SNMP 게이트웨이를 이용한 WBEM 기반의 통합관리 시스템

[1] 이소정, [1] 유선미, [2] 주홍택, [1] 홍원기, [3] 안창원

[1] 포항공과대학교 컴퓨터공학과

[2] 계명대학교 컴퓨터공학과

[3] 한국전자통신연구원 디지털 홈 연구단

{annie, sunny81, jwkhong}@ postech.ac.kr, juht@kmu.ac.kr, ahn@etri.re.kr

요 약

WBEM은 DMTF에 의해서 표준화되고 있는 네트워크 및 시스템 관리 기술 중에 하나로서 현재 산업계의 각광을 받고 있다. 한편 이미 설치되었거나 설치되고 있는 많은 장치가 SNMP만을 지원하고 있는 것도 현실이다. WBEM이 통합관리에 초점을 두고 있는 만큼 SNMP라는 서로 다른 도메인을 통합시킬 필요가 있다. 본 논문에서는 SNMP 게이트웨이에 의한 WBEM기반 통합 네트워크 관리 시스템을 제안한다. 먼저 WBEM/SNMP Gateway를 위한 요구사항을 분석하고, 그것을 충족시키는 WBEM기반 관리 시스템 구조를 제시하고 이 구조의 핵심 요소인 WBEM/SNMP Gateway 그리고 SNMP Provider를 설계결과를 제시한다. WBEM/SNMP Gateway는 관리 정보 모델 변환과 관리 동작 변환이라는 두가지 메커니즘을 제공한다. 마지막으로 유효성 검증을 위해 구현한 WBEM/SNMP Gateway와 WBEM Manager의 프로토타입을 설명한다.

1. 서론

거대해지고 복잡해지는 네트워크 관리의 필요성이 높아짐에 따라, 다양한 단체에서 표준화 작업이 이루어지고 있다. 그 중에서도 DMTF (Distributed Management Task Force)[1]는 기업과 인터넷 환경을 위한 관리 표준을 개발하고 채택하여 상호운영성을 주도하는 산업계 표준화 기관이다. 현재 DMTF에서는 시스템 및 네크워크 관리와 관련하여 CIM(Common Information Model)[2], WBEM(Web-Based Enterprise Management)[3] 등의 기술을 표준화하고 있다.

WBEM은 여러 회사에서 제작한 시스템, 네트워크 장비, 어플리케이션 등을 관리하기 위한 표준이다. WBEM은 CIM으로 관리 정보를 정의하고, CIM으로 정의된 관리 정보를 CIM-XML[4]로 Encoding하여 HTTP[5]를 사용하여 전송한다. 현재 Microsoft나 SUN의 서버 시스템 그리고 Cisco의 Network 장비들이 WBEM 에이전트를 탑재하고 있는 것을 보면 WBEM은 더욱 확산될 것이라 예상된다. 그러나 현재까지는 대부분의 장비가 SNMP[6] Agent를 탑재하고 있었으므로, 서로 다른 Domain간에 통합 관리의 필요성은 매우 높다.

WBEM으로 통합 관리하기 위해서는 WBEM/SNMP Gateway가 필요하다. 그러나 현재까지는 WBEM/SNMP Gateway의 open source가 제공되지 않는 실정이며, 단지 SNMP Provider가 Microsoft, SUN에서 상업용으로만 제공되고 있다. 그러므로 현재 대학이나 공공 연구소 같은 곳에서 WBEM와 SNMP의 통합에 대한 연구에 어려움이 있다. 그러므로 본 논문에서는 WBEM/SNMP Gateway를 설계하고 프로토타입을 개발한 결과를 제시한다. 또한 WBEM/SNMP Gateway는 서로 다른 관리 도메인 사이에서 상호 운영되어야 하므로, MIB을 MOF로 바꾸는 관리 정보 모델 변환(Specification Translation)방법과 CIM Operation을 SNMP Operation으로 바꾸는 관리 동작 변환 (Interaction Translation)방법을 고안한 결과도 제공한다.

본 논문의 구성은 다음과 같다. 2 장의 관련 연구에서는 WBEM 표준 기술과 현재 제공되고 있는 SNMP Provider들 그리고 open source 구현물 중의 하나인 Pegasus에 대해 소개한다. 3 장에서는 기존의 WBEM 구현물들에 대해 Benchmarking Test한 결과를 보여준다. 또한 4 장에서는 WBEM/SNMP Gateway를 위한 요구사항을 분석하고 구조를 설계하며 각각의 Specification Translation 방법과 Interaction Translation 방법에 대해 설명한다. 5 장에서는 프로토타입 구현을 설명하며, 6 장에서 결론과 앞으로의 연구방향을 제시함로써 본 논문을 마무리한다.

2. 관련연구

이 장에서는 본 논문의 기술적 토대인 WBEM의 핵심 기술들과 현재 Microsoft와 SUN에서 제공하는 SNMP Provider들에 대해 소개한다.

2.1 WBEM 개요

이 장에서는 본 논문의 바탕인 WBEM 핵심 기술들에 대해서 소개하고, 현재 Microsoft WMI와 SUN Solaris에서 제공하는 SNMP Provider들에 대해 간략히 설명한다.

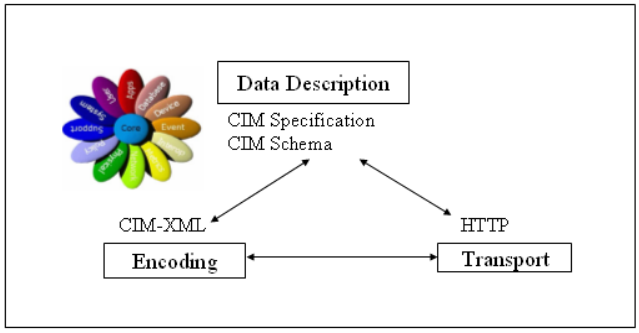

그림 1. WBEM 기술들간의 관계

그림 1에서 묘사되어 있듯이 WBEM은 세가지 기술로 구성된다. 이 세가지 기술은 시스템 관리 정보에 대한 표준 관리 데이터 모델인 CIM(Common Information Model), 데이터 Encoding 기법으로 사용되는 CIM-XML 그리고 관리 명령과 전송 프로토콜로 사용되는 HTTP이다. DMTF는 CIM Schema를 Management Object Format(MOF)[7], XML 그리고 UML[15]의 세 가지로 표현한다.

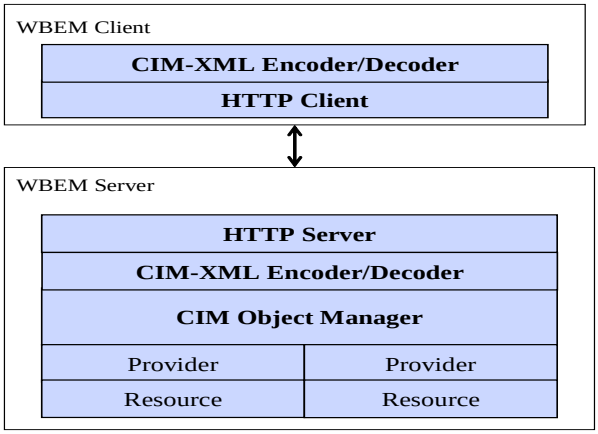

그림 2. WBEM 기반 관리 시스템 구조

그림 2는 WBEM 기반 관리 시스템 구조이다. WBEM 기반 관리 시스템은 WBEM Client와 WBEM Server를 포함한다. WBEM Client는 HTTP Client를 통해 WBEM Server에 연결하며, CIM-XML Encoder/Decoder를 통해 요청할 메시지를 만들거나 응답되어온 Message를 parsing한다.

마찬가지로 WBEM Server는 HTTP Server를 통해 메시지를 주고 받으며, CIM-XML Encoder/Decoder는 요청된 CIM-XML Message를 parsing하여 수행해야 할 CIM Operation를 걸러내고 다시 응답메시지를 만들 수 있다. CIMOM(CIM Object Manager)은 각 관리 정보에 대한 CIM Operation을 적절한 Provider에게 forwarding하고, Provider에서 실제로 관리 자원으로부터 필요한 정보를 가져온다.

2.2 현재 제공되는 SNMP Provider 소개

2.2.1 Microsoft WMI SNMP Provider

WMI[8]는 Microsoft Windows 운영체제에 탑재되는 Web-based Enterprise Management System이며, WBEM 표준에 적합한 WBEM Server이다.

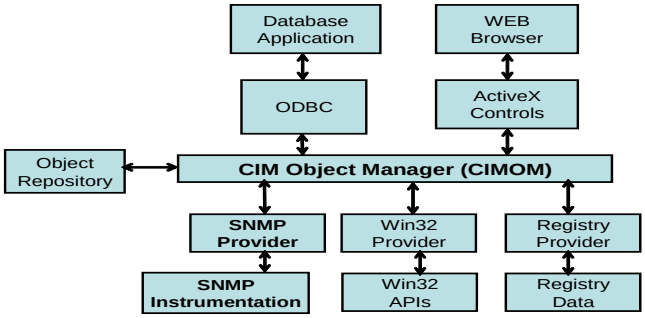

그림 3. WMI의 구조

그림 3은 WMI의 구조를 보여준다 WMI는 CIMOM과 Object Repository를 가지고 있고, Database Application이나 WEB Browser 등과 같은 형태로 CIMOM과 통신할 수 있다. Win32 Provider는 Win32 환경에서 운영체제의 구성 요소들을 관리하며, 마찬가지로 Registry Provider는 Registry data에 접근하여 값을 CIMOM에게 보내주고 받을 수 있다.

WMI는 SNMP MIB Object variable들을 읽거나 쓸 수 있는 SNMP Provider를 포함한다. SNMP Provider는 각각의 object들의 값을 CIM Class instances들의 property 값들에 Mapping시킬 수 있다. 또한, SNMP trap은 자동적으로 WMI event에 Mapping될 수 있다.

WMI는 smi2smir라는 utility를 포함한다. 변환된 결과는 SNMP Module Information Repository (SMIR)라고 불리는 SNMP schema database에 넣어질 수 있다. 이때 MIB 파일들은 NOTIFICATION-TYPE Macro, OBJECT-TYPE Macro, TEXTUAL-CONVENTION Macro, Trap-TYPE Macro와 같은 매크로들을 통해 MIB으로부터 MOF로의 Mapping 방법을 결정한다. 그 중에서도 OBJECT-TYPE macro는 MIB object의 기본적인 특징들을 나타내는데 필수적인 조항들을 포함한다.

Microsoft WMI의 SNMP Provider의 단점은 MIB에서 MOF로의 변환 결과가 WMI에 의존적이라는 것이다. ToInstance와 같은 flavor는 다른 WBEM 구현물에는 선언되어 있지 않은 새로운 flavor이므로 다른 WBEM 사용하려면 MOF를 또 수정해야 하는 불편함이 있다. 또한

WMI SNMP Provider의 Interaction Translation 기능을 위한 source code가 공개되어 있지 않다. 무엇보다도 WMI의 SNMP Provider는 Windows 기반의 OS에서만 제공이 된다는 한계가 있다.

2.2.2 SUN Solaris의 SNMP Provider

SUN은 Solaris Operating Environment 용 WBEM 구현물인 Solaris WBEM Services[9] 및 SUN WBEM SDK[10][11]를 제공한다.

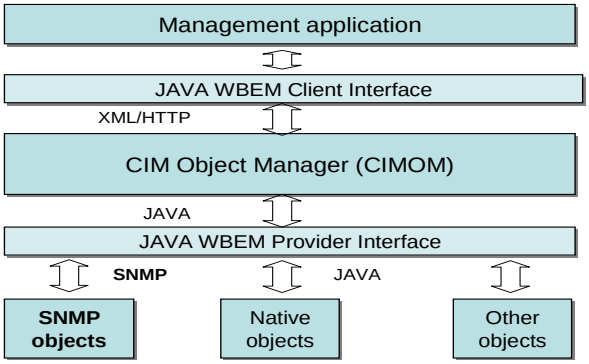

그림 4. WBEM Services의 구조

그림 4는 WBEM Services의 구조를 보여준다. WBEM Management application에서는 JAVA Client API를 통해 CIMOM으로 접근할 수 있다. 또한, CIMOM에서는 JAVA Provider API들을 통해 SNMP Object, Native Object 등의 다양한 관리 객체들로부터 정보를 가져오거나 바꿀 수 있다.

SUN Solaris에서는 SNMP Object들을 관리하기 위해 SNMP Provider가 제공되며, MIB파일을 MOF파일로 생성시켜줄 수 있는 MIB2MOF utility가 포함된다. 이때, MOF file은 각각의 SNMP group들을 CIM class로 표현한다. MIB2MOF는 기본적으로 DMTF에서 표준화된 Qualifier들을 사용하여 DMTF의 MOF 파일들과 매우 유사하다.

SUM Solaris SNMP Provider의 단점 또한 Interaction Translation을 위한 source code가 공개되어 있지 않으며 SNMP Provider에 대한 충분한 문서가 제공되지 않는 것이다. 또한 기본적으로 Solaris SNMP Provider를 사용하려면 Solaris 기반의 OS이어야 한다는 한계가 있다.

2.3 OpenPegasus 소개

본 연구에서는 Pegasus[12]를 확장하여 WBEM/SNMP Gateway 프로토타입을 구현하였다. Pegasus는 앞에서 설명한 Microsoft WMI나 SUN WBEM과는 달리 DMTF의 WBEM 기술인 CIM과 CIM-XML 표준에 따라 구현한 open source이다. Pegasus는 TOG(The Open Group)가 주도적으로 이끌어가고 있는 프로젝트이며, 코드와 문서들을 모두 자유롭게 이용할 수 있고 현재 기업체들로부터 가장 많은 관심을 받고 있다.

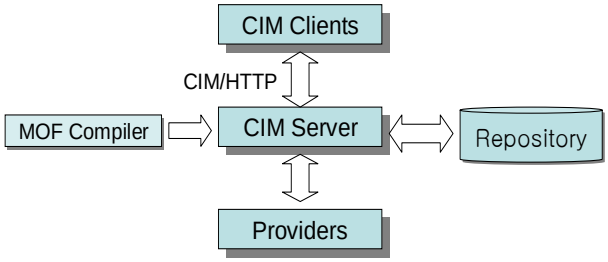

그림 5. Pegasus의 구조

그림 5 는 Pegasus의 구조를 나타낸다. Pegasus는 WBEM 표준에 따라 CIM Client와 CIM Server 사이에서 CIM/HTTP로 통신한다. 또한 CIM Server는 Repository내의 CIM schema들에게 접근할 수 있으며, Client가 알고자 하는 관리 정보에 대한 Provider를 호출할 수 있다. Pegasus는 MOF Compiler 기능을 하는 cimmof utility를 가지고 있다. Pegasus는 C++로 구현되어 있으며, repository가 XML[13]로 구현되어 있다.

3. WBEM 구현물 비교

본 연구에서는 기존의 WBEM 구현물을 확장하여 WBEM/SNMP Gateway 프로토타입을 구현하였다. WBEM 구현물들 중 우수한 하나를 선택하기 위하여 Benchmarking 테스트를 실시하였고 본 장에서는 이 결과를 제시한다. Pegasus와 WBEM Services는 개발자들이 가장 많이 활용하고 있는 open source들이다. 이 두 WBEM 구현물에 대해 Pentium 4, 1.6GHz, 256MB RAM인 Linux 환경에서 다음과 같은 Benchmarking 테스트를 실시하였다.

Operation	# of classes	100	200	400	800	1600
Enumerate Instances	WBEM	377	342	372	359	348
	Pegasus	181	183	184	185	315
Enumerate Instances Names	WBEM	222	222	224	227	244
	Pegasus	41	41	41	44	48
Create Instance	WBEM	113	114	114	114	115
	Pegasus	15	16	17	18	24

표 1. 첫번째 벤치마킹 테스트 결과 (시간 단위: ms)

표 1은 Repository 안에 들어간 Class들의 수를 100, 200, 400, 1600 으로 증가시키면서 EnumerateInstances, Enumerate InstancesNames, CreateInstance operation을 수행했을 때의 수행 시간을 측정한 결과이다. EnumerateInstances를 수행했을 경우에는 WBEM이 Pegasus보다 1~2 배 이상 수행 시간이 길며, EnumerateInstancesNames와

CreateInstance의 경우에는 WBEM이 Pegasus보다 5~6 배 이상 수행 시간이 더 걸리는 것을 확인하였다.

Operation	# of instances/ class	100	500
EnumerateInstances	WBEM	377	1670
	Pegasus	181	863
EnumerateInstances Names	WBEM	222	1069
	Pegasus	41	185
CreateInstance	WBEM	113	121
	Pegasus	15	31

표 2. 두번째 벤치마킹 테스트 결과 (시간 단위: ms)

표 2는 class 당 instance들의 수를 100 에서 500 으로 증가시키면서 EnumerateInstances, EnumerateInstancesNames, CreateInstance를 수행했을 때의 latency를 측정한 결과이다. 이 때, class들의 수는 1600 개이다. 각 class당 instance의 수가 100 개인 경우에는 WBEM의 수행 시간이 Pegasus보다 길다. 또한 각 class당 instance들의 수가 500 개로 늘어났을 경우에 WBEM 수행 시간의 증가률은 Pegasus의 수행 시간 증가율보다 크다.

결국 이 두 가지 Benchmarking 테스트를 통해 class나 instance의 수를 늘였을 때 WBEM Services보다 Pegasus의 성능이 더 좋음을 알 수 있다. WBEM/SNMP Gateway는 다수의 SNMP Agent를 관리하기 때문에 Benchmarking 테스트는 Scalability에 중점을 두었으며, Pegasus를 확장하여 WBEM/SNMP Gateway 프로토타입을 구현한다.

4. WBEM/SNMP Gateway 설계

이 장에서는 WBEM 기반 전체 서버 관리 시스템 구조, WBEM/SNMP Gateway의 구조 그리고 SNMP Provider 구조를 설명한다. 또한 WBEM/SNMP Gateway의 관리 정보 변환, 관리 동작 변환 방법을 제시한다.

4.1 요구사항 분석

- 관리 정보는 CIM Schema로 정의되고, XML로 Encoding된 후, HTTP payload에 Embedded되어 전송된다.
- WBEM/SNMP Gateway는 기존에 사용되어었던 SNMP device를 통합하기 위한 기능을 제공해야 한다.
- WBEM/SNMP Gateway는 에러가 발생했을 때, Trap 정보를 export할 수 있는 메커니즘을 제공해야 한다.
- 전체 관리 시스템에서 보안 메커니즘을 제공해야 한다.
- WBEM/SNMP Gateway는 Scalability와 Flexibility를 제공해야 한다.
- 관리자가 언제 어디서나 관리가 가능한 Web-based User Interface를 제공해야 한다.

4.2 기능 구조 설계

4.2.1 전체 관리 시스템 구조

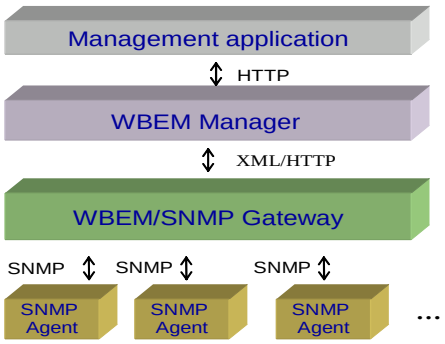

그림 6. WBEM기반 전체 관리 시스템 구조

그림 6은 WBEM 기반 서버 관리 시스템의 전체 구조이다. 기존의 Manager-Agent 구조를 따르며, WBEM/SNMP Gateway가 이들 사이의 상호운영성을 제공한다. WBEM Manager는 WBEM Client, WBEM/SNMP Gateway는 WBEM Server에 해당된다. 관리자들은 관리 어플리케이션을 통해 WBEM Manager에 접근하며, WBEM Manager와 WBEM/SNMP Gateway는 XML/HTTP를 통해, WBEM/SNMP Gateway는 SNMP를 통해 SNMP Agent를 관리할 수 있다.

4.2.2 WBEM/SNMP Gateway 구조

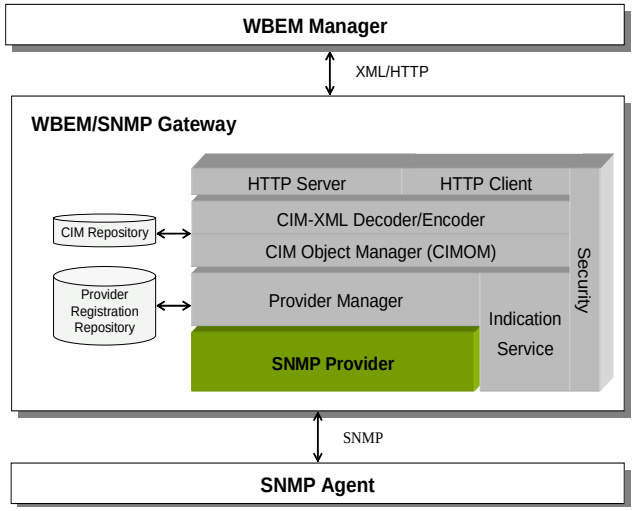

그림 7. WBEM/SNMP Gateway 구조

그림 7은 WBEM/SNMP Gateway의 구조이다. WBEM/SNMP Gateway내의 HTTP Server는 WBEM Manager로부터 메시지를 받으며, HTTP Client를 통해 trap 정보를 Manager에게 보낼 수 있다. WBEN/CIM

Encoder/Decoder는 받은 메시지를 변환한 후, 이를 CIMOM에게 전달한다. 그러면 CIMOM은 적절한 provider를 통해 CIM operation을 결과값이 다시 WBEM Manager에게 반환된다.

Provider Manager는 다양한 Provider들을 Provider Registration Repository에 저장함으로써 관리한다. 이들은 Instance Provider, Method Provider, Association Provider 그리고 Indication Provider를 포함한다.

SNMP Provider는 다른 Provider들과 마찬가지로 Provider의 등록 절차를 거쳐 서비스가 제공되며, CIM operation들을 SNMP Operation으로 변환함으로써 Interaction translation을 수행할 수 있다. Indication Service는 SNMP Trap를 받기 위해 가입한 SNMP Agent들의 리스트를 생성하고 변경 삭제하는 일을 수행한다.

또한 보안적인 이슈로써 manager와 agent 사이에 HTTP over SSL(HTTPS)[16]를 사용하면 보다 안전한 통신을 할 수 있으며, WBEM Client가 사용자 ID와 패스워드를 전달함으로써 사용자 인증을 할 수 있다. 또한 사용자는namespace를 접근하기 위한 읽기/쓰기 권한을 필요로 할 수 있다.

4.2.3 SNMP Provider의 기능과 구조

그림 8은 SNMP Provider의 두 가지 기능을 나타낸다. Specification Translation은 MIB 파일을 MOF 파일로 바꾸는 것이며, Interaction Translation은 CIM Operation을 SNMP Operation으로 변환하는 것이다.

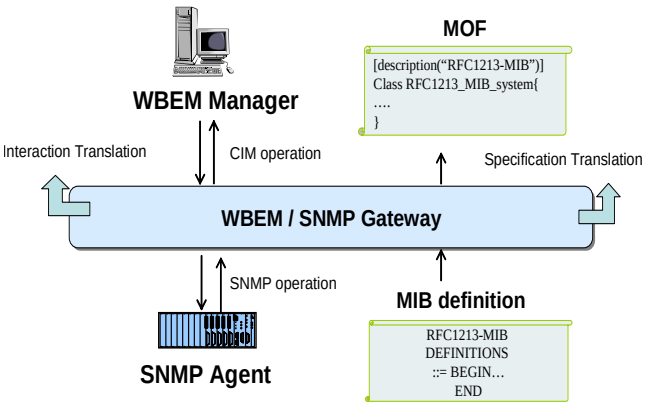

그림 8 SNMP Provider의 기능

그림 9는 SNMP Provider의 구조이다. WBEM/SNMP Gateway는 MIB으로 정의된 관리 정보를 MOF로 변환시켜줄 수 있는 MIB2MOF Translator를 포함한다. 변환된 결과로써 나온 MOF 파일은 MOF Compiler를 통해서 Repository에 등록된다. 또한 WBEM/SNMP Gateway는 CIM operation을 SNMP operation으로 시켜줄 수 있는 SNMP Provider를 포함한다. SNMP Provider는 CIMOM으로부터 들어온 CIM Operation들을 적절한 CIM Request Handler를 통해 처리되게끔 할 수 있다. CIM Request Handler는 각 CIM Operation에 Mapping되는 SNMP Operation을 호출하고, SNMP Stack에서 직접SNMP Agent에 대해 SNMP Request를 보낸다. SNMP Agent는 해당하는 MIB variable을 WBEM/SNMP Gateway에게 보내고 그 값을 다시 WBEM Manager에서 확인할 수 있다.

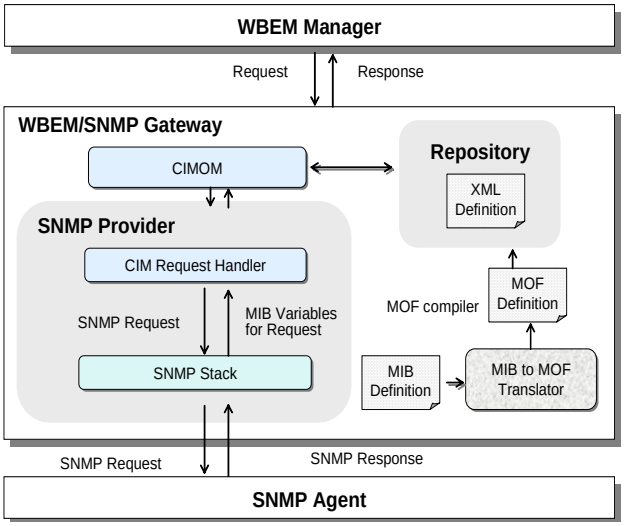

그림 9. SNMP Provider의 구조

4.3 관리 정보 모델 변환

Specification Translation을 위해 SNMP SMI[17] 데이터타입은 해당되는 CIM 데이터타입으로 변환된다. 표 3과 같이 SNMP MIB의 SYNAX 정보인 OBJECT 데이터 타입은 CIM 데이터 타입으로 Mapping된다.

SNMP SMI Datatype	SNMP Ver.	CIM Datatype
INTEGER	v1	Sint32
OCTET STRING	v1	String
OBJECT IDENTIFIER	v1	String
IpAddress	v1	String
Counter	v1	Unit32
Gauge	v1	Unit32
TimeTicks	v1	Unit32
Opaque	v1	Sint8[ ]
DisplayString	v1	String
NetworkAddress	v1	String
Counter32	v2	Unit32
Counter64	v2	Unit64
Integer32	v2	Sint32
Gauge32	v2	Unit32
Unsigned32	v2	Unit32
TruthValue	v2	Sint32
BITS	v2	String

표 3. 데이타타입 매핑 방법

표 4에서 보여주듯이, SNMP MIB의 ACCESS, STATUS, DESCRIPTION 정보는 DMTF에서

표준화한 CIM specification의 qualifier를 통해 선언한다.

OBJECT TYPE MACRO		CIM Qualifier
ACCESS	Read – Only	Read
	Read – Write	Read, Write
	Write-only	Write
	Not-accessible	(write nothing)
STATUS	Mandatory	Required
	Optional	(write nothing)
	Obsolete	(write nothing)
	Deprecated	(write nothing)
DESCRIPTION		Description

표 4. OBJECT TYPE MACRO와 CIM Qualifier의 매핑 방법

본 연구에서는 현재 DMTF에서 제공되고 CIM Class들 중 하나를 상속을 받아 새롭게 Extended class를 정의함으로써 SNMP object관리 정보의 계층 구조를 정의한다. 그림 10과 같이, RFC1213-MIB의 system group, interface group은 새롭게 정의한 mib-2 클래스를 상속 받았으며, ManagedElement클래스의 자식 클래스인 ManagedSystemElement 클래스로부터 다시 상속 받는다.

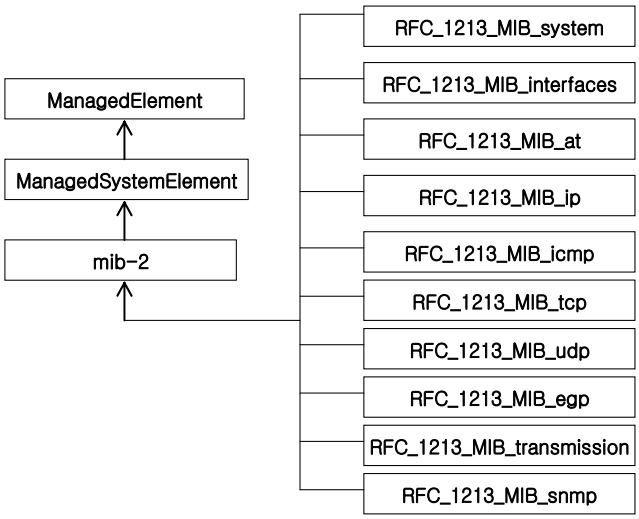

그림 10. RFC1213-MIB을 위한 CIM Class들간 상속 관계

그림 11은 RFC1213-MIB의 system group내용을, 그림 12는 interface group의 내용을 MOF로 변환한 것이다. 각 SNMP MIB 트리의 group node들은 CIM 클래스 이름으로, 각 leaf node들은 클래스의 속성으로 들어간다. 또한, SNMP MIB의 데이터타입과 OBJECT TYPE MACRO는 CIM의 데이터타입과 Qualifier로 대체된다.

```
[description("RFC1213-MIB")]
class mib_2:CIM_ManagedSystemElement{
 [Key, Read, Description("."), Required]  string sysIPaddress;
};

[description("RFC1213-MIB"), group_objectid("1.3.6.1.2.1.1")]
class SNMP_RFC1213_MIB_system : mib_2{

      [object_identifier("1.3.6.1.2.1.1.1"), Read, Description("."), Required]
        string sysDescr;
      [object_identifier("1.3.6.1.2.1.1.2"), Read, Description("."), Required]
        string sysObjectID;
      [object_identifier("1.3.6.1.2.1.1.3"), Read, Description("."), Required]
        uint32 sysUpTime;
      [object_identifier("1.3.6.1.2.1.1.4"), Read, Write, Description("."), Required]
        string sysContact;
      [object_identifier("1.3.6.1.2.1.1.5"), Read, Write, Description("."), Required]
        string sysName;
      [object_identifier("1.3.6.1.2.1.1.6"), Read, Write, Description("."), Required]
        string sysLocation;
      [object_identifier("1.3.6.1.2.1.1.7"), Read, Write, Description("."), Required]
        sint32 sysServices;
};
```

그림 11 RFC1213-MIB system group의MOF 정의

```
class SNMP_RFC1213_MIB_interfaces : mib_2{
 [object_identifier("1.3.6.1.2.1.2.1"), Read, Write, Description("."), Required]   sint32 ifNumber;
};

[description("RFC1213-MIB"), module_name("RFC1213-MIB"), group_objectid("1.3.6.1.2.1.2")]
class SNMP_RFC1213_MIB_ifTable : SNMP_RFC1213_MIB_interfaces{

 [object_identifier("1.3.6.1.2.1.2.2.1.1"), Read, Description("."), Required]    sint32 ifIndex;
 [object_identifier("1.3.6.1.2.1.2.2.1.2"), Read, Description("."), Required]    string ifDescr;
 [object_identifier("1.3.6.1.2.1.2.2.1.3"), Read, Description("."), Required]    string ifType;
 [object_identifier("1.3.6.1.2.1.2.2.1.4"), Read, Description("."), Required]    sint32 ifMtu;
 [object_identifier("1.3.6.1.2.1.2.2.1.5"), Read, Description("."), Required]    uint32 ifSpeed;
 [object_identifier("1.3.6.1.2.1.2.2.1.6"), Read, Description("."), Required]    string ifPhysAddress;
 [object_identifier("1.3.6.1.2.1.2.2.1.7"), Read, Description("."), Required]    sint32 ifAdminStatus;
 [object_identifier("1.3.6.1.2.1.2.2.1.8"), Read, Description("."), Required]    sint32 ifOperStatus;
 [object_identifier("1.3.6.1.2.1.2.2.1.9"), Read, Description("."), Required]    uint32 ifLastChange;
 [object_identifier("1.3.6.1.2.1.2.2.1.10"), Read, Description("."), Required]   uint32 ifInOctets;
 [object_identifier("1.3.6.1.2.1.2.2.1.11"), Read, Description("."), Required]   uint32 ifInUcastPkts
 [object_identifier("1.3.6.1.2.1.2.2.1.12"), Read, Description("."), Required]   uint32 ifInNUcastPkts;
 [object_identifier("1.3.6.1.2.1.2.2.1.13"), Read, Description("."), Required]   uint32 ifInDiscards;
 [object_identifier("1.3.6.1.2.1.2.2.1.14"), Read, Description("."), Required]   uint32 ifInErrors;
 [object_identifier("1.3.6.1.2.1.2.2.1.15"), Read, Description("."), Required]   uint32 ifInUnknownProtos;
 [object_identifier("1.3.6.1.2.1.2.2.1.16"), Read, Description("."), Required]   uint32 ifOutOctets;
 [object_identifier("1.3.6.1.2.1.2.2.1.17"), Read, Description("."), Required]   uint32 ifOutUcastPkts;
 [object_identifier("1.3.6.1.2.1.2.2.1.18"), Read, Description("."), Required]   uint32 ifOutNUcastPkts;
 [object_identifier("1.3.6.1.2.1.2.2.1.19"), Read, Description("."), Required]   uint32 ifOutDiscards;
 [object_identifier("1.3.6.1.2.1.2.2.1.20"), Read, Description("."), Required]   uint32 ifOutErrors;
 [object_identifier("1.3.6.1.2.1.2.2.1.21"), Read, Description("."), Required]   uint32 ifOutQLen;
 [object_identifier("1.3.6.1.2.1.2.2.1.22"), Read, Description("."), Required]   string ifSpecific;
};
```

그림 12. RFC1213-MIB interface group의 MOF 정의

본 논문에서 정의한 MOF는 DMTF 표준에 따라서 정의를 하였으므로 벤더나 장비에 의존적인 한 WBEM 구현물에서만 사용되는 것이 아니라 WBEM 표준을 따르는 모든 구현물에서 사용될 수 있다.

4.4. 관리 동작 변환

SNMP Agent들을 관리하기 위해서는 CIM Operation들을 SNMP Operation들로 변환할 수 있는 Interaction Translation 방법이 필요하다. 표 5는 CIM Operation을 각각 SNMP의 operation으로 Mapping하는 방법이다.

기본적인 읽기를 위한 EnumerateInstances, GetInstance 그리고 GetProperty는 MIB tree의 leaf node인 property정보를 가져온다. EnumerateInstance, GetInstance 그리고 GetProperty Operation의 차이는 다음과 같다. EnumerateInstance의 경우에는 생성된 모든 Instance들에 대한 property정보를 가져온다. 그러므로 SNMP Agent 당 하나의 instance를 생성한다고 할 때, 이 operation으로 여러 agent들의 관리 정보를 가져올 수 있다. 그러나 GetInstance의

경우에는 특정한 키를 포함한 오직 하나의 Instance에 대한 정보를 가져온다. 키 값이 IP주소라고 하면, 하나의 SNMP Agent에 대한 관리 정보만 가져올 수 있다. 마지막으로 GetProperty는 하나의 Instance로부터 하나의 property 정보를 가져온다. 그러므로 EnumerateInstances operation은 SNMP Get, GetNext operation으로 변환되고, GetInstance, GetProperty operation은 SNMP Get operation으로 변환된다.

기능	CIM Operation	SNMP Operation
기본적인 읽기	GetClass, EnumerateClasses, EnumerateClassNames, GetInstance, EnumerateInstances, EnumerateInstanceNames, GetProperty	SNMP Get, SNMP GetNext
기본적인 변경	SetProperty	SNMP Set
스키마 조작	CreateClass, ModifyClass, DeleteClass	없음
인스턴스 조작	CreateInstance, ModifyInstance, DeleteInstance	SNMP Set
Association 순회	Associators, AssociatorNames, References, ReferenceNames	없음
Qualifier 선언	GetQualifier, SetQualifier, DeleteQualifier, EnumerateQualifier	없음
질의	ExecQuery	없음

표 5. Interaction Translation을 위한 매핑 방법

인스턴스 조작이나 기본적인 변경을 위한 ModifyInstance와 SetProperty operation은 MIB tree의 leaf node에 해당하는 property 값을 바꿀 수 있다. 그러므로 이 operation들은 SNMP Set Operation으로 변환된다.

그러나 기본적인 읽기 기능을 하는 것들 중 GetClass, EnumerateClasses, EnumerateClassNames, EnumerateInstanceNames 그리고 기본적인 스키마 조작 기능을 하는 CreateClass, ModifyClass, DeleteClass의 경우에는 SNMP Agent에 접근하지 않으므로 SNMP Operation으로 변환할 필요가 없다. 또한 Association 순회와 Qualifier 선언 또는 질의를 하는 경우에는 해당되는 SNMP Operation이 없으므로 대체되지 않는다.

5. 프로토타입 구현

이 장에서는 앞에서 설계한 WBEM/SNMP Gateway와 WBEM Manager의 프로토타입 구현에 대해 설명한다.

5.1 WBEM/SNMP Gateway 구현

Specification Translation을 위해서 우리는 MIB2MOF Translator를 구현하였다. 이것을 통해 MIB으로 정의된 모든 관리 정보는 MOF로 변환될 수 있다. 그러면 MOF compiler를 통해 MOF파일을 compile하여 repository에 관리 정보를 등록할 수 있다.

Interaction Translation을 위해서는, SNMP Provider의 backend interface를 구현해야 한다. WBEM/SNMP Gateway 포함되어 있는 SNMP Provider는 NET-SNMP[14] API를 통해 RFC1213-MIB의 System Group 정보를 가져오거나 변경할 수 있다. SNMP Provider는 Instance Provider이며, Instance Provider에 해당하는 function이 구현된다.

5.2 WBEM Manager 구현

WBEM Manager 을 위해 사용자가 언제어디서나 접속할 수 있는 Web User Interface를 구현하였다.

사용자는 화면에 보여지는 메뉴를 통해 원하는 CIM operation을 수행할 수 있다. 즉, 사용자는 화면에서 보여지는 입력창을 통해 관리하고자 하는 Agent의 IP Address, 속성이름 혹은 변경하고자 하는 속성값을 파라미터로 원하는 Operation을 수행할 수 있다. 예를 들어, GetProperty를 수행하기 위해서는 Agent IP Address를 입력하고 왼쪽 트리에서 가져오고자하는 속성이름을 선택해야 하며, SetProperty를 수행하기 위해서는 추가로 변경하고자 하는 속성값 또한 입력해야 한다.

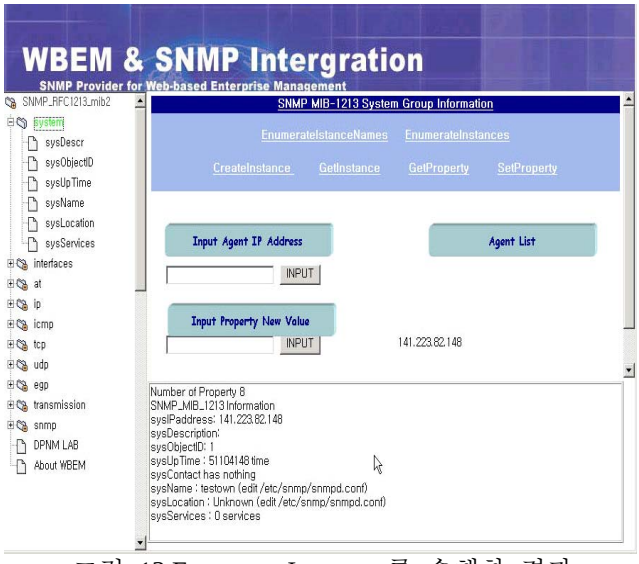

그림 13 EnumerateInstances를 수행한 결과

그림 13은 입력 파라미타 없이 WBEM Manager UI에서 EnumerateInstances를 수행한 결과이다.

6. 결론

본 연구에서는 DMTF의 서버 관리 표준인 WBEM 기술에 대해 파악하고, 현재 제공되고 있는 Microsoft WMI와 SUN Solaris의 SNMP Provider들에 대해 설명하였다. 또한, WBEM 기반의 전체 관리 시스템, WBEM/SNMP Gateway의 구조 그리고 SNMP Provider의 구조를 설계하고, WBEM/SNMP Gateway의 관리 정보 모델 변환 방법과 관리 동작 변환 방법을 제시하였다. 마지막으로 WBEM/SNMP Gateway와 WBEM Manager의 프로토타입을 구현하였다.

기존의 Microsoft의 WMI나 SUN Solaris에서도 WBEM/SNMP Gateway 기능을 제공해주고자 하는 노력이 있었으나, 관리 정보 모델 변환의 경우 변환 결과가 구현물에 의존적이었다. 뿐만 아니라 관리 동작 변환의 경우에는 OS나 장비에 의존적으로만 수행이 되고, 소스 코드가 공개되어있지 않아 개발이나 확장이 어렵다. 그러므로 본 논문에서는 어떤 환경에서나 실행될 수 있고 DMTF의 표준을 따르는 WBEM/SNMP Gateway를 개발하였다.

향후의 연구계획은 표준 SNMP MIB뿐만 아니라 Enterprise MIB에 대해서도 수용해나가는 것이며, 많은 SNMP Agent들을 관리하여 우리가 설계, 구현한 WBEM/SNMP Gateway에 대한 유효성을 검증하는 것이다.

또 다른 한가지로는 CIM Server의 기능별, 모듈별 성능 테스트를 통한 Lightweight한 CIM Server에 대한 연구를 하는 것이다.

참고 문헌

[1] DMTF, http://www.dmtf.org
[2] DMTF, "Common Information Model (CIM) Specification, Version 2.2", June 1999
[3] WBEM, “WBEM Initiative”, http://www.dmtf.org/wbem
[4] DMTF, “Representation of CIM in XML, Version 2.0”, June 1999
[5] R. Fielding, J. Gettys, J. Mogul, H. Frystyk Nielsen, L. Masinter, P. Leach and T. Berners-Lee, "Hypertext Transfer Protocol - HTTP/1.1", RFC 2616, IETF HTTP WG, June 1999.
[6] J. Case, M. Fedor, M. Schoffstall, and J. Davin (Eds), “A Simple Network Management Protocol (SNMP)”, RFC 1157, IETF, May 1990.
[7] DMTF, "Management Object Format (MOF)," http://www.dmtf.org/education/mof
[8] Microsoft, Windows Management Instrumentation (WMI),http://msdn.microsoft.com/library/default.asp?url=/library/enus/wmisdk/wmi/wmi_reference.asp.
[9] Sun Microsystems, Solaris WBEM Service, http://www.sun.com/software/solaris/8/ds/ds-wbem24/.
[10] Sun Microsystems, Java Web Based Enterprise Management (WBEM Services)
[11] http://wbemservices.sourceforge.net
[12] Pegasus, http://www.pegasus.org
[13] Tim Bray, Jean Paoli and C. M. Sperberg-McQueen, "Extensible Markup Language (XML) 1.0," W3 Recommendation REC-xml-19980210, Feb. 1998
[14] Net-SNMP, http://net-snmp.sourceforge.net/.
[15] OMG(Object Management Group),“UML(Unified Modeling Language)”,http://www.uml.org/
[16] Apache-SSL, http://www.apache-ssl.org
[17] Frank Strauss et al, ”A Library to Access SMI MIB Information”, http://www.ibr.cs.tu-bs.de/projects/libsmi/.