

# CORBA/SNMP 통합 관리 시스템 구축을 위한 게이트웨이 구축 방안

## (An Implementation Methodology of a Gateway for Inter-Domain Management Between CORBA and SNMP)

강 영 민 <sup>\*</sup>    홍 원 기 <sup>\*\*</sup>  
(Young-Min Kang) (Won-Ki Hong)

**요 약**    엔터프라이즈 네트워크의 크기가 커지고 그 관리 범위가 넓어짐에 따라 분산 망 관리 시스템의 수용을 요구하게 되었고 또한 표준 프레임워크들의 상호 연결 및 작용이 필요하게 되었다. 이 논문에서는 네트워크 관리 프레임워크인 SNMP도메인과 분산 객체 기술의 하나인 CORBA도메인간의 통합 관리를 위한 구현 방안을 제시한다. 표준에서 사용하는 직접 변환은 객체 지향적인 특성을 표현하고 있지 않으며, action을 위한 속성의 정의가 다른 속성과 구별되지 않는 등의 한계를 지니고 있다. 이러한 한계점은 추상적 변환 방법을 통해 새로운 클래스의 정의, 명확한 메소드 정의라는 방법으로 충분히 극복될 수 있다. 직접 변환의 편리함과 추상적 변환의 효율적인 관리 정보의 구축이라는 장점을 이용한 직접 변환과 추상적 변환의 혼용을 통한 정보 모델 변환 방법을 이용한 게이트웨이의 개발 방법을 제시한다. 이러한 통합 관리 시스템 구축은 CORBA/SNMP 간의 효율적인 통합 관리를 제공한다.

**Abstract**    Today's enterprise networks are composed of multiple types of interconnected networks. Enterprise Networks interoperability between these technologies is needed. To enable interworking, it is necessary to be able to map between the relevant object models and to build on this to provide a mechanism to handle protocol conversion on the domain boundaries.

In this paper, we describe a gateway between management application in the CORBA domain and agent in the SNMP domain and various integration methods. The main function of the gateway is to dynamically convert the method invocations on object reference in CORBA domain to SNMP messages for MIB entries at remote agents. We also present translation methods from SNMP MIB to CORBA IDL using Direct translation and Abstract translation. JIDM algorithm has no notion of containment and inheritance relationships between object classes and is difficult to understanding between management attribute and SNMP Action attribute. Abstract translation over come these problems. New superclasses define for common attributes and define explicit CORBA method for SNMP Action. It is a methodology for obtaining the CORBA-compliant management agents from already existing SNMP agents.

### 1. 서 론

오늘날의 컴퓨팅 환경은 그 미래의 발전 방향을 예측할 수 없을 정도로 빠르게 변화, 발전하고 있다. 엔터프

라이즈 네트워크는 여러 장비와 시스템 그리고 다양한 업무 수행을 위한 서비스들로 구성이 되어 있다. 이러한 환경은 시스템의 증가와 운영되는 서비스 프로그램들의 다양화 및 관리 프레임워크의 이질성 등으로 인해 그 규모 및 복잡성이 날로 증가가 되고 있다. 이렇게 거대하고 복잡한 네트워크를 기술적 측면에서나 경제적 측면에서 효과적으로 관리하는 문제는 시급히 해결해야 할 당면 과제이다 [1].

효과적이며 안정된 관리를 위해 네트워크 및 시스템의

<sup>\*</sup> 비 회 원 : 포항공과대학교 컴퓨터공학과  
ymkang@postech.ac.kr

<sup>\*\*</sup> 종신회원 : 포항공과대학교 컴퓨터공학과 교수  
jwkhong@postech.ac.kr

논문접수 : 1999년 1월 18일

심사완료 : 1999년 8월 30일

관리는 관리하고자 하는 도메인의 특성에 맞추어 현재 표준화가 진행되고 있다. 통신망 관리를 위해서는 OSI의 개념을 확장한 Telecommunication Management Network (TMN) [2]을 사용하고 있으며, 단순한 네트워크 자원의 관리는 Simple Network Management Protocol (SNMP) [3]를 통해 하고 있다. 또한 요즘은 소프트웨어 측면의 분산 시스템 관리 및 통합 관리를 위한 통합 기술로 Common Object Request Broker Architecture (CORBA) [4]를 사용하고 있다.

각 기관에서 제정한 표준들은 분산 환경보다는 상하 개념에 의한 관리 구조만을 고려하였으며, 상호 작용에 대해 지원하지 못하는 서로 비 호환의 관리 기준들이다. 이로 인해 관리자는 서로 다른 도메인간의 정보에 대한 통합된 정보를 획득하지 못하고 있다.

서로 다른 프레임워크간의 통합 관리라고 하는 것은 관리자 또는 개발자에게 서로 다른 도메인간에 존재하는 정보를 하나의 방법으로 통합하고, 관리하고자 하는 데이터를 단일 방법으로 다룰 수 있는 방법이다 [5]. 이러한 통합 관리를 위해서는 몇 가지 요구되는 사항이 있다. 즉 관리하고자 하는 도메인상의 관리 정보를 파악하여 통합해 주어야 하며, 그 관리 정보를 관리하기 위한 여러 오퍼레이션에 대한 통합 과정이 이루어져야 한다 [6].

이 논문에서는 네트워크 관리 프레임워크인 SNMP 도메인과 분산 객체 기술의 하나인 CORBA도메인간의 통합 관리를 위한 구현 방안을 제시한다. 표준에서 사용하는 직접 변환은 객체 지향적인 특성을 표현하고 있지 않으며, action을 위한 속성의 정의가 다른 속성과 구별되지 않는 등의 한계를 지니고 있다. 이러한 한계점은 추상적 변환 방법을 통해 새로운 클래스의 정의, 명확한 메소드 정의라는 방법으로 충분히 극복될 수 있다. 직접 변환의 편리함과 추상적 변환의 효율적인 관리 정보의 구축이라는 장점을 이용한 직접 변환과 추상적 변환의 혼용을 통한 정보 모델 변환 방법을 이용한 게이트웨이의 개발 방법을 제시한다. 이러한 통합 관리 시스템 구축은 CORBA/SNMP 간의 효율적인 통합 관리를 제공한다.

이 논문의 구성은 다음과 같다. 먼저 2장에서는 통합 관리 시스템 구축과 관련된 연구를 알아본다. 3장은 통합 관리를 위한 시스템 구축 방안에 대해 설명한다. 4장은 이 논문의 핵심으로 실질적인 통합 관리 시스템을 위한 게이트웨이의 설계에 대해 설명한다. 5장은 구현에 대한 설명을 하고 마지막으로 6장에서 결론을 맺고 향후 계획에 대해 설명한다.

## 2. 관련 연구

Joint Inter-Domain Management (JIDM) Group은 X/Open과 Network Management Forum (NMF)의 도움 아래 구성된 working group이다. 이 그룹은 서로 다른 관리 기술을 사용하는 도메인간의 통합에 대한 연구를 하고 있다. CORBA, CMIP그리고 SNMP기술에 관심을 두고 있으며, 이 중 CORBA/SNMP와 CORBA/CMIP의 통합에 대한 연구를 주로 행한다[6].

JIDM 그룹은 다른 도메인간의 완벽한 통합 관리를 위해 두 부분의 매핑 작업에 대한 표준을 발표하였다. 한 부분은 Specification Translation이라고 하는 부분으로 통상 ST라 부른다 [5, 7]. 이 ST의 목적은 서로 다른 도메인간의 정보 모델을 호환이 가능하도록 변환하는 작업이다. JIDM은 ASN.1 [8]과 GDMO [9], SMI [3]를 IDL로 변환하는 알고리즘을 발표하였다.

또 한 부분은 Interaction Translation, IT라고 하는 부분으로 프로토콜 변환을 담당해 주는 부분이다[5, 7]. 즉 각 도메인간은 정보 모델 뿐 아니라 사용 프로토콜 또한 호환이 되지 않으므로 이에 대한 해결을 담당하는 부분이다. JIDM은 이 두 가지 작업에 대한 표준을 발표했으며, 통합 관리를 요구하는 많은 곳에서 JIDM의 기준을 따라 제품을 만들거나 연구 중에 있다.

### 2.1 Specification Translation (ST)

ST는 GDMO나 SMI를 CORBA IDL로 변환을 하기 위한 메커니즘이다. 이러한 ST에는 ASN.1, SNMP SMI와 CMIP의 GDMO에 대한 변환 방법이 있다.

#### 2.1.1 ASN.1의 IDL 변환 알고리즘

ASN.1은 CCITT와 ISO에 의해 제정된 표준으로 응용 프로그램의 데이터를 사용하는 플랫폼에 상관없이 추상적으로 정의하거나 구조를 정의하는데 사용하는 언어로써 SNMP와 CMIP에서 관리 정보를 표현하는데 공

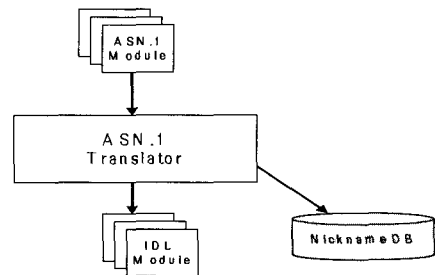


그림 1 ASN.1의 IDL 변환

통적으로 사용되고 있다. 아래의 (그림 1)은 이러한 변환의 구조이다.

2.1.2 GDMO의 IDL 변환 알고리즘

하나의 GDMO 템플릿은 하나의 인터페이스로 정의되며, 템플릿의 속성은 그 Interface에 대한 오퍼레이션의 Set으로 변환된다. 그리고 Action들은 Interface의 Operation으로 변환된다. Notification은 이벤트 처리를 위한 오퍼레이션이 된다. GDMO에 정의된 모든 관리 객체 클래스는 IDL로 전환되며, GDMO에서 정의된 클래스간의 상속 구조도 인터페이스에 그대로 정의가 된다. 이렇게 생성된 인터페이스에 대한 속성과 오퍼레이션에 대한 구현은 IT에서 담당하면 된다.

2.1.3 SMI의 IDL 변환 알고리즘

기본적인 SNMP의 데이터 타입, 매크로 그리고 trap에 대한 것은 CORBA IDL로 변환한다. 다음과 같은 기본 알고리즘을 따른다.

- 모든 SNMP group은 IDL 인터페이스가 된다. group에 정의된 스칼라 데이터 타입은 객체 클래스의 속성이 된다.
- SNMP 테이블은 IDL 인터페이스가 된다. 테이블에 정의된 열 변수들은 해당 클래스의 속성으로 정의 된다.
- SNMP trap은 CORBA의 이벤트 서비스를 통해 정의된다.

3. 통합 관리 시스템 구축을 위한 접근 방법

이번 장은 관리 정보 통합과 기능상 통합을 통해 정의되는 통합을 위한 다양한 접근 방법에 대해 알아보도록 하겠다.

3.1 변환과 대치를 통한 접근 방법

정보 모델 변환 (Information Model Translation)을 통한 방법과 정보 모델 대치 (Information Model Mapping)를 통한 방법이다.

• 정보 모델 변환

이 방법은 원천 정보 모델을 목적 정보 모델로 변환 (Translation)하여 정보 모델을 통합하는 방법이다. 원천 정보 모델의 구문 하나하나를 목적 정보 모델의 구문으로 변환하는 방법이다. JIDM의 ASN.1/SMI/GDMO를 CORBA IDL로 변환하는 방법이 이와 같은 접근 방법의 대표적인 예이다 [5, 7].

• 정보 모델 대치

이 방법은 관리 대상 프레임워크의 정보 모델 전체를 이미 존재하는 통합 관리를 위한 통합 관리 프레임워크의 정보 모델로 대치 (Mapping)하는 방법이다. 예로 IBM

에서 개발한 System View 라는 Agent가 이러한 접근 방법 [14]을 택하고 있다.

3.2 변환 (Translation)원칙에 따른 접근 방법

직접적인 변환과 추상적인 변환 방법이 있다.

• 직접 변환 (Direct Translation)

직접 변환 방법은 원천 모델의 관리 정보들을 정해진 변환 규칙, 즉 알고리즘에 의해 목적 정보 모델로 구문상 변환 (Syntax Translation)을 하는 방법이다. 이와 같은 방법은 JIDM의 ASN.1/SMI/GDMO를 IDL로 변환하는 작업에서 사용하는 방법이다.

• 추상적인 변환 (Abstract Translation)

추상적인 변환의 목적은 원천 정보 모델을 목적 정보 모델로 바꿀 경우 불필요하거나 연관이 없는 관리 정보를 삭제하고 변환 시 손실되는 유용한 정보를 삽입한다든가 아니면 관리자가 외부에서 새로운 관리 정보 및 기능을 넣음으로써 보다 효율적인 관리 정보를 만들어 내는 것이다. 즉 구문상 일대일 변환을 통한 변환 방법이 아니라 문맥상 의미(Semantic)를 부여해 변환하고자 하는 것이다 [13].

4. CORBA/SNMP 통합 관리를 위한 게이트 웨이의 설계

이 장에서는 실제 구현되어야 할 CORBA/SNMP 게이트웨이의 요구 사항에 대하여 알아보고 이를 충족시키기 위해 적합한 접근 방법을 도출한다. 또한 도출된 접근 방법을 기반으로 한 기본 설계에 대해 살펴보도록 하겠다.

4.1 CORBA/SNMP 통합 관리 시스템의 요구 사항

구현될 시스템은 CORBA 도메인과 SNMP 도메인간의 통합 관리를 위한 통합 메커니즘, 즉 게이트웨이를 구현하는 것이다. 게이트웨이를 구현하기 위해 CORBA 도메인의 매니저와 SNMP 도메인의 에이전트간의 관리 정보 모델과 관리 기능에 대한 요구 사항을 알아 보도록 하겠다.

4.1.1 통합 관리 시스템의 요구 사항

서로 다른 영역을 관리하는 프레임워크들을 통합하기 위해서는 통합의 대상이 되는 영역들간의 관리 정보에 대한 통합, 관리 시스템 운영을 위한 관리 기능 통합 및 관리 시스템의 관리 정보 및 운영을 위해 제공되는 구조적인 통합이 요구된다.

• 정보모델 (Information Model) 통합

관리 정보에 대한 통합이라고 하는 것은 통합 관리를 하고자 하는 영역들의 관리 자원들을 매니저가 관리 가능한 관리 정보 모델로 바꾸는 것이다.

- 관리 기능 (Operation) 통합

관리 시스템의 기능에 대한 통합이라는 것은 위에서 언급한 관리 정보 통합을 기반으로 매니저 시스템이 관리를 위해 수행되어야 하는 기능에 대한 통합이다.

- 관리 구조 통합

관리 자원들의 모델링 접근 방법 및 구문상 개념에 대한 파악이 이루어져야 한다. 먼저 모델링 접근 방법이라고 하는 것은 관리 프레임워크가 어떠한 모델링 접근 방법을 가지고 있는지는 것이다. 구문상 개념 파악이라는 것은 관리 대상을 나타내는 방법으로 무엇을 통해 관리 정보를 기술해 나가느냐 하는 것이다.

#### 4.1.2 게이트웨이의 관리 정보

CORBA 도메인의 정보와 SNMP 도메인의 정보를 정확히 파악해 CORBA 매니저가 SNMP 에이전트를 관리 할 수 있게 하는 단일 관리 정보, 즉 IDL을 도출해 낼 수 있어야 한다 [7].

#### 4.1.3 게이트웨이의 관리 기능

CORBA 매니저는 SNMP 에이전트의 정보를 획득하거나 변경을 위한 오퍼레이션이 필요하다. 즉 SNMP의 Get/Set 오퍼레이션이 필요하게 된다. 그러나 두 도메인간의 사용 프로토콜이 다르기 때문에 직접적으로 CORBA 매니저가 SNMP 에이전트를 이용할 수가 없다. 그렇기 때문에 게이트웨이는 두 도메인간의 원활한 오퍼레이션을 수행해 줄 수 있어야 한다. 또한 Get/Set에 대한 기본적인 오퍼레이션 외에 SNMP 에이전트에서 발생하는 trap 또는 notification에 대한 정보를 CORBA 매니저에게 정확히 알려줄 수 있는 기능을 제공해야 한다 [7].

### 4.2 CORBA/SNMP 통합 관리 시스템 구축을 위한 접근 방법

통합 관리를 위한 여러 접근 방법은 이미 3장에서 언급하였다. 이러한 접근 방법 중 CORBA 도메인과 SNMP 도메인을 효과적으로 관리할 수 있는 통합 접근 방법에 대해 알아보도록 하겠다.

- 변환과 대치를 통한 게이트웨이의 구현 방법

CORBA 도메인과 SNMP 도메인간은 도메인간의 정보 모델이 서로 상이하고 CORBA상에 이미 관리 정보 모델로 규정한 어느 정보 모델도 존재하지 않기 때문에 정보 모델 대치 방법은 택할 수가 없다. 그렇기 때문에 SNMP의 정보를 CORBA 도메인의 정보인 IDL로 변환하는 정보 모델 변환 방법을 사용한다.

- 변환 원칙에 따른 게이트웨이의 구현

직접 변환에 대해 살펴보면 SNMP 도메인의 관리 정보 모델인 SMI를 CORBA 매니저 정보인 IDL로 일

대일 변환해야 한다. 또 한 가지 고려 사항은 추상적인 변환 방법에 대한 것이다. CORBA 도메인과 SNMP 도메인의 경우도 그 관리 정보 모델이 매우 상이함으로 추상적인 변환 방법을 이용할 수 있을 것이다.

- 직접 변환, 추상적 변환의 혼용을 통한 게이트웨이의 구현

두 도메인인 CORBA와 SNMP간의 정보 모델이 서로 상이하기 때문에 직접 변환만을 가지고는 부족하다. 즉 변환기를 이용하여 SNMP SMI를 CORBA IDL로 직접 변환하게 되면 속성들은 IDL에서 정의되는 인터페이스와 속성들로 표현이 이루어지지만 이러한 직접 변환을 통해 나온 IDL이 CORBA의 사용 환경과 구문상 속성을 모두 정확히 표현하지는 못하고 있다.

우선 직접 변환 알고리즘은 객체 지향 기법의 상속이나 객체간의 관계 등 객체 지향 기법의 특성을 제대로 표현하지 못하고 있다. 또한 SNMP는 기본적으로 명확히 action을 취할 수 있는 오퍼레이션이 정의되어 있지 않기 때문에 [3] 관리자 입장에서 관리 대상에 대한 관리를 위해 필요시하는 명확한 action 오퍼레이션의 정의가 들어있지 못하게 된다. 그렇기 때문에 매니저는 단순히 SNMP에서 지원하는 Get/Set의 오퍼레이션만을 취할 수 있다.

이러한 SMI를 IDL로 직접 변환함으로써의 한계는 사용자와의 상호 작용을 통해 충분히 극복될 수 있는 부분이다. 즉 위에서 언급한 단점들을 사용자가 파악하여 필요 부분이나 유용한 정보 부분을 추가할 수 있다면 SMI를 IDL로 바꾸었을 때 CORBA 매니저는 보다 유용한 정보 모델을 이용할 수 있기 때문에 보다 효과적으로 SNMP 도메인을 관리 할 수가 있다. 바로 이 부분이 추상적 변환 방법이 된다.

#### 4.3 정보 모델 변환의 기본 구조

통합 관리 시스템 구축을 위한 게이트웨이는 정보 모델 변환 부분과 상호 연동을 위한 두 부분으로 구성된다. 우선 정보 모델 변환 메커니즘의 기본 구조는 그림 1과 같다.

사용자에게서 원천 파일을 입력받아 목적 모델로 바꾸기 위한 변환 모듈과 모듈을 통해 나온 정보에 사용자가 필요한 정보를 추가하거나 삭제할 수 있는 사용자 인터페이스 부분으로 구성된다.

변환 모듈과 사용자 상호 작용을 거쳐 원하는 정보 모델을 만들어 낸다. 변환 모듈은 변환 알고리즘에 의한 자동화 모듈이며, 사용자 인터페이스에 대한 부분은 관리자의 수동작업을 통해 이루어진다. 변환 모듈은 JIDM의 변환 알고리즘을 이용한다. JIDM 알고리즘은 몇가지

한계점이 있는데 다음과 같다.

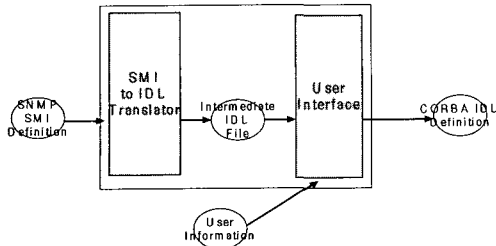


그림 2 정보모델 변환 메커니즘의 구조

- 첫번째로 객체 지향 방법에 관한 문제이다. JIDM의 변환 알고리즘은 CORBA 객체를 생성하나 객체 지향 기법에서 가장 먼저 논하는 객체들간의 상속이라는 개념이 없다. 즉 group이나 테이블로 정의되어 있는 것을 단지 클래스로 정의만 할 뿐 상위 클래스에 대한 정의 및 각 클래스간의 연결성에 대한 점은 고려를 하고 있지 않다. 따라서 객체 지향적인 방법을 사용하고 있는 CORBA 도메인 환경에서는 JIDM의 변환 알고리즘을 통해 나온 것이 적절하지 않다.

- 다음은 action에 대한 문제이다. SNMP는 에이전트에게 명확히 action을 가할 수 있는 특별한 메커니즘을 가지고 있지 않다. 그러나 SNMP set 오퍼레이션을 통해 어떠한 action을 취할 수 있게 하는데 이것을 action invocation이라 한다. 그러나 이 방법은 명확한 오퍼레이션 호출에 의한 것이 아니기 때문에 CORBA 도메인에서는 문제가 될 수 있다.

CORBA 도메인은 명확한 오퍼레이션 호출로 action을 취하는데 반해 JIDM의 변환 방법을 통하게 되면 일반적인 속성들과 action invocation을 위한 속성들 사이의 구분이 힘들어 지기 때문에 CORBA 매니저는 명확한 action 오퍼레이션을 취할 수 없게 된다. 따라서 명확히 action을 취할 수 있게 해준다면 사용자는 일반적인 속성과의 명확한 구분과 SNMP 에이전트의 효율적인 관리를 이룰 수 있다.

사용자 인터페이스 부분에서 해야 하는 작업은 이러한 직접 변환의 한계점을 극복할 수 있는 작업들로서 아래와 같다.

- 새로운 Superclass의 정의 : 이 정의는 목적 정보 모델을 보다 더 객체 지향적인 특성을 가진 정보 모델로 만들기 위한 작업이다. 몇몇의 객체 클래스들을 살펴보면 포함하고 있는 속성들이 성격(타입)은 같으나 이름

이 틀린 것을 포함하고 있다. 이러한 공통적인 성격을 갖는 속성들을 묶어 상위 클래스로 정의하게 되면 공통의 속성들을 이용하는 객체들은 이 공통의 속성을 묶어 정의한 상위 클래스를 상속받아 쓸 수 있게 된다. 주로 이렇게 공통의 특성을 담고 있는 것은 이름을 표현하는 부분, 설명 부분, 상태 표현 등의 부분이 주로 속하게 된다. 이렇게 공통 부분에 대한 상위 클래스를 정의함으로써 객체 확장에 이용될 수 있는데 새롭게 추가되는 클래스들은 상위 클래스를 단순히 상속받음으로써 클래스를 구성할 수 있다.

- SNMP의 action은 어떤 속성의 값을 set할 때 일어나게 되는데 명확한 action 오퍼레이션을 요구하는 CORBA 도메인에서는 다른 방법을 취해야 한다. 에이전트가 속성 값을 변경시킬 때 지원할 수 있는 기능이 라면 매니저도 그 속성을 보고 원할 때는 언제든지 그 action을 수행할 수 있게 해주어야 한다. 예를 들어 컴퓨터 시스템을 관리하고자 하고 그 컴퓨터의 관리 속성인 CPUaction이 있다고 생각해 보자. 속성 CPUaction의 값이 1 또는 2, 3, 4로 set됨에 따라 어떤 암시적인 action이 수행된다고 하면 관리자(사용자)는 CPUaction이라는 속성 정의만을 보고서는 action을 취하는 속성인지 아닌지에 대해 action을 취하지 않는 다른 일반적인 속성과 구별하기란 힘들다. 그렇기 때문에 만약 관리자가 사용자 인터페이스를 통해 enable(), disable(), lock() 그리고 unlock()등의 명확한 오퍼레이션을 매니저 부분에 정의하여 삽입할 수 있다면 CPUaction을 1로 setting하는 것을 enable()이란 메소드를 호출하는 것으로 한다면 매니저는 명확한 action 오퍼레이션을 취할 수 있으면서 다른 일반적인 속성들과의 구분도 할 수 있을 것이다.

이외에 효율적인 관리를 위해서라면 필요로 하는 부분이 있다면 언제든지 추가, 삭제가 가능하다. 이러한 객체 지향 모델의 구축을 좀 더 정형화된 방법으로 살펴해보도록 하겠다.

#### 4.3.1 객체 지향 모델의 구축 방안

변환기를 통해 나온 CORBA IDL을 보다 CORBA 환경에 적합하기 위해 위에서 언급한 내용을 추가하는 다음의 과정을 거친다. 즉 객체 지향 모델링을 위해서는 객체들의 상속(Inheritance)과 집합(Aggregation)관계를 표현해 주어야 한다 [15].

(1) 객체들을 추출 : CORBA IDL에서 정의되는 모든 객체들을 정리한다. 이렇게 함으로써 모든 객체의 속성들을 정리하여 다음에 이어지는 과정을 원활히 할 수 있게 한다. 또한 이 단계에서

action invocation을 위한 속성들을 찾아 적당한 메소드로 정의함으로써 CORBA 매니저가 명확한 action을 취할 수 있게 해준다.

- (2) 객체들간의 상속(Inheritance) 정의 : 정리된 객체들의 속성을 파악하여 다음을 찾아낸다. 즉 공통의 속성들이다. 이름은 틀리나 의미와 기본 타입이 같은 공통적인 속성들이 정의되어 있다. 이러한 공통의 속성을 묶어 상위 클래스로 정의하고 나머지 객체들은 새롭게 정의한 상위 클래스를 상속 받도록 한다.
- (3) 집합(Aggregation)관계 정의 : (2)의 과정에서 정의되지 않은 객체들은 집합 관계로 표현을 함으로써 객체들간의 구성을 정의해 준다. 이처럼 (2),(3) 단계를 통해 객체들은 상속이나 집합 등의 연결 관계가 정의되어지므로 보다 객체 지향적인 정보 모델을 갖게 된다.
- (4) CORBA IDL의 수정 : 위에서 정의한 상속이나 집합 관계를 Object Modeling Technique (OMT) [28]로 표현함으로써 쉽게 새로운 IDL을 얻을 수 있다.

그림 2는 위의 과정을 도식화한 것이다.

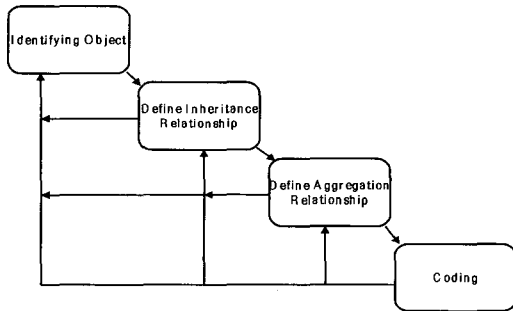


그림 3 객체 지향 모델의 구축 과정

#### 4.3.2 객체 지향 모델의 구축 예

직접 변환의 한계를 극복하고 보다 CORBA 환경에 적합한 정보 모델 변환의 예를 살펴보기 위해 RFC1213 MIB II [21]를 기반으로 한 Sun Workstation MIB을 대상으로 하겠다. Sun MIB은 RFC1213에서 정의한 system, interface, ip, icmp, tcp, upd 와 snmp group 외에 sun system관리를 위한 sunSystem과 SunHostPerf그룹을 가지고 있다. 다음의 그림 3은 직접 변환기를 통해 나온 각각의 인터페이스의 일부분을 OMT notation으로 표현한 것이다. 그림에서 보듯이 객

체간의 상속이라든가 하는 개념은 볼 수가 없다. 단지 Sun MIB이 각각의 인터페이스로 구성된다는 집합 관계만이 표현된다.

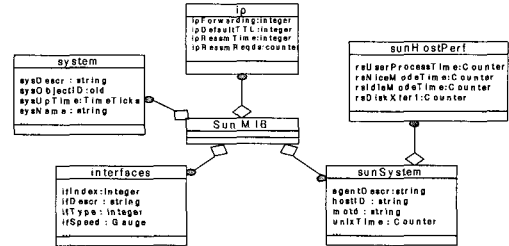


그림 4 직접 변환을 통해 나온 객체 모델

직접 변환기를 통해 나온 IDL을 통해 각각의 인터페이스와 속성들을 정리해 보면 Counter 타입의 In/Out속성들이 각 그룹마다 대칭적으로 정의되어 있다. 특히 이들 중 몇몇은 여러 그룹에서 공통적으로 쓰이고 있으며, 기본 타입도 같다. 이러한 속성을 묶어 일반적인 In/Out 인터페이스의 메소드로 정의하여 다른 인터페이스에서 이를 상속받아 쓰도록 한다.

또한 read-write 속성 중 action을 위한 것들은 메소드로 정의한다. 예로 ip 그룹의 ipForwarding 속성은 1, 2로 setting 함에 따라 Forwarding 또는 not\_Forwarding을 하게 된다. 이러한 속성들을 Forwarding()과 not\_Forwarding() 메소드로 정의를 하여 CORBA 매니저가 바로 이용할 수 있도록 한다.

그림 4의 OMT notation [28]은 이러한 과정을 통해 나온 모델 중 일부분을 보여준다.

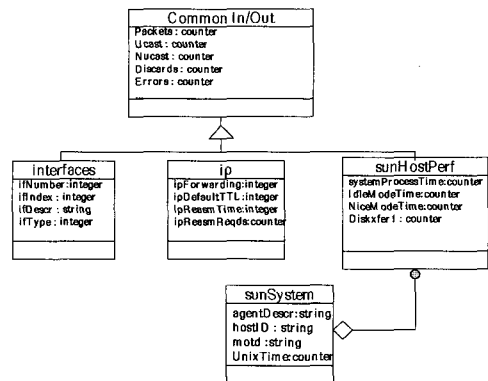


그림 5 객체 지향 모델의 구축 예

4.4 상호 연동 변환의 기본 구조

이번 절은 상호 연동에 대한 부분으로 CORBA 매니저에서 정보 모델 변환으로 얻어진 SNMP MIB 객체를 이용하여 SNMP 도메인 관리를 위한 오퍼레이션 수행을 가능하게 하는 것이다 [7]. CORBA 매니저는 ORB를 통해 기본적인 오퍼레이션인 Get/Set을 호출하며, 통합 메커니즘인 게이트웨이는 이를 받아 SNMP Get/Set을 호출함으로써 두 도메인간의 상호 작용을 가능케 한다. 또한 SNMP 에이전트에서 발생하는 Trap에 대한 정보는 일반적인 오퍼레이션의 수행과 반대로 게이트웨이가 이를 받아 CORBA 매니저에게 알려 주어야 한다. 이러한 기능의 기본적인 구조는 그림 5와 같다 [12].

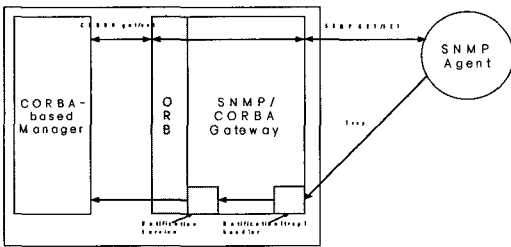


그림 6 상호 작용 변환의 기본 구조

두 도메인간의 상호 작용을 위해 게이트웨이는 3개의 기본적인 구성 요소로 구성이 된다.

- 관리할 대상의 객체를 찾는 MIB Repository
- CORBA 매니저 어플리케이션과 SNMP 에이전트 사이의 오퍼레이션을 수행하는 Request Handler
- SNMP 에이전트에서 발생하는 Trap을 CORBA 이벤트로 바꾸어 매니저에게 알려주는 Trap Management 이다.

그림 6은 이러한 구성 요소들로 구성된 게이트웨이의 기본 구조이다.

MIB repository의 경우 관리할 객체를 찾기 위해 CORBA의 naming service [16]를 이용한다. 또한 Trap management 모듈의 경우도 event service [17]를 이용함으로써 보다 쉽고 효과적으로 처리한다.

4.4.1 MIB Repository

MIB repository는 게이트웨이에서 다룰 수 있는 정적인 MIB구조를 제공한다.

이와 같은 리스트는 CORBA의 Naming service를 이용하여 구성하거나 Object Factory를 정의하여 구현할 수 있다. Object Factory는 관리해야 될 모든 객체 리스트를 가지고 있어 요구를 하게 되면 객체의 참조

포인터를 리턴한다. MIB repository는 다음의 기능을 제공한다. MIB의 그룹, 테이블 및 속성들의 리스트를 제공한다.

특정 SNMP 에이전트 접근을 위해 CORBA 도메인 내에서 MIB에 접근하기 위한 접근 생성 및 운용을 제공한다. 즉 어떤 특정 에이전트의 정보를 액세스하기 위해 CORBA 매니저는 게이트웨이에게 특정 에이전트의 MIB을 요청해야 하고, 게이트웨이는 Naming tree 또는 Object Factory를 이용하여 MIB에 접근할 수 있는 객체 참조 포인터를 생성해 주며, 매니저는 이 객체 참조 포인터를 이용하여 관리 대상이 되는 에이전트에게 요청 (Request)을 함으로써 원하는 정보를 얻어 올 수 있게 된다.

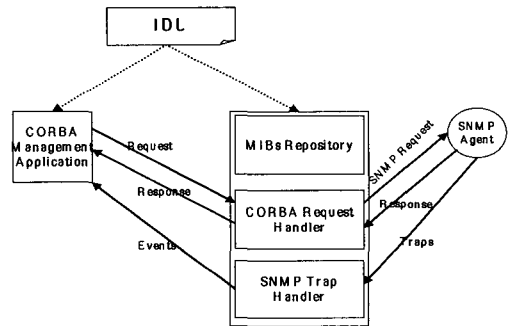


그림 7 게이트웨이의 기본 구조

4.4.2 CORBA 요청 관리

CORBA 요청 관리 부분은 CORBA 매니저로부터의 요청을 SNMP 에이전트에게 전달을 하는 부분이다. 매니저가 일단 게이트웨이의 naming service를 통해 객체 참조 포인터를 얻게 되면 매니저는 해당 객체 참조 포인터를 이용하여 SNMP 에이전트 변수에 대해 get/set 오퍼레이션을 부를 수 있고 응답을 얻을 수 있게 된다.

4.4.3 SNMP trap 관리

Trap관리 부분은 SNMP 에이전트에서 발생하는 trap정보를 CORBA의 event 서비스를 이용해 CORBA 도메인에 전달하는 부분이다. 기본적인 기능은 Event 채널에 대한 참조 포인터를 얻어 event 채널과 바인딩 후 event를 기다리게 된다.

4.5 정보 모델 변환 방법의 비교

이 절은 CORBA/SNMP또는 CORBA/CMIP의 통합 관리에 대해 많은 연구를 하고 있는 JIDM의 표준과

이 논문에서 제시하는 정보 모델 변환 방법과 비교해 보겠다.

JIDM의 Specification Translation (ST)는 잘 정의된 변환 알고리즘이나 단순히 SNMP MIB을 CORBA IDL로 변환하는 부분만을 정의하고 있기 때문에 객체 지향 속성을 반영하지 못하고 있다. 또한 SNMP의 action을 정의하는 속성에 대한 구별도 정의되지 않기 때문에 매니저는 명확한 action을 수행할 수 없다. 그러나 이 논문에서 제시하는 새로운 Super class의 정의와 action을 나타내는 속성에 대해 CORBA 메소드로 정의하는 방법을 사용함으로써 SNMP MIB을 보다 객체 지향적인 방법으로 사용할 수 있다.

공통의 속성을 묶어 새로운 클래스로 정의함으로써 MIB에 새로운 그룹 또는 Table의 추가 시 공통의 속성을 제외한 특정 속성에 대해서만 정의하고 나머지 부분은 상속받음으로써 쉽게 정보 모델을 확장 적용할 수 있다. 또한 action에 대한 부분도 CORBA 메소드로 명확히 구분짓기 때문에 사용자는 일반적인 SNMP의 관리 정보를 나타내는 속성과도 쉽게 구별지을 수 있기 때문에 명확한 action을 수행할 수 있게 한다. 표 1은 JIDM의 정보 모델 변환과의 비교 내용을 정리한 것이다.

표 1 정보 모델 변환 방법의 비교

	직접 변환과 추상적변환의 혼용	JIDM의 직접 변환 방법
객체 지향의 특성	공통의 속성을 묶어 새로운 Super class를 정의함으로써 보다 더 상속등의 객체 지향 특성을 반영	그룹이나 테이블을 객체로만 정의하기 때문에 객체 지향 기법의 어떠한 특성도 반영하지 못함.
SNMP의 Action	명확한 CORBA 메소드로 정의를 하기 때문에 메소드 호출로 Action 수행 가능	Action을 나타내는 속성이 다른 관리 속성과 구별이 되지 않는다.

## 5. 시스템 구현

앞 장에서 CORBA/SNMP 통합 관리 시스템 구현을 위한 접근 방법에 대해 설명하였다. 이 장은 위에서 정의한 접근 방법을 통해 어떻게 통합 관리 시스템을 위한 게이트웨이를 구현하였는가 설명한다.

### 5.1 구현 환경

게이트웨이를 구현하기 위한 하드웨어 플랫폼으로

Solaris 2.5를 운영 체제로 쓰는 Sun Ultra 1을 이용하였다. 게이트웨이를 구성하는 정보 모델 변환 메커니즘은 컴파일러 생성 툴인 flex와 yacc [18]을 이용하여 SmpIDL이라는 SMI를 IDL로 바꾸는 컴파일러를 구현하였다. 상호 작용 변환을 위한 모듈은 CORBA와 C++을 이용하여 구현하였다. 상호 작용 변환의 3가지 구성 요소들은 CORBA IDL로 정의하였다. 그리고 CORBA제품으로 현재 가장 널리 쓰이고 있는 IONA Orbix 2.3 [19]을 이용하였다.

### 5.2 정보 모델 변환 메커니즘

SMI 모델을 IDL 모델로 변경하고자 한다면 우선 SMI에 대한 어휘 분석과 구문 분석이 이루어져야 한다 [20]. 여기서 사용자가 정의한 정보 테이블에 구문 분석을 통해 나온 정보를 저장하고 나중에 이 테이블을 이용하여 IDL 파일을 생성하게 된다. 어휘 분석을 위해서는 flex를 사용하여 정의하였다.

또한 변환기를 통해 나온 IDL 화일에 마지막으로 사용자와의 상호 작용 부분이 추가 된다. 즉 IDL 파일을 보다 CORBA 환경에 적합하게 만들기 위한 작업으로 4.3에서 정의한 부분을 추가한다. SMI에서 정의되어 있는 속성들 중 공통적인 부분에 대해 상위 클래스를 정의한다. 그리고 implicit한 action을 명확한 CORBA 메소드로 바꾸어 준다.

### 5.3 상호 연동의 구현

3개의 구성 요소를 IDL로 정의하여 IDL 컴파일러에 의해 생성되는 C++ 코드를 이용하여 상호 작용 모듈을 구현한다. 즉 MIB repository을 위한 부분과 CORBA Request와 SNMP Trap을 처리할 수 있는 부분을 CORBA IDL을 이용하여 정의하였다.

### 5.4 게이트웨이의 이용 예

정보 모델 변환과 상호 작용 변환을 통해 구현된 게이트웨이가 어떻게 쓰이는가 알아 보도록 하겠다.

그림 7은 매니저가 어떻게 관리 객체에 대한 참조 포인터를 얻어오고 요청을 보내는지 보여준다.

우선 MoFactory에 바인더하여 객체 참조 포인터를 얻은 다음 얻고자 하는 관리 정보를 요청하면 된다. 아래의 프로그램 코드는 MIB II [21]의 system그룹의 정보를 가져오는 것으로 단계별로 코드화한 것을 보여 주고 있다.

첫번째 단계로 factory에 바인딩하여 관리 대상 객체를 생성한다. 두번째 단계는 시스템 그룹에 대한 포인터를 얻는다. 세번째 단계는 호스트 이름과 community 값을 넘겨줌으로써 에이전트에 접근한다. 마지막으로 시스템 그룹 속성들에 대한 정보를 가져온다.



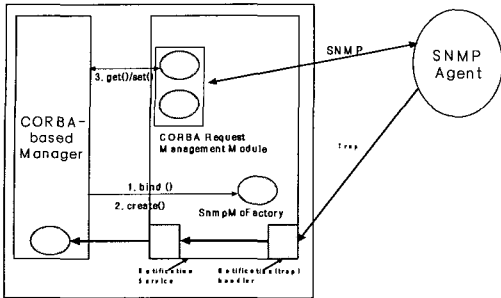


그림 8 게이트웨이의 이용 예

```

..
MoFactory_var var;
CORBA::Object* ptr;
RFC1213_MIB::system* ptr2;
char* host_name="tigris.postech.ac.kr";
char* community="public";
var = snmpMOProxyFactory::_bind();
ptr = var->create("system");
ptr2= RFC1213_MIB::system::_narrow(ptr);
ptr2->host(host_name); // assign host value
ptr2->community(community);
cout << "sysName= " << ptr2->sysName() << endl;
cout << "sysContact=" << ptr2->sysContact() << endl;
cout << "sysDescr= " << ptr2->sysDescr() << endl;
cout << "sysObjectID=" << ptr2->sysObjectID() << endl;
cout << "sysLocation=" << ptr2->sysLocation() << endl;
cout << "sysServices=" << ptr2->sysServices() << endl;
}
ptr->delete( ptr2);
..

```

7. 결론 및 향후 과제

오늘날과 같은 엔터프라이즈 환경에서는 거대하고 복잡한 네트워크를 기술적, 경제적 측면에서 효과적으로 관리하는 문제는 해결해야 할 당면 과제가 되었다.

효과적이며 안정된 관리를 위해 많은 곳에서 표준화가 진행되고 있다. 그러나 이러한 표준들은 특정 관리 영역에만 쓰이므로 다양한 표준을 양산하게 되었다. 이로 인해 날로 커져만 가는 네트워크 환경에서 다양한 기술들이 존재하게 되었고, 효율적인 관리를 위해 여러 기술을 통합하여 사용할 수 있는 관리 시스템의 요구가 나오게 되었다.

이 논문에서는 통합 관리 시스템을 구축하기 위해 요구되는 사항들을 정의해 봄으로써 통합에 필요한 요소와 접근 방법에 대해 설명하였다. 구현된 게이트웨이는

정보 모델 변환과 상호 작용 변환 두가지를 지원함으로써 통합을 위해 요구되는 사항을 만족시킨다. 또한 여러 접근 방법들의 장점을 포함하는 구현 방법을 택했기 때문에 보다 효과적이다.

JIDM의 통합 관리를 위한 정보 변환과 상호 작용 변환이 표준으로 나와있지만, 정보 모델 변환의 경우 직접 변환 방식을 채택하고 있기 때문에 CORBA 도메인 환경의 특성을 잘 나타내지 못한다.

이 논문에서는 보다 CORBA 환경에 친화적인 정보 모델 변환 방식을 사용하고, 간단하면서도 모든 기능을 표현할 수 있는 상호 작용 변환 방법을 제시함으로써 해결하였다.

그러나 이 논문에서 제시한 게이트웨이는 CORBA와 SNMP만을 고려하였기 때문에 더 보완을 해야 할 부분이 있는데, OSI의 CMIP도 관리 대상이 될 경우 또는 그 외의 proprietary한 대상들도 추가될 경우 각각의 게이트웨이에 대한 구현 방법과 서로 다른 게이트웨이를 매니저가 효과적으로 이용하게 해주는 중간층, 즉 Proxy coordinator에 대한 연구가 필요하다. 뿐만 아니라 CORBA가 아닌 non-CORBA를 이용한 구현과 비교를 위한 prototype의 정의가 필요하다. 또한 사용자 인터페이스 구현을 위한 정형화된 방법도 찾아야 할 것이다.

참고 문헌

- [1] 박성욱, "웹 기반의 엔터프라이즈 네트워크 트래픽 모니터링 및 분석", Masters Thesis, POSTECH, GSIT,1998.
- [2] ITU-T Recommendation M.3100: Principles for a Telecommunications Management Network, International standard, 1992.
- [3] W, Stallings,"SNMP, SNMPv2 and RMON," 2nd Edition, Addison-Wesley, 1996.
- [4] OMG. The Common Object Request Broker: Architecture and Specification Revision 2.0. OMG, July 1995. OMG TC Document.
- [5] Joint-Inter-Domain Working Group, X/Open and Network Management Forum, "CORBA/TMN Interworking," May 1998.
- [6] Open Group, "Inter-Domain Management: Specification Translation-intro," <http://www.opengroup.org/onlinepubs/8349099/chap01.html>,1997.
- [7] Joint-Inter-Domain Working Group, X/Open and Network Management Forum, "CORBa/TMN Interworking SNMP Part," May, 1998.
- [8] ISO/IEC, CCITT, Specification of Abstract Syntax Notation One (ASN.1), ISO/IEC 8824, CCITT Recommendation X.208, 1998.

- [9] International Standards Organization, Information Technology-Open Systems Interconnection-Guidelines for the Definition of Managed Objects (GDMO), ISO/IEC IS 10165-4, June 1992.
- [10] Smile Company, <http://www.orbycom.fr/products.html>.
- [11] UH Communications, <http://www.login.dknet.dk/uh/products.html>.
- [12] Subrata Mazumdar, "Inter-Domain Management between CORBA and SNMP" DSOM'96, L'Aquila, Italy, October 1996.
- [13] Qinzhen Kong, Graham Chen, "Integrating CORBA and TMN Environments" CiTR Technical Journal-Volume 1, [http://www.citr.com/02.TechnicalJournal/01.Volume\\_1/01.Papers/TechnicalJournal.html](http://www.citr.com/02.TechnicalJournal/01.Volume_1/01.Papers/TechnicalJournal.html).
- [14] IBM SystemView Agent, <http://www.support.tivoli.com/sva/shaover.html>.
- [15] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, and William Lorenson. *Object-Oriented Modeling and Design* Prentice-Hall, 1991.
- [16] OMG, CORBA service: Naming service Specification, <http://www.omg.org/corba/sectrans.htm>, 1995.
- [17] OMG, CORBA service: Event service Specification, <http://www.omg.org/corba/sectrans.htm>, 1995.
- [18] 김성환, 김영달, 김창현, 김휘강, "UNIX 시스템 툴 잘쓰기", PC어드밴스, pp 548-612, May 1995.
- [19] IONA, Orbix 2, IONA Technologies Ltd., Release 2.0.
- [20] Michael Sample, "Snacc 1.2: A High Performance ASN.1 to C/C++/IDL Compiler," February 1993. <http://www.fokus.gmd.de/ovma/freeware/snacc/entry.html>
- [21] K. McCloghrie, M. Rose, Management Information Base for Network Management of TCP/IP-based internets: MIB-II, RFC1213, March 1991.



홍 원 기

1983년 Univ. of Western Ontario 전산학 학사 졸업. 1985년 Univ. of Western Ontario 전산학 석사 졸업. 1991년 Univ. of Waterloo 전산학 박사 졸업. 1991년 ~ 1992년 Univ. of Waterloo Post-Doc Fellow. 1992년 ~ 1995년 Univ. of Western Ontario 연구교수. 1995년 ~ 현재 포항공대 컴퓨터공학과 부교수. 관심분야는 네트워크 및 시스템 관리, CORBA, 분산멀티미디어 시스템



강 영 민

1997년 숭실대학교 전자계산학 학사 졸업. 1999년 포항공과대학교 정보통신대학원 석사 졸업. 1999년 ~ 현재 제일정보통신 기술연구소 근무. 관심분야는 분산처리 및 네트워크 관리