Design and Implementation of CORBA-based Network Management Applications within TMN Framework

Moon- Sang Jeong , Jeong-Hwan Kim, Joon-Hyub Kwon and Jong-Tae Park
School of Electronic and Electrical Engineering
Kyungpook National University
1370, Sankyuk-Dong, Buk-Gu
Taegu, Korea 702-701,
Tel: +82-53-950-5543
FAX: +82-53-950-5505
(msjeong, jhkim, jhkwon)@ain.kyungpook.ac.kr, park@ee.kyungpook.ac.kr

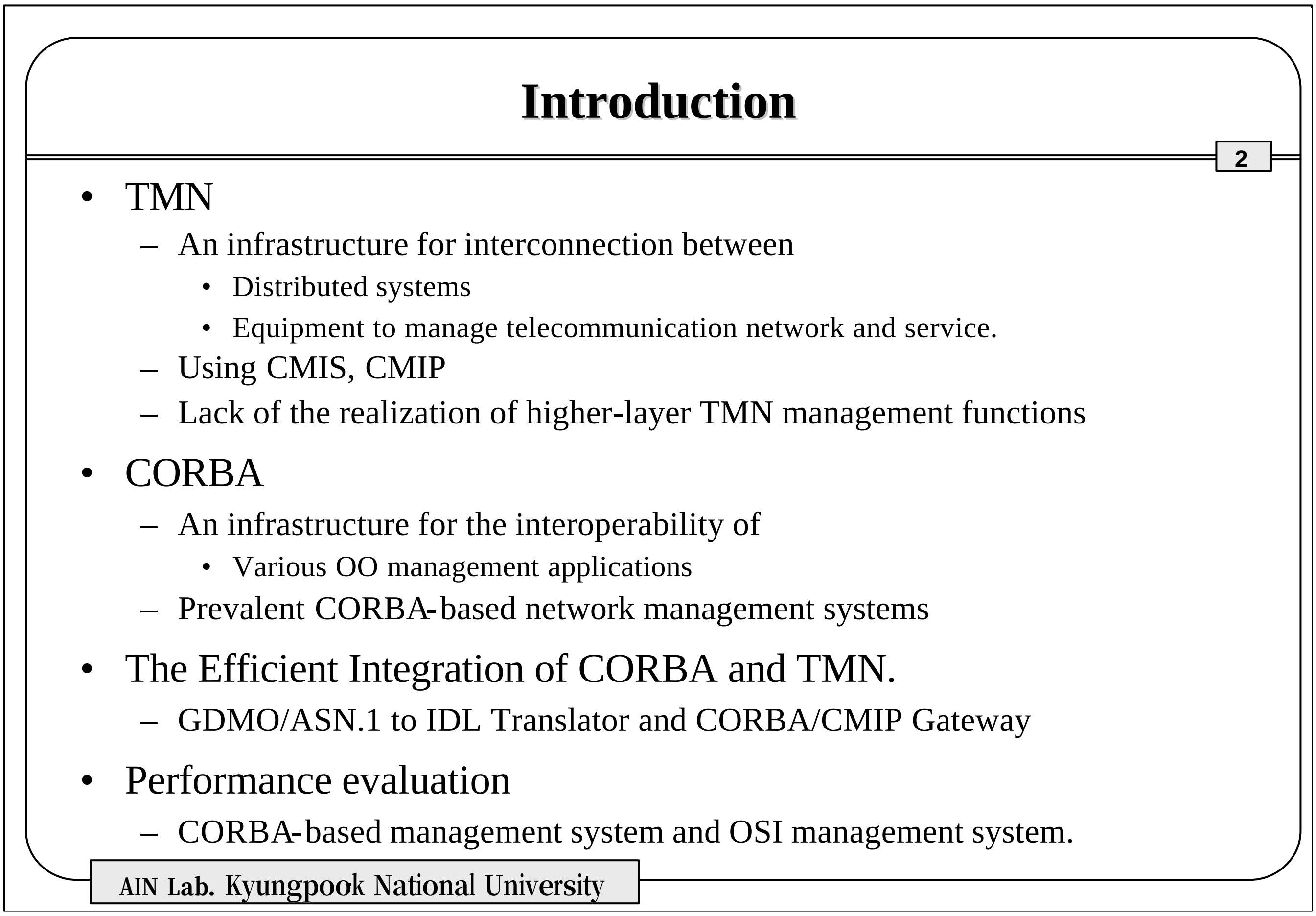
Seong-Beom Kim, Chan-Kyou Hwang
Network Management Technology Research Lab.
Korea Telecom,
62-1, Whaam-Dong, Yusung-Gu,
Taejeon, Korea 305-348
(sbkim, hwang)@nmtl.kotel.co.kr

Abstract

There are recently worldwide growing interests for applying CORBA technology for the realization of higher layer Telecommunication Management Network (TMN) management functions. In this paper, we have developed a translator and a gateway for the integration of CORBA technology within TMN framework. GDMO/ASN.1 to CORBA IDL translator has been designed and implemented for translating TMN management information into CORBA IDL interface. The CORBA/TMN gateway has been designed for the realization of the interaction translation specification of JIDM task force. In particular, OSIEventReport interface for handling OSI event report in CORBA-based network management application has been designed in detail. Finally, we compare the performance of CORBA-based management system with the performance of OSI management system.

Keywords **:** *TMN, CORBA-based management, Integrated network management, OSI management, Network management architecture, GDMO, IDL, Gateway, Performance evaluation*

Contact Person : *Jong-Tae Park (park@ee.kyungpook.ac.kr)*



A Telecommunication Management Network (TMN) [1] is an infrastructure which provides interfaces for interconnection between various types of operation systems (OSs) and/or telecommunications equipment to manage telecommunication network and service, and a management information is exchanged through these interfaces. TMN architecture uses internally CMIS and CMIP for exchanging management information. CMIP provides object-oriented information model and powerful information searching abilities using scoping and filtering. CMIP also provides more diverse event notification services using event forwarding discriminator (EFD) and Log Control functions. However, less effort has been made for the realization of higher-layer TMN management functions for service and business management.

On the other hand, CORBA [2] has been widely adopted for developing distributed systems in many areas of information technology. CORBA provides the infrastructure for the interoperability of various object-oriented management applications in a distributed environment. With CORBA, users can transparently access to the management information, independently of software or hardware platforms. CORBA can support location transparency, and the integration of management information and services. Therefore, it could enhance the portability of applications that are developed across multiple network management platforms.

In this paper, We propose a platform architecture for the efficient integration of CORBA and TMN, where CORBA-based management functions as well as TMN-based management functions can be realized efficiently. The system is based on the JIDM's work, but some extensions are made for the implementation of gateways and management applications. CORBA/CMIP gateway has been designed and implemented in which OSI event report function has been designed with detailed API. We show architectures and implementation methods for the interaction of CORBA and CMIP, and for the management applications in CORBA-based network management systems. We describe the performance evaluation of CORBA-based management system and OSI management system.

Related Works

3

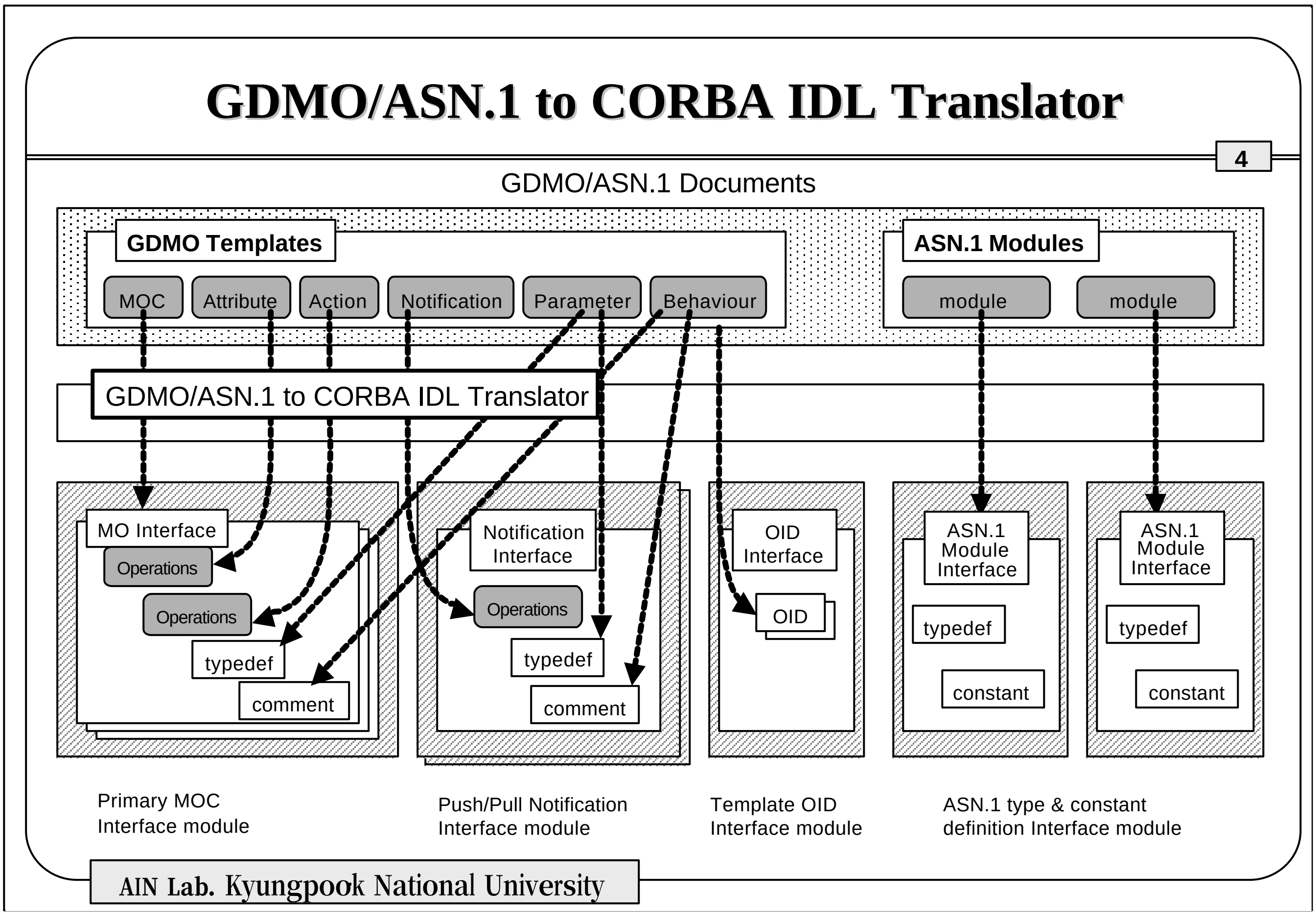
- OMG, ISO/OSI, ITU-T, Lucent Technologies, HP, DSET
- Joint Inter-Domain Management (JIDM) -X/Open, NMF
  - Specification Translation
  - Interaction Translation
  - De facto standard
- ACTS PROSPECT
  - Cooperative, integrated service management
  - Inter-domain management
- CiTR
  - Integrated service management environment with a CORBA-based and a TMN-based management environment
- Field Experience
  - ORBYCOM, HP, KNU, Telefonica I+D, UHC, Alcatel, IBM, ISR Global Telecom

AIN Lab. Kyungpook National University

Recently, there are many efforts for developing CORBA-based network management system which requires the integration of CORBA and TMN technologies. These include the efforts by industrial consortia such as OMG, NMF, X/Open, ISO/OSI, ITU-T, Lucent Technologies, CiTR, HP, DSET, ACTS PROSPECT project, and others [3], [4], [5].

The most representative work of CORBA/TMN integration is that of Joint Inter-Domain Management (JIDM) task force [6] sponsored by Open group and Network Management Forum – currently TeleManagement Forum. In the CORBA/TMN integration, the JIDM works can be divided into two main parts: Specification Translation and Interaction Translation. Specification Translation describes translation algorithm between CORBA IDL [7] and GDMO/ASN.1 which are management information modeling languages used in TMN architecture. Interaction translation describes the dynamic converting mechanisms between the protocols in one domain and the protocols within another domain, without either party being necessarily aware of the conversion process [8].

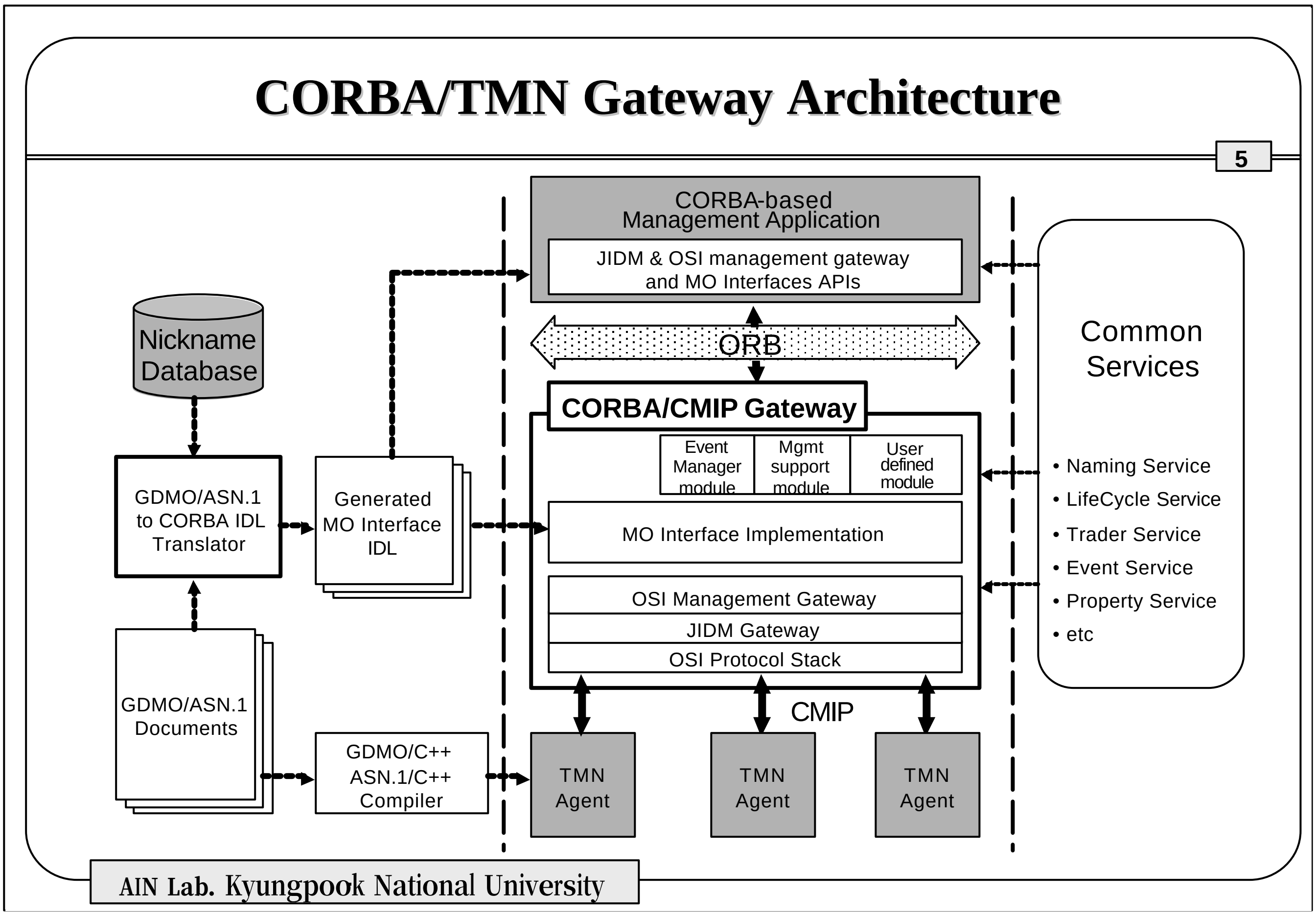
There are some approaches for the implementation of GDMO/ASN.1 to IDL translator, which are not fully suppoprted the JIDM Sepcification Translation document. ACTS PROSPECT project in Europe has tried to give solutions to issues such as cooperative service management, integrated service management and inter-domain management in multiple service provider environments, using CORBA technology within TMN framework [9]. They developed CORBA/CMIP gateway for the integration of CMIP-based network management with CORBA-based service management. They had proposed a management system architecture that had extended JIDM's architecture with some extensions. The researchers at CiTR in Australia had proposed an integrated service management environment with a CORBA-based distributed service management environment and a TMN-based network management environment [10]. They represented the high-level network management entities as CORBA objects for service management network, and defined relationships between CORBA objects and network objects at the lower layer.



OSI information model defines managed object (MO) that represents a management information for network resource . In OSI architecture, MO is described by GDMO [11] and its data definition is described by ASN.1 [12], [13]. GDMO has 9 generic template structures: managed object class (MOC), package, attribute, action, notification, parameter, behavior, attribute group, and name binding. MOC template is on top for representing managed objects about real resources, and the other templates can be derived from it.

CORBA has interface definition language (IDL) for the description of distributed objects, and it is composed of interfaces for representing distributed objects and operations for its behaviors. There does not exist containment relationship between CORBA objects, and these objets can be accessed by directory structure using CORBA naming service. So, it is necessary to develop a conversion mechanism of management information between OSI and CORBA.

GDMO/ASN.1 documents are translated into several IDL modules by translator. GDMO/ASN.1 to CORBA IDL translator generates Primary MOC Interface module for representing MOC information template, and Template OID Interface module for having OID information about all GDMO templates. In Primary MOC Interface module, each GDMO MOC template is mapped to MO Interfaces, and attribute and action templates are mapped to operations. Notification Interface modules have operations for handling events about notification templates. Parameter and behavior templates are mapped to type definitions and comments, respectively. Each ASN.1 module is mapped to IDL module, and its type and value assignments are mapped to type definitions and constants, respectively. This mapping algorithm is described in Specification Translation by JIDM task force. GDMO documents refer to type assignments and OID values in ASN.1 documents, and refer to external GDMO documents templates. So, GDMO/ASN.1 to CORBA IDL translator must perform conversion of management information using its related GDMO and ASN.1 documents for correct translation.



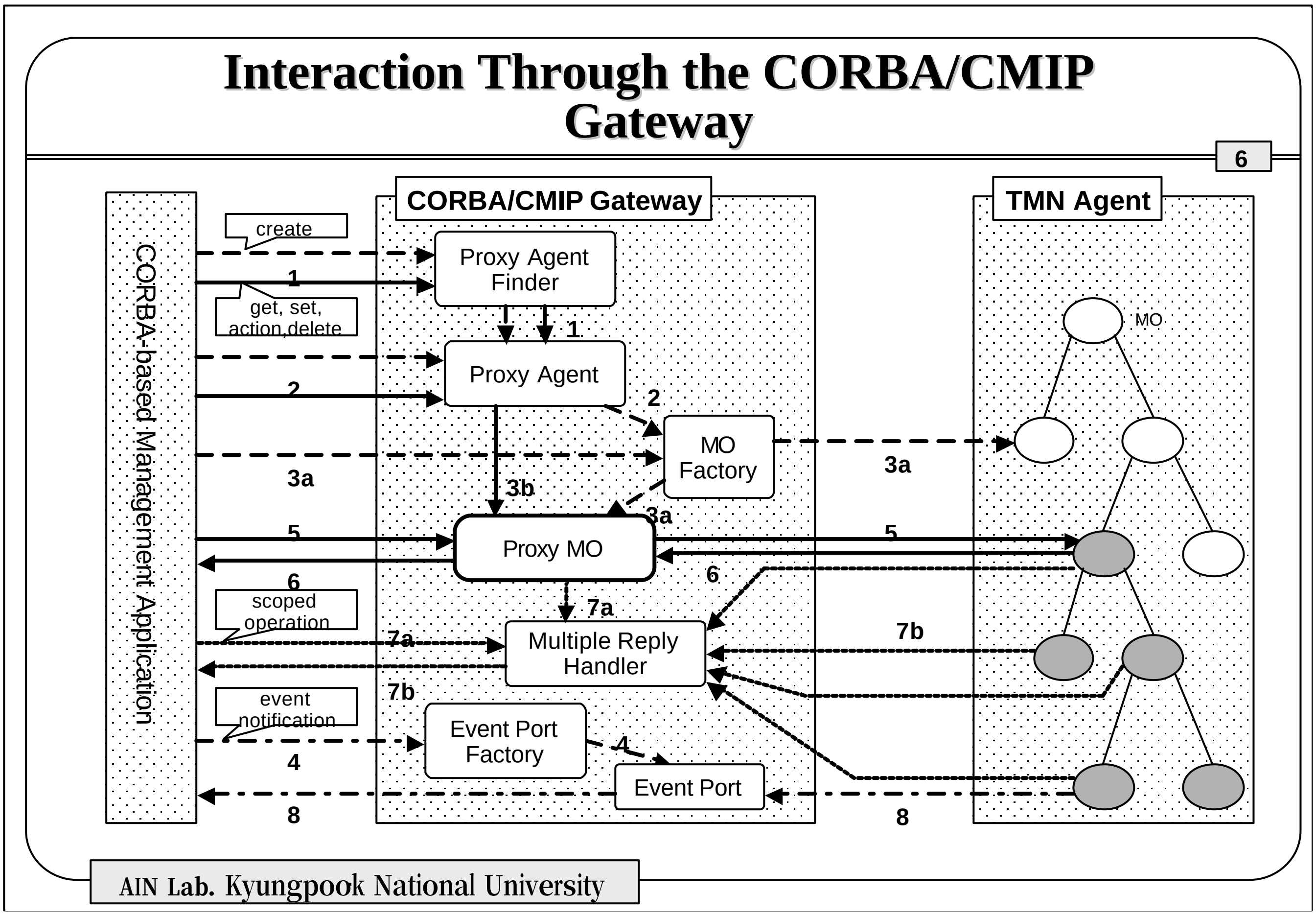
OSI management architecture defines manager-agent role for organizational model. Manager requests an operation to an agent associated with managed objects and the agent returns response to the manager. In OSI management architecture, CMIS primitives provide services for management operations such as create, delete, get, set, action and event notification, and CMIP defines the transmission procedure of management information and the syntax of CMIS management services [14].

In CORBA, the object implementation component defines operations that implement a CORBA IDL interface [15]. The client component is the program entity that invokes an operation on an object implementation. ORB core provides a transparent communicating mechanism between clients and target object implementations.

When CORBA-based management system is applied in OSI management architecture's organization and communication model, it needs CORBA/CMIP gateway for interaction of two inter-domain communication protocols and management operations. For such interoperation, we implement CORBA/CMIP gateway based on JIDM Interaction Translation.

The gateway has OSI protocol stack for communicating with OSI-based TMN agent, and JIDM and OSI management gateway for supporting protocol conversion between CORBA and CMIP. These can support all processing of CMIS primitives.

JIDM and OSI management gateways are responsible for only management protocol conversion and CMIS primitive conversion between CORBA-based management system and TMN agents. Therefore, CORBA/CMIP gateway has the object implementation of IDL code, which is generated by GDMO/ASN.1 to CORBA IDL translator for supporting the operations of CORBA-based management system to a managed system. It provides managed object (MO) Interface APIs and gateway APIs to CORBA-based management application and organizes CORBA-based management system using these.



This figure shows the interaction process between CORBA-based Management Application and TMN agent through the CORBA/CMIP gateway, which is based on JIDM Interaction Translation specification.

The gateway's operations may be classified into 4 cases: create operation, single operation, scoped operation and event notification. Create operation finds or creates Proxy Agent through Proxy Agent Finder. It also creates MO on TMN agent by means of MO Factory and creates Proxy MO on gateway. Single operation finds or creates Proxy Agent by means of Proxy Agent Finder, establishes connection with Proxy MO, and transmits each operation and receives response through Proxy MO. In case of scoped operation, the gateway obtains the reference of Base MO, and transmits scoped operation to agent. And multiple reply handler receives multiple responses from agent, and sends it back to management application. So that, the gateway needs not support containment relationship, Finally, Event Port created by Event Port Factory receives the event notification from TMN agent, and transfers it to management application.

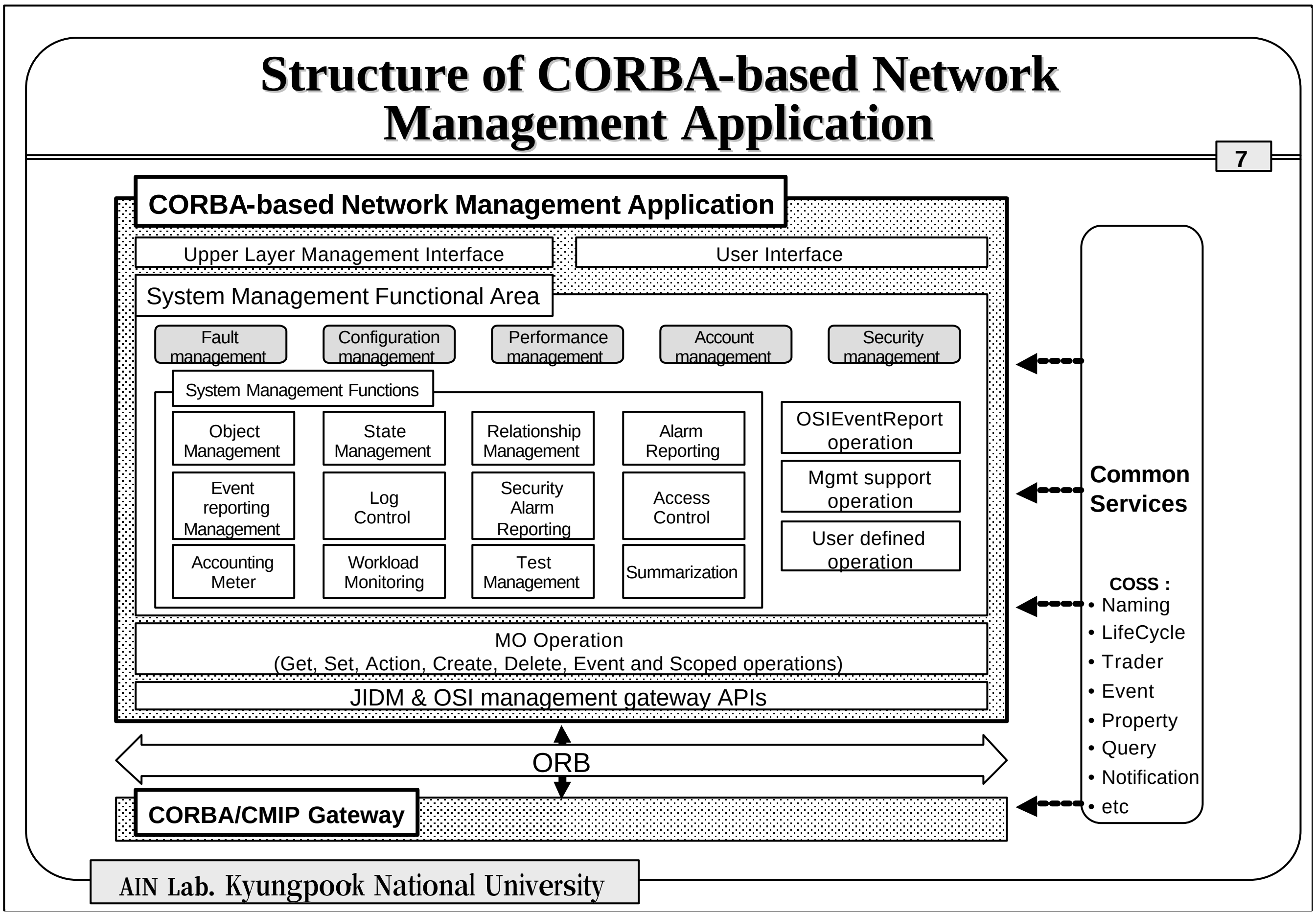
In summarize, CORBA/CMIP gateway performs the following procedures:

1. Send a request to the Proxy Agent Finder to find or create a Proxy Agent
2. Send a request to the Proxy Agent to find or create a Managed Object Factory

3a. Send a request to the Managed Object Factory to create both a TMN Managed Object and an associated Proxy MO

3b. Find a CORBA object reference to an already existing managed object and create an associated Proxy Managed Object

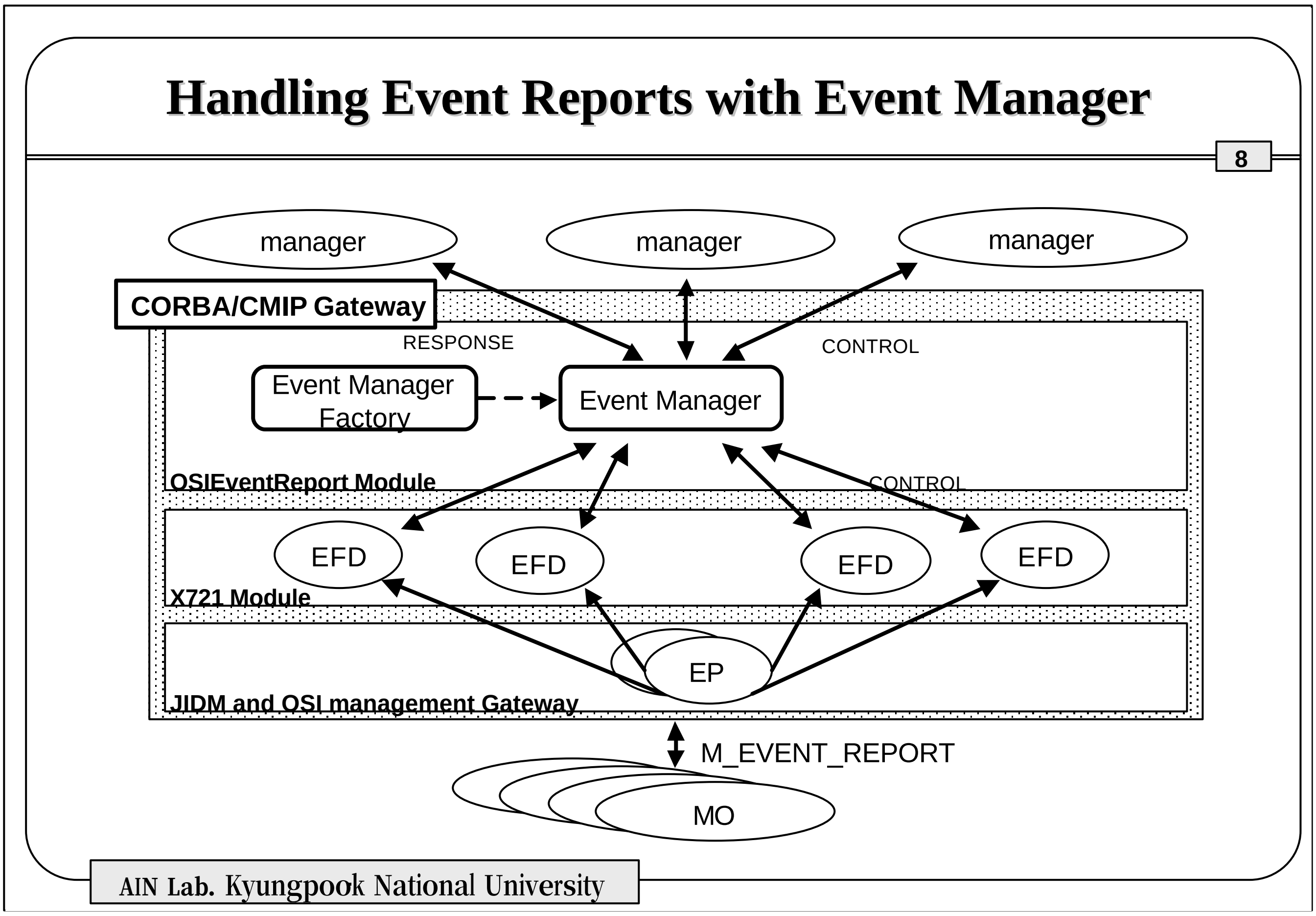
4. Send a request to the Event Port Factory to create an Event Port, then connect an event consumer with the Event Port to receive the event.
5. Send CMIS request (get, set, action, and delete) from the CORBA-based management application to the TMN agent through the Proxy Managed Object
6. Receive responses from the TMN Managed Object synchronously or asynchronously
7. Receive multiple responses from TMN agent, and send it back to management application.
8. Receive an event from the TMN Managed Object by either using push or pull model.



OSI management architecture defines five system management functional areas (SMFA), fault management, configuration management, accounting management, performance management, and security management. Each functional area provides modular development from network management tools, and is customized by a network provider. Also, OSI management architecture's functional model defines various generic functions, system management functions (SMF) which help to develop several network management auxiliary functions [16], [17].

CORBA-based network management application should support these five management areas and provide several management functions. The figure above shows the CORBA-based network management application architecture in which all the management functions and other operations may be realized by using CORBA IDL.

In the figure above, CORBA-based network management application performs general MO operations and management operations associated with SMFAs. First, management application establishes connections with gateway for MO using several CORBA common services, such as naming, lifecycle and trader services. And then it performs CMIS operations using JIDM and OSI management gateway APIs provided by CORBA/CMIP gateway and MO Interface APIs provided by GDMO/ASN.1 to CORBA IDL translator. Using such MO operations and user-defined operations, SMFs may be composed and perform the management functions of functional areas. In addition, management application provides user interface for supporting requests and responses of management behaviors. Upper layer management interface provides additional functionality which is necessary to the realization of TMN service management layer.



In CORBA-based management application, we design OSIEventReport as an example of management function. Event Manager enables an open system to establish and control the discrimination and the forwarding of event reports to other open systems [18]. The figure above shows Event Manager's handling mechanism for event reports.

The event reporting management function provides the capability for setting up a long-term event reporting relationship between two open systems.

Event report that is generated in an agent side experiences several procedures at management system. Events which are generated by status change, attribute value change, creating or deleting MO are transferred to Event Port (EP) in JIDM and OSI management gateway. And Event Port transmits event reports to EFDs that are created by event manager and to existing EFDs. EFDs that receive event reports manipulate them according to attributes, and event manager may forward, suspend, resume, or delete it. EFD initializes attributes when created and also may modify attributes value by external system manager.

Managers may use operations provided by event manager due to exported interfaces. Manager may create and initialize a new discriminator by calling create operation on event manager in order to accept a desirable event. Once a discriminator is created, an EFD that inherits from it is created. Event reports that are generated externally are transferred to every EFD through event ports on CORBA/CMIP gateway. IF event reports are transferred to EFD, then MOC, MO instance, event type, and attributes specific to event type such as instance severity, backed up status, and probable cause are tested by testFilter. Only if EFD satisfies filter, EFD shall transfer event reports to destination.

module OSIEventReport


```
#ifndef OSIEventReport_idl
#define OSIEventReport_idl
#include "X721.idl"
#include "OSIMgmt.idl"

module OSIEventReport {
interface EventManager {

  // operations for discriminator
  X721::discriminator createDC(
    in X721Att::SimpleNameType did,
    in X721Att::DiscriminatorConstructType dc,
    in X721Att::AdministrativeType as,
    in X721Att::OperationalStateType os,
    in X721Att::AvailabilityStatusType avs,
    in X721Att::StartTimeType strt,
    in X721Att::StopTimeType stpt,
    in X721Att::IntervalsOfDayType iod,
    in X721Att::WeekMask wm,
    in X721Att::SchedulerName sn
    )raises(ATTRIBUTE_ERRORS);
  void deleteDC(
    in X721::discriminator dd
    )raises(ATTRIBUTE_ERROS);
  void suspendDC(
    in X721::eventForwardingDiscriminator sefd
    )raises(ATTRIBUTE_ERRORS);
  void resumeDC(
    in X721::eventForwardingDiscriminator refdd
    )raises(ATTRIBUTE_ERRORS);
  void modifyDC(
    in X721::discriminator md
    )raises(ATTRIBUTE_ERRORS);
  boolean discriminateDC(
    in X721Att::DiscriminatorConstructType dc,
    in X721Att::AdministrativeType as,
    in X721Att::OperationalStateType os,
    in X721Att::AvailabilityStatusType
    )raises(ATTRIBUTE_ERRORS);

// operations for eventforwardingdiscriminator
  X721::eventForwardingDiscriminator createEFD(
    in X721::discriminator d,
    in X721Att::DestinationType dest,
    in X721Att::BackUpDestinationListType blist,
    in X721Att::ActiveDestinationType adest,
    in X721Att::ConfirmedModeType cm
    )raises(ATTRIBUTE_ERRORS);
  void deleteEFD(
    in X721::eventForwardingDiscriminator defd
    )raises(ATTRIBUTE_ERROS);
  void modifyEFD(
    in X721::eventForwardingDiscriminator mefd
    )raises(ATTRIBUTE_ERRORS);
  boolean testFilterEFD(
    in X721::eventForwardingDiscriminator efd
    )raises(ATTRIBUTE_ERRORS);
  };
interface EventManagerFactory {
  EventManager createEventManager (
    in Key k,
    in Criteria creation_criteria
    )raises (InvalidKey,
            InvalidCriteria,
            CannotMeetCriteria,
            AlreadyExists);
  };
};
#endif
```

AIN Lab. Kyungpook National University

Event report management model

It describes the conceptual components that provide for remote event reporting and local processing of potential event reports. The model also describes the control messages, event reporting messages and retrieval messages.

The conceptual event pre-processing function receives local notifications and forms the potential event reports. Conceptually, these potential event reports are distributed to all event forwarding discriminators that are contained within the local open system. A potential event report is perceived as a discriminator input object for the purposes of discrimination by the event forwarding discriminators only and is not visible from outside the local system.

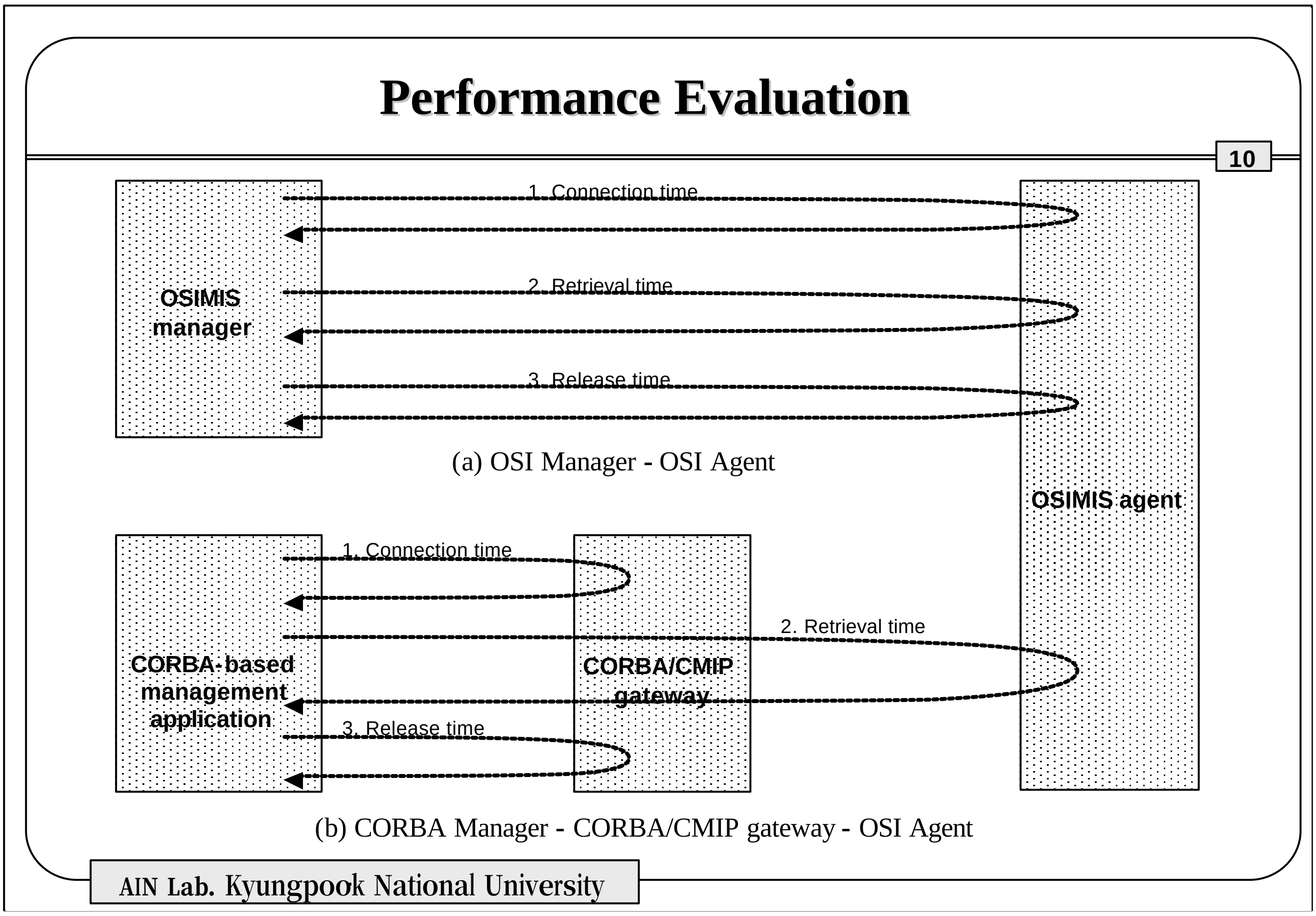
The event forwarding discriminator is used to determine which event reports are to be forwarded to a particular destination during specified time periods. It may also be used to specify the mode (confirmed or non-confirmed) for

forwarding events. Each event forwarding discriminator may contain a scheduling capability determining the intervals during which event reports will be selected for forwarding. Each event forwarding discriminator contains a discriminator construct which specifies the characteristics a potential event report must satisfy in order to be forwarded. Event reports that have been selected are forwarded to the destination as soon as possible. The event forwarding discriminator is itself a managed object and can therefore emit notifications. These notifications are processed as potential event reports by all event forwarding discriminators including the one that generated the notification.

Event reporting management functions

Specified interface performs the following functions:

- Initiation of event forwarding;
- Termination of event forwarding;
- Suspension of event forwarding;
- Resumption of event forwarding;
- Modification of event forwarding conditions;
- Retrieval of event forwarding conditions.



(a) OSI Manager - OSI Agent

(b) CORBA Manager - CORBA/CMIP gateway - OSI Agent

We evaluate the performance of the implemented CORBA-based network management system for CORBA/TMN integration which is compared with OSI manager/agent system for a case study. We measure the operation execution time of CORBA-based management system and OSI system, both of which are accessing an identical agent system. We also measure the overhead of CORBA-based management system compared with OSI system.

To measure the operation execution time of CORBA-based management system, we use OSIMIS agent system [19] running in SUN SPARC 1000 system. OSIMIS OSI agent has X.721 MIB, Unix MIB and M.3100 MIB [20]. We measure the operation execution time of OSI management system for the performance comparison with CORBA-based management system. We use mibdump program which performs M-GET operation in OSIMIS for the performance measurement of OSI management system. CORBA-based management system is implemented by using Orbix 2.3 [21] and its OSI protocol stack is implemented by using ISODE 8.0[22].

We measure the elapsed time until management systems receive replies in response to CMIS primitive request for the same agent. Both managers and agent are placed at the same system, so that the effect of other communication traffic to the total elapsed time is minimized. Figure above shows the system architecture for experimentation. Figure (a) shows the structure for the measurement of the operation execution time from OSI manager to OSI agent, and it consists of three part: connection time for performing connection between OSI manager and agent, retrieval time for the execution of CMIS operation to the connected agent, and release time for closing connection. Figure (b) shows the structure for the measurement of operation execution time from CORBA-based management application to OSI agent via CORBA/CMIP gateway, and it also consists of three part: connection time for performing connection between CORBA-based management application and gateway, retrieval time for the execution of CMIS operation from CORBA-based management application to OSI agent, and release time for closing connection. We measure the total execution time for each case. In CORBA-based management system, we also measure the operation execution time in case of using connected proxy managed object when management application holds its reference. We compare the performance of CORBA-based management system with the performance of OSI management system using these measured times. Experiment is divided into the single operation and the scoped operation of CMIS M-GET primitive.

Execution Times of CORBA and OSI Management System

11

Figure A (for Single Operation)

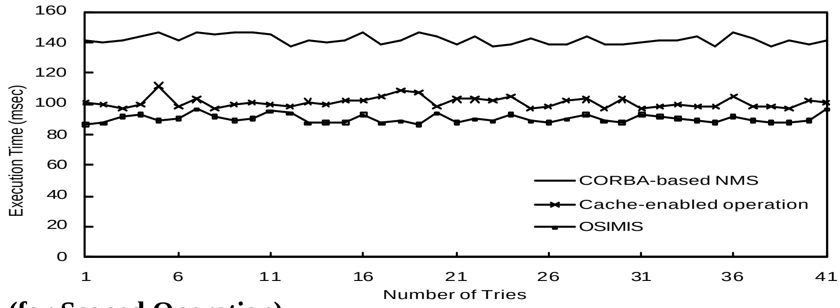
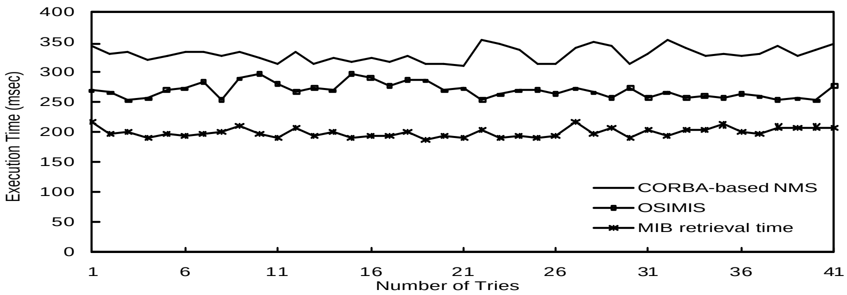


Figure B (for Scoped Operation)



AIN Lab. Kyungpook National University

Experimental Results for Single Operation

Figure A shows the experimental results of single operation for M-GET operation in CORBA and OSI management system. In case of OSI management system, it took average 54.179msec for connection time, 27.144msec for retrieval time and 9.867msec for release time, which sums to a total of 91.19msec for the execution of single operation. In CORBA-based management system, it takes average 41.818msec for connection time and 97.776msec for retrieval time, summary to a total of 141.624msec for the execution of single operation. And it took average 100.678msec in case of using connected proxy managed object that had been not destroyed.

Experimental Results for Scoped Operation

Figure B shows the experimental results of scoped operation for M-GET operation in CORBA and OSI management system. In case of OSI management system, it took average 58.107msec for connection time, 199.500msec for retrieval time that takes to get a response from a agent to and 12.209msec for release time, which sums to a total of 269.816msec for the execution of single operation. In CORBA-based management system, it takes average 44.735msec for connection time and 278.195msec for retrieval time, summary to a total of 330.583msec for the execution of single operation.

Analysis

When it comes to the comparison of figure A with figure B, the elapsed time difference of the operation execution on CORBA-based NMS with the other is lager than in figure B. Generally, single operation execution time is shorter than scoped operation execution time. In addition to this, execution time on CORBA-based NMS is slower than the others. From simulation's outcome's point of view, execution time is good affirmative.

Performance Comparison of CORBA-based management system and OSI system

12

	Single Operation			Scoped Operation	
	OSIMIS	CORBA-based NMS		OSIMIS	CORBA-based NMS
		Normal	Connected PMO		
Connection time	54.179ms	41.818ms	NA	58.107ms	44.735ms
Retrieval time	27.144ms	97.776ms	NA	199.500ms	278.195ms
Total time	91.19ms	141.624ms	100.678ms	269.816ms	330.583ms
Retrieval overhead	29.76%	69.04%	NA	73.94%	82.64%
Total overhead	100	155.31	110.40	100	122.52

AIN Lab. Kyungpook National University

Table shows the performance comparison with CORBA-based management system and OSI system. Retrieval overhead is defined to be the proportion of retrieval time over total time. In case of single operation, it took 29.76% in OSI system and it took 69.04% in CORBA-based management system. In case of scoped operation, it took 73.94% in OSI system and it took 82.64% in CORBA-based management system. In CORBA-based management system, the retrieval time for the operation of CMIS primitive includes the total execution time of OSI management system so that it is larger than the total execution time of OSI system alone. However, it is note that the overhead of CORBA-based system for a scoped operation is not so much as the case for a single operation. As shown in the table, for the case of scoped operation, the retrieval time was also a dominating factor 73.74% for OSI management system, and 82.64% for CORBA-based system for the total execution time. Connection times in CORBA-based management system were similar for single and scoped operations. Suppose the total execution time of OSI management system is 100, COABA-based management system has the overhead of 155.31 in single operation and of 122.52 in scoped operation. For the case of using connected proxy MO, the overhead was 110.40.

By looking into the experimental results, CORBA-based management system has some overhead in connection time for the connection of CORBA-based management application with CORBA/CMIP gateway. Therefore it may be necessary to find out the methods to minimize the connection time for system performance improvement. JIDM Interaction Translation optionally defines a caching mechanism to provide CORBA-based management application with fast and efficient access to the values of attributes of a managed object. The caching mechanism enables CORBA ManaedObject object to maintain a local store of attribute values, thus eliminating the need to contact the real underlying managed object when this information is requested. Any managed object that has been configured with the ability to cache may also be optionally configured with the ability to dynamically update its cached attribute values in response to change notifications received from the underlying managed object. This mechanism is known as tracking. Using these mechanisms, the performance of CORBA-based management system may be improved.

Conclusion & Future Work

- Integration of CORBA technology within TMN framework.
  - GDMO/ASN.1 to IDL Translator and CORBA/CMIP Gateway
- OSIEventReport module, using CORBA IDL.
- Compared CORBA-based management system with OSI system.
  - Overhead but negligible for scoped operations.
  - Overcome with various techniques
- Improving performance with caching, tracking mechanism.
- Implementing a CORBA-based management platform
  - Extending management functional interfaces
  - Extending the architecture for the management of IMT-2000 and high-speed networks.

In this paper, we have developed a translator and gateway for the integration of CORBA technology within TMN framework. We have designed a platform architecture for CORBA/TMN integration. We have designed and implemented GDMO/ASN.1 to CORBA IDL translator architecture for the conversion of management information, and CORBA/CMIP gateway for the interaction between CORBA and CMIP in detail. For implementation of GDMO/ASN.1 to CORBA IDL translator, we have developed a new directional path finding algorithm for efficient and correct reordering of GDMO and ASN.1 documents. GDMO/ASN.1 to CORBA IDL translator satisfies full specification of JIDM Specification Translation document. We have also designed and implemented CORBA-based management application architecture. OSI event report management function has been developed by defining a module, OSIEventReport module, using CORBA IDL. Finally, we have evaluated the performance of CORBA-based management system in comparison with OSI management system. In CORBA-based management system, the operational procedures between CORBA/CMIP gateway and TMN agent include the whole operational procedure of OSI management system so that CORBA-based management system may have some overhead for the connection of CORBA-based management application with CORBA/CMIP gateway. The overhead of CORBA-based management system was shown to be negligible for scoped operation.

We are planning to improve the performance of CORBA-based management system by using caching and tracking mechanism. We are planning to extend management functional interfaces for CORBA-based management system, and to implement a CORBA-based network management platform. We are also planning to extend the architecture for the management of other emerging network and services such as IMT-2000 and high-speed networks. The GDMO/ASN.1 to IDL translator is now available at http://ain.kyungpook.ac.kr/nexus on your request.

References

[1] ITU-T Recommendation M.3010, *Principles for a Telecommunication Management Network*, May 1996.

[2] OMG, *CORBA 2.0/IIOP Specification*, Technical Document formal/97-02-25, February 1997.

[3] J. T. Park, S. H. Ha, and James W. K. Hong, "Design and Implementation of TMN SMK System Using CORBA ORB," *Journal of Network and Systems Management,* Plenum Press, New York, To be published in June 1998.

[4] J. T. Park, J. H. Lee, J. W. Hong, Y. M. Kim and S. M. Kim, "A VPN Management Architecture for Supporting CNM Services in ATM Networks," *Proceeding of the IEEE/IFIP International Symposium on Integrated Network Management*, May 1997, pp.44-57.

[5] J. W. Baek, T. J. Ha, J. T. Park, J. W. Hong, and S. B. Kim, "ATM Customer Network Management Using WWW and CORBA Technologies," *Proceeding of the IEEE/IFIP Network Operations and Management Symposium*, New Orleans, Louisiana, February 1998, pp.120-129.

[6] N. Soukouti and U. Hollberg, "Joint Inter Domain Management: CORBA, CMIP and SNMP," *Proceeding of the 5th IFIP/IEEE International Symposium on Integrated Network Management*, May 1997, pp.153-164.

[7] X/Open and NMF, *Inter-domain Management: Specification Translation,* Open Group Preliminary Specification P509, March 1997.

[8] OMG CORBA/TMN Interworking RFP, *JIDM Interaction Translation*, Relevant Document telecom/98-10-10, October 1998.

[9] PROSPECT Consortium, *D33A: Concepts for CORBA/TMN interworking*, Deliverable Number: AC052/GDM/WP3/DS/S/003//b1, May 1997.

[10] Graham Chen and Qinzheng Kong, "Integrated TMN Service Provisioning and Management Environment," *Proceeding of the 5th IFIP/IEEE International Symposium on Integrated Network Management,* May 1997, pp.99-112.

[11] ITU-T Recommendation X.722, *Information technology - Open Systems Interconnection - Structure of management information: Guidelines for the definition of managed objects (GDMO)*, January 1992.

[12] ITU-T Recommendation X.208, *Open Systems Interconnection Model and Notation - Specification of Abstract Syntax Notation One (ASN.1),* 1988.

[13] ITU-T Recommendation X.680, *Information Technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation*, July 1994.

[14] ITU-T Recommendation X.711, *Information Technology – Open Systems Interconnection – Common Management Information Protocol: Specification*, 1992.

[15] OMG, *Object Management Architecture Guide*, Third Edition, Wiley & Sons, 1995.

[16] ITU-T Recommendation X.700, *Management framework for Open Systems Interconnection (OSI) for CCITT applications*, September 1992.

[17] ITU-T Recommendation M.3400, *TMN management functions*, April 1997.

[18] ITU-T Recommendation X.734, *Information Technology - Open Systems Interconnection – Systems Management: Event Report Management Function*, September 1992.

[19] University College London Computer Science Department, *The OSI Management Information Service User's Manual, Version 4.0*, for system version 4.0, 1994.

[20] ITU-T Recommendation M.3100, *Generic Network Information Model*, July 1995.

[21] IONA Technologies, *Orbix Programmer's Guide, Orbix 2, Distributed Object Technology*, October 1997.

[22] Marshall T. Rose, Julian P. Onions, and Colin J. Robbins, *The ISO Development Environment User's Manual, – Volume 1: Application Services, Version 7.0*, July 18, 1991.