

---

# Object Web을 이용한 CORBA기반의 OSI 망관리

김 강 석\*, 송 왕 철\*

## OSI Network Management of CORBA-based using Object Web

Gang-Suk Kim\*, Wang Cheol Song\*

### 요 약

CORBA는 어플리케이션들이 네트워크, 언어, 컴포넌트 경계, 운영체제 간에 걸쳐 그들의 범위를 확장할 수 있게 하는 분산 객체 기반의 미들웨어 환경을 제공한다. 자바는 모든 주요 운영체제 위에서 작업을 수행할 수 있는 이식 가능한 객체를 제공할 수 있다. 웹과 CORBA, 그리고 자바 매핑은 자바에서 제공하는 모바일코드 기능과 CORBA에서 제공하는 객체 컴포넌트 기술을 적용하여 대규모 정보 시스템을 웹 위에서 구축할 수 있게 해준다. 이러한 개발환경을 Object Web이라고 정의하고 있다.

본 논문에서는 효율적인 OSI 망관리를 위해 분산 객체 기술인 CORBA를 기반으로 하여 Object Web을 이용한 CORBA 관리자과 CMIS-레벨 게이트웨이 모델을 적용한 CORBA/CMIS 게이트웨이를 설계하고 그 구조를 기반으로 실제 시스템을 구현하였다. 제시된 구조에서 CORBA 관리자는 Object Web을 기반으로, OSI 영역내의 관리객체들을 관리하며, 게이트웨이는 CORBA IDL들을 CMIS 서비스로의 변환과 역변환을 담당한다. 이를 위해 OSI 대리자를 CORBA 대리자로 추상화하였다.

### Abstract

The CORBA provides a object-oriented middleware that lets applications extend their range across networks, languages, component boundaries, and operating systems. Java can provide a portable object that works in any operating system. Mapping Java into CORBA helps large scale information system operate through the Web by using the mobile code property of Java and object component technique of CORBA. This development environment is called the Object Web. This paper proposes an architecture of OSI network management with a CORBA manager using Object Web and an CORBA/CMIS gateway which apply a CMIS-level gateway model

---

\* 제주대학교 정보공학과  
접수일자 1999년 1월 14일

to gateway. In this architecture, CORBA manager is based on Object Web and can manage MOs in the OSI domain. The gateway has a role to translate CORBA IDLs into CMIS services and vice versa. In addition, an OSI agent is abstracted into a CORBA agent.

## I. 서론

망관리의 목적은 통신망을 효율적으로 운용하고, 망이 정상적으로 동작하고 있는가를 보장해 주는데 있다. 현재까지 여러 종류의 통신망 관리의 국제 표준은 상호 연관성을 가지면서도 주요 관리 대상에 따라 각각 독립적으로 연구되어 왔다. ITU-T의 TMN(Telecommunication Management Network)은 통신망의 전송 시스템 및 교환 시스템을 관리할 목적으로 개발되었고[1], ISO의 OSI(Open Systems Interconnection)관리는 OSI 자원을 관리할 목적으로[2], 또한 IETF의 SNMP(Simple Network Management Protocol)는 TCP/IP등의 인터넷을 관리할 목적으로 개발되었다[3].

분산 시스템 분야에서는 최근 소프트웨어와 하드웨어의 이질성을 해결하기 위하여 미들웨어에 대한 연구가 OMG(Object Management Group)[4]를 중심으로 진행되고 있으며, 객체 지향 기술을 바탕으로 분산 객체 시스템의 표준으로 CORBA(Common Object Request Broker Architecture)를 제안하고 있다[5]. 한편, 통신 서비스 이용자들은 보다 신속하고 다양한 서비스의 제공을 요구하고 있으며, 통신망 환경 또한 비대화·복잡화되어 가고 있다. 이에 따라 NMF(Network Management Forum)[6]의 OMNIPoint에서는 기존의 ITU-T/TMN, ISO/OSI, OMG/CORBA, OSF/DME 등에서 제시된 다양한 관리구조의 통합에 대한 연구를 진행하고 있고[7], OMG에서도 CORBA를 이용한 망관리 구조를 연구하고 있으며[8], X/Open과 NMF의 JIDM(Joint Inter-Domain Management) 태스크포스는 OMG/CORBA, ISO/OSI, IETF/SNMP간의 관리를 위해 정보모델과 프로토콜의 변환에 대한 연구를 진행중이다[9].

본 논문에서는 플랫폼에 독립적인 관리기능의 구현을 위하여 Object Web을 이용한 CORBA 관리자, 그리고 관리자가 OSI 영역(Domain)내의 관리객체

들을 일관성 있게 관리할 수 있고, 객체 클래스의 수를 최소화하기 위한 CMIS(Common Management Information Service)게이트웨이 구조를 제시하였다. 게이트웨이는 기존 대리자를 수정하지 않은 채 CORBA 대리자(Agent)로 보여지도록 하였다. 이는 OSI의 관리구조로 CORBA 기반의 구조가 사용될 때 게이트웨이를 이용하여 기존의 자원을 CORBA 대리자로 동작하도록 함으로써 현재 사용중인 자원 또한 CORBA를 이용하여 관리가 가능하다는 장점을 얻을 수 있다.

## II. OSI 망관리 시스템

OSI시스템 관리에서 기본적인 기능이 관리자와 대리자간의 정보 교환을 통한 망관리이다. 이 시스템 관리를 목적으로 정보와 명령을 교환하기 위하여 이용하는 기능이 CMISE(Common Management Information Service Element)이다. CMISE는 관리 응용과의 인터페이스 서비스를 담당하는 CMIS와 PDU(Protocol Data Unit)및 관련절차를 규정한 CMIP(Common Management Information Protocol)으로 이루어진다.

### 1. CMISE

CMISE서비스는 기본적으로 요청받은 서비스에 대하여 실패, 성공이나 요청을 받았다는 응답을 확인해주는 확인 서비스들과 응답을 하지 않는 비확인 서비스로 나눌수 있으며, 다음 두 개의 ISO 어플리케이션 프로토콜을 사용한다.

- ACSE(Association Control Service Element) : 어플리케이션간의 어소시에이션을 설정하고 해제한다.
- ROSE(Remote Operations Service Element) : 어플리케이션간의 상호작용을 요구하거나 응답한다.

2. CMIS

CMIS는 네트워크 관리 어플리케이션이 수행시킬 수 있는 오퍼레이션이다. 여기에는 시스템간 연결 및 해제를 위한 서비스인 A-Associate, A-Reales, A-Abort가 있고 대리인에서 관리자에게 관리 객체에 발생한 특정 상황을 통지하는 M-EVENT-REPORT 서비스가 있다. 그리고 실제로 관리객체들에 대한 정보를 주고받는 서비스인 관리운용서비스에는 여섯 가지 서비스가 있다. 이들 서비스는 다음 세 가지 범주로 나눌 수 있다.

1) Association Service

통신을 하기 위해서 해당 시스템의 어플리케이션간의 연결 및 해제를 하기 위해서 필요하며 ACSE를 이용한다.

2) Management-notification Service

관리 객체들에 대한 특정사건을 보고하는데 이용하는 것으로 필요한 속성, 행동 및 작동절차 등에 대한 것은 각 관리객체들에 정의되어 있다. 이를 위한 서비스로 M-EVENT-REPORT가 있다.

3) Management-operation Service

시스템 관리를 위해서 정보를 주고 받는데 이용되는 M-GET, M-SET, M-CREATE, M-DELETE, M-ACTION, M-CANCEL-GET과 같은 여섯 가지 서비스가 있으며 관리객체에 대한 사항은 관리객체들에 정의된 것에 의존한다.

M-EVENT-REPORT를 포함한 일곱 가지 서비스 중에서 M-GET, M-CREATE, M-DELETE, M-CANCEL-GET은 반드시 응답을 받아야하는 확인 서비스이고 나머지는 서비스를 요청할 때에 확인/비확인을 위한 파라미터를 주어야 한다. M-CANCEL-GET은 이미 요청된 M-GET을 중지시킬 때 사용되는 서비스로써 이를 사용하는 이유는 MIB(Management Information Base)이 수정 도중에 문제가 발생하면 MIB의 일관성을 보증하기가 어렵기 때문이다.

3. CMIP

CMIP은 관리정보를 전송하는 절차 즉, CMISE 사이에 CMIS 서비스를 완성시키기 위해서 교환하는 CMIP PDU를 만들고 전송하는 것에 대해 정의해 놓은 것이다. 양단의 CMISE 사용자들이 정보교환을 위해서 시스템을 연결하는 데 ACSE를 이용하는데 이때는 CMIP이 이용되지 않는다. 관리 서비스를 위해서 CMISE는 PDU를 교환하기 위해 CMIP를 채용한다. 그리고 CMIP는 CMIP PDU전송을 위해서 ROSE를 이용한다.

4. 관리자와 대리자

그림 1은 OSI 시스템 관리에서의 관리자와 대리자사이의 상호관계를 보여주고 있다.

관리자와 대리자간에는 CMIP 프로토콜을 사용하여 망 관리 정보를 주고받는다. 관리 객체는 망에서 감시의 대상이 되는 자원들을 객체로 모델링한 것

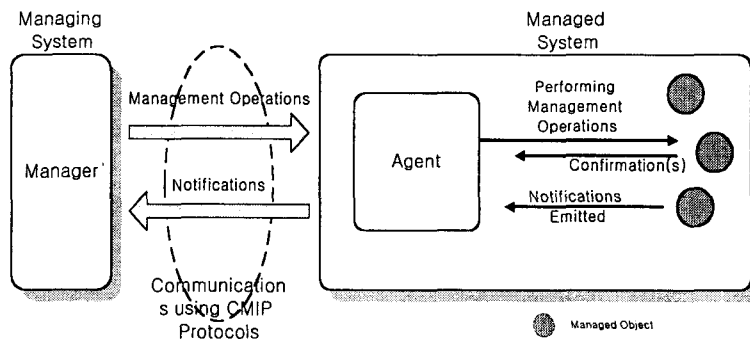


그림 1. OSI 시스템 관리에서의 관리자와 대리자  
Fig. 1. Manager and agent for OSI system management

인데, 관리객체의 속성이 자원의 상태를 반영하고 있고 특정한 사건이 있을 경우 사건에 대한 통지를 대리자에게 보낸다. 대리자는 관리자로부터의 요청에 대하여 관리 객체의 값을 읽어 반환하고, 관리 객체에서 발생한 사건을 수신하여 관리자로 전달해 준다. 관리자를 사용하여 운용자는 망을 운용하고, 관리 객체에서 발생한 사건을 수신하면 이를 운용자에게 알려 적절한 조치를 취할 수 있도록 한다.

### Ⅲ. 분산 객체 기술을 이용한 망관리

#### 1. CORBA 개념

OMG에서 객체지향 기술을 기반으로 이종의 분산된 환경 하에서 응용 프로그램들이 서로 통합할 수 있는 표준기술로 OMA를 제안하였다. OMA는 ORB(Object Request Broker), COSS(Common Object Services Specification), Common Facilities, 어플리케이션 객체 등으로 구성된다. OMA중에서 객체간의 통신 메커니즘을 제공하는 컴포넌트로서 위치가 정해져 있는 ORB의 규격이 CORBA이다. ORB의 기능을 요약하면, 클라이언트가 발행하는 요구를 목적지의 객체로 보내 그 응답을 반환하는 것이다. 이것을 어플리케이션으로부터 이용가능하게 하기 위해서 CORBA에서는 다음의 사양을 설정하고 있다.

- 객체인 인터페이스를 공통으로 기술하기 위한 인터페이스 정의 언어(IDL)
- C언어 등의 프로그램 언어로 어플리케이션을 기술하기 위한 언어 매핑
- 객체의 생성, 요구의 발송 등 각종 서비스를 제공하는 컴포넌트 기능과 인터페이스
- 객체간의 통신을 위한 프로토콜

CORBA객체는 CORBA IDL로 기술된 인터페이스를 통해서 상호작용 한다. CORBA IDL로 기술된 객체의 인터페이스는 IDL 컴파일러를 통해서 객체 통신을 위한 클라이언트 스텝 코드와 구현 스텝 코드 코드가 생성된다. 이 코드들은 통신을 위해 필요한 요소들을 포함하며, 사용자는 생성된 코드들을 이용해서 응용을 구현할 수 있다. 이와 같이, 설계와 구현을 분리함으로써 사용자는 네트워크와 구현 언어에 대해서 고려하지 않아도 된다.

#### 2. Object Web

인터넷을 대중화하는데 있어 중심적인 역할을 한 것이 웹이며, 이는 사용자에게 매우 쉽고, 일관된 인터페이스를 제공한다. 정보의 분산 여부에 무관하게 웹은 모든 종류의 정보의 다목적 저장소의 역할을 한다. 또한 웹 브라우저는 이미 거의 모든 플랫폼에서 작동한다. 이는 거의 모든 플랫폼에서 망관리 시스템에 접근하는 것이 가능하다는 것을 의미하며, 망 자원을 공통적이며 일관된 방법으로 관리하는 것을 가능하게 한다. 웹 기반의 망관리는 WBEM(Web-Based Enterprise Management)[10]과 JMAPI(Java Management API)[11]에서 연구가 활발히 진행중이다.

한편, 웹에서 CORBA와 자바에서 제공하는 다양한 분산서비스들과 객체지향 패러다임을 이용하기 위해 시스템 변형 없이, 웹에 CORBA를 통합한 환경을 Object Web이라 한다[12]. Object Web은 하나의 독자적인 시스템이 아니라 현재 분산 개발 환경을 선도하고 있는 기술들을 통합하여 상승효과를 얻는 방법이다[13].

#### 3. Object Web을 이용한 OSI 관리구조

망관리 시스템은 망 자원의 종류나 관리 방법 등에 무관하게 망 자원을 관리자에게 관리 가능하도록 보여주어야 한다. 따라서 어떤 자원이 어떤 프로토콜을 이용하여 관리되는지에 무관하게 어떤 자원으로 볼 수 있어야 하며 그 수단이 되는 프로토콜은 관리자에게 숨겨져야 한다. 따라서, 관리된다는 측면에서의 망 자원을 추상화시킬 수 있다면 복잡한 망을 보다 쉽게 관리할 수 있는 방법을 제공해 줄 수 있다[14].

OSI 시스템 관리는 TMN과 같은 환경에서 망이나 망 요소를 관리하기에 적합하지만, 그 자체로는 분산되고 이질적인 환경에서 서비스관리를 하는데는 문제가 있다. 반면, CORBA는 분산 객체 환경을 제공해 주는 기능 때문에 전기통신망 서비스 관리에 적합한 차세대 기술로 여겨지고 있다.

그림 2.는 Object Web을 이용한 OSI 망관리 시스템의 구조를 보여주고 있다..

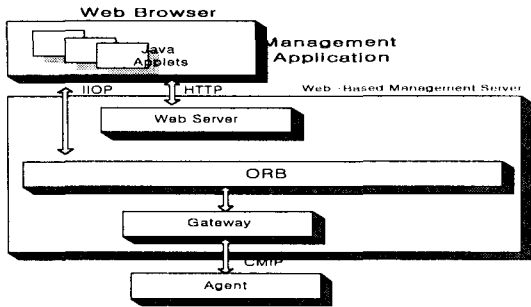


그림 2. Object Web을 이용한 OSI 망관리 시스템 구조  
 Fig. 2. Architecture of OSI network management using object web

이 시스템은 관리응용과 웹 기반 관리 서버, 대리자로 구성되어 있다. 관리응용은 자바 인터프리터와 ORB를 탑재한 웹 브라우저에서 실행되어진다. 관리응용은 자바로 구현된 CORBA 클라이언트로, 웹 서버로부터 다운로드되어 실행되며 망관리를 위해 게이트웨이에 바인딩 된다. 웹 기반 관리 서버는 웹 서버, 게이트웨이로 구성된다. 웹 서버는 HTML 문서와 관리자 기능을 갖는 자바 애플릿 등을 웹 브라우저에게 제공한다. 게이트웨이는 SNMP나 CMIP과 같은 망관리 프로토콜과 CORBA 기술을 이용하는 관리자 사이의 상호응용에 사용되는 메소드를 제공한다. 또한 프로토콜에 의존적인 관리정보 구조와 상

호작용을 위한 기능들을 CORBA 인터페이스로의 변환, 역변환 역할도 수행하게 된다. 망관리 대리자는 MIB을 처리하여 망을 관리하며, CORBA와의 변환을 위해 게이트웨이가 필요하다.

#### IV. 관리자과 게이트웨이 설계

##### 1. 설계 구조

본 논문에서 설계한 망관리 구조는 그림 3과 같으며 크게 관리 영역, 피 관리 시스템 영역으로 구분된다. 사용자 인터페이스로는 Object Web을 사용할 수 있는 웹 브라우저를 이용하며, 다수의 웹 브라우저를 통하여 하나의 피 관리 시스템에 접근하는 것이 가능하다.

피 관리 시스템 영역은 크게 CORBA/CMIS 게이트웨이, 클라이언트와 이벤트 처리기, 그리고 OSI 대리자로 구성되어진다. CORBA/CMIS 게이트웨이는 CORBA 서버의 역할을 수행하며, 대리자와는 CMIP 프로토콜을 사용하여 통신한다. 본 논문에서의 CORBA/CMIS 게이트웨이는 OSI CMIS 서비스들을 CORBA IDL 객체로 매핑하는 방식을 사용하였다. 클라이언트는 CORBA 클라이언트로서 웹 브라우저로 다운로드 되면 IIOIP를 통하여 관리자 기능을 수행하게된다. 이벤트 처리기는 OSI 대리자로부터 발생하는 이벤트를 감지하여 CORBA 클라이언

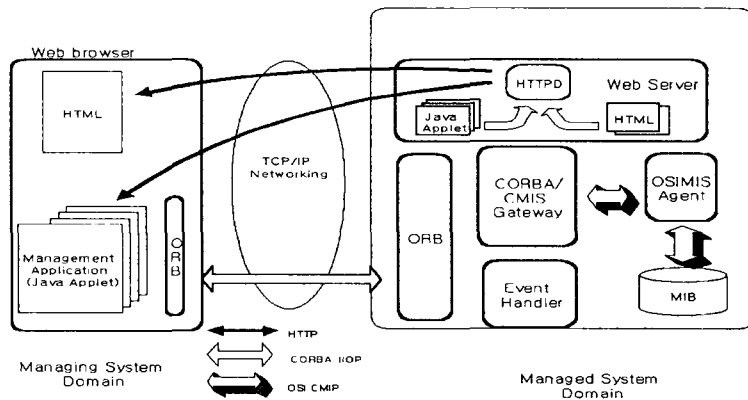


그림 3. Object Web을 이용한 CORBA기반 OSI 망관리의 설계구조  
 Fig. 3. Design Architecture of OSI network management using object web

트로 이벤트를 통보한다.

그림 4는 관리 시스템과 피 관리 시스템 간의 통신 절차를 나타낸다.

- ① 망 관리자는 웹 브라우저를 이용하여 웹 서버로 접속을 시도한다. 이때 웹 브라우저로부터의 메시지는 HTTP를 통하여 URL의 형태로 웹 서버로 전달된다.
- ② 웹 서버는 웹 브라우저로부터 수신한 URL을 분석하여 망 관리자가 요청한 HTML 문서를 웹 브라우저로 보낸다. 이 경우, HTML 페이지는 삽입된 자바 애플릿에 대한 참조를 포함한다.
- ③ 웹 브라우저는 HTML 문서내의 애플릿 태그를 참조하여 웹 서버로부터 자바 애플릿을 검색한다. 웹 서버에 의해 검색된 애플릿은 바이트 코드 형태로 스텝과 함께 웹 브라우저로 다운로드 된다.
- ④ 웹 브라우저로 다운로드된 애플릿은 자바의 런타임 보안 점검 모듈을 통해 실행된 후 메모리로 로드되어 CORBA 서버 객체를 호출한다. 이때, 대리자와 관리자간의 변환을 담당하는 게이트웨이와 바인딩 한다.
- ⑤ 망 관리자는 바인딩된 게이트웨이에게 망관리 요청을 전달한다. 망관리 요청을 받은 게이트웨이는 대리자를 호출하며, 대리자로부터 수신한 응답은 관리자로 반환한다.

## 2. 대리자와 게이트웨이간의 인터페이스

본 절에서는 OSI의 망관리 요청 종류와 CORBA/MAN게이트웨이를 설계하는데 필요한 CMIS 서비스 인수 구조에 따른 CORBA IDL 인수 구조 및 이들 간의 매핑구조를 자세히 설명하였다.

### 1) 망관리 요청의 종류

망관리 요청은 관리자가 대리자에게 요청하는 것과 대리자가 관리자에게 요청하는 것으로 크게 구분할 수 있다. 관리자가 대리자에게 요청하는 경우는

- ① 대리자 내에 존재하는 관리객체의 값을 읽을 때
- ② 대리자 내에 존재하는 관리객체의 값을 변경할 때
- ③ 대리자 내에 존재하는 관리객체에 특정한 행동을 취할 때
- ④ 대리자 내에 관리 객체를 새로 생성할 때
- ⑤ 대리자 내에 존재하는 관리객체를 삭제할 때 등이 있다.

대리자가 관리자에게 요청하는 경우는

- ⑥ 대리자에게서 발생한 사건을 관리자에게 알리 고자 할 때이다.

표 1에서 CMIP 프로토콜이 앞의 여섯 가지 망관리 요청들을 제공하기 위한 CMIS 서비스를 보여주고 있다[15].

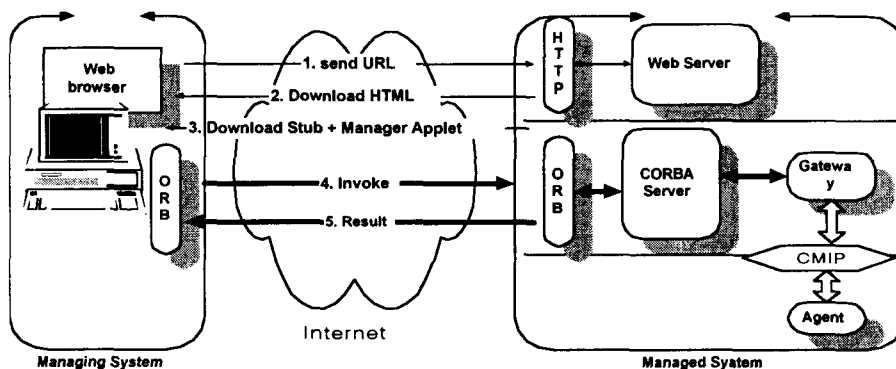


그림 4. object web을 이용한 OSI 망관리 시스템의 통신절차

Fig. 4. Communication step of OSI network management system using object web

표 1. CMIS 서비스의 종류  
Table 1. Kind of CMIS service

CMIS Service	
①	M-GET
②	M-SET
③	M-ACTION
④	M-CREATE
⑤	M-DELETE
⑥	M-EVENT-REPORT

2) CMIS 서비스 인수와 IDL 인수 구조

(1) CMIS 서비스 인수

표 2의 CMIS 서비스 인수는 앞 절에서 기술한 망관리 요청 기능을 수행하는 CMIS 서비스들 중에서 CMIS GET 서비스와 서비스에 대응되는 CORBA IDL 타입을 기술하였다.

표 2의 첫 번째 부분은 서비스 파라미터이며 두 번째 부분은 서비스 요청 파라미터들의 조건과 이들 요청 파라미터들이 지시 프리미티브에 어떠한 조건으로 매칭 되는지를 나타낸다. 세 번째 부분은 요청 서비스에 대한 응답 파라미터들의 종류와 조건들을 나타낸다. 본 논문에서는 이러한 서비스 파라미터들의 특성들을 기반으로 하여 서로 관련이 깊은 파라미터들을 모아서 하나의 구조체로 사용하였다. 표 2에서 보는바와 같이 CMIS\_GETREQ라는 CORBA IDL 구조체 안에 Base-object class, Base-object instance, Scope, Filter, .. 등과 같이 대리자에게로 전달되어야 할 파라미터들을 각각을 구조체 필드로 정의하였다.

(2) CMIS 서비스 인수 구조에 따른 게이트웨이의 IDL 인수 구조

본 논문에서 게이트웨이의 모든 망관리 서비스 메소드는 CMIS 서비스 종류와 파라미터에 따라 정의하였다. CORBA IDL로의 매핑은 표 2에서 본 바와 같이 CMIS 서비스 파라미터 구조에 따라 적절한 CORBA IDL로 정의하였고, 정의된 CORBA IDL의 구조체 일부분을 아래에 기술하였다.

표 2. CMIS GET 서비스 파라미터와 대응되는 IDL type name

Table 2. CMIS GET service parameters and vs. IDL type name

Parameter Name	Request/Indication	Response/Confirm	IDL type name
Invoke identifier	M	M	RequestID
Linked identifier	-	C	CMIS_GET_LR_LIST
Base-object class	M	-	CMIS_GETREQ
Base-object instance	M	-	
Scope	U	-	
Filter	U	-	
Access control	U	-	
Synchronization	U	-	
Attribute-identifier list	U	-	
Managed-object class	-	C	CMIS_GETRES
Managed-object instance	-	C	
Current time	-	U	
Attribute list	-	C	
Errors	-	C	CMIS_ERROR
			CMIS_ERRORInfo

```

.....
// GET Request Argument
struct CMIS_GETREQ {
    CMIS_MID          GET_req_class;
    CMIS_MN           GET_req_instance;
    CMIS_EXTER        GET_req_accessControl;
    CMIS_SYNC         GET_req_sync;
    CMIS_SCOPE        GET_req_scope;
    CMIS_FILTER        GET_req_filter;
    ASN1_Integer      GET_req_nattrs;
    CMIS_MID_LIST     GET_req_attrs;
};
.....

```

위의 구조체는 GET 메소드에 사용되는 CORBA IDL 파라미터 구조체이며 GET 메소드의 요구 인수를 보여주고 있다.

### 3) 인터페이스의 정의

게이트웨이 인터페이스는 대리자와 게이트웨이간에 존재하는 인터페이스이며 앞 절에서 기술한 망 관리 요청을 수행하는 메소드와 관련 파라미터를 정의하여야 한다. 아래에 그에 따른 대리자와 게이트웨이간의 인터페이스를 IDL로 정의한 것 중에 일부분을 기술하였다.

```

.....
interface CMISService {
short CMIS Get (in RequestID      GTrequestID,
                in CMIS_GETREQ    GTgetReq,
                out CMIS_GET_LR_LIST GTlinkedList,
                out CMIS_GETRES    GTgetRes,
                out CMIS_ERROR      GTerrors,
                out CMIS_ERRORInfo  GTerrorInfo);
.....
}
    
```

각 메소드의 매개변수는 서비스 메소드마다 파라미터들이 다르며, 입출력 방향은 관리자에서 게이트웨이로 파라미터를 보낼 때에는 in으로, 게이트웨이에서 관리자로 보내는 것은 out으로 되어있다.

## V. 관리 시스템의 구현

### 1. 구현 환경

본 논문의 구현은 SOLARIS 2.5.1 운영체제하에서, OSI 대리자는 UCL(University College of London)에서 개발된 OSI 망 관리 플랫폼인 OSIMIS(OSI Management Information Service)를 이용하여 구현하였다. 그리고 하부 통신은 ISODE(ISO Development Environment)를 이용하였으며 CORBA 플랫폼으로는 Visigenic[16]의 Visibroker for C++ and Java 3.3을 이용하였다. 자바의 애플릿을 구현하기 위해 JDK 1.1.6을 사용하였고 웹 서버는 Netscape Enterprise를 사용하였다.

### 2. 관리자의 구현

본 논문에서 구현한 관리자는 웹 브라우저에서 실행이 되어져야하기 때문에 게이트웨이 인터페이

스 정의를 Visibroker의 idl2java 컴파일러로 컴파일하여 클라이언트 스텝 코드를 생성하였으며 이 소스코드를 이용하여 관리자 애플릿을 작성하였다. 관리자 애플릿은 자바로 작성되었으며 게이트웨이와의 통신을 위해 CORBA 2.0 규격에 따라 관리코드를 작성하였다. 따라서 구현된 관리자 애플릿은 CORBA 규격과 자바를 사용하여 웹 브라우저에서 실행되는 Object Web방식으로 구현함으로써 관리자에게 웹 브라우저 위에서 관리자 애플릿이 실행하여 관리정보를 주고받도록 하였다. 관리자 애플릿은 관리자에게 대리자를 관리할 수 있는 인터페이스를 제공하며, 구현된 게이트웨이에 의해 지원되는 인터페이스를 통하여 대리자로 관리정보를 보낼 수 있으며 또한 관리자로부터 자료를 입력받을 수도 있다. 이때 사용되는 메소드는 Get, Set, Action, Create, Delete 중에 하나이며, 웹 브라우저를 이용하여 OSI 객체 및 그 속성들에 대하여 관리할 수 있게 하였다.

아래에 스텝을 기초로 구현된 관리자 애플릿의 일부를 기술하였다.

```

.....
// Variable creation and instance
int GTrequestID=0;
CMIS_GETREQ GTgetReq = new CMIS_GETREQ();
.....
// Assignment to Variable
GTrequestID = GTrequestID + 1;
// Request
Getstatus = CMISGET(GTrequest, GTgetReq,
                    GTlinkedList, GTgetres, Gerrors,
                    GTerrorInfo);
.....
    
```

### 3. 게이트웨이 구현

CORBA에서 CMIP/CMIS를 통하여 OSI시스템과 상호작용하기 위해서는 관련 객체 모델들 사이의 매핑과 도메인간에 프로토콜 변환을 처리하는 메커니즘이 제공되어야 한다. 이러한 상호작용변환을 담당하는 개체를 게이트웨이라고 하며, 사용되는 망관리 프로토콜에 따라 다수의 게이트웨이가 존재할



수 있다. 본 논문에서는 OSI 대리자와 관련 있는 게이트웨이를 CORBA/CMIS 게이트웨이라고 정의하였고 CMIS-level 인터페이스를 기반으로 하는 게이트웨이 모델을 사용하여 OSI기반의 대리자와 CORBA기반의 관리자 사이의 변환을 담당하도록 구현하였으며, 망관리 프로토콜에 의존적인 대리자를 관리자 측면에서 CORBA 대리자로 볼 수 있도록 추상화하였다. 구현된 CORBA/CMIS 게이트웨이는 게이트웨이 인터페이스를 정의한 IDL을 컴파일했을 때 생성되는 스킴레톤을 기초로 구현되었으며 구현 결과 중의 일부를 아래에 기술하였다.

```

.....
// Convert the string Argument to CMIS parameter
// Object Class
if(GTgetReq.GET_req_class.MID_mid_type      =
GlobalForm) (
    Obj_class.Mid_type = MID_GLOBAL;
    strObjClass      = CORBA::String_dup(GTgetReq.
        GET_req_class.mid_mid_global);
    Obj_class.mid_global = name2oid(strObjClass);
}
else
.....
// Request Identification
requestID = GTrequestID;
// Perform the CMIS M-GET request
mi = &mis;
status = M-GET(msd, requestID, &Obj_class,
    objInst, scope, filter, NULLMACCESS,
    Sync, nattrs, mi);
request_Invoke = mi->mi_invoke;
.....

```

CORBA/CMIS 게이트웨이에는 Get, Set, Action, Create, Delete, Event 등 CMIS 서비스의 메소드가 구현되어져있다. 즉, Get 메소드의 구현에서는 CMIS의 M-GET 서비스를 호출하며 이러한 서비스를 호출할 때 그 매개변수가 되는 것들은 ISODE와 OSIMIS에서 제공되는 정보모델 변환 함수를 사용하여 CORBA IDL 타입을 적절한 정보모델로 변환하였다.

본 논문에서 구현한 CORBA/CMIS 게이트웨이는 CORBA 관리자에 대한 관리 대리자로 동작하도록 하였다. 즉, 관리자 애플릿으로부터 관리서비스를 요청 받아 적절한 CMIS API를 통하여 OSI 대리자에게 동작 요청을 전달하고 OSI 대리자는 게이트웨이를 통해 전달된 관리자의 오퍼레이션 요청을 수행한 다음, 다시 CMIS API를 통해 그 결과를 IOP를 이용하여 관리자 애플릿에 전달한다. 이때 게이트웨이는 클라이언트 역할을 하게되며 관리자 애플릿 위에서 적절한 동작을 호출하여 그 응답을 관리자 애플릿에 보내게된다.

#### 4. 시험 및 검증

본 논문에서 구현된 시스템에서 OSI 대리자는 피 관리 시스템에서 실행중이며, OSIMIS에서 제공해주는 대리자를 이용하였다. 이 대리자에 의해 유지되는 MIB은 OSIMIS에서 기본적으로 제공되는 Unix-MIB를 이용하였다. 구현된 시스템은 관리자 애플릿과 CORBA/CMIS 게이트웨이이며, 게이트웨이는 피 관리 시스템에서 실행된다. 망의 관리자는 어디에서든지 웹 브라우저를 이용하여 게이트웨이에 접근함으로써 대리자들의 정보를 읽어들이거나 변경할 수 있다.

##### 1) 기능 시험

그림 5는 관리자가 PC 또는 터미널에서 웹 브라우저를 이용하여 웹 서버로부터 관리자 애플릿을 가져온 결과를 보여주고 있다. 다운로드된 관리자 애플릿은 웹 브라우저로부터 인증을 받아 자바 인터프리터 또는 자바 가상 머신에 의해 실행이 되어진다. 그림 5는 관리자 애플릿이 게이트웨이에 바인딩 한 후, Get 메소드를 실행하여 게이트웨이에서 가져올 수 있는 정보를 IOP를 통해 Object Web 방식으로 가져온 결과를 보여 주고 있다. 그림 5의 왼쪽 창의 자료는 OSI 대리자에서 인스턴스화된 객체를 보여주고 있으며 uxObj1은 OSIMIS에서 제공하는 객체이다. 오른쪽 창은 왼쪽 창에서 선택된 객체의 속성을 보여주고 있으며 이들은 시스템 관리 서비스기 가져올 수 있는 정보를 IOP를 통하여 Object Web방식의 관리자에게 전달하게될 것이다.

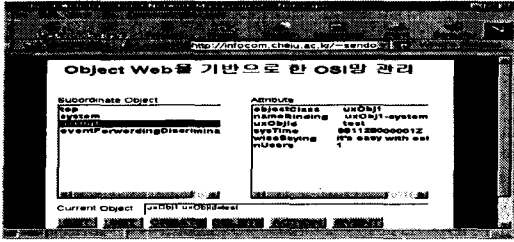


그림 6. OSI 대리자 내의 관리객체에 대한 Get 요구의 결과  
 Fig. 6. Result of get requesting the value of managed object in OSI agent

## VI. 결 론

대규모 망 관리에서는 이질적인 통신망 자원, 다양한 통신망 하부 구조 그리고 각 시스템의 언어에 관계없이 효율적으로 망을 운용·관리할 수 있는 기술이 필요하다.

Object Web은 웹, 자바 그리고 CORBA를 통합하여 제공한다. 이는 망관리에서 CORBA에서 제공되는 다양한 서비스 외에도 기존의 웹에서 제공하고 있는 서비스들을 함께 지원한다는 것이며, 특히 자바는 운영체제에 독립적이고, 모빌코드 속성을 제공함으로써 어플리케이션 개발에 많은 유연성을 제공해 준다. 따라서 본 논문에서는 효율적인 망관리를 위해 Object Web을 도입하여 웹 방식의 사용자 인터페이스를 갖는 동적인 관리자를 구현하였다. 또한 CORBA를 하부 구조로 하는 CORBA/CMIS 게이트웨이 구조를 제시하였으며, 각각의 요소들이 정상적으로 동작하는지 확인하였다.

본 논문에서는 OSI 관리 환경만을 고려하였지만, 다양한 망관리 환경과의 통합을 지원하기 위한 더 많은 고찰이 필요하다.

향후 연구과제로 망에 존재하는 여러 게이트웨이들 간의 통합 연동을 위한 CORBA 연합 트레이더 연구가 진행될 필요가 있다.

## 참고문헌

[1] 조영현, 김영명, 석승학, "TMN을 향한 첫걸음", 하이테크정보, pp. 11-28, 1997.

[2] A. Tang, S. Scoggins, "OPEN NETWORKING WITH OSI", PTR Prentice Hall, pp.17-33, 1994.

[3] W. Stallings, "SNMP, SNMPv2, and CMIP: The Practical Guide to Network-Management Standards", ADDISON-WESLEY, pp. 65-77, 1993.

[4] OMG, <http://www.omg.org/>, 1998.

[5] OMG, "The Common Object Request Broker: Architecture and Specification", OMG Document, Version 2, July 1995.

[6] NMF, <http://www.nmf.org/>, 1998.

[7] NMF, "OMNIPoint Integration Architectute - Delivering a Management System Framework to Enable Service Management Solutions", Technical report, August 1994.

[8] OMG Telecom Task force, "CORBA-based Telecommunication Network Management.", OMG White Paper, May 1996.

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

[10] WBEM Consortium, WBEM homepage, <http://wbem.freerange.com/>, July 1996.

[11] Sun Microsystems Inc., "Java Management API Architecture", June 1996.

[12] R. Orfali, D. Harkey, "Client/Service Programming with JAVA and CORBA", Sccond Edition, John Wiley & Sons Inc., 1998.

[13] 왕창중, 이세훈, "CORBA 프로그래밍", 도서출판 대림, pp. 121-142, 1998.

[14] 손경찬, "CORBA와 WWW을 이용한 통합 망 관리 시스템의 설계 및 구현", 경북대학교 전자공학과 정보통신공학 석사논문, pp. 10-11, 1996.

[15] M. Sloman, "Network and Distributed System Management", ADDISON-WESLEY, pp. 197-225, 1996.

[16] Visigenic, <http://www.inprise.com>, 1998.

김 강 석(Gang-Suk Kim)

송 왕 철(Wang Cheol Song)

1989년 2월 연세대학교 전자공학과 공학사

1991년 2월 연세대학교 전자공학과 공학석사

1995년 8월 연세대학교 전자공학과 공학박사

1996년 2월~현재 제주대학교 정보공학과

\* 주관심분야 : TMN, 분산망관리, CORBA, mobile agent 등