme for identifying the management objects results in a tree referred to as a management information tree (MIT). The tree structure lends itself to the application of scoping feature where a group of objects may be identified in a single CMIP request. This is done by identifying the start of a search (base object) and the level of objects for selection.

Using this paradigm several information models have been developed in standards groups — American National Standards Institute (ANSI), European Telecommunications Standards Institute (ETSI), ITU, and ISO — as well as consortia such as the Network Management Forum (NMF) and ATM Forum. Other public specifications are also available in Bellcore Generic Requirements.

PUTTING IT ALL TOGETHER

Several concepts have been introduced at a high level in the previous sections. It is difficult to provide an in-depth explanation of these concepts to the reader in a short article. This section gives an overview of the steps required to exchange network management information in this paradigm based on the various aspects presented earlier.

ASSOCIATION SETUP AND RELEASE

In this approach, prior to transferring any network management information it is first necessary to set up an association between the application entities in the manager and agent systems. The context for the information exchange, such as whether the data is relevant to network management versus banking versus directory queries, is agreed between the communicating systems prior to the data transfer. Additional information, such as use of protocol features for that association and authenticating the peer application, may also be included in the setup phase. This phase allows the two systems to agree on what is expected in the association. The extent of the detail exchanged during the setup phase may vary depending on several factors (e.g., agreements outside the mechanized interface between the suppliers of the two systems, policies to be followed, business-level requirements). The initial setup phase forms a contract between the two applications and an exchange violating this contract may result in aborting the association.

The release phase signals the completion of the exchange between the application entities. Either entity may release the association. Depending on the type of release, the association may be terminated normally or abnormally.

NM INFORMATION TRANSFER

After the association is established, the data transfer is achieved using the protocol infrastructure defined earlier. The data is exchanged either as a request from the manager to perform one of the operations defined in CMISE followed optionally by a response from the agent, or autonomous notification from the agent which may be acknowledged by the manager. The request reply protocol definition includes a sequence number to facilitate correlating response and request. Depending on the NM application and the resource being managed, the message exchanged includes the reference to the resource and the information relevant for that resource within the specific application.

Two simple examples are given below to illustrate the components of the message.

EXAMPLES

Assume that a managed object class, circuitPack, is modeled to support protection switching capabilities of a resource such as a controller card. In the example above, let us assume that if SCNU1A has a failure, a protection switch will occur to a mated card, SCNU1B. A model of an equipment may be defined to support the fault management function for reporting alarms. The equipment managed object class will include a notification called "equipment alarm." Let us suppose that the information to be included with the notification are severity of the alarm, probable cause, if it is backed up and reference to the backed up entity. The components of the message invoked by the agent are shown in Fig. 5.

The first field in the above structure is a sequence number to facilitate correlation if an acknowledgment is returned for this event report. The value 0 corresponds to an operation value assigned by CMIP for event reports (notifications) that do not have an acknowledgment. Since this event report does not require a response, the sequence number is not utilized for correlation purposes. Following the operation value field is the identification of the managed object class (circuitPack in this example). The specific equipment is referenced using the local name by assigning the value "SCNU1A" to the attribute equipment ID. The latter is also referred to as the *naming attribute*. The type of event reported from the managed object is identified in the next field as an equipment alarm. The event time (which may optionally be present) specifies the time the event occurred. As part of the definition of the equipment alarm notification, the information model will specify the associated information: severity of the alarm, probable reason, and the identity of the protecting unit if there was a protection switch (backed up) when the equipment failed.

Let us now consider how the various concepts introduced earlier are included in the above exchange. The service used for reporting the event is the M-EVENT-REPORT service discussed in the fourth section. The protocol infrastructure, CMIP, specifies the operation value for event reports and allocates fields for providing the reference to the managed resource and the type of event, and leaves a hole so that event-specific information can be included. Alarm surveillance functional requirements define different types of alarms along with the appropriate parameters, such as severity, that must be included when reporting an alarm. The information model determines that the circuitPack managed object class is a representation of a card in a system and is expected to report an equipment alarm when it fails. The syntax for the above fields

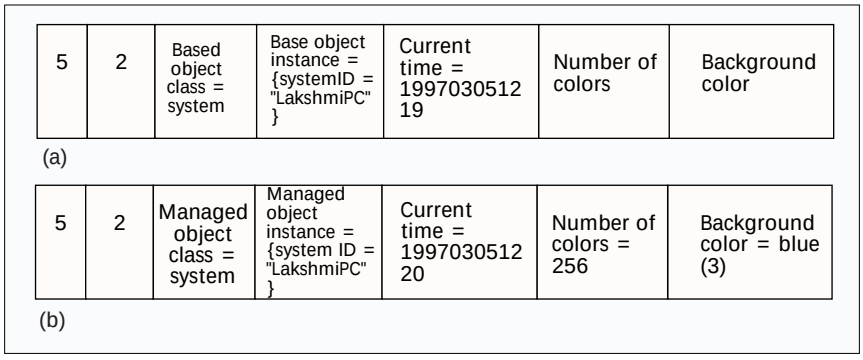

Figure 6. *Examples: a) Get Request; b) Get Response.*

are defined in either protocol or information model standards. Given these specifications, an implementation has to map resources within a product such as a network element to the available information models. In the above example a controller card was modeled as a circuit pack, and the alarm indicates failure of the working card. As part of the mapping, it was determined that the severity should only be minor because a protecting card will be used until the primary one is repaired. This mapping of relevant values and fields varies depending on the product architecture and what services are planned with the product.

In contrast to the above example, consider a case where the manager requests an operation to be performed by the agent. Assume that a managed object class system models a personal computer (PC). Depending on the type of monitor used, the number of colors and the background color may vary. Figure 6 defines a request from the manager to read these attributes and the response from the agent.

It is assumed in this example that the attribute number of colors has the syntax of an integer, and that the syntax for background color is a list of enumerations — 0 for green, 1 for yellow, and so on.

The service used to request the values of the attributes is the M-GET service discussed in the fourth section, and the operation value assigned for the Get operation is 2 in CMIP. The information model defines the system managed object class. The function performed supports the configuration management area, where the manager learns about the properties of the resource being managed.

Work in Progress

Distributed NM Application

The initial focus for OSI systems management was on interface specifications between peer management application entities. The system management standards, as mentioned above, were developed to manage the communication entities of the end systems, including the application entities. Industry growth in distributed processing has identified two aspects that must be addressed as part of a management application: management of the distributed application processes, and taking advantage of the distributed processing technology to implement management application. The latter also includes environments where managed resources may be distributed. To support these two requirements, ITU and ISO groups are jointly progressing the draft standard "Open Distributed Management Architecture."

This work applies the reference model developed as part of open distributed processing to management. The architectural descriptions are developed in terms of five viewpoints: enterprise, information, computational, engineering, and technology. These viewpoints facilitate specifications at varying abstraction levels and provide a structured method for documenting requirements, starting with business needs through the technology used in implementation. Standards usually do not address the technology viewpoint as it pertains to implementation. The information viewpoint describes the relevant information without being concerned about how it is represented or exchanged at the object interface. The computational viewpoint introduces object interfaces and the signature of the interface. The engineering viewpoint defines the interfaces using a specific methodology or protocol. Existing OSI systems management specifications using GDMO and text can be considered to address both the computational and engineering viewpoints. The enterprise and information viewpoints are specified as requirements in text without making a clear distinction.

As part of ODMA, a client/server model has been introduced for operations and notifications. Another concept introduced is the explicit object-oriented modeling of the managing role. In the previous sections the emphasis was on modeling the managed resources, and details of the manager role process were not explicitly modeled. A table mapping the existing concepts to those introduced as part of distributed management architecture is included in the draft standard.

The move toward distributed processing also introduces the need to support different types of transparencies: location, relocation, migration, access, failure, persistence, replication, and transaction. Without going into the details, as an example, with location transparency an agent role application does not reveal the location in space of the managed resource. This allows relocation of the managed object without requiring the managing application to be aware of where it is present (thus leading to migration transparency).

New functions such as operation dispatching and notification dispatching have been introduced in support of distributed management applications. Future work plans include additional functions, development of notations to represent the viewpoints, and use of CORBA IDL[9] in support of ODMA functions.

System Management Functions

Introduction of new system management functions is another area where work is ongoing in standards. Various management areas and examples of functions associated with them were introduced in the third section. Several functions and information models to support the requirements were completed as part of the initial set of joint ITU Recommendations ISO Standards in 1992. These documents formed part of X.730 series (ISO/IEC 10164-Parts 1 to 7) and addressed some of the functions in the areas of configuration, fault, common and security. A second series of documents were then completed to address performance, security, accounting and common management areas. New functions are still being introduced even though at a much slower rate. Some of these functions are:

- The Management Knowledge Management Function (X.750) provides a model for the sharing management knowledge between the managing and managed systems.
- The Management Domain Management Function addresses creation and administration of management domains, including the policies enforced.

[9] *The Common Request Broker Architecture was developed for distributed processing by OMG. The notation used to represent the object definitions is in the Interface Definition Language (IDL).*

- The Command Sequencer is where the manager can specify a sequence or collection of commands to be executed by the agent at a later time.
- The Enhanced Event Control function expands the existing event report control mechanism where criteria can be set to forward events to different destinations.

The current mechanism for forwarding events did not specify what to do with the events if the communication link to the destination is not available (the assumption is to drop it into a bit bucket). The enhancements propose queuing the events until destinations are made available and then forwarding them.

PROGNOSIS AND SUMMARY

Even though system management functions, corrigenda to correct errors, and amendments to existing functions may continue to evolve at a slower pace than the initial set of standards, it must be recognized that the required foundation for building implementations exists now. OSI systems management protocol and information models have had a sluggish start with respect to availability of interoperable implementations in spite of the fact that a set of completed standards were available in 1992. Some of the reasons for this result are the complexity of the approach, which comes naturally with the flexibility and powerfulness of the approach; the lack of tools that allow the developers to be productive without requiring them to climb a steep learning curve; a simpler approach available for internet management; and the upfront costs associated with establishing the necessary infrastructure support in agent systems.[10] This situation is changing, and tools are coming into the market that will increase developers' productivity without requiring them to first become expert in this technology [1]. Using the tools, products have been deployed for applications such as trouble administration function (TMN X interface) and Integrated Digital Loop carrier (TMN Q3 interface) in the last couple of years. Assuming this trend continues, wider deployment of CMIP-based products is expected. There are also efforts to develop products using the protocol developed as part of the CORBA effort, taking advantage of the existing information models for the CMIP paradigm. The jury is still out on commercial deployment of CORBA for network management.

This article has given a bird's eye view of the concepts and specifications available from the standards effort on OSI systems management. In summary, system management architecture defines a framework that can be used with different engineering solutions. OSI system and network management is a peer-to-peer interface with manager and agent roles taken by the systems during a communications exchange. This is different from other approaches where the manager/agent role assumed by a system is fixed. In addition, OSI management offers a powerful information modeling paradigm using object-oriented principles. This approach is scalable and can be applied to different networking technologies (present and future) as well as new services. These standards have been found very useful not just for managing OSI protocol entities in an end system, but also for managing the telecommunications network.

[10] *Network Management is usually considered as the cost of doing business in order to sell the system. This makes it a difficult business case to justify the associated expense.*

REFERENCES

[1] M. Feridun *et al.*, "Implementing OSI Agents/Managers for TMN," *IEEE Commun. Mag.*, Sept. 1996, vol. 34, no. 9, pp. 62–67.

ADDITIONAL READING

[1] ISO/IEC 7498-4, Information Processing Systems — Open Systems Interconnection — Basic Reference Model - Part 4: Management Framework.
[2] CCITT Rec. X.701|ISO/IEC 10040:1992,. "Information Technology — Open Systems Interconnection - Systems Management Overview," 1989.
[3] CCITT Rec. X.710, "Common Management Information Service Definition for CCITT Applications," 1991, 1997.
[4] CCITT Rec. X.711|ISO/IEC 9596-1(E), "Information Technology-Open Systems Interconnection — Common Management Information Protocol Specification — Part 1: Specification," 2nd ed., 1991, 1997.
[5] CCITT Rec. X.720|ISO/IEC 10165-1, "Information Technology — Open Systems Interconnection — Structure of Management Information: Management Information Model," 1992.
[6] CCITT Recommendation X.721|ISO/IEC 10165-2, "Information Technology — O pen Systems Interconnection — Structure of Management Information: Generic Management Information," 1992.
[7] CCITT Rec. X.722|ISO/IEC 10165-4, "Information Technology — Open Systems Interconnection — Structure of Management Information: Guidelines for the Definition of Managed Objects," 1992.
[8] CCITT Rec. M.3010, "Principles for a Telecommunications Management Network (TMN)."
[9] CCITT Rec. X.730, 740, and 750 series|ISO/IEC 10164 Parts 1–*n*, "Information Technology — Open Systems Interconnection — Systems Management Functions."
[10] ISO/IEC 13244|ITU Draft Rec. X.703, "Open Distributed Management Architecture," 1997.
[11] L. Raman, "CMISE Functions and Services," *IEEE Commun. Mag.*, vol. 31, no. 5, May 1993.
[12] S. M. Klerer, "Information Modeling," *IEEE Commun. Mag.*, vol. 31, no. 5, May 1993.
[13] D. Sidor, "TMN Standards Satisfying Today's Needs While Preparing for Tomorrow," this issue.

BIOGRAPHY

LAKSHMI RAMAN (lakshmi_raman@adc.com) is director of systems engineering for the Access Platform Product in the Broadband Communications Division of ADC Telecommunications. She chairs the working party in ITU SG 4 responsible for development and maintenance of OSI systems management standards as well as some of the TMN Recommendations. She also chairs the Management Services subworking group of T1M1.5, developing information models for over ten years. Prior to ADC, she worked at Bellcore, where she was responsible for the network operations protocol and standards group. She has a Ph.D in solid state physics and a Master's in physics and computer engineering.