

Object Web 基盤의 OSI 망 管理

김 강 석* · 송 왕 철**

OSI Network Management based on the Object Web

Gang-Suk Kim* and Wang-Cheol Song**

Abstract

The Object Web translates the existing CORBA IDL into the Java language and uses IIOP for interaction between a client and a server under the exiting Web environment. The client and the server through the Object Web, do not use HTTP, a web protocol, but IIOP that is an inter-object communication protocol provided by the CORBA. This paper proposes an architecture which consists of a gateway, a CORBA manager and an OSI agent. The 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 service and vice versa. Hence, an OSI agent can be abstracted into a CORBA agent. In this architecture, any client with the Web browser can manage OSI MOs in distributed environment. The client can invoke the management operations through CORBA server after it download the Java applet from HTTP server.

Key words : Object Web, OSI, Network management

1. 서 론

정보 통신 기술의 급속한 발달과 함께 통신 서비스 사용자들에게 보다 신속하고 다양한 서비스를 제공하기 위해 통신망 환경은 비대화·복잡화되어 가고 있고, 다양하고 이질적인 통신망 구성요소(NE : Network Element)가 통신망 내에 공존하게 되었으며, 망 내에서 서로 다른 영역을 관리하는 운용 시스

템(OS: Operation System)들간의 연동기능이 미비하여 전체 운용 시스템의 효율성이 점차 저하되고 있다. 한편, 통신망의 거대화화 및 네트워크 기술의 발전으로 인하여 대형 서버중심의 클라이언트/서버 컴퓨팅 환경에서 벗어나 개방형 통신망에 접속된 다양한 플랫폼을 이용한 분산 컴퓨팅 환경으로 바뀌어가고 있으며, 분산객체(distributed object) 통합 기술에 관한 연구¹⁾가 활발하게 이루어지고 있다. 분산 시스템 분야에서는 최근 소프트웨어와 하드웨어의 이질성을 해결하기 위하여 미들웨어에 대한 연구가 OMG(Object Management Group)를 중심으로 진행되고 있으며, OMG에서는 객체 지향 기술을 바탕으로 분

* 제주대학교 대학원
Graduate School, Cheju Nat'l Univ.

** 제주대학교 정보공학과
Dept. of Information Eng., Cheju Nat'l Univ.

산 객체 시스템의 표준으로 CORBA(Common Object Request Broker Architecture)를 제안하고 있다.²⁾ 이러한 CORBA를 이용함으로써 분산 객체를 통합하는데 있어서 기존에 개발되고 운용되던 리가시 시스템들을 통합하여 단일 인터페이스를 제공하고, 각 리가시 시스템들을 담당하는 객체의 위치나 구현 내용에 상관없이 클라이언트는 이들 객체들이 제공하는 서비스를 이용할 수 있으므로 위치, 구현의 투명성이 보장된다. 또한 새로운 객체와의 연결이 쉽고, 기존 리가시 시스템의 변경이 일어나는 경우에 이를 담당하는 CORBA객체에만 변경이 일어나므로 클라이언트는 여기에 상관할 필요가 없어 확장성과 유연성이 보장된다.

분산 컴퓨팅 환경에서의 효율적인 망 관리를 위해 TMN(Telecommunication Management Network)³⁾의 관리구조로 OMG/CORBA, ISO/ODP에서 사용하고 있는 분산객체기반 구조가 채택될 것으로 예상된다. 이에 따라, NMF(Network Management Forum)의 OMNIPoint에서는 기존의 ITU-T/TMN, ISO/OSI, OMG/CORBA, OSF/DME 등에서 제시된 다양한 관리구조의 통합에 대한 연구를 진행하고 있고, OMG에서도 CORBA를 이용한 망 관리 구조를 연구하고 있으며⁴⁾, X/Open과 NMF의 JIDM(Joint Inter Domain Management) 태스크포스는 OMG/CORBA, ISO/OSI, IETF/SNMP간의 관리를 위해 정보모델과 프로토콜의 변환에 대한 연구를 진행 중이다.⁵⁾

한편, 웹 브라우저는 거의 모든 플랫폼 상에서 동작하며 일관된 사용자 인터페이스를 가진다. 따라서 망 관리 시스템의 사용자 인터페이스로서 웹 브라우저를 사용하면 다양한 장점을 얻을 수 있다. 웹 브라우저에서 자바와 IIOP(Internet Inter-ORB Protocol)를 이용함으로써 운영체제에 독립적이며, 모빌 코드 속성을 제공함으로써 어플리케이션 개발에 많은 유연성을 제공해줄 뿐만 아니라 IIOP를 통하여 직접 CORBA 객체와 통신할 수 있다. 즉 기존의 CORBA의 IDL 언어를 자바 언어로 변환하고 기존의 웹 환경에서 클라이언트와 서버 사이의 통신은 IIOP를 사용하는 것으로 Object Web이라 한다.⁶⁾ 따라서 Object Web의 클라이언트와 서버는 웹 상에서 웹의 프로토콜인 HTTP를 사용하지 않고 CORBA가

제공하는 객체간 통신 프로토콜인 IIOP를 통해서 작동한다.

본 논문에서는 Object Web을 기반으로 한 관리자와 관리자가 OSI 도메인내의 관리 객체들을 일관성 있게 관리할 수 있는 게이트웨이 구조를 제시하였으며, 그 기반기술로 CORBA를 이용하였다. CORBA/CMIP 게이트웨이는 기존의 대리자를 수정하지 않은 채 CORBA 대리자로 보여지도록 하였다. 이는 TMN의 관리구조로 CORBA 기반의 구조가 사용될 때 게이트웨이를 이용하여 기존의 자원을 CORBA 대리자로 동작하도록 함으로써 현재 사용중인 자원 또한 CORBA를 이용하여 관리가 가능하다는 장점이 있다.

II. 망 관리 개념과 Object Web

2.1. 망 관리의 개요

망 관리의 목적은 통신망을 효율적으로 운용하고 망이 제대로 동작하고 있는가를 보장해 주는데 있다. 이를 위해서 망 관리 시스템은 망을 구성하는 다양한 장비로부터 망의 사용에 관한 정보를 수집하여 운용자에게 보고한다. 망을 관리할 때의 대상은 사설교환기(PBX), 컴퓨터, 모뎀, 동축케이블, 광섬유 등과 같은 물리적인 망의 자원뿐만 아니라 대역폭(bandwidth), 서비스 품질(Quality of Service), 통신 프로토콜, 응용 소프트웨어 등의 논리적인 자원도 포함한다.

Fig. 1은 망을 관리할 때 관리자(manager)와 대리자(agent), 관리객체(managed object)간의 상호작용을 보여 준다. Fig. 1에서 관리자와 대리자간에는 망 관리 프로토콜을 사용하여 망 관리 정보를 주고받는다. 관리객체는 망에서 감시의 대상이 되는 자원들을 객체로 모델링한 것인데, 관리객체의 속성(attribute)이 자원의 상태를 반영하고 있고 특정한 사건이 있을 경우 사건에 대한 통지를 대리자에게 보낸다. 대리자는 관리자로부터의 요청에 대하여 관리객체의 값을 읽어 반환하고, 관리객체에서 발생한 사건을 수신하여 관리자로 전달해 준다. 관리자를 사용하여 운용자는 망을 운용하고, 관리객체에서 발생한 사건을 관리자가 수신하면 이를 운용자에게 알려 적절한 조치를

취할 수 있도록 한다.

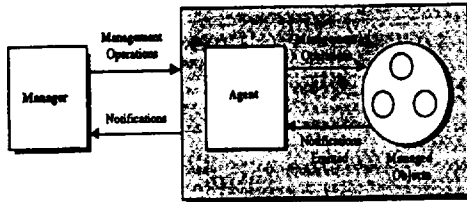


Fig. 1 Interaction between manager and agent and object

2.2. OSI 및 CORBA의 통합방법

현재까지는 TMN의 관리구조를 OSI 관리모델을 기반으로 모델링 하였으나 통신망이 커질수록 분산 객체 기반 구조를 적용하도록 하는 것이 필요하다. CORBA는 현재 거의 업계표준으로 인정되는 추세이며, NMF와 X/Open은 공동으로 JIDM 태스크포스를 조직하여 CMIP/ SNMP/CORBA 도메인간의 관리에 대한 연구를 진행중이다.

CORBA와 TMN의 통합을 위한 방법 중 대표적인 것으로 GDMO-to-IDL, Value-Added, Abstract Object mapping 등이 있다.⁷⁾

2.2.1. GDMO-to-IDL 게이트웨이

이 방법은 CORBA 환경과 OSI 망 관리 환경 사이의 게이트웨이를 만드는 것이다. 즉, GDMO 객체가 정의되고 CMIP 프로토콜이 사용되어지는 OSI 환경 하에서의 대리자를 만들고, CORBA IDL을 기초로 하는 객체가 정의되고 통신메커니즘으로 ORB가 사용되는 CORBA 환경에서의 관리자를 만들어, 그 사이를 이어주는 게이트웨이를 제공해 주는 것이다. 이를 위해서는 다음 두 level사이에서 작업이 이루어져야 한다.

- object definition level에서 각 GDMO객체는 IDL에서의 매핑을 필요로 한다. 이 매핑은 정의된 객체에서 수행될 오퍼레이션, 통지, 그리고, 가능한 객체 자체의 행위들을 포함 한다.

- Communication level에서 ORB에서 제공된 프로토콜을 사용하는 관리자에서 보내진 메시지나 요구들은 대리자에게 보내지기 전에 CMIP으로 변환될

필요가 있다. 또한, CMIP을 사용하는 대리자에 의해서 보내지는 응답이나 메시지는 ORB가 지원하는 프로토콜로 변환되어진다.

2.2.2. Value-Added 게이트웨이

이 방법은 GDMO 객체들과 CORBA IDL 객체들 사이의 일대일 매핑을 지원한다. 그러나, GDMO 객체 오퍼레이션들과 IDL 객체 메소드들 사이에는 일대일 매핑을 갖지 않기 때문에, 각 IDL 객체를 위해서 서로 다른 메소드들이 정의되어야 한다. 그래서, 사용자 정의 메소드를 허용하여 extra value를 덧붙일 수 있도록 하는 것이다.

2.2.3. Abstract Object mapping

이 방법은 망 관리 도구로서의 CMIB과 서비스 관리 도구로서의 CORBA를 통합하는 방식이다. 따라서 적당한 부분의 관리 계층에서 CORBA 객체의 한 집합을 정의한다. GDMO 객체들의 집합은 단일 CORBA 객체로 매핑이 되며, CORBA 객체와 GDMO 객체 사이에는 관계가 정의되어 나중의 참조를 위해서 기록된다.

2.3. 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 망 관리 시스템에 접근하는 것이 가능하다는 것을 의미하며, 망 자원을 공통적이며 일관된 방법으로 관리하는 것을 가능하게 한다.

최근 들어 인터넷 기술분야에서 급속도로 확산되고 있는 분야 중 하나가 인트라넷이란 분야이다. 이 작은 정보 고속도로는 외부와의 연결에도 가장 안전한 연결방법을 제공하며 기존의 인터넷에서 개발된 정보 시스템을 그대로 이용할 수 있게 해준다. 예를 들어, 특정 기관에서 TCP/IP를 응용한 인트라넷을 구성한다면 기존에 인터넷에서 사용했던 전자우편이나 유즈넷 뉴스, 웹 기술 등을 재사용 할 수 있다. 이를 통해 가장 저렴하고 효율적인 정보 네트워크를 구성할 수

있게 된다. 이 때 해당 인트라넷은 기존의 조직이나 기관에서 사용했던 응용 프로그램이나 시스템을 새로 구축하는 시스템과 통합해야만 한다. 이 문제를 해결하기 위한 방법으로 가장 적합한 것이 바로 CORBA를 이용하는 것이다.

Object Web이란 웹에서 CORBA와 자바에서 제공하는 다양한 분산서비스들과 객체지향 패러다임을 이용하기 위해 시스템 변형 없이, 웹에 CORBA를 통합한 환경이다. Object Web은 하나의 독자적인 시스템이 아니라 현재 분산 개발 환경을 선도하고 있는 기술들을 통합하여 상승효과를 얻는 방법이다.

웹과 CORBA를 상호 운용하는 방법으로는 먼저 웹과 CORBA사이에 상호 운용에 필요한 데이터 변환을 처리해주는 게이트웨이를 두는 방법과 웹 클라이언트나 웹 서버에서 직접 CORBA의 IIOP를 지원하는 방법이 있다. 또 다른 방법으로는 웹 브라우저에서 수행 가능한 애플릿을 작성할 수 있는 언어인 자바언어로 CORBA 자체를 구현하는 것이다.

Object Web을 이용하여 망 관리 시스템을 구성할 경우의 이점들은 다음과 같다.

- 사용자 인터페이스를 독립
- 플랫폼 독립성 제공
- 프로토콜 독립성 제공
- 자바를 이용한 이동 관리자와 이동 대리자 기능 제공

III. 관리자와 게이트웨이 설계

3.1. 설계 구조

시스템 관리는 크게 사용자 인터페이스, 관리 시스템 도메인, 피 관리 시스템 도메인으로 구분된다. 사용자 인터페이스로는 웹 브라우저를 이용하며, 다수의 웹 브라우저를 통하여 하나의 관리 시스템에 접근하는 것이 가능하다.

관리 시스템 도메인은 크게 클라이언트와 게이트웨이로 구성되며 클라이언트는 CORBA 클라이언트 역할을, 게이트웨이는 CORBA 서버의 역할을 수행한다. 피 관리 시스템 도메인은 CMIP 프로토콜을 사용하는 대리자와 실제 관리되는 물리적·논리적 자원들로 구성된다.

OSI 대리자는 기존에 사용되고 있던 시스템일 수 있으며 기존의 OSI 관리자를 사용할 수도 있다. 또한 CMIP외에도 SNMP 또는 전용의 망 관리 프로토콜도 사용될 수 있다.^(8,9)

본 논문에서 설계한 망 관리 시스템은 Fig. 2와 같으며 아래의 절차에 따라 동작한다.

1 단계 : 망의 운용자는 웹 브라우저를 이용하여 웹 서버로 메시지를 보낸다. 이때 웹 브라우저로부터의 메시지는 HTTP를 통하여 URL의 형태로 웹 서버로 전달된다.

2 단계 : 웹 서버는 웹 브라우저로부터 수신한

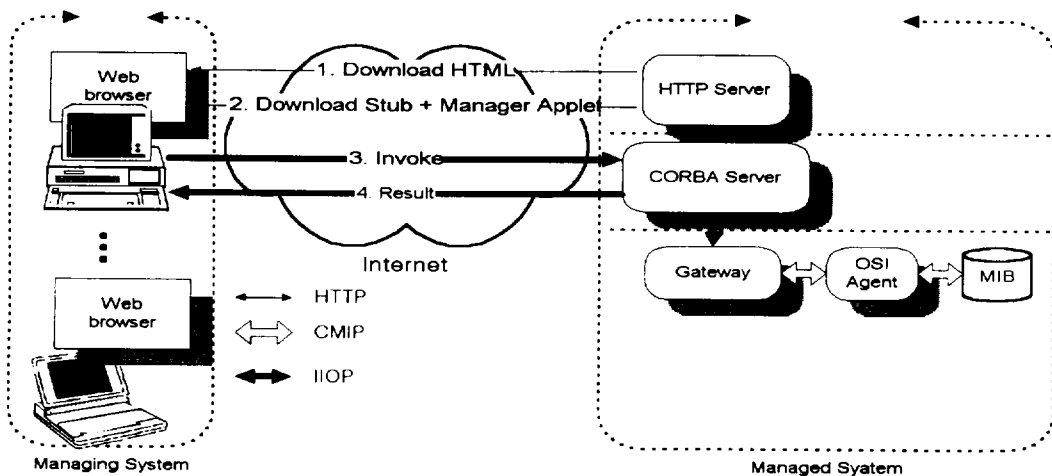


Fig. 2 Architecture of OSI network management system using object web

URL을 분석하여 망 운영자가 요청한 HTML 문서를 웹 브라우저로 보낸다. 만약 망 관리와 관련된 요청이 아니라면 전체 과정은 끝난다. 망 관리와 관련된 요청이라면 반환 받은 HTML 문서내의 자바에플릿이 실행되어 해당 관리자가 웹 브라우저로 다운로드된다.

3 단계 : 웹 브라우저로 다운로드된 관리자는 대리자와 관리자간의 변환을 담당하는 게이트웨이와 바인딩 한다.

4 단계 : 망 운용자는 바인딩된 게이트웨이에게 망 관리 요청을 전달한다. 망 관리 요청을 받은 게이트웨이는 대리자를 호출하며 대리자로부터 수신한 응답은 관리자로 반환한다. 망 운용자가 망 관리를 종료할 때까지 4 단계를 반복한다.

3.2. 관리자과 게이트웨이간의 인터페이스

3.2.1. 망 관리 요청의 종류

망 관리 요청은 관리자가 대리자에게 요청하는 것과 대리자가 관리자에게 요청하는 것으로 크게 구분할 수 있다.

관리자가 대리자에게 요청하는 경우는 (1) 대리자 내에 존재하는 관리객체의 값을 읽어올 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. 또 대리자가 관리자에게 요청하는 경우는 (6) 대리자에게서 발생한 사건을 관리자에게 알리고자 할 때이다.

Table 1 Kind of CMIP message

	CMIP
(1)	M-GET
(2)	M-SET
(3)	M-ACTION
(4)	M-CREATE
(5)	M-DELETE
(6)	M-EVENT-REPORT

현재 망 관리 프로토콜에서 표준으로 사용하고 있는 것은 CMIP과 SNMP가 있으며 Table 1에서

CMIP 프로토콜이 위의 여섯 가지 기능을 제공하기 위한 망 관리 메시지를 비교하였다.

3.2.2. 인터페이스의 정의

게이트웨이 인터페이스는 관리자와 게이트웨이간에 존재하는 인터페이스이며 앞 절에서 기술한 (1) ~ (6)의 기능을 수행하는 메소드를 정의하여야 한다. 아래에 관리자와 게이트웨이간의 인터페이스를 IDL로 기술하였다.

```
interface CMIService {
    short Get (in RequestID           GRequestID,
              in CMIS_GETREQ         GGetReq,
              out CMIS_GET_LR_LIST   GLinkedList,
              out CMIS_GETRES        GGetRes,
              out CMIS_ERROR         GTerrors,
              out CMIS_ERRORInfo     GErrorInfo):
```

IV. 관리 시스템의 구현

4.1. 구현 환경

본 논문의 구현을 위해 OSI 대리자는 OSIMIS (OSI Management Information Service)를 이용하여 구현되었다. OSIMIS는 유럽의 RACE 프로젝트의 일환으로 영국의 UCL(Univisity College of London)에서 개발된 망 관리 플랫폼이며 하부 통신은 ISODE (ISO Development Environment)를 이용하였다. CORBA 플랫폼으로는 Visigenic사의 Visibroker를 이용하였고 웹 서버는 Apache을 사용하였다.

4.2. 관리자의 구현

관리자는 웹 브라우저에서 실행이 되어져야하기 때문에 게이트웨이 인터페이스 IDL을 IDL-to-JAVA 컴파일러로 컴파일 하여 자바 클라이언트 코드를 생성한다.

자바 클라이언트 코드는 웹 브라우저로 다운로드 되어 게이트웨이와 바인딩을 시도하고 바인딩에 성공하면 자바 클라이언트 코드는 관리자로 동작을 하게 되고 게이트웨이의 메소드를 호출한다. 그 메소드는 Get, Set, Action, Create, Delete중의 하나이다.

4.3. 게이트웨이 구현

게이트웨이는 상호작용변환을 담당하는 개체이며 사용되는 망 관리 프로토콜에 따라 다수의 게이트웨이가 존재할 수 있다. 게이트웨이는 CMIP 대리자와 같은 망 관리 프로토콜에 의존적인 대리자를 관리자 측면에서 CORBA 대리자로 볼 수 있도록 해준다. 게이트웨이는 게이트웨이 인터페이스를 통하여 바인딩된 관리자로부터의 망 관리 요청을 적절히 변환하여 대리자를 호출한다. 또한 대리자로부터의 망 관리 정보에 대한 응답을 적절히 변환하여 관리자에게 반환한다. 본 논문에서는 OSI 대리자와 관련 있는 게이트웨이 객체를 CMIP 게이트웨이 객체라고 정의한다. 게이트웨이 객체는 게이트웨이 인터페이스를 정의한 IDL을 컴파일 했을 때 생성되는 골격(skeleton)을 기초로 구현되며 C++로 컴파일한 결과 중의 일부는 아래와 같다.

```
// 일부 생략
virtual CORBA::Short Get(
    RequestID GTrequestID,
    const CMIS_GETREQ& GTgetReq,
    CMIS_GET_LR_LIST_ptr & GTlinkedList,
    CMIS_GETRES_ptr & GTgetRes,
    CMIS_ERROR& GTerrors,
    CMIS_ERRORInfo_ptr & GTerrorInfo
) = 0;
```

4.4. CMIP 게이트웨이 객체

CMIP 게이트웨이 객체에서는 Get, Set, Action, Create, Delete의 모든 메소드가 구현되어야 한다. 예를 들어 Get 메소드의 구현에서는 CMIP의 M-GET서비스를 호출하며 이러한 서비스를 호출할 때 그 매개변수가 되는 것들은 ISODE와 OSIMIS에서 제공되는 정보모델 변환 함수를 사용하여 적절한 정보모델로 변환한다.

본 논문에서 구현한 CORBA/CMIP 게이트웨이는 CORBA 관리자에 대한 관리 대리자로 동작한다. 게이트웨이는 Object Web을 지원하는 관리자에 의해 발급된 관리 응용을 발급 받아 적절한 CMIP API를 통하여 OSI 대리자에게 동작 요청을 전달한다. OSI 대리자는 게이트웨이를 통해 전달된 관리자의 오퍼레

이션 요청을 수행 한 다음, 다시 CMIP API를 통해 그 결과를 게이트웨이에게 반환한다. 게이트웨이는 관리자에게 응답을 할 때는 클라이언트 역할을 하게 되며 관리자 상에서 적절한 동작을 호출하여 그 응답을 관리자에게 IIOP를 통해 전달한다.

4.5. 시험 및 검증

4.5.1. 시스템 구성도

본 논문에서 구현된 시스템은 Fig. 3과 같이 구성되어 있다. OSI 대리자는 피 관리 시스템(infocom.cheju.ac.kr)에서 실행 중이며 이들은 OSIMIS에서 제공해주는 대리자를 이용하였다. 이 대리자에 의해 유지되는 MIB (Management Information Base)은 Unix-MIB이며 시험의 편의를 위해 더 이상의 MIB확장은 하지 않았다.

구현된 시스템은 관리자와 CORBA/CMIP 게이트웨이이며, 게이트웨이는 호스트 A에서 실행중이다. 망의 운용자는 어디에서든지 웹 브라우저를 이용하여 게이트웨이에 접근함으로써 대리자들의 정보를 읽어들이거나 변경할 수 있다.

OSI 망 관리 플랫폼은 OSIMIS 4.0을 사용하였다. 게이트웨이 인터페이스를 위해 CORBA플랫폼으로 Visibroker for C++ 3.2를, 관리자를 위하여 Visibroker for Java 3.2, JDK 1.1.6을 이용하였으며 게이트웨이를 구현하기 위해서는 OSIMIS와 ISODE에서 제공하는 루틴을 이용하였다.

4.5.2. 기능 시험

Fig. 4는 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. Fig. 4는 관리자 애플릿이 CORBA/CMIP게이트웨이에 바인딩 한 후, Get 메소드를 실행하여 게이트웨이에서 가져올 수 있는 정보를 IIOP를 통해 Object Web방식으로 가져온 결과를 보여 주고 있다. Fig. 4의 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 서비스가

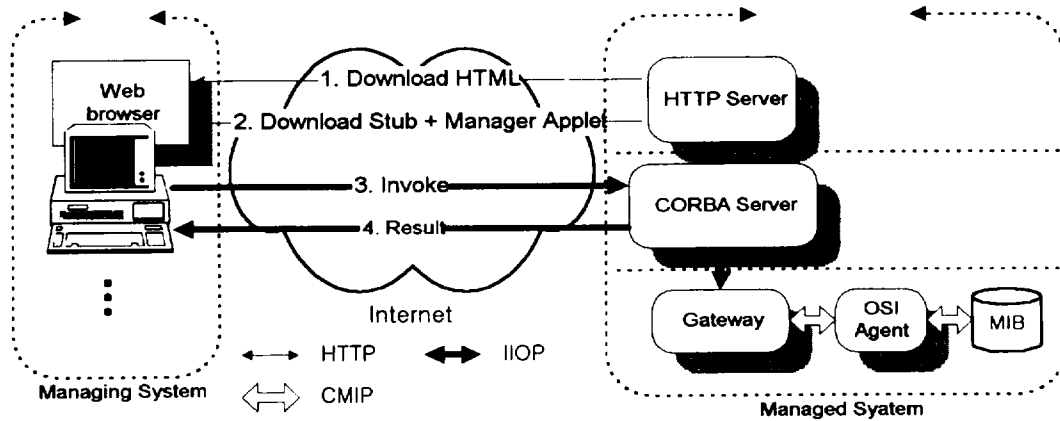


Fig. 3 System configuration

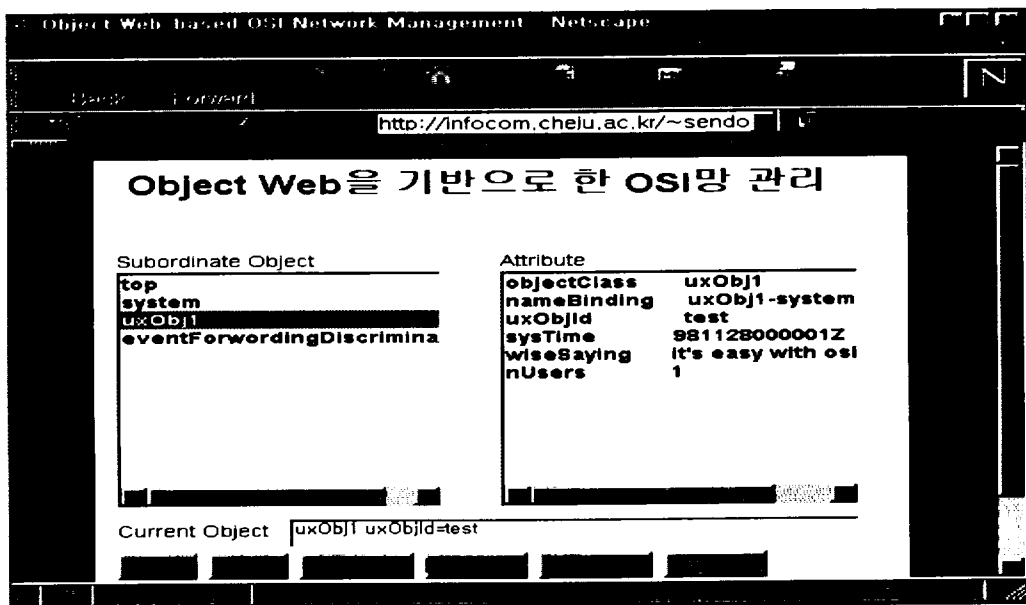


Fig. 4 Result of get requesting the value of managed object in OSI agent

가져올 수 있는 정보를 IIOP를 통하여 Object Web 방식의 관리자에게 전달하게 된 것이다.

V. 결 론

본 논문에서는 Object Web방식의 사용자 인터페이스와 CMIP 프로토콜을 사용하는 대리자를 관리할

수 있는 게이트웨이 구조를 제시하였으며, 그 하부 구조로 CORBA를 이용하였다. 또한 설계된 구조를 기반으로 관리자와 게이트웨이를 구현하였으며, 각각의 요소들이 정상적으로 동작하는지 확인하였다. 게이트웨이 구현에 CORBA를 사용하여 객체들 사이에 투명하게 서비스를 요청하고 응답할 수 있도록 하였고 OSI 대리자를 추상화하여 CORBA 대리자로 보여지도록 함으로써 현재 사용중인 자원 또한 CORBA

를 이용하여 관리가 가능하도록 하였다.

Object Web을 기반으로 한 망 관리는 CORBA에서 제공되는 다양한 서비스 외에도 기존의 웹에서 제공하고 있는 서비스들을 함께 지원하는데 특히 자바는 운영체제에 독립적이며, 모빌코드 속성을 제공함으로써 어플리케이션 개발에 많은 유연성을 제공해 준다. 따라서 본 논문에서는 효율적인 망 관리를 위해 Object Web을 도입하여 관리자틀 구현하였고 CORBA를 이용하여 게이트웨이를 구현하였다. 그러나 본 논문에서는 OSI 대리자에 대한 게이트웨이만을 고려하였고 다양한 망 관리 프로토콜을 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) Guy H.Genilloud, 1996, Towards Distributed Architecture for System Management for of Doctor Sciences, Swiss Federal Institute of Technology in Lausanne.
- 2) Object Management Group, December 1997, The Common Object Request Broker: Architecture and Specification, OMG.
- 3) 조영현, 김영명, 석승학, 1997, TMN을 향한 첫걸음, 하이테크정보, 서울.
- 4) OMG Telecom Special Interest Group, January 1996, CORBA-Based Telecommunication Network Management System, OMG White Paper Draft 2, OMG.
- 5) JIDM taskforce Draft Document: January 1996, "Inter-Working between OSI Management, SNMP and CORBA", NMF.
- 6) Robert Orfali and Dan Harkey, 1997, Client/Service Programming with Java And CORBA, John Wiley & Sons, inc., Canada.
- 7) Qinzheng Kang, Graham Chen, 1996, Integrating CORBA and TMN Environments, IEEE/IFIP Network Operations and Management Symposium, pp. 86-96.
- 8) ISO/IEC 9596, 1990, Common Management Information Protocol(CMIP)
- 9) RFC 1157, May 1990, A Simple Network Management Protocol(SNMP).