DISTRIBUTED NETWORK MANAGEMENT USING CORBA/TMN

GRAHAM CHEN, MICHAEL NEVILLE AND QINZHENG KONG

***Abstract*—This paper presents different technical approaches to integrate OSI network management environment with CORBA to offer a distributed network and service management environment. Based on this integration, the telecommunication services can be deployed, provisioned and managed in the same environment.**

The paper also reports an experiment platform we built based on the integration strategies. The platform integrates Orbix CORBA with HP OpenView DM to offer Video-on-Demand services and to provide its management functions. We present an evaluation of the functionality and performance of this platform.

***Keywords*—CORBA, TMN, OSI Management, Network and Service Management**

***Source of Publication*—Proceedings of the 7th IFIP/IEEE International Workshop on Distributed Systems Operation and Management (DSOM) 1996.**

1 INTRODUCTION

One of the main challenges facing today's telecommunication industry is that of increasing revenue by providing high-quality services while, at the same time, reducing costs by introducing new network management technologies.

While OSI network management remains a major technology for building TMN-based network management systems and applications, CORBA is a useful technology for adding value to the TMN management world. It offers the facilities for building distributed network management applications which provide the telecommunication industry with scalability, reliability and better performance.

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

This paper discusses the role of CORBA in the TMN management domain and a range of CORBA/TMN integration issues. It also presents our experiences in developing a prototype distributed network management environment by integrating CORBA with a TMN environment.

2 TMN NETWORK MANAGEMENT

The rapid growth of global computer and telecommunication networks has increased the demand for distributed and large-scale network management applications. Contemporary network management applications are expected to manage geographically dispersed networks and services composed of complex, heterogeneous computing and telecommunications resources. Such applications are expected to achieve consistent, reliable and transparent access to these resources. A crucial issue for the industry is the use of new technologies to guarantee that a global network of resources can be shared and managed by a variety of applications in a consistent and efficient manner.

The increased demand for managing large-scale, distributed networks in a heterogeneous environment is yet to be met due to the limitations of existing network management technologies. The current models of network and systems management (OSI and SNMP) were not designed to meet such stringent demands.

Figure 1 shows the monolithic application architecture of a typical OSI-based management application:

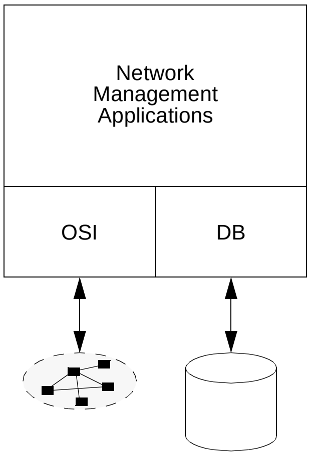

Figure 1: The architecture of a typical OSI-based management application

Due to its non-distributed nature, the OSI architecture cannot meet demands for scalability and performance. Typically, an OSI-based management application cannot cope with the requirement to manage additional resources, and to support more management functions or applications.

3 DISTRIBUTED NETWORK MANAGEMENT

Different technology development bodies, such as standards bodies and industrial consortia, are currently seeking new models for network management. To provide management systems which will satisfactorily manage the current generation of telecommunication networks and service applications, any new model must support ***common requirements***—which are of interest to distributed platforms and ***special requirements***—which are of interest to network and service management platforms.

3.1 COMMON REQUIREMENTS FOR DISTRIBUTED PLATFORMS

Distribution—To support highly distributed management applications, a management platform needs to provide a distributed environment as well as distributed services. Telecommunication applications are always geographically distributed and, therefore, their management applications should also be distributed.

Scalability—The ever-increasing size of the resources, information and networks which distributed platforms are expected to manage means that a management platform itself needs to be scalable.

Heterogeneity—The extreme diversity of resources and computer systems in the management domain means that a management platform needs to handle heterogeneous systems.

Common Services—In order to reduce the cost and complexity of application development, a management platform should provide common services which different network management applications can simultaneously access.

3.2 SPECIAL REQUIREMENTS FOR NETWORK AND SERVICE MANAGEMENT PLATFORMS

Consistency—Data consistency is a very important issue in distributed network and service management applications. A network and service management platform must be able to support a consistent data model.

Reliability and Performance—Service management applications must have high degree of reliability, availability (7x24 is a common requirement) and performance. This is particularly important as the size and complexity of networks and services increase. Also, a network and service management platform must support applications which tolerate component failures.

Service Integration—The ability to provide service integration is a new issue for network and service management platform. Such a platform must provide functions which allow for the introduction and management of new services—as well as for the integration of new services with existing ones.

Interoperability—A network and service management platform needs to be inter-operable in different business domains and administrative authorities which may have different enterprise policies, business objectives, underlying, organizational infrastructures and technological frameworks.

3.3 A New Architecture for Distributed Network Management

All of the preceding factors have greatly increased the complexity of today's distributed management applications and have also increased the complexity of management functions for such applications.

Figure 2 shows a distributed Network Management architecture which satisfies all of the preceding requirements:

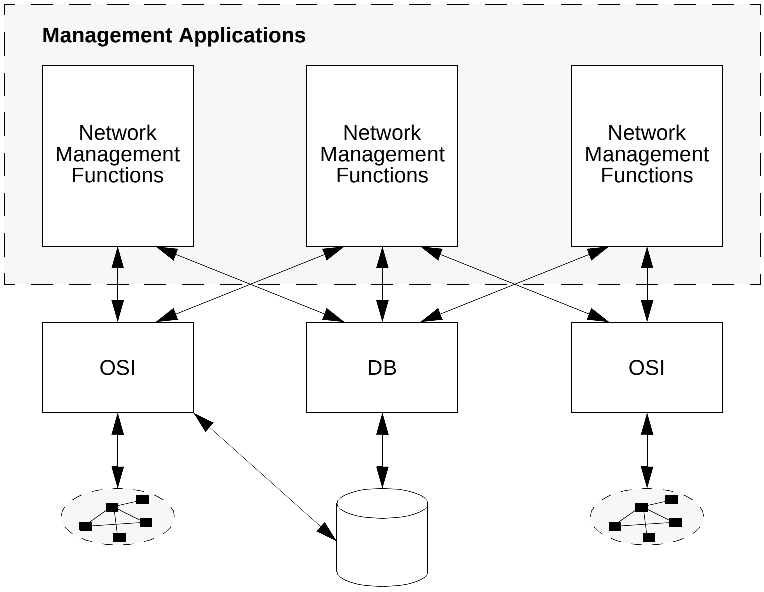

Figure 2: Distributed Network Management Architecture

Such architecture requires a more powerful technology framework than its predecessors. To face this new challenge, both standards bodies and industry are looking in new directions to provide a framework to meet today's network management requirements.

CORBA provides a suitable technology framework to provide the distribution infrastructure for this architecture. With its ORB and distributed services, CORBA offers an ideal environment for building management applications which require generic and transparently distributed facilities. Furthermore, CORBA is the strategic technology framework to support service management environments.

4 Integrating CORBA and TMN

For OSI network management, CMIP has distinct advantages for the management of the large numbers of relatively unsophisticated objects which represent a physical network. CMIP was designed to perform such tasks and has been entrenched in telecommunication networks for many years. The cost of replacing CMIP in this environment is prohibitively expensive. A more practical approach is to combine CORBA and OSI network management to provide an integrated solution which provides the benefits of both technologies.

The major problem in integrating CORBA and OSI-based TMN network management is in the area of object modelling. These two technologies use quite different approaches to modelling the required managed objects.

In OSI network management, the OSI standards define a set of systems management functions, GDMO is used to define a set of management objects and ASN.1 is used to define the data types used in objects.

In CORBA, the Interface Definition Language (IDL) is used to define object interfaces and the Common Object Services Specification (COSS) is used to define a set of object services.

Combining CORBA and TMN requires integration of these disparate object models as well as their related management functions and object operations.

5 APPROACHES TO CORBA/TMN INTEGRATION

5.1 THE IDL-GDMO GATEWAY APPROACH

One approach of combining CORBA and OSI-based TMN is to build a gateway between the CORBA environment and the OSI network management environment, and to map every GDMO object and every management operation in the TMN environment into the CORBA environment. This mapping is referred to as one-to-one mapping.

A scenario for such approach is to develop managers in the CORBA environment and agents in the OSI environment. On the agent side, a GDMO object is defined for each type of managed object and the CMIP communication protocol is used. On the manager side, object definitions are based on CORBA IDL and the ORB is used as the communication mechanism.

Figure 3 provides a typical system architecture for the IDL-GDMO gateway approach.

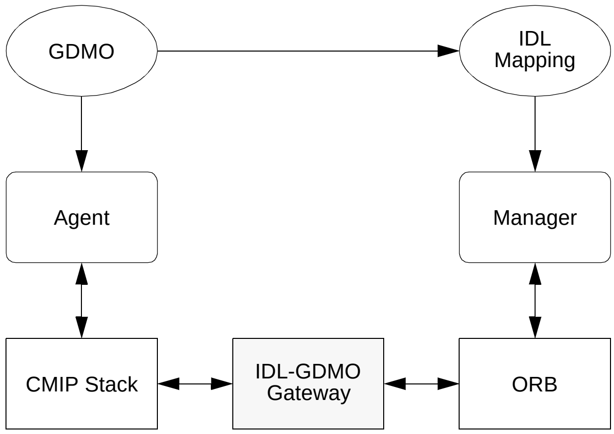

Figure 3: IDL-GDMO Gateway

To conduct communications between a manager and an agent, agreement must be made at two levels:

- **at the object definition level**—where each GDMO object requires an IDL mapping. This mapping includes the object's definition (such as attribute types and values), its operations (such as get, set, create, delete, action), the notifications it can issue, the relationships it can have with other objects (such as its position in a containment tree), and possibly its behaviours.
- **at the communication protocol level**—where messages or requests, which are sent by a manager using protocols that the ORB supports, need to be translated into CMIP before being passed to the agent. Messages or responses sent by a CMIP agent need to be re-translated to the protocols supported by the ORB before being passed to the manager.

Each GDMO object requires an equivalent IDL object definition and each operation in a GDMO object requires an equivalent IDL method definition. Creating a standard set of mapping definitions is difficult.

It is possible to generate IDL objects automatically from original OSI-GDMO definitions. These help to develop an ORB-CMIP Gateway (some NMF projects have taken this approach).

5.2 CHARACTERISTICS OF THE IDL-GDMO GATEWAY APPROACH

Problems with the IDL-GDMO gateway approach include difficulties in handling operations which have filter, scope, and synchronisation conditions. Other problems include CORBA's difficulties in handling linked replies and notifications. CORBA does not readily support the concepts of filtering and scoping, and needs to introduce proxy objects for notifications.

Also, because the mapping between GDMO objects and IDL objects is a pure one-to-one mapping, it does not add any value to the existing network management world. In fact, this approach diminishes some of advantages of both CORBA and CMIP. For example, the flexible object management capability of CORBA and the powerful filtering and scoping capabilities of CMIP are both compromised when the IDL-GDMO gateway approach is taken.

5.3 THE ABSTRACT OBJECT DEFINITION APPROACH

An alternative approach is to introduce a set of CORBA objects at an appropriate level in the management hierarchy. A group of GDMO objects is then mapped into a single CORBA object at a higher level.

CORBA IDL can also be used to define a set of service objects. These services are implemented on top of a set of GDMO network element objects. The relationships between the IDL service objects and the GDMO network element objects are specified and available for later reference.

Requests on a CORBA object are mapped into a set of operations which are performed on a set of GDMO objects according to the relationship between them. For example, a single service activation request is mapped into a set of configuration or creation operations on a set of network element objects.

Events occurring on a GDMO object can be reflected at the CORBA object level. For example, when a network element generates an alarm, relationship records specify which services are affected and to what degree. This allows service level management tasks to occur, for example, informing service customers about a degradation in the quality of a service.

CORBA and CMIP can be used together to manage networks and services. For example, the OSI CMIP-based communication protocol can be used for network management-related communications between a manager and an agent. The CORBA communication protocol can provide service management related communications between management applications and CORBA IDL-defined service objects.

Figure 4 shows an integrated TMN management environment using an abstract object definition approach.

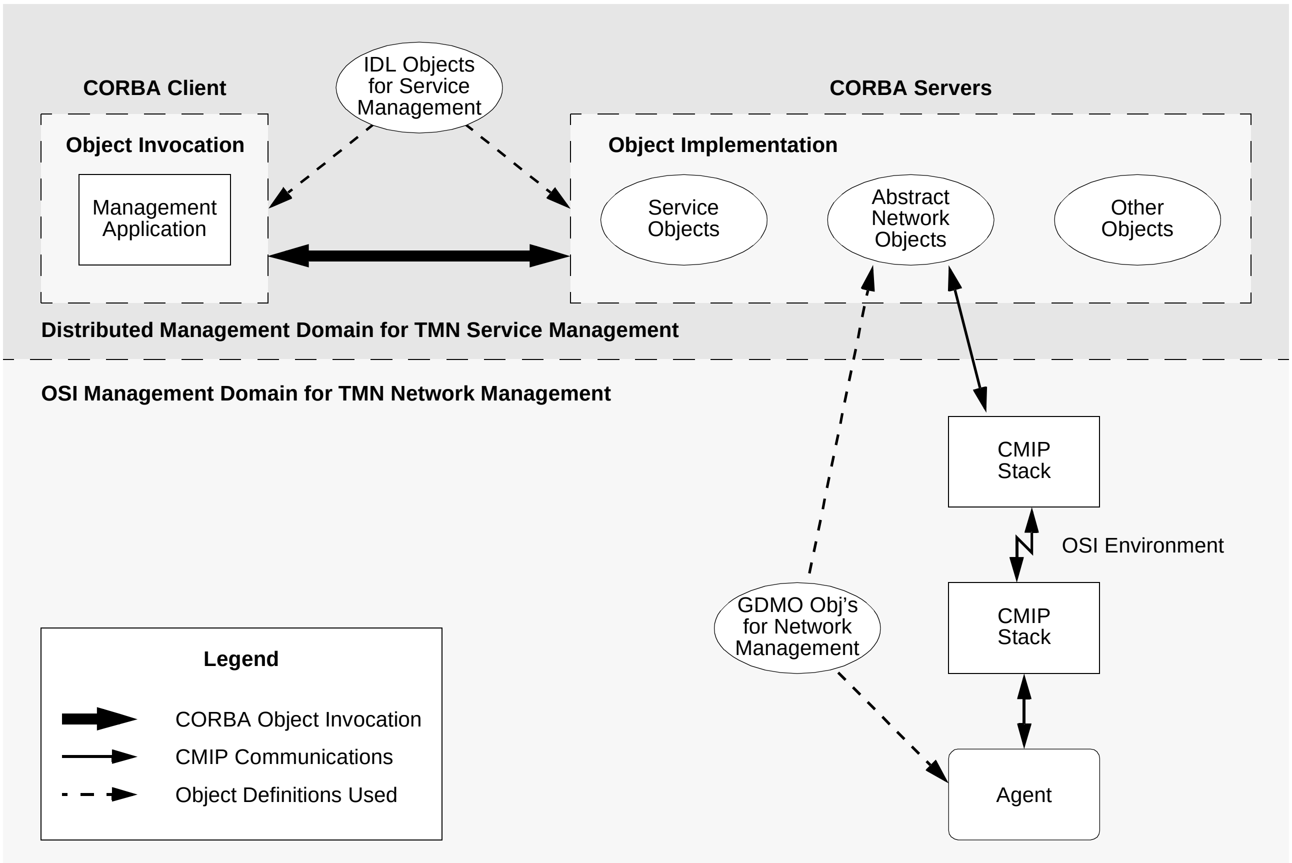

Figure 4: Integrated Management Environment

5.4 CHARACTERISTICS OF THE ABSTRACT OBJECT DEFINITION APPROACH

This approach is not based on one-to-one mappings between GDMO objects and CORBA objects. Further, GDMO operations, such as get operations with filter and scope conditions, are not simulated in the CORBA object environment.

This approach makes the best use of both CORBA and CMIP technologies by using CORBA for service management and CMIP for network management. The integration between service management and network management is supported by representing high-level network management entities using CORBA objects, and by defining relationships between CORBA objects and network objects.

Industry and standards bodies have not yet defined high-level TMN service objects, however, in the future, such objects will play an important role in achieving service level interoperability and standard specifications. The TINA consortium is currently positioned as the best organisation to define these objects for the telecommunications industry. Once specified, they can be represented as CORBA objects.

6 PROTOTYPE IMPLEMENTATION

In order to demonstrate the feasibility of using CORBA for distributed network management we built a prototype system using CORBA/TMN integration. The prototype, known as the Video-On-Demand (VOD) Service Demonstrator:

- evaluated alternative CORBA and TMN integration approaches
- evaluated the scalability and reliability aspects of CORBA/TMN systems
- investigated CORBA and TMN integration for the implementation and management of network-based services

The Demonstrator is a software mock-up of a simplistic VOD service. It is targeted at exercising different aspects of CORBA/TMN integration rather then trying to provide a complete VOD service.

Figure 5 shows the architecture of the VOD Demonstrator system.

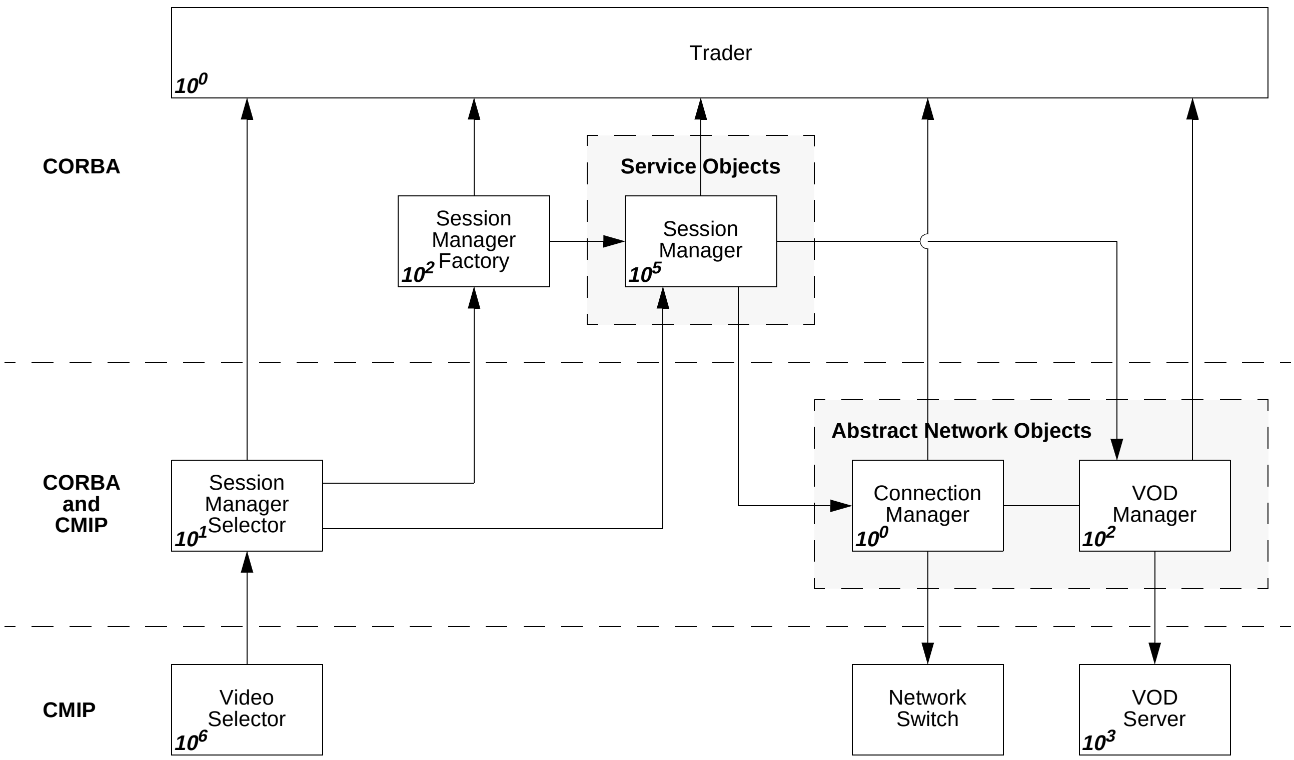

Figure 5: VOD Service Architecture

Both of integration approaches discussed in Section 5, the IDL-Gateway approach and the Abstract Object Definition approach, are demonstrated by the prototype. Abstract Object Definition mapping was used in modelling the Connection Manager and one-to-one object mapping was used in modelling the VOD Manager and VOD Server.

The system was designed to be tolerant to failure of the network connecting the management systems. The system demonstrated recovery of the management path when one of the workstations managing the service failed.

The scalability of the system was tested by measuring the performance of the system when running on single and multiple hosts.

IONA’s Orbix 1.3 was used as the CORBA platform and was integrated with HP’s OpenView DM 3.3 as the OSI environment.

6.1 RESULTS

The prototype demonstrates that different integration approaches may be required in different cases. In the case of using a one-to-one object mapping between the GDMO and IDL, there is still a requirement to introduce extra methods into the target object model—in our case, the IDL object model—to provide value added services.

The VOD demonstrator showed the following key features of CORBA/TMN integration:

- the CMIP stack dominates system performance
- CORBA’s memory usage is high

The CMIP stack used in this system was HP Openview DM 3.3.1. The newer HP Openview DM release 4.1 is reported to have twice the performance of this version. Potentially, this could halve the number of CMIP hosts required for the system.

The high memory cost of CORBA objects and their references makes CORBA usage for small objects inefficient. Therefore, CORBA is more suited for larger and more abstract objects.

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

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

The Abstract Object 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, however, it significantly reduces the number of CORBA objects and their associated requests. The Abstract Object Definition approach optimises the performance characteristics of the CORBA/TMN integration.

7 CONCLUSION

Through this study, we are confident that the TMN/CORBA integration concept is sound and will assist the telecommunication industry to build more reliable, efficient and functional platforms to manage distributed network and service environments. In addition, we make the following recommendations:

- OSI and CORBA are not interchangeable technologies. They are suitable to different domains. In the area of integration, we do not recommend the reconstruction of the OSI management architecture using CORBA.
- Leverage the integration technology to achieve a distributed network and service management platform.
- Develop more generic and reusable services to augment the functionality of the platform. These include fault tolerant, systems management and transaction processing services.

ACKNOWLEDGMENTS

The authors would like to thank Pushkar Piggott, Norm Lawler, Ian Rose, Mike Sharrott and Andrew Baker for their assistance.

REFERENCES

[1] ISO/IEC IS 9595, ITU-T Recommendation X.710, Information Technology—Open Systems Interconnection - Common Management Information Service Definition, 1991.

[2] ISO/IEC JTC1/SC21/N9988: Open Distributed Management Architecture, working draft.

[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.

[6] Integrating CORBA and TMN Environments, Qinzheng Kong and Graham Chen, Proceedings of IEEE/ IFIP Network Operations and Management Symposium, Kyoto, 1996.
[7] CORBA/TMN Integration, CiTR Pty Ltd, Internal Technical Report, 1996.
[8] OMNIPOINT V3, Network Management Forum, 1996.