The Integration of CORBA-Based Management with OpenView DM

GLENN HOLLIMAN, QINZHENG KONG AND MICHAEL NEVILLE

Abstract—**The increasing complexity of network infrastructures and network-based services has exceeded the capability of the current management models. There is a need to evolve these models to provide the scalability, reliability and extensibility required to manage network infrastructure and services and implementation of network-based services. This paper considers the integration of a CORBA environment with a TMN environment to manage network infrastructure and services, and the implementation of services. It also presents our experiences gained from a prototype implementation of a distributed network management environment using CORBA integrated into a TMN environment.**

Source of Publication—**Proceedings of the International OpenView Forum Conference, St. Louis, USA, June 1996.**

1 INTRODUCTION

While OSI network management remains a major technology for building TMN-based network management systems and applications, CORBA is a useful technology that can add tremendous value to the TMN management world. It offers functionality for building *distributed* network management applications. These distributed applications can provide scalable, reliable and extensibility. Distribution also allows development of solution that provide improved performance.

The integration of TMN network management technology and CORBA holds the key to the problems of distributed network management.

2 NETWORK MANAGEMENT AND NETWORK-BASED SERVICES

2.1 NETWORK MANAGEMENT

Telecommunication services are supported by distributed telecommunication and computer networks. The management of such networks is critical to the quality of service delivered to customers. In past years, a lot of effort has been put into developing management technology and international standards for managing such networks. TMN (ITU-T standards for Telecommunication Management Network) and OSI based Network Management are the result of such effort.

TMN and OSI NM technology have been specifically developed for network management. Object oriented technology is used in TMN and OSI Network Management. Each network element is modelled as a managed object. Object instances are maintained in a Management Information Base (MIB) which has a tree structure and is referred to as containment tree. Each object is identified by a unique object ID which clearly specifies the location of the object in a containment tree. Each object is characterised by its packages which include attributes and the operations which can be performed on these attributes. All the attributes of the managed objects are clearly visible to management entities.

A set of common operations are defined. The values of attributes of object can be read and updated using the get and set operations. Objects can be created and deleted explicitly by a create or a delete operation.

For example, GDMO definition for ATM Managed Element is:

```
managedElement MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 (1992) | ISO/IEC 10165-2: 1992":top;
    CHARACTERIZED BY
    managedElementPackage;
managedElementPackage PACKAGE
    BEHAVIOUR
        managedElementBehaviour;
    ATTRIBUTES
        managedElementId GET,
        "Rec. X.721 (1992)": systemTitle GET-REPLACE,
```

```
        alarmStatus GET,
        "CCITT Rec. X.721 (1992)": administrativeState GET-REPLACE,
        "CCITT Rec. X.721 (1992)": operationalState GET,
        "CCITT Rec. X.721 (1992)": usageState GET;
    NOTIFICATIONS
        "CCITT Rec. X.721 (1992)": environmentalAlarm,
        "CCITT Rec. X.721 (1992)": equipmentAlarm,
        "CCITT Rec. X.721 (1992)": communicationsAlarm,
        "CCITT Rec. X.721 (1992)": processingErrorAlarm;;

CONDITIONAL PACKAGES
        ... ...
REGISTERED AS { m3100ObjectClass 3 };
```

This GDMO definition defines that the *managedElementId*, *operationalState*, and *usageState* attributes are read-only attributes to a management entity while the value of *systemTitle* and *administrativeState* can be read and replaced by a management entity. Both "get" and "replace" are standard operations.

2.2 NETWORK-BASED SERVICES

The telecommunication industry is currently experiencing considerable change in its business environment. The combined effort of the telecommunication and computer industries has resulted in considerable growth in communication and data service offerings. This growth has reduced the gap between telecommunication services and information management services. The demand for more services, and the integration of these services, is increasing rapidly.

This trend has put enormous pressure on the existing telecommunication industry. It has forced the industry to adjust its IT strategy from network management-oriented to more service-oriented. It has also caused the industry to employ new technologies to support integrated services and to support interoperability with different telecommunication companies in order to provide better services.

Network-based services are modelled differently from networks and their elements. A n*etwork* is normally modelled by a large numbers of relatively simple objects, such as objects derived from the ManagedElement Object Class. These objects and their relationships model the real network elements, and the network topology and hierarchy. They are relatively simple in the sense that these objects normally have a set of states that are visible to a remote management process. Most objects support only simple operations such as allowing a management process to look at or to change the current value of its attributes.

Network-based services, on the other hand, are normally modelled by a relatively small number of very complex objects. These objects represent high-level abstractions of the network, and relationships between network components (network elements) and computerised services supported by the network. For instance, a Video-On-Demand service may have a small number of instances representing the distributed service offerings and management components, but it is supported by a large number of underlying network elements and various management information systems. Accessing Video-On-Demand service may invoke very complicated functions including accounting, performance, interoperability, quality of services, and so on. Services often need to be created, implemented, and deployed rapidly while networks are relatively stable by comparison.

Although the existing TMN and OSI technology can be used to manage services, it was not specially developed for service management and requires extra features to meet service management requirements. People are also looking at technology developments to see whether new technologies, especially distributed object technology, can be used for service management.

2.3 PLATFORMS

2.3.1 TMN

Network management platforms, such as the HP OpenView DM platform and many others, have been developed to provide a network management environment. These platforms were based on the OSI Common Network Management Protocol (CMIP) and provide the capability to exchange management information between different application entities. OSI CMIP provides support for the peer-to-peer communication paradigm.

The standardisation of the protocol layer only solves some of the problems in building network management applications. The standardisation of the application programming interface has not been addressed. Without this standardisation it is difficult to obtain interoperability of network management applications between platforms. Currently, many TMN network management and network element management applications are built on top of these network management platforms.

2.3.2 CORBA

CORBA is a distributed object technology with many features common to all distributed systems. For example, it supports location transparency of objects and transparency network representation of data. CORBA technology supports an environment for defining objects and invoking operations on objects. It also supports departing object implementations from their interface definitions: a service interface can be specified in CORBA Interface Definition Language (IDL). There can be multiple implementations of that service. For example, the different implementations of a service might offer different qualities of service, be implemented in different languages, or be implemented on different platforms.

The language independence provided by CORBA allows interoperability of clients and services that are implemented in different languages. Language bindings for CORBA IDL have been defined for C, C++, and Smalltalk.

The Common Object Services Specification (COSS) [5] defined by the OMG provides the basic services required for building distributed applications. The transaction service and relationship service are particularly applicable to network and network-based service management. The transaction service offers a mechanism for synchronising different activities/operations within applications. The relationship service can be used to store the relationships between managed objects.

The distribution facilities provided by CORBA allow distribution of management applications and services. Distribution can increase scalability of the platform by allowing distribution of the load to multiple machines. Distribution can also improve performance by locating services in proximity to their clients, rather then in a central location.

All these features, which are important to the industrial need for a highly extensible platform, make CORBA technology a good candidate for an environment to implement service offerings and service management.

Examples of service objects in the Video-On-Demand (VOD) application include a connection manager, which manages the connection of the underlying switches, and a VOD manager, which manages the equipment used for the video services. Interfaces to these objects can be specified in CORBA IDL as:

```
Connection Manager

interface ConnectionManager: SupervisedObject {
    ChannelID isConnected( in SelectorID );
    void connect( in SelectorID, in ChannelID )
        raises( SelectorAlreadyConnected,
            ChannelAlreadyConnected );
    void disconnect( in SelectorID )
        raises( NotConnected );
};

VOD Manager

interface VodManager: SupervisedObject {
    ChannelID select( in VideoID )
        raises( ServerFailure );
    Time progress( in VideoID )
        raises( NotSelected, ServerFailure );
    void play( in ChannelID, in Time )
        raises( NotSelected, ServerFailure );
    void pause( in ChannelID )
        raises( NotSelected, ServerFailure );
};
```

3 INTEGRATION OF NETWORK AND SERVICE MANAGEMENT

As the use of CORBA in network and service management expands, it is likely that developers will find that at high abstraction levels CORBA is a more appropriate technology for developing distributed management applications than the OSI model. The CORBA technology offers more functionality and therefore increased productivity when developing distributed management applications.

However, the TMN CMIP interface is likely to remain the standard interface for network management for some time to come. This is due both to the amount of investment in existing applications, the number of standards based on GDMO and CMIP, and also the management features for network elements. Hence, there will be a significant requirement for an integration environment between these two management technologies.

To integrate CORBA and TMN is to integrate these two different object models and the related management functions and object operations. There are a number of approaches that can be used to support integration between CORBA and TMN network management environments. Three approaches are discussed below. For more detail see [1].

3.1 IDL-GDMO GATEWAY APPROACH

One way of interfacing CORBA and OSI based TMN is to build a gateway between the CORBA environment and the OSI network management environment. To be able to conduct communications between a manager and an agent, agreement must be made at two levels:

- At the object definition level, each GDMO object needs to have a mapping in IDL. This mapping includes the object definition (such as attribute types and values), operations that can be performed on the object (such as get, set, create, delete, action), notifications the object can issue, and possibly the behaviours of the object.
- At the communication protocol level, messages or requests sent by a manager using protocols supported by the ORB need to be translated into CMIP before being passed to the agent. Messages or responses sent by the agent using CMIP need to be translated back to the protocols supported by the ORB before being passed to the manager.

For each GDMO object, its equivalent IDL object definition is required. Also, for each operation in a GDMO object, an equivalent method in IDL definition is required. A compiler can be developed to generate such mapping. The Joint Inter-Domain Management Group (JIDM) have taken this approach.

There are a number of problems that need to be solved during the mapping, such as the handling of CMIP get operation with filter, scope, and synchronisation conditions, the handling of linked replies, and the handling of notifications.

3.2 VALUE ADDED APPROACH

Instead of having a one-to-one mapping between GDMO object operations and IDL object methods, value can be added by allowing different methods to be defined for each IDL object. In other words, extra value can be added by allowing user-defined methods.

One example of a value-added method is to reboot a router. Instead of defining a simple reboot method, a time signature can be added, so the operator can request to reboot the router at certain time, say, at midnight.

Compared to the simple gateway approach in Section 3.1 this approach is more powerful and flexible. However, it requires extra.

3.3 ABSTRACT OBJECT DEFINITION APPROACH

A third approach is to introduce a set of CORBA objects at the appropriate level of the management hierarchy. A group of GDMO objects can be mapped into a single CORBA object at a higher level. A set of methods can be defined for each CORBA object.

For example, a set of service objects can be defined using CORBA IDL. These services are implemented on top of a set of GDMO network element objects. The relationship between the IDL service objects and the GDMO network element objects can be specified. These relationships can be recorded for later references.

Requests on a CORBA object are mapped into a set of operations performed on a set of GDMO objects. Events generated by a GDMO object can be reflected at CORBA object level using the COSS Event Service.

There are significant advantages to building higher levels of abstraction in the CORBA world, as the abstract objects can be used to build services that are closer to business needs and are more resilient to infrastructure changes. The CORBA environment is suited to fewer, more complex, coarse grain objects, while the TMN environment is suited to larger numbers of simpler, finer grain objects.

3.4 COMBINING APPROACHES

The Abstract Object Definition approach and IDL-GDMO Gateway approach are not mutually exclusive. In many cases the IDL-GDMO Gateway approach can be used very effectively. However, even when IDL-GDMO Gateway techniques are used to map all TMN objects into the CORBA space, the ability to provide high level abstraction is still required. For instance, a billing record is a much higher abstraction than network usage. Thus, even if the network usage record is represented as a CORBA object, there is still the need to map this record to the billing record, as the billing record contains more semantics (such as tariff policy, special offers, and so on).

The higher abstraction can also be built in the IDL-GDMO Gateway approach. The TMN objects can be mapped into CORBA objects (proxies). Higher abstraction CORBA objects can then be built to interact with the proxy objects. The CORBA object definition environment can then be used to define new objects (with higher level abstractions) and the relationship service can be used to associate different object instances.

4 INTEGRATION EXAMPLE OF NETWORK ELEMENT AND SERVICE MANAGEMENT

The Video-On-Demand (VOD) Service Demonstrator is a prototype system built to investigate the integration of CORBA and TMN for implementation and management of network based services. The Demonstrator is a software mock-up of a simple VOD service. The implementation is targeted at exercising different aspects of CORBA/TMN integration rather then trying to provide a complete VOD service.

The Abstract Object Definition approach and the IDL-GDMO Gateway approach to integrating CORBA and TMN are used by the VOD demonstrator.

4.1 ARCHITECTURE

The architecture is designed to demonstrate both the IDL-GDMO Gateway of CMIP into CORBA and the Abstract Object Definition approach.

The demonstrator has three sections—the TMN section, the CORBA section, and the integration section (CORBA and TMN).

The TMN section consists of the CMIP agents for the Video Selector, the Network Switches and the VOD servers. The CORBA section has a number of CORBA objects to implement the VOD Service. The architecture of the demonstrator is given below:

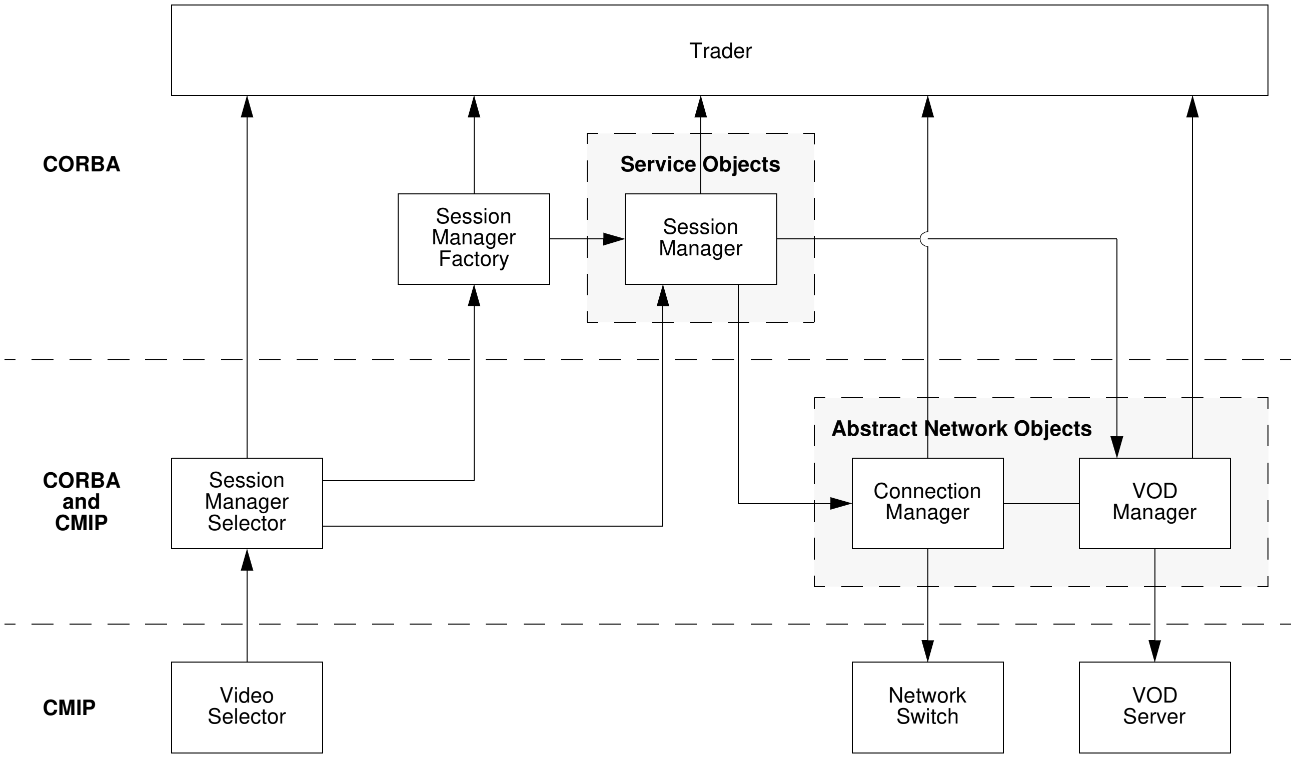

Figure 1: Architecture of the VOD Service Demonstrator

4.2 MANAGEMENT SECTIONS

4.2.1 TMN SECTION

The TMN section consists of the CMIP agents for the Video Selector, the Network Switches and the VOD servers. It reflects the network and network element management part of the demonstrator.

Network Switch—The CMIP agent for the network. The Network Switch is not implemented as it does not demonstrate any functionality that isn't demonstrated in other components of the demonstrator.

VOD Server—The CMIP agent that simulates the behaviour of a VOD Server.

Video Selector—An X application that supports a graphical interface. The graphical interface provides the functions that would normally be provided by the hardware at the VOD customer premises, namely the set top box. It simulates the Video Selector CMIP agent.

In addition to the X application, a command line version of the Video Selector is used to simulate up to 10^5 video selectors in a single process. This application injects CMIP events into the systems at the required rate to simulate the peak and average load for video selectors.

4.2.2 CORBA SECTION

The CORBA section has a number of CORBA objects to implement the VOD Service. It represents the service management aspects in telecommunication applications.

Session Manager Factory—A CORBA object used to create Session managers.

Session Manager—A CORBA object that is responsible for managing a VOD service session. It receives events from the Video Selector (via the Session Manager Selectors) and co-ordinates the control of the VOD Server and the Connection Manager.

Trader—A CORBA trading service.

Supervisory Console—An X application that queries the Supervisory interface on a CORBA object and displays the result. CORBA Objects may be located from the trader, or from the result of a query on the Supervisory interface.

A Supervisory interface is provided on all the CORBA objects. This interface allows the Supervisor Console to monitor the state of the CORBA object.

4.2.3 INTEGRATION SECTION

This section contains a number of CORBA objects which play an important role in the CORBA TMN integration

VOD Manager—Manages VOD servers. It fulfils requests to play a video by making CMIP requests to a VOD Server. The VOD Manager is an example of the one-to-one mapping approach.

Session Manager Selector—Captures the events generated by the Video Selector. The Session Manager Selector creates Session Managers when required and forwards events from the Video Selector to the Session Manager managing that Video Selector.

Connection Manager—A CORBA object used to manage the network connections. The Connection Manager is an example of an Abstract Object Definition mapping approach.

The Connection Manager would use the CMIP interface to Network Switches to configure connections between the Video Selectors and the VOD Servers for transmission of the video data.

4.3 PLATFORM ENVIRONMENT

4.3.1 NETWORK MANAGEMENT ENVIRONMENT

The HP OpenView DM platform is used as the network management platform. It is used to provide OSI CMIP communications between managers and agents for network and network element management. Management operations and notifications, such as Actions and Events are used in the demonstrator.

When a customer selects a VOD service, a CMIP event is generated. This event is passed by the VOD Selector to the Session Manager Selector. The Session Manager Selector invokes a CORBA request on the Session Manager. The Session Manager then implements the functionality to provide the service requested by the customer. The Session Manager uses the Connection Manager to ensure that the network connection required to provide the service is configured and the VOD Manager to configure the VOD Server to play the requested video.

The VOD Manager and the Connection manager use CMIP requests to configure the network and control the VOD Server.

4.3.2 SERVICE MANAGEMENT ENVIRONMENT

Iona Orbix is used as the CORBA platform for service management. The trading service is used to find required CORBA services, such as Session Manager Factories, VOD Managers and Connection Managers. The Session Manager Factory service allows dynamic creation of Session Managers (CORBA objects). The Session Managers are used to manage a single Video Session.

The ability to detect failed objects and trade for new services provides a suitable environment to build services that are capable of fault recovery. Using the demonstrator it was possible to show the recovery of the system from CORBA object, process and network failures.

4.3.3 DIFFERENT METHODS ARE REQUIRED FOR THE OBJECT MAPPING

In the VOD demonstrator, both the IDL-GDMO Gateway approaches and the Abstract Object Definition approach are used. It shows that, different mapping approaches should be used to meet different requirement. The Abstract Object Definition approach and IDL-GDMO Gateway approach are not mutually exclusive, both can be used very effectively.

IDL-GDMO GATEWAY APPROACH

The VOD Manager CORBA interface is an example of the IDL-GDMO Gateway approach. Each VOD Server object modelled by GDMO in the CMIP domain is mapped onto a CORBA object, the VOD Manager, in the CORBA domain. Any CORBA request on the CORBA object is translated as an action on the CMIP object.

ABSTRACT OBJECT DEFINITION APPROACH

The Connection Manager is an example of the Abstract Object Definition approach. A Connection Manager is an abstraction of a set of network elements (e.g. router and switch) and the connections among them. At the service level, more focus is on the state of a connection for a VOD service as a whole rather than the state of each elements. A fault on a network element is viewed as an element fault by a network manager, but is viewed as poor quality of service by a service manager. It is often the case that certain level of abstraction of encapsulation is required at service level for better efficiency.

4.3.4 EVALUATION OF APPROACHES

As a comparison of the IDL-GDMO Gateway approach and the Abstract Object Definition approach consider an implementation of the Connection Manager to manage an ATM switch. It is estimated that to manage an ATM switch that can support 2,000 PVCs requires manipulation of 10,000 GDMO objects within the switch. To set up an individual PVC is estimated to require 8 CMIP requests on 5 GDMO objects.

The IDL-GDMO Gateway approach would require mapping of the 10,000 GDMO objects into 10,000 CORBA objects. Establishment of a PVC would involve 8 CORBA requests on 5 CORBA objects and 8 CMIP requests on 5 GDMO objects.

The Abstract Object Definition approach would involve 1 CORBA request on 1 CORBA object and 8 CMIP requests on 5 GDMO objects.

The Abstract Object Definition approach does not reduce the CMIP load but significantly reduces the number of CORBA objects required and the number of CORBA requests. Use of the Abstract Object Definition approach makes best use of the performance characteristics of the CORBA and TMN.

5 SUMMARY

CORBA and TMN integration is an important issue in telecommunication service and network management. The key issue of integration is the object mapping between different object models. Different mapping approaches can be used between CORBA IDL and CMIP GDMO. An integration example, the VOD demonstrator, demonstrates the integration idea. Experience shows that both IDL-GDMO Gateway and Abstract Object Definition approaches are both required and different approaches can co-exist.

REFERENCES

[1] Kong, Q. and Chen G. Integrating CORBA and TMN Environments, IEEE/IFIP Network Operations and Management Symposium, Kyoto, Japan, April 1996.

[2] Sukiyaki, N. Interworking between OSI management SNMP and CORBA, JIDM draft document, January, 1996.

[3] Object Models, Object Management Group, Draft 0.3, January 12, 1995.

[4] CORBA The Common Object Request Broker: Architecture and Specification, Object Management Group, Revision 1.2, Draft 29 December 1993.

[5] Common Object Services Specification (COSS 1), Object Management Group, Volume I, March 1, 1994.