

CORBA/SNMP 게이트웨이 설계 및 구현

이 길 행[†] · 허 정 석^{††} · 김 명 균^{††}

요 약

본 논문은 CORBA/SNMP 게이트웨이의 설계 및 구현에 대해 기술한다. 게이트웨이는 기존의 망관리 시스템들을 CORBA를 이용하여 통합하는데 이용된다. 본 논문의 통합 망관리 시스템은 웹과 CORBA 기술을 이용하여 구현된다. 웹은 분산 응용프로그램들에 플랫폼에 독립적이고 사용하기 쉬운 사용자 인터페이스를 제공하고, CORBA는 서로 다른 망관리 모델들을 효과적으로 통합하기 위해 사용된다. 기존의 피관리 객체들을 그대로 수용하기 위해, 제안된 통합 망관리 시스템은 서로 다른 망관리 모델 사이에 변환 게이트웨이를 사용한다. 통합 망관리 서버는 망관리 연산을 위해 기능이 확장된 웹 서버와 변환 게이트웨이들로 구성되는데, 본 논문에서는 CORBA/SNMP 게이트웨이의 설계 및 구현에 대해 기술한다. CORBA/SNMP 게이트웨이는 CORBA와 SNMP 관리 모델 사이의 관리정보 및 관리연산 변환을 위한 정적, 동적 변환 기능을 수행한다. CORBA/SNMP 게이트웨이는 SNMP MIB에 대한 CORBA 뷰를 제공하여, CORBA 관리자들로 하여금 SNMP 에이전트들을 CORBA 연산들을 통해 접근할 수 있도록 하여준다. 또한 CORBA/SNMP 게이트웨이는 SNMP 에이전트에서 발생한 통지(trap)를 받아, 그 통지를 받기를 원하는 CORBA 관리자들에 전달하는 역할을 수행한다.

Design and Implementation of a CORBA/SNMP Gateway

Gil-Hang Lee[†] · Jeong-Suk Heo^{††} · Myung-Kyun Kim^{††}

ABSTRACT

This paper describes the design and implementation of a CORBA/SNMP gateway. The gateway is used to integrate traditional network management systems using CORBA. The integrated network management system uses World Wide Web (WWW) and CORBA. WWW provides a platform-independent and easy-to-use interface to distributed applications and CORBA is used to integrate different network management models seamlessly. To support the traditional managed objects without modification, we use a translation gateway between two different management models. The integrated network management server consists of a WWW server with enhancement for network management and translation gateways for converting management operations between different management models. This paper describes the design and implementation of a CORBA/SNMP gateway. The CORBA/SNMP gateway provides a syntactic and dynamic translation between CORBA and SNMP management models. The gateway provides a CORBA view to SNMP MIBs and allows the CORBA managers to access SNMP agents via CORBA operations. The CORBA/SNMP gateway also delivers the traps occurred in the SNMP agents to the CORBA managers that want to receive the trap information.

1. 서 론

통신망은 점점 더 규모가 커지고 많은 서로 다른 제

작자들로부터 제작된 장비들을 갖는 복잡한 망으로 점차적으로 진화해 왔다. 이러한 복잡한 통신망을 효율적으로 관리하기 위해서는 표준화된 망관리 프로토콜이 필요한데, 이를 위해 국제표준화 그룹인 ITU(International Telecommunication Union)와 ISO(International Standards Organization)는 OSI 시스템 관리모델[1]을 제정하고, IAB(Internet Architecture Board)에서는 Internet

* 본 연구결과는 정보통신부의 정보통신 우수시범학교 지원사업에 의하여 수행된 것입니다.

† 준 회 원 : 한국전자통신연구원 책임연구원

†† 정 회 원 : 울산대학교 컴퓨터정보통신공학부 교수

논문접수 : 2000년 2월 15일, 심사완료 : 2000년 7월 4일

관리를 위해 SNMP[2] 기반 망관리 모델을 제정하였다. 그러나 이러한 기존의 망관리 모델들은 간단한 망관리에는 효율적이지만, 복잡한 망관리 및 서비스 관리와 같은 복잡한 응용에는 한계를 갖고 있다. 망관리 시스템 개발자들은 이러한 한계를 극복하기 위한 새로운 기술을 찾기 위해 많은 노력을 해왔는데, 개방형 분산 객체 기술인 OMG(Object Management Group)의 CORBA(Common Object Request Broker Architecture) [3]는 응용 서비스 레벨에서 잘 정의된, 통일된 API(Application programming Interface)를 제공함으로써 이러한 문제를 해결할 수 있는 좋은 대안으로 여겨지고 있다. 이에 따라 OMG Telecom 작업그룹[4]과 X/Open과 NMF(Network Management Forum)[5] 등 표준화 단체들과 기업체들, 그리고 많은 연구자들은 CORBA와 같은 분산 객체 기술을 이용하여 망관리 시스템을 통합하기 위한 연구를 진행하고 있다 [6-10].

본 논문에서는 이러한 분산 객체 지향 기술인 CORBA를 이용하여 기존의 망관리 시스템들을 통합하기 위한 방법으로 변환 게이트웨이를 이용한 방법에 대해 기술한다. 이러한 게이트웨이를 이용한 통합 망관리 시스템은 기존의 망관리 객체들을 적은 비용으로 효과적으로 수용할 수 있도록 하여준다. 본 논문의 통합 망관리 시스템은 웹과 CORBA 기술을 이용하여 구현된다. 웹은 분산 응용프로그램들에 플랫폼에 독립적이고 사용하기 쉬운 사용자 인터페이스를 제공하고, CORBA는 서로 다른 망관리 모델들을 통합하는데 사용된다. 기존의 피관리 객체들을 그대로 수용하기 위하여, 제안된 통합 망관리 시스템은 서로 다른 망관리 모델 사이에 변환 게이트웨이를 사용한다. 통합 망관리 서버는 망관리 연산을 위해 기능이 확장된 웹 서버와 변환 게이트웨이들로 구성되는데, 본 논문에서는 CORBA/SNMP 게이트웨이의 설계 및 구현에 대해 기술한다. CORBA/SNMP 게이트웨이는 CORBA와 SNMP 관리 모델 사이의 관리정보 및 관리연산 변환을 위한 정적, 동적 변환 기능을 수행한다. CORBA/SNMP 게이트웨이는 SNMP MIB에 대한 CORBA 뷰를 제공하여, CORBA 관리자들로 하여금 SNMP 에이전트들을 CORBA 연산들을 통해 접근할 수 있도록 하여준다. 또한 CORBA/SNMP 게이트웨이는 SNMP 에이전트에서 발생한 통지(trap)를 받아, 그 통지를 받기를 원하는 CORBA 관리자들에 전달하는 역할을 수행한다. 본 논문에서는 피관리 객체들 뿐 아니라 게이트웨이 서비스 객체들과 서로 다른 유형의 통지를 처리하기 위한 이벤트핸들러들을 CORBA 명명트리에 등록

함으로써 모든 객체들을 일관성있게 CORBA 명명서비스를 이용하여 접근할 수 있도록 하였다.

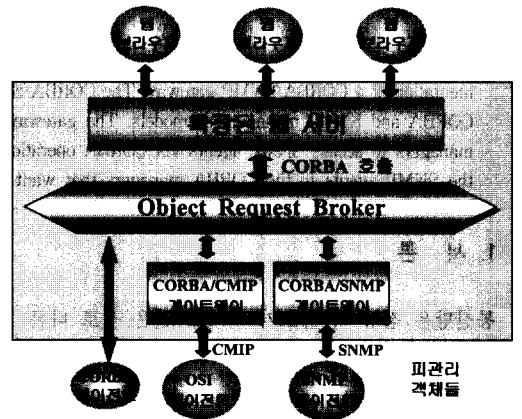
다음절에서는 웹과 변환 게이트웨이를 이용한 통합 망관리 시스템의 전체 구조에 대해 기술하고, 3절에서는 CORBA/SNMP 게이트웨이의 구조 및 각 구성요소의 기능에 대해 기술하고, 4절에서는 CORBA/SNMP 게이트웨이를 이용한 망관리 연산 수행 절차 및 게이트웨이의 구현에 대해 기술하고, 마지막으로 5절에서는 결론 및 향후 연구에 대해 기술한다.

2. 통합 망관리 시스템 구조

제안된 통합 망관리 시스템 구조는 (그림 1)과 같다. 통합 망관리 서버는 망관리 연산을 위한 기능이 확장된 웹 서버와 서로 다른 망관리 연산 사이의 변환을 위한 변환 게이트웨이들로 구성된다. 망관리 서버는 웹 브라우저들로부터 망관리 연산 요청을 URL 형태로 받게된다. 이를 위해 본 논문에서는 아래와 같이 망관리 연산들을 URL 형태로 사상한다.

`http://<웹서버주소>/<망관리도메인>/<망관리연산>?<인자>`

여기서 <웹서버주소>는 망관리 연산 기능이 확장된 통합 망관리 서버의 주소를 나타내고, <망관리도메인>은 피관리 대상인 망관리 에이전트가 속한 망관리 도메인을 나타낸다. 망관리 도메인 이름은 OSI 관리 도메인은 *cmip*, SNMP 관리 도메인일 경우는 *snmp*로 나타낸다. <망관리연산>은 해당 망관리 도메인의 연산을 나타내는데, SNMP 관리 도메인의 경우는 *get* 또는



(그림 1) CORBA 기반 통합 망관리 시스템 구조

set 그리고 OSI 관리 도메인의 경우는 *m-get*, *m-set*, *m-create*, *m-action* 등의 연산을 나타낸다. <인자>는 해당 망관리 연산에 필요한 인자를 나타낸다.

망관리 연산 기능이 확장된 웹 서버는 각 웹 브라우저들로부터 http 요청을 받아 해석하여, 그 요청이 망관리 연산일 경우에는 해당 연산을 수행한다. 망관리 서버는 요청된 망관리 연산이 CORBA 피관리 객체에 대한 연산일 경우에는 ORB를 통해 바로 CORBA 객체 호출을 하지만, 피관리 객체가 CORBA 객체가 아닐 경우에는 해당 변환 게이트웨이를 호출한다. 각 변환 게이트웨이들은 서로 다른 망관리 도메인 사이에서 관리정보 모델과 관리연산의 변환들과 같은 정적, 동적 변환을 수행한다. 본 논문에서는 CORBA/SNMP 게이트웨이의 설계 및 구현에 대해서 기술한다.

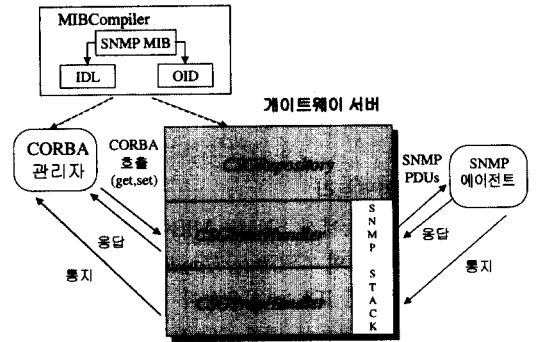
3. CORBA/SNMP 게이트웨이의 설계

CORBA/SNMP 게이트웨이는 SNMP 에이전트의 MIB에 대한 CORBA 뷰를 게이트웨이 내에 생성하여 각 CORBA 관리자들이 SNMP MIB 변수들을 CORBA 연산을 통해 접근할 수 있도록 하여 준다. 또한 CORBA/SNMP 게이트웨이는 각 SNMP 에이전트들로부터 발생한 통지(trap)들을 받아 그 통지를 받기를 원하는 CORBA 관리자들에 전달하여 주는 역할을 수행한다. 본 논문에서는 게이트웨이의 구현 및 사용이 용이하도록 하기 위하여 CORBA의 명명서비스(naming service)와 이벤트 서비스(event service)를 사용하여 구현하였다.

3.1 CORBA/SNMP 게이트웨이 구조

CORBA/CMIP 게이트웨이의 구조는 크게 (그림 2)와 같이 4부분으로 구성된다.

*MIBCompiler*는 SNMP MIB에 대한 CORBA 뷰를 생성하기 위해 필요한 IDL 파일을 생성한다. *MIBCompiler*는 MIB 명세를 입력으로 받아 그 MIB 명세에 포함된 그룹 및 테이블 엔트리들에 대한 IDL 명세를 생성한다. 해당 그룹 및 테이블 엔트리들에 대한 객체들은 이 IDL 명세를 IDLtoJava 컴파일러를 통해 컴파일하여 생성된 Java 객체를 확장하여 생성한다. 또한 *MIBCompiler*는 변환시에 손실되는 MIB 그룹 및 변수들에 대한 OID(Object Identifier)와 같은 정보들을 실행시에 보완하기 위하여 OID 파일을 생성한다.



(그림 2) CORBA/SNMP 게이트웨이 구조

SNMP MIB 명세에서 CORBA IDL로 변환하기 위한 표준 명세가 X/OPEN과 NMF의 JIDM(Joint Inter-Domain Management) 사양변환에 정의되어 있는데, 기본적인 변환 규칙은 다음과 같다:

SNMP ASN.1	CORBA IDL
MIB	모듈
그룹	인터페이스
테이블 엔트리	인터페이스
그룹 내의 각 변수	해당 인터페이스 내의 속성
테이블 엔트리 내의 변수	해당 인터페이스 내의 속성

CSGRepository(CORBA/SNMP Gateway Repository)는 MIB 내의 그룹의 수 또는 테이블의 수 등과 같은 MIB에 대한 정적인 정보들을 유지한다. 또한 *CSGRepository*는 *MIBCompiler*에 의해 생성된 정보들을 이용하여 SNMP 변수들에 대한 이름(OID)과 해당 변수들에 대한 CORBA 객체 이름(CosNaming) 사이의 변환을 위한 매소드들을 가진다.

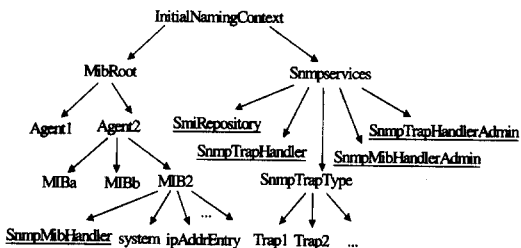
CSGInterHandler(CORBA/SNMP Gateway Interaction Handler)는 CORBA 관리자로부터 망관리 연산을 받아 처리하는 기능을 수행한다. *CSGInterHandler*는 먼저 *MIBCompiler*에 의해 생성된 IDL 명세를 이용하여 각 MIB 내의 그룹 및 테이블 엔트리들에 대한 CORBA 객체를 생성하고, 생성된 객체들을 CORBA 명명트리에 등록한다. 명명트리에 등록된 각 SNMP MIB 객체들은 CORBA 관리자들에 의해 CORBA 객체로 인식되어 CORBA 객체 연산을 통해 접근되게 된다.

CSGTrapHandler(CORBA/SNMP Gateway Trap Handler)는 SNMP 에이전트들에서 발생한 통지를 처리하는 역할을 수행한다. *CSGTrapHandler*는 SNMP 에이전트의 통지를 받아 CORBA의 이벤트 서비스들

통해 전달한다. CORBA의 이벤트 서비스에는 Push와 Pull의 두 가지 이벤트 전달 모델이 있는데, 본 논문에서는 SNMP 통지 전달 방법과 유사한 CORBA의 Push-Push 모델을 사용한다.

3.2 CORBA 명명트리

CORBA 관리자들에게 SNMP MIB에 대한 CORBA 뷰를 제공하기 위하여 *CSGInterHandler*는 *MIBCompiler*에 의해 생성된 IDL을 이용하여 각 MIB 내의 그룹 및 테이블 엔트리들에 대한 CORBA 객체들을 생성하고, 생성된 객체들을 CORBA 명명트리에 등록시킨다. 본 논문에서는 이 CORBA 명명트리를 확장하여 CORBA/SNMP 게이트웨이 내의 서비스 객체들도 명명트리에 등록함으로써, SNMP MIB 객체 및 게이트웨이 서비스 객체들을 모두 명명트리를 통해 접근하도록 하였다. 이를 위한 전체 명명트리의 구조는 (그림 3)과 같다. 명명트리의 루트 *InitialNamingContext* 아래 *MibRoot*와 *SnmppServices* 명명 컨텍스트를 두고, SNMP MIB 객체들은 *MibRoot* 컨텍스트 아래에 등록을 하고 게이트웨이 서비스 객체들은 *SnmppServices* 컨텍스트 아래에 등록을 한다. 또한 서로 다른 SNMP 통지들을 구분하기 위하여 *SnmppServices* 컨텍스트 아래에 *SnmppTrapTypes* 명명 컨텍스트를 두고, 이 컨텍스트 아래에 각 통지 타입(각 통지를 전달하는 이벤트 채널)들을 등록한다.



(그림 3) CORBA 명명트리 구조

3.3 CORBA/SNMP 게이트웨이 모듈

CORBA/SNMP 게이트웨이는 *CSGRepository*, *CSGInterHandler*, 그리고 *CSGTrapHandler*의 3개의 모듈로 구성되어 있다.

3.3.1 *CSGRepository* 모듈

CSGRepository 모듈은 SNMP MIB들에 대한 정적

인 정보들과 SNMP OID과 CORBA 이름 사이의 변환을 위한 메소드들을 포함하고 있으며, 다음과 같은 인터페이스들로 구성되어 있다.

- *OidRepository* 인터페이스 : SNMP OID에서 CORBA 이름으로 변환하기 위한 메소드(*get_scoped_name*, *get_next_scoped_name*)들과 CORBA 이름에서 SNMP OID로 변환하기 위한 메소드(*get_oid*, *get_next_oid*)들을 포함하고 있다.
- *SmiVariableDef* 인터페이스 : OID나 액세스형 등과 같은 SMI 변수들에 대한 정보들을 포함하고 있다.
- *SmiEntryDef* 인터페이스 : 사양변환을 통해 SMI 모듈의 그룹이나 테이블 엔트리들로부터 생성된 IDL 인터페이스들에 대한 정보를 정의하고 있으며, 자신의 OID, 인터페이스내의 변수들의 수, 인터페이스내의 변수들의 리스트 등과 같은 정보들을 포함하고 있다.
- *SmiModuleDef* 인터페이스 : SMI 모듈에 대한 정보를 정의하고 있으며, 모듈내의 *SmiEntryDef* 리스트 등과 같은 정보들을 포함하고 있다.

3.3.2 *CSGInterHandler* 모듈

CSGInterHandler 모듈은 CORBA 관리자로부터 SNMP MIB 객체들에 대한 CORBA 메소드 호출을 받아 SNMP 메시지로 변환한 뒤 해당 에이전트에게 전달하고, 그 SNMP 에이전트로부터 응답 메시지를 받아 다시 해당 CORBA 관리자에게 전달하는 역할을 수행한다. 이를 위해 *CSGInterHandler*는 다음과 같은 인터페이스들로 구성되어 있다.

- *SnmppMibHandlerAdmin* 인터페이스 : 이 인터페이스는 *SnmppMibHandler* 객체를 생성하고 삭제하는 메소드들로 구성되어있는데, CORBA 관리자가 SNMP MIB 변수들을 접근하기 위하여 가장 먼저 접근하는 객체로 게이트웨이 초기화 시에 생성되어 명명트리에 등록이 된다.
- *SnmppMibHandler* 인터페이스 : 이 인터페이스는 SNMP MIB 객체들을 생성하고 초기화하기 위한 메소드들을 가진다. 생성된 MIB 객체들은 관리자에게 의해 접근될 수 있도록 CORBA 명명트리에 등록이 된다. 또한 *SnmppMibHandler* 인터페이스는 *CompositeObject* 객체를 생성하기 위한 메소드를 갖는다.

- *CompositeObject* 인터페이스 : SNMP에서는 하나의 SNMP 메시지를 통해 여러 변수들을 접근할 수 있도록 함으로써 관리자와 에이전트 사이의 트래픽을 줄일 수 있다. CORBA 환경에서 이러한 기능을 제공하도록 하기 위해 본 연구에서는 *CompositeObject* 객체를 사용한다. *CompositeObject* 인터페이스는 *CompositeObject* 객체에 변수를 추가, 삭제, 또는 변경하기 위한 메소드들과 생성된 *CompositeObject* 객체를 접근하기 위한 메소드(*get*, *set*)들을 가진다.
- *MibQueryHandler* 인터페이스 : 이 인터페이스는 CORBA 관리자로부터의 요구에 실제 SNMP 메시지를 구성하고 해당되는 에이전트들에게 전달하는 기능을 수행한다.

3.3.3 CSGTrapHandler 모듈

CSGTrapHandler 모듈은 SNMP 에이전트들에서 발생한 통지들을 받아 그 통지 정보를 받고자 원하는 관리자들에게 전달하는 기능을 수행한다. 이 통지 정보 전달을 위해 본 게이트웨이에서는 CORBA의 이벤트 서비스를 이용하는데, CORBA의 Push와 Pull 이벤트 전달 모델 중에서 Push 모델을 사용한다. 이를 위해 *CSGTrapHandler*는 다음과 같은 인터페이스들로 구성되어 있다.

- *SnmptTrapHandlerAdmin* 인터페이스 : 이 인터페이스는 *SnmptTrapHandler* 객체를 생성하고 삭제하는 메소드로 구성되어 있는데, CORBA 관리자가 SNMP 에이전트로부터 통지를 받기 위하여 가장 먼저 접근하는 객체로 게이트웨이 초기화 시에 생성되어 명명트리에 등록이 된다.
- *SnmptTrapHandler* 인터페이스 : 이 인터페이스는 CORBA 관리자들이 받기를 원하는 통지들을 등록하는 메소드와 에이전트들로부터 받은 통지들을 CORBA 이벤트 서비스를 통해 해당 관리자들에게 전달하기 위한 메소드를 가지고 있다. 각 통지들은 명명트리에 등록되며, *SnmptTrapHandler* 객체에 의해 같은 형의 통지들은 하나의 이벤트채널을 통해 그 통지를 받기 원하는 관리자들에게 전달된다.

4. CORBA/SNMP 게이트웨이 구현

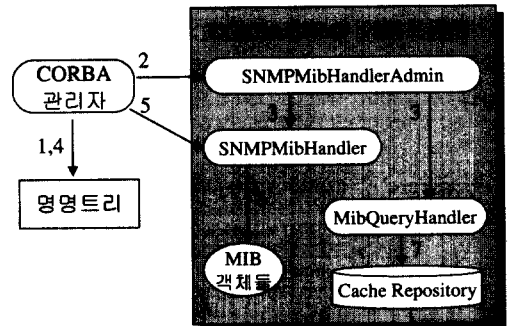
본 절에서는 CORBA/SNMP를 이용하여 관리자들이 SNMP 변수들을 접근하는 절차 및 SNMP 에이전트에

서 발생한 통지들을 해당 관리자들에게 전달하는 과정과 또한 그 구현에 대해 기술한다.

4.1 SNMP MIB에 대한 CORBA 뷰 생성

CORBA/SNMP 게이트웨이는 SNMP MIB를 접근하기 위하여 먼저 각 MIB에 대한 CORBA 뷰를 생성하고, 생성된 MIB 객체들을 명명트리에 등록한다. 이를 위한 절차는 (그림 4)와 같다.

1. CORBA 관리자는 명명트리를 통해 *SnmptMibHandlerAdmin* 객체를 찾는다.
- 2,3. CORBA 관리자는 *create_SnmptMibHandler()*와 *create_MibQueryHandler()* 메소드를 통해 *SnmptMibHandler*와 *MibQueryHandler* 객체들을 생성하고 명명트리에 등록한다.
- 4,5,6. CORBA 관리자는 SNMP MIB에 대한 CORBA 뷰를 생성하기 위하여 명명트리로부터 해당 *SnmptMibHandler*를 찾고, *create_and_init_Mib()* 메소드를 호출한다. 이 메소드를 통해 새로 생성된 MIB 객체들은 명명트리에 등록된다.
7. *MibQueryHandler* 객체는 생성시에 기존에 접근된 MIB 변수들을 저장하기 위한 *CacheRepository*를 초기화한다.



(그림 4) SNMP MIB에 대한 CORBA 뷰 생성 절차

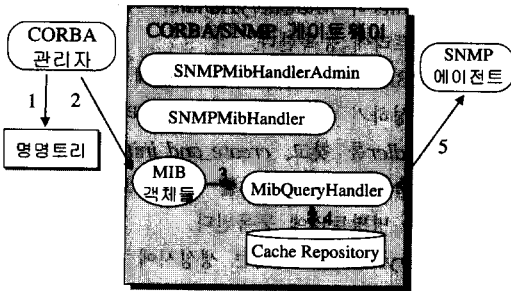
4.2 SNMP MIB 변수 접근

생성되어 명명트리에 등록된 SNMP MIB 객체들은 관리자에게 의해 CORBA 객체로 인식되어 CORBA 명명 서비스를 통해 접근될 수 있는데, 이를 위한 절차는 (그림 5)와 같다.

1. CORBA 관리자는 명명 서비스를 통해 접근하고자

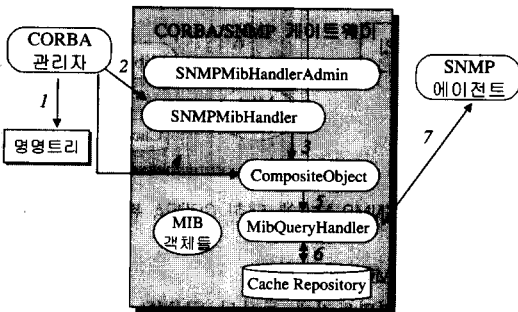
하는 SNMP 변수를 가지고 있는 객체에 대한 참조자를 얻어온다.

2. CORBA 관리자는 위 과정 1에서 얻어온 객체에 대한 *get()* 또는 *set()* 접근 메소드를 호출한다.
3. 해당 객체에 대한 접근 호출은 *MibQueryHandler*의 *get_a_variable()* 또는 *set_a_variable()* 메소드를 호출한다.
- 4.5. *MibQueryHandler* 객체는 접근 메소드의 인자로 주어지는 SNMP MIB 변수를 *CacheRepository*에서 찾아 있으면 바로 그 값을 되돌려 주고, 없으면 그 메소드에 대한 SNMP 메시지를 생성하여 해당 에이전트에 전송한다.



(그림 5) SNMP MIB 변수 접근 절차

CORBA 관리자는 *CompositeObject*를 통해 한번의 객체에 대한 접근 호출을 통해 여러 개의 SNMP 변수들을 접근할 수 있는데, 이를 위한 절차는 (그림 6)과 같다.



(그림 6) CompositeObject 객체를 이용한 SNMP MIB 변수 접근 절차

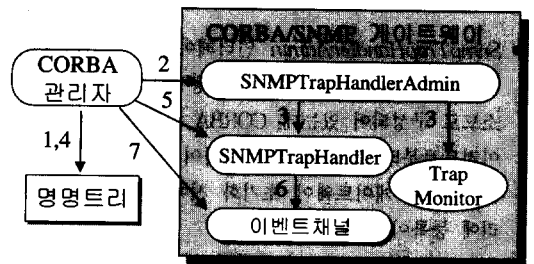
1. CORBA 관리자는 명명 서비스를 통해 접근하고자 하는 *SnmplibHandler* 객체에 대한 참조자를 얻어

온다.

- 2,3. CORBA 관리자는 *SnmplibHandler* 객체의 *create_CompositeObject()* 메소드를 통해 *CompositeObject*를 생성한다.
4. CORBA 관리자는 *CompositeObject* 객체의 *add_var()* (또는 *delete_var()*, *update_var()*) 메소드를 통해 접근하고자 하는 MIB 변수들을 설정하고, 설정이 끝나면 그 *CompositeObject* 객체에 대한 *get()* 또는 *set()* 접근 메소드를 호출한다.
5. *MibQueryHandler* 객체는 접근 메소드의 인자로 주어지는 SNMP MIB 변수들을 먼저 *CacheRepository*에서 찾아 있으면 바로 그 값을 되돌려 주고, 없으면 그 메소드에 대한 SNMP 메시지를 생성하여 해당 에이전트에 전송한다.

4.3 SNMP 통지의 등록 및 전달

CORBA/SNMP 게이트웨이는 SNMP 에이전트들에서 발생한 통지들을 CORBA의 이벤트 서비스를 통해 해당 관리자들에게 전달한다. 각 관리자가 SNMP 에이전트들로부터 통지를 받으려면 먼저 게이트웨이의 *SnmplibTrapHandler*에 자신이 받고자 하는 통지를 등록을 하여야 하는데, 이를 위한 절차는 (그림 7)과 같다.



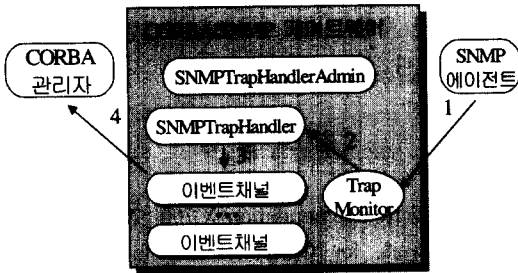
(그림 7) SNMP 통지 등록 절차

- 1,2,3. CORBA 관리자는 명명 서비스를 통해 *SnmplibTrapHandlerAdmin* 객체 참조자를 찾고, 그 객체로부터 *SnmplibTrapHandler* 객체와 *SnmplibTrapMonitor* 객체를 생성한다(이미 존재할 경우 그 객체의 참조자를 반환한다). *SnmplibTrapMonitor* 객체는 게이트웨이에 도착한 통지를 화면에 표시하기 위해 사용하였다.
- 4,5,6. CORBA 관리자는 *SnmplibHandler*의 *register_event()* 메소드를 통해 자신이 받고자 원하는 통지를 등록한다. *SnmplibTrapMonitor* 객체는 요청된 통지가 이미 존재하면 그 통지를 전송하기 위한 이

벤트채널을 반환하고, 존재하지 않으면 그 통지를 전송할 이벤트채널을 새로 생성하고 그 이벤트채널에 대한 객체 참조자를 명명트리에 등록하고 해당 관리자에게 반환한다.

7. CORBA 관리자는 반환된 이벤트채널에 Push 소비자로서 연결을 설정한다.

게이트웨이에 등록된 통지들은 해당 이벤트채널을 통해 관리자들에게 전달이 되는데, 이를 위한 절차는 (그림 8)과 같다.



(그림 8) SNMP 통지 전달 절차

1.2. *SnmptTrapMonitor* 객체는 게이트웨이에 도착한 통지를 화면에 표시하고, 그 통지 정보의 전달을 위해 *SnmptMibHandler*의 *send_event()*를 호출한다.

3.4. *SnmptMibHandler* 객체는 *SnmptTrapMonitor* 객체로부터 전송된 통지의 형을 분석하여 그 통지를 전송할 이벤트채널을 명명트리로부터 찾고, 그 이벤트채널을 통해 통지 정보를 해당 관리자들에게 전송한다.

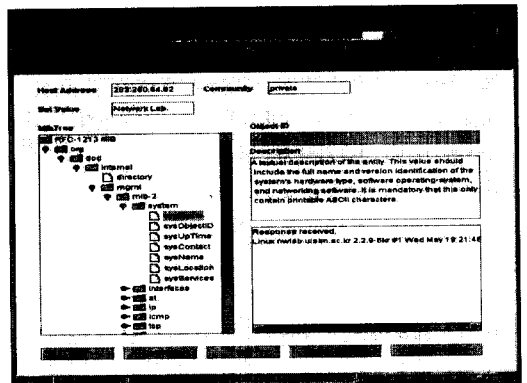
4.4 CORBA/SNMP 게이트웨이의 구현

본 연구에서는 앞 절에서 기술한 CORBA/SNMP 게이트웨이를 윈도우 NT에서 구현하였다. ORB는 VisiBroker for Java 3.4를 사용하였고, SNMP MIB 컴파일러는 Lucent사의 SMItOIDL 컴파일러를 사용하였다. CORBA/SNMP 게이트웨이는 호스트 203.250.64.80에 구현을 하였으며, 구현된 게이트웨이의 기능을 테스트하기 위하여 SNMP 에이전트는 교내의 203.250.64.254 라우터와 리눅스 SNMP 에이전트(203.250.64.82)를 사용하였고, 대상 피관리 객체는 MIB2의 system 그룹과 ip 그룹에 대해 테스트를 하였다. 본 논문은 CORBA/SNMP 게이트웨이의 설계 및 구현에 대해 기술하였고, 이에 따라 본 절에서는 게이트웨이의 기능을

보이기 위해 CORBA 관리자에서 피관리 객체를 접근 (get 또는 set)하는 연산과 피관리 객체에서 발생한 통지가 CORBA 관리자에게 전달되는 결과만을 기술하였다.

(그림 9)는 피관리 객체를 접근할 때 사용하는 화면으로, 화면 구성은 일반 MIB 브라우저와 같은 형태로 구성하였다. (그림 9)는 리눅스 에이전트(203.250.64.82)의 system 그룹내의 sysDesc 변수에 대해 get 연산을 수행한 결과를 보여준다. 각 필드의 의미는 아래와 같다.

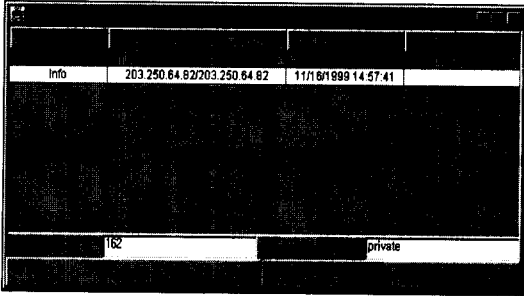
- Host Address : SNMP 에이전트의 호스트 주소를 나타낸다.
- Community : SNMP 에이전트에 요청시 전송하는 community 값을 나타낸다.
- Set Value : Set 연산시 해당 변수에 set 할 값을 나타낸다.
- MibTree : MIB의 트리 구조를 나타낸다.
- ObjectID : MibTree에서 선택된 피관리 객체의 OID 값을 나타낸다.
- Description : MibTree에서 선택된 피관리 객체에 대한 설명(description)을 나타낸다.
- Get, GetNext, Set : 각각 Get, GetNext, Set 연산을 나타낸다.
- Trap Monitor : 선택된 피관리 객체의 값을 주기적으로 폴링하여 값의 변화를 그래프로 보여준다.
- Trap Monitor : 등록된 에이전트들에 대한 통지들을 보여준다.



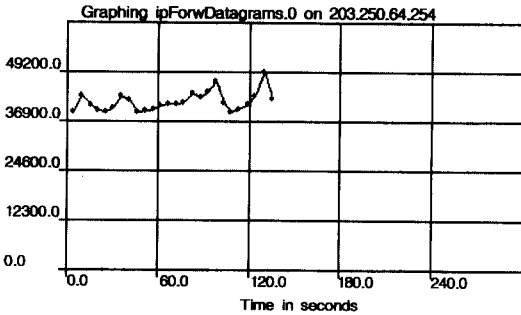
(그림 9) SNMP 변수에 대한 접근(get) 연산 결과

(그림 10)은 하나의 CORBA 관리자가 203.250.64.82 리눅스 에이전트와 라우터 203.250.64.254로부터 통지를 받도록 등록하고, 이 에이전트들로부터 발생한 통

지를 받아 그 결과를 화면에 표시한 것을 보여준다. (그림 11)은 교내 라우터 203.250.64.254에 대해 ip 그룹의 포워딩(forwarding)된 데이터그램의 수(ipForwardDatagrams)를 5초 간격으로 폴링한 결과를 보여준다.



(그림 10) SNMP 통지 정보 화면표시 결과



(그림 11) SNMP 변수(ipForwDatagrams) 주기적 폴링 결과

5. 결론 및 향후 연구

본 연구에서는 웹과 CORBA를 이용한 통합 망관리 시스템에 대해 기술하였다. 웹은 플랫폼에 독립적이고 사용하기 쉬운 인터페이스를 제공하고, CORBA는 서로 다른 망관리 모델들을 효과적으로 통합하도록 하여 준다. 제안된 망관리 서버는 망관리 기능이 확장된 웹 서버와 서로 다른 망관리 모델들 사이의 변환을 위한 변환 게이트웨이들로 구성된다. 이러한 변환 게이트웨이를 사용함으로써 기존의 피관리 객체들을 변경하지 않고 사용할 수 있도록 함으로써 쉽게 통합할 수 있도록 하여 준다. 본 연구에서는 CORBA와 SNMP 관리 모델 사이의 변환을 위한 CORBA/SNMP 게이트웨이의 설계 및 구현에 대해 기술하였다. 제안된 게이트웨이는 SNMP MIB 객체들에 대한 CORBA 뷰를 게이트웨이

에서 제공함으로써 CORBA 관리자들로 하여금 SNMP MIB 변수들을 CORBA 객체처럼 인식하고 접근하도록 하였다. 그리고 SNMP MIB 객체들 뿐 아니라 게이트 내의 *SnmplibHandler*와 *SnmplibTrapHandler* 등과 같은 서비스 객체들도 CORBA 명명트리에 등록하여 CORBA 명명 서비스를 통해 접근할 수 있도록 하였으며, 또한 서로 다른 종류의 통지들을 처리하기 위하여 통지 유형을 명명트리에 등록하여 같은 유형의 통지들 여러 관리자들이 하나의 이벤트채널을 통해 전달받을 수 있도록 하였다.

본 연구의 CORBA/SNMP 게이트웨이는 현재 SNMPv1에 대해서만 구현이 되었으며 SNMPv2를 제공하도록 확장하고 있다.

참 고 문 헌

- [1] CCITT Recommendation X.700 : Management Framework for Open Systems Interconnection (OSI) for CCITT Applications, 1992.
- [2] J. Case, M. Fedor, M. Schoffstall, and J. Davin, A Simple Network Management Protocol (SNMP), RFC 1157, 1990.
- [3] OMG, The Common Object Request Broker : Architecture and Specification, Revision 2.0, 1995.
- [4] OMG Telecom Task Force, CORBA-Based Telecommunication Network Management System, 1996.
- [5] Joint Inter-domain Working Group, Inter-domain Management : Specification Translation, Apr. 1997.
- [6] S. H. Chae, et. al, "Implementation of GDMO to IDL translator and CORBA/CMIP gateway for TMN/CORBA integration," APNOMS'98, Sendai, Japan, Sept. 1998.
- [7] J. W. Hong, et. al, "Web-based intranet services and network management," IEEE Communications Magazine, Vol.35, No.10, pp.100-110., Oct. 1997.
- [8] Jae-Yong, Kim, et. al, "TMN-based integrated network management using web and CORBA technologies," Proc. of the 6-th IFIP/IEEE International Symposium on Integrated Network Management, Boston, USA., 1999.
- [9] Qinzhen Kong and Graham, "Integration TMN and

CORBA environments," Proc. of IEEE/IFIP Network Operations and Management Symposium, Kyoto, 1991.

- [10] Georg Pavlou, "A novel approach for mapping the OSI-SM/TMN model to ODP/OMG CORBA," Proc. of the 6-th IFIP/IEEE International Symposium on Integrated Network Management, Boston, USA., 1999.



이길행

e-mail : ghlee@nice.etri.re.kr

1984년 전남대학교 계산통계학과 (공학사)

1986년 한국과학기술원 전산학과 (공학석사)

1996년 한국과학기술원 전산학과 (공학박사)

1986년~현재 한국전자통신연구원 책임연구원
관심분야 : 망관리 및 보안



허정석

e-mail : heojs@uou.ulsan.ac.kr

1976년 서울대학교 전기공학과 (공학사)

1986년 서울대학교 컴퓨터공학과 (공학석사)

1995년 부산대학교 컴퓨터공학과 (공학박사)

1998년~현재 울산대학교 컴퓨터정보통신공학부 교수
관심분야 : 근거리통신망 프로토콜, 이동통신망



김명균

e-mail : mkkim@uou.ulsan.ac.kr

1984년 서울대학교 컴퓨터공학과 (공학사)

1986년 한국과학기술원 전산학과 (공학석사)

1996년 한국과학기술원 전산학과 (공학박사)

1986년~현재 울산대학교 컴퓨터정보통신공학부 조교수
관심분야 : 연결망, 망관리 및 보안, 인터넷응용