WBEM 표준의 데이터 전송방식 분석

Analysis of the Data Transport Mechanism for WBEM Standard

박종근 (J.G. Park) 서버플랫폼연구팀 연구원
안창원 (C.W. Ahn) 서버플랫폼연구팀 선임연구원
조희남 (H.N. Cho) 서버플랫폼연구팀 연구원
김성운 (S.W. Kim) 서버플랫폼연구팀 팀장

목 차

Ⅰ. 서론
Ⅱ. WBEM 개요
Ⅲ. 데이터 전송방식
Ⅳ. 결론

분산 네트워크 환경을 기반으로 다양한 시스템과 장비가 서로 연결되어 사용되고 있는 현재의 IT 환경의 문제점을 해결하기 위해 DMTF는 선도적인 기업 및 교육 기관과 협력하여 이기종 시스템간 상호호환성을 제고하기 위한 시스템 자원 관리 표준 규격을 제정해 나가고 있으며, 그 중 하나가 WBEM 표준이다. 이 표준에서는 공통 정보 모델인 CIM을 비롯하여 데이터를 인코딩하고 전송하기 위한 XML과 HTTP를 표준으로 채택하고 있으며, CIM 데이터를 더욱 효과적으로 표현하고 전송하기 위해 필요한 XML 문서의 형식과 태그 그리고 HTTP 헤더와 오류 코드 등을 추가로 정의하고 있으며, 이들 표준의 적용에 있어 필요한 몇몇 요구사항들도 정의하고 있다. 본 고에서는 표준 규격을 바탕으로 WBEM 표준에서 채택하고 있는 데이터의 전송방식에 대해 자세히 소개하고, 끝으로 현재 WBEM 표준이 갖고 있는 문제점과 주요 이슈를 함께 살펴보고자 한다.

Ⅰ. 서론

분산된 네트워크 환경을 기반으로 다양한 시스템과 장비가 서로 연결되어 사용되고 있는 현재의 IT 환경은 시스템과 장비에 종속적인 관리 도구를 사용함으로써 이기종 시스템 사이에 상호호환성이 보장되지 않는 문제점을 갖고 있다. 즉, 장비 제공 업체나 관리 소프트웨어 업체가 독자적인 데이터 모델과 기술을 바탕으로 개발하여 제공하는 전용 관리 도구를 사용함으로써, 각 장비마다 서로 다른 관리 도구와 방식을 적용해 오고 있다. 결국 이로 인해 시스템 관리 및 유지에 많은 비용이 발생하여 이는 고스란히 기업의 부담으로 작용하고 있다.

결국 이러한 상황을 효과적으로 대처하기 위해 분산 시스템 관리 분야의 표준화를 주도하고 있는 DMTF는 선도적인 기업 및 교육 기관과 협력하여 이기종 시스템간의 상호호환성을 제고하기 위한 시스템 자원 관리 표준 규격을 제정해 나가고 있으며, 그 중 하나가 WBEM 표준이다.

WBEM 표준에서는 이기종의 플랫폼과 동일 플랫폼상의 여러 시스템 자원들을 효과적으로 제어하고 감시하며 관리하기 위해 공통 정보 모델인 CIM을 비롯하여 데이터를 인코딩하고 전송하기 위한 XML과 HTTP를 표준으로 채택하고 있다. 특히 이미 검증되어 널리 활용되고 있는 XML과 HTTP를 표준 방식으로 채택함으로써 XML과 HTTP가 갖고 있는 다양한 장점을 고스란히 활용하고 있다. 그러나 CIM 데이터를 더욱 효과적으로 표현하고 전송하기 위해 필요한 XML 문서의 형식과 태그 그리고 HTTP 헤더와 오류 코드 등을 추가로 정의하고 있으며, 이들 표준의 적용에 있어 필요한 몇몇 요구사항들도 정의하고 있다.

본 고에서는 WBEM 표준에서 채택하고 있는 데이터의 전송방식에 대해 분석하고 이를 자세히 소개하고자 한다. 이를 위해 “CIM Operations over HTTP” 규격에 대해 현재 제정된 내용[1]과 앞으로 개정될 내용[2]을 바탕으로, CIM 시스템에서 주고 받는 CIM 메시지와 이 메시지를 HTTP로 통신하기 위해 사용하고 있는 확장 헤더를 소개한다. 또한 HTTP 메시지 생성과 관련된 여러 고려사항들을 함께 정리하고, 끝으로 현재 WBEM 표준이 갖고 있는 문제점과 주요 이슈를 함께 살펴보고자 한다.

Ⅱ. WBEM 개요

DMTF에서 제정한 WBEM 표준은 CIM을 기반으로 분산 네트워크 환경에서 효율적인 시스템 관리 프레임워크를 제시하고 있는 표준이다. 본 장에서는 우선 WBEM을 정의한 DMTF 표준화 단체와 WBEM 표준에 대해 간략히 살펴본 다음 WBEM에서 채택하고 있는 XML과 HTTP에 대해서도 간단히 살펴보기로 한다.

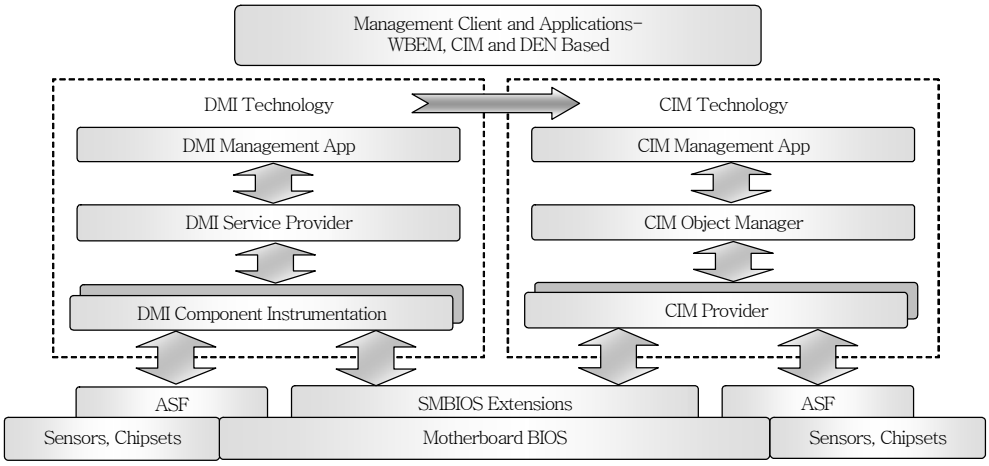
1. DMTF 개요

DMTF는 기업 환경 및 인터넷 환경에서 요구되는 관리 표준을 개발하고 상호운용을 위한 통합 기술을 선도하는 산업체 조직이다.

1992년에 설립된 DMTF는 표준 규격을 개발하고 개정해 나감에 있어 선도적인 기업과 관련 산업의 수요자인 회원들을 중심으로 작업반(working group)을 구성하여 협력해 나가고 있다. DMTF의 운영은 이사회(board member)에 의해 이루어지고 있는데 여기에는 3Com, Cisco Systems, Dell Computer Corp., HP, IBM, Intel, Microsoft, NEC, Novell, Oracle, Sun Microsystems, Symantec, VERITAS Software 등이 참여하고 있다[3].

현재 DMTF에서 제정하고 있는 표준으로는 CIM, WBEM, DEN, DMI, ASF, SMBIOS 등이 있으며, 이들 표준의 개념적 위치는 (그림 1)과 같다.

(그림 1)에 따르면 ASF와 SMBIOS 등과 같은 저수준(low-level)의 관리 인터페이스가 DMI와 CIM 기반 기술과 상호 작용하여 시스템 관리를 가능하게 하는 일련의 관계를 파악할 수 있다. DMTF의 전략은 DMI 기반 기술을 CIM/WBEM 기반 기술로 통합



(그림 1) DMTF 관리 표준 및 관계

함으로써 하나의 일관된 관리 하부구조를 구축하는 것이다[4].

다음 절에서 소개할 WBEM과 CIM을 제외한 나머지 표준의 특징을 간략히 살펴보면, DEN은 CIM을 LDAP 프로토콜과 X.500 모델을 지원하는 하나의 디렉토리로 매핑하는 표준이며, DMI는 네트워크로 연결되어 있는 데스크톱 컴퓨터, 노트북, 서버를 관리하기 위한 표준 프레임워크의 구축에 관한 것이다. 그리고, ASF는 운영체제가 정상적으로 동작하지 않을 때 발생하는 관리 공백을 해결하기 위해 설계되었으며, 컴퓨터 시스템을 대상으로 원격 통제 및 경보 인터페이스를 정의하고 있다. 끝으로 SMBIOS는 주기판 또는 시스템 제조업체가 제품에 대한 하드웨어 관련 관리 정보를 표준 형태로 제공하기 위한 방법을 기술하고 있다.

DMTF의 각 표준에 대해서는 안창원 외[5]에 더욱 상세히 소개되어 있다.

2. CIM 표준

공통 정보 모델인 CIM은 비즈니스 컴퓨팅 및 네트워킹 환경을 기술하기 위한 개념적인 모델로서, 플랫폼 독립적인 동시에 기술 중립적으로 관리 정보를 교환하기 위해 제정된 획기적인 표준이다. 여기서 비즈니스 컴퓨팅 및 네트워킹 환경이라 함은 관리 대상 개체 및 그들의 상태, 운용, 조합, 구성, 관계 등을 모두 포함한다. CIM 모델은 특정 영역에만 국한되지 않으며, 네트워크를 기반으로 클라이언트부터 서버에 이르는 종단간(end-to-end) 관리를 지향하고 있다[6].

기본적으로 CIM은 관리 정보 및 서비스 의미체계에 대한 단일 모델을 정의하여 모든 요소를 이 모델의 의미체계에 매핑한다. 그리고 장비의 상세 정보부터 서비스 구성에 이르기까지 비즈니스 컴퓨팅 및 네트워킹의 모든 요소를 단일 의미체계로 제공한다. 이로 인해 데이터를 재사용할 수 있으며, 제품간의 정보를 일관성 있게 표현할 수 있다.

CIM의 또 다른 장점으로는 모델 자체의 유연성과 확장성이다. 즉, 사용자가 특정 관리 영역을 기술하기 위해 정보 모델을 추가해 나갈 수 있다.

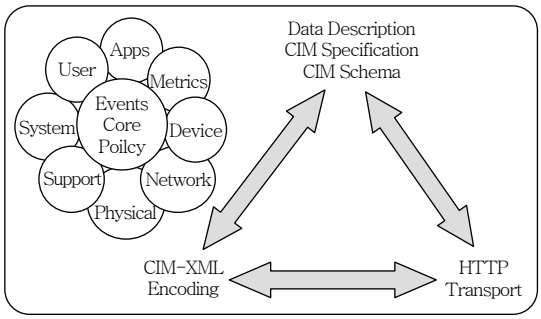
CIM 스키마(schema)에는 크게 핵심모델(core model)과 이를 확장한 일련의 공통모델(common model)로 구성된다. 핵심모델은 모든 관리 영역에 적용될 수 있는 모델이며, 공통모델은 시스템, 서비스, 네트워크, 응용 프로그램, 사용자, 데이터베이스 등과 같이 관리를 필요로 하는 네트워크 수준부터 운영체제 및 응용 프로그램에 이르는 주요 기술 영역에 대해 공통적인 정보를 기술하고 있는 모델이다. 이 외에도 CIM 스키마는 사용자의 요구에 따라 확장한 확장모델(extension model)을 사용할 수 있

으며, 이 모델은 기본적으로 핵심모델이나 공통모델로부터 상속받는 것을 원칙으로 하지만 특정 기술에 국한된 사항은 독립적으로 정의될 수도 있다.

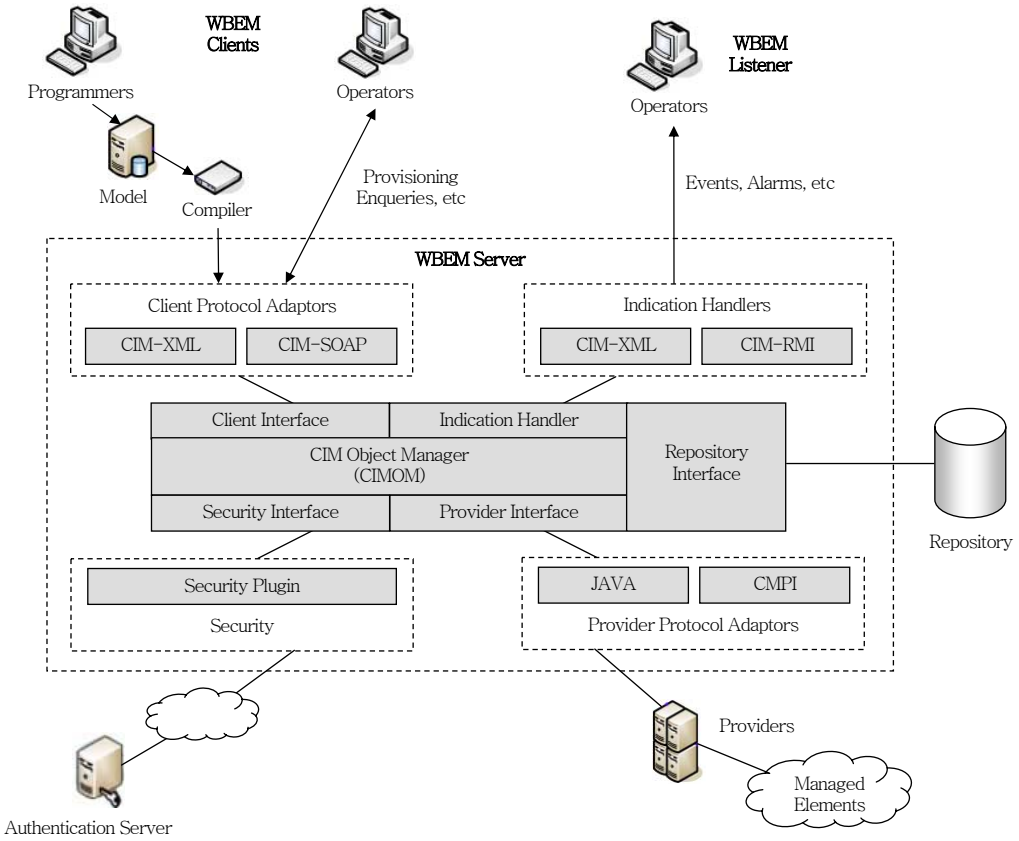
3. WBEM 표준

분산 네트워크 환경에서 효율적인 시스템 관리 프레임워크를 제시하고 있는 WBEM 표준은 1996년 7월 BMC Software, Cisco Systems, Compaq, Intel, Microsoft를 주축으로 개발되기 시작하였다. 초기에는 WBEM에서 필요한 새로운 전송 프로토콜인 HMMP를 개발하는 데 역량을 집중하였으나, HTTP의 헤더를 단순히 확장하는 것만으로도 충분히 그 목적이 달성될 수 있었기 때문에 HMMP 프로토콜의 개발은 중단되고 대신 HTTP를 사용하기로 결정하였다. 이후에 이 그룹은 DMTF의 CIM을 WBEM의 관리 정보를 표현하는 표준으로 정하였으며, 1998년 6월에는 WBEM의 탄생을 주도했던 기업들이 DMTF의 이사회에 가입함으로써 DMTF에서 WBEM의 지속적인 개발을 추진해 오고 있다[7].

WBEM 표준은 데이터 모델인 CIM과 WBEM 프로토콜인 CIM-XML이 최종 확정되어 활용되고 있



(그림 2) WBEM 구성 표준 관계



(그림 3) WBEM의 구조

으며, CIM-XML은 다시 XML DTD 형식에 따라 CIM 데이터를 XML로 표현하는 "Representation of CIM using XML" 표준과 WBEM의 전송방식에 대한 "CIM Operations over HTTP" 표준으로 이루어진다. (그림 2)는 이들 표준의 관계를 도식화한 것이다.

이 외에도 응용 프로그램이 WBEM 관리 시스템을 식별하고 서로 연동되기 위한 방법을 제공하는 "WBEM Discovery using SLP and WBEM URI mapping"과 CIM 기반의 관리 환경으로부터 데이터를 추출하기 위해 사용되는 "CIM Query Language"에 대한 표준화가 추가로 진행중이며 현재 초안(preliminary version)이 개발된 상태이다.

WBEM 시스템을 구성하기 위해서는 WBEM 클라이언트와 적어도 하나 이상의 WBEM 서버가 있어야 한다. WBEM 서버의 구성 요소는 다수의 WBEM 프로바이더와 CIM 서버인 CIMOM, 그리고 WBEM 클라이언트와의 통신을 위한 인터페이스로 구성된다.

WBEM 클라이언트는 대개 관리를 위한 응용 프로그램으로 표현되지만, WBEM 정보를 이용하는 여러 형태로도 존재할 수 있다. WBEM 클라이언트는 관리를 위한 정보를 얻기 위해 직접 프로바이더에게 접근할 수는 없으며 반드시 프로바이더를 관리하며 상호작용하는 CIMOM을 통하여 정보를 얻을 수 있다. 이와 같은 WBEM의 구조를 도식화하면 (그림 3)과 같다[8].

4. XML과 HTTP 개요

WBEM 표준의 CIM-XML은 CIM 데이터를 XML로 인코딩하고, 인코딩된 데이터는 HTTP를 이용하여 전송하도록 규정하고 있다.

XML은 구조화된 정보를 표현하는 언어로서, HTML을 대체할 목적으로 1996년 W3C에서 제안한 언어이다. 근본적으로 HTML에서 사용되는 연결(link) 기능 등을 확장함과 동시에 SGML의 데이터 모델링 능력을 강화한 언어로서, HTML과 SGML의 장점을 모두 갖고 있는 것이 특징이다.

DMTF에서는 CIM 데이터를 XML로 표현하기 위해 메타 스키마 매핑(meta schema mapping) 기법을 사용한다. 이는 XML 스키마로 CIM 메타 스키마를 표현함으로써, 클래스(class)와 인스턴스(instance) 그리고 한정사(qualifier)와 같은 CIM 선언들과 HTTP에 사용되는 CIM 메시지를 XML로 표현할 수 있다.

XML과 함께 WBEM에서는 XML로 인코딩된 데이터를 주고 받기 위해 HTTP를 전송 프로토콜로 사용한다. HTTP는 1989년 Tim Berners-Lee가 최초로 하이퍼텍스트 기반으로 연결된 정보 시스템의 장점을 언급한 이후 1990년 하반기에 Robert Cailliau와 함께 최초의 웹 브라우저와 서버를 개발하고 여기에 사용될 프로토콜로 개발되었으며, 현재 HTTP 규격은 v. 1.1까지 정의되어 있다.

WBEM 표준에서는 HTTP v. 1.0과 HTTP v. 1.1에 기반하여 메시지를 생성한다. 그러나 CIM에 특화된 정보를 전달하고 CIM 메시지의 의미를 구체화하기 위해 추가적인 확장 헤더와 오류 코드를 정의하고 있으며, 표준 HTTP 헤더에 대해서도 몇몇 추가 요구사항을 정의함으로써 WBEM의 전송 프로토콜로서 효과적으로 이용되고 있다. 더욱이 Basic Authentication 및 Digest Authentication과 같은 Web Authentication 방법과 SSL, TLS, 그리고 SHTTP 프로토콜 등과 같이 HTTP를 위해 개발된 다양하면서도 막강한 보안 기능을 별도의 추가 개발 없이 그대로 이용할 수 있음으로써 WBEM에 효과적인 보안 기능을 제공하고 있다.

Ⅲ. 데이터 전송방식

WBEM 표준에서는 CIM 스키마를 이용하여 관리 정보를 제공할 수 있거나 이용할 수 있는 제품을 CIM 시스템(products)이라고 하며, 여기에는 CIM Client, CIM Server 그리고 CIM Listener가 있다. 그리고 CIM 메시지란 이들 CIM 시스템 사이에 주

고 받는 메시지로서, 이 메시지를 주고 받기 위해서는 XML DTD 형식에 따라 CIM 데이터를 XML로 인코딩한 뒤 HTTP를 이용하여 전송한다.

이와 같이 WBEM에서 CIM 데이터 정보를 주고 받기 위해 정의한 전송 방식에 대해 "CIM Operations over HTTP" 규격[1],[2]을 중심으로 현재 표준으로 제정되어 있는 내용과 앞으로 개정될 내용을 함께 소개하기로 한다. 그리고 끝으로 WBEM 표준에서의 데이터 전송 방식이 갖고 있는 문제점과 이에 대한 주요 이슈를 정리한다.

1. 요구사항

CIM 메시지를 XML로 표현하고 이를 HTTP 메시지로 생성(encapsulation)하는 방법은 다양할 수 있다. 따라서, WBEM에서는 서로 다른 CIM 구현물 사이의 상호운용성 관점에서 XML 인코딩과 HTTP 메시지 생성에 대한 표준화와 관련하여 다음과 같이 최소한의 요구사항을 제시하고 있다[1],[2].

먼저 CIM 메시지를 XML로 표현할 때에는 다음과 같은 기준이 적용된다.

- 각각의 CIM 메시지는 완전하게 XML로 기술된다. 이때 간결함보다는 완전함이 우선시된다.
- CIM 메시지는 관리의 목적으로 CIM 구현물이 효과적으로 통신할 수 있도록 충분한 기능성을 제공한다.
- CIM의 구현 범위가 다양할 수 있도록 CIM 메시지를 기능 범주(functional profiles)별로 구분한다. 다만 기능 범주는 상호운용성이 보장되는 한 최소한으로 유지한다.

그리고 CIM 메시지를 HTTP 메시지로 생성할 때에는 다음과 같은 기준이 적용된다.

- HTTP 1.0에 기반한 시스템을 고려하여, 메시지 생성은 HTTP 1.0과 HTTP 1.1을 모두 지원할 수 있도록 설계한다.
- 메시지 생성에 있어 HTTP 1.0이나 HTTP 1.1과 충돌이 발생하는 어떠한 요구사항도 추가하지 않는다.
- 메시지 생성은 현재의 기본적인 HTTP 토대 위에서 가능해야 한다. 일부 HTTP에 대한 개선이 있을 수 있지만, 강제사항으로써 메시지 생성에 관련된 개선은 허용되지 않는다.
- 메시지 생성에 있어 HTTP tunneling이나 URL munging의 사용은 피한다.
- 메시지 생성에 있어 방화벽과 프록시 서버의 효율적인 제어를 위해 HTTP 헤더에 핵심이 되는 CIM 메시지 정보를 포함한다.
- HTTP 메시지 내에서 CIM 메시지는 분명하고 모호하지 않게 표현한다.

2. CIM 메시지

CIM 메시지는 CIM 시스템 간의 정보를 교환하기 위해 사용되는 잘 정의된 요청 데이터 패킷 또는 응답 데이터 패킷이며, 여기에는 CIM Operation 메시지와 CIM Export 메시지가 있다. CIM Operation 메시지는 대상 이름공간(namespace)상에서 어떤 동작(operation)을 호출하는 데 사용되는 CIM 메시지이며, CIM Export 메시지는 대상 이름공간이나 또는 대상 이름공간 밖의 개체에 대한 정보를 교환하기 위해 사용되는 CIM 메시지이다.

이때 CIM Operation 요청 메시지를 전송하고, CIM Operation 응답 메시지를 받아서 처리하는 클라이언트를 CIM Client라고 하고, 이와는 반대로 CIM Operation 요청 메시지를 받아서 처리한 다음 CIM Operation 응답 메시지를 전송하는 서버를 CIM Server라고 한다. 이와는 달리 CIM Export 요청 메시지를 받아서 처리한 다음 CIM Export 응답 메시지를 전송하는 서버를 CIM Listener라고 한다.

CIM 메시지를 XML로 표현하기 위해 XML DTD에서는 <CIM> 항목 아래에 <MESSAGE> 항목을 정의하고, 메시지의 성격에 따라 서로 다른 하위 항목을 정의하고 있다. 따라서, CIM Operation 메시지의 경우, 단일 요청 또는 응답 메시지에 대해 각각 <SIMPLEREQ>와 <SIMPLERSP> 하위 항목

을 정의하고, 다중 요청 또는 응답 메시지에 대해서는 각각 <MULTIREQ>와 <MULTIRSP> 하위 항목을 정의한다. 반면에 CIM Export 메시지의 경우, 단일 메시지에 대해 각각 <SIMPLEXPREQ>와 <SIMPLEXPRSP> 하위 항목을 정의하고, 다중 메시지의 하위 항목으로는 각각 <MULTIEXPREQ>와 <MULTIEXPRSP>를 정의하고 있다.

단 모든 CIM 요청 메시지는 <MESSAGE> 항목의 ID 속성값을 반드시 가져야 하며, 이에 대응되는 CIM 응답 메시지는 요청 메시지와 동일한 ID 속성값을 가져야 한다. 또한 PROTOCOLVERSION 속성의 값으로 반드시 CIM Operations over HTTP [1],[2] 규격의 버전 번호를 사용한다.

가. CIM Operation 메시지

모든 CIM Operation 메시지는 하나 이상의 메소드 호출로 정의된다. 이와 관련된 메소드에는 CIM Operation을 모델링하기 위해 표준 규격에서 정의한 23개의 Intrinsic 메소드와 특정 스키마 내의 CIM 클래스 또는 CIM 인스턴스 메소드로 정의되어 있는 Extrinsic 메소드가 있다.

단일 CIM Operation 요청 메시지의 메소드가 Intrinsic이면 XML로 <SIMPLEREQ> 항목 아래에 호출되는 메소드 이름을 담고 있는 <IMETHODCALL> 항목이 존재하며, 그 하위에 메소드가 실행될 대상 이름공간을 담고 있는 <LOCALNAMESPACEPATH> 항목이 따른다. 이와는 달리 메소드가 Extrinsic이면 XML로 <METHODCALL> 하위 항목이 있고 그 아래에 메소드가 호출되는 클래스 또는 인스턴스를 의미하는 <LOCALCLASSPATH> 또는 <LOCALINSTANCEPATH> 항목이 따른다.

요청 메시지에 대응되는 단일 CIM Operation 응답 메시지는 <SIMPLERSP> 항목 아래에 각각 <IMETHODRESPONSE> 항목과 <METHODRESPONSE> 항목으로 표현된다. 반면에 다중 CIM Operation 메시지의 경우 <MULTIREQ>, <MULTIRSP> 항목 모두 각각 둘 이상의 <SIMPLEREQ>, <SIMPLERSP> 항목으로 표현된다.

〈표 1〉 CIM Intrinsic Methods

기능 범주	종속관계	메소드
Basic Read	NONE	GetClass EnumerateClasses EnumerateClassNames GetInstance EnumerateInstances EnumerateInstanceNames GetProperty
Basic Write	Basic Read	SetProperty
Schema Manipulation	Instance Manipulation	CreateClass ModifyClass DeleteClass
Instance Manipulation	Basic Write	CreateInstance ModifyInstance DeleteInstance
Association Traversal	Basic Read	Associators AssociatorNames References ReferenceNames
Query Execution	Basic Read	ExecQuery *ExecuteQuery*
Qualifier Declaration	Scheme Manipulation	GetQualifier SetQualifier DeleteQualifier EnumerateQualifier

WBEM 표준에서 정의하고 있는 23개의 Intrinsic 메소드는 기능별로 <표 1>과 같이 구분되며, 각 기능 범주는 상호 종속관계를 갖는다. 여기에서 종속관계란, 예를 들어, 'Basic Write'가 'Basic Read'에 종속관계를 갖는 경우, 'Basic Write'를 지원하는 CIM 서버는 반드시 'Basic Read'도 지원해야 함을 의미한다. 단, ExecuteQuery 메소드는 새로이 개정될 규격[2]에서 추가될 메소드이다.

나. CIM Export 메시지

CIM Export 메시지는 정보의 교환을 위해 주고받는 메시지로서, 모든 CIM Export 메시지 요청은 하나 이상의 Export 메소드 호출로 정의된다.

단일 CIM Export 요청 메시지는 XML로 <SIMPLEEXPREQ> 항목 아래에 호출되는 메소드 이름을 담고 있는 <EXPMETHODCALL> 항목이 존재하며, 이에 대응되는 단일 CIM Export 응답 메시지는 <SIMPLEEXPRSP> 항목 아래에 <EXPME-

THODRESPONSE> 항목으로 표현된다.

반면에 다중 CIM Export 메시지의 경우 CIM Operation 메시지와 마찬가지로, <MULTIEXPREQ>, <MULTIEXPRSP> 항목 모두 각각 둘 이상의 <SIMPLEEXPREQ>, <SIMPLEEXPRSP> 항목으로 표현된다.

표준에서는 Export 메소드로서 CIM Listener로 단일 CIM Indication을 내보내는 ExportIndication 메소드를 유일하게 정의하고 있다. 그러나, 개정될 규격[2]에는 비동기 요청 메시지에 대한 응답을 CIM Listener에게 전송하는 역할을 담당하는 ExportAsyncResponse 메소드가 추가될 예정이다. 여기에서 비동기 요청 메시지란 장시간동안 실행되어야 하는 CIM Operation 요청 메시지(예, 디스크 포맷, 데이터베이스 생성, 파일 시스템 백업/복원 등)에 대해 일단 응답 메시지로 오류 또는 요청 메시지의 수신 확인만을 전송하고, 실제 실행 결과는 나중에 별도로 전송해 주는 메시지를 의미한다. 이때 실제 결과를 포함하고 있는 응답 메시지는 요청 메시지를 전송한 클라이언트 이외의 다른 곳으로 전송될 수 있으며 이때 전혀 다른 인코딩 방식과 전송 프로토콜을 사용할 수도 있다. 이와 같은 특성은 HTTP의 세션이 상대적으로 짧기 때문에 도입된 개념으로, 마찬가지로 새로이 개정될 규격[2]에 도입될 예정이다.

3. HTTP 메시지 생성

본 절에서는 CIM 메시지를 어떻게 HTTP 메시지로 생성하는지에 대해 소개하기로 한다. 일반적으로 CIM 메시지는 다양한 HTTP 요청 메소드와 함께 사용될 수 있지만, CIM Operations in HTTP [1],[2]에서는 HTTP POST 요청 메시지만을 정의한다. 단, HTTP 확장 프레임워크[9]를 사용하는 경우에는 HTTP POST와 M-POST 요청 메시지를 사용하며, 이를 통해 인터넷 프록시나 방화벽은 CIM 메시지 호출에 대해 더욱 개선된 필터링 제어와 관리 유연성을 제공할 수 있다.

일반적으로 하나의 CIM 메시지는 CIM Operation 요청이나 응답 또는 CIM Export 요청이나 응답 중의 하나이다. WBEM 표준에서는 CIM 메시지의 의미를 구체화하기 위해 POST 메시지의 HTTP 헤더에 다음과 같은 확장 헤더를 사용하고 있다.

• CIMOperation

모든 CIM Operation 요청 또는 응답 메시지에 나타나야 하며, HTTP 메시지에 CIM Operation 요청 또는 응답이 포함되어 있음을 의미한다.

• CIMExport

모든 CIM Export 요청 또는 응답 메시지에 나타나야 하며, HTTP 메시지에 CIM Export 요청 또는 응답이 포함되어 있음을 의미한다.

• CIMProtocolVersion

모든 CIM 메시지에 나타날 수 있으며, 데이터 전송에 사용된 CIM Operations over HTTP 규격의 버전을 의미한다. 만일 이 헤더가 생략되면 1.0으로 가정한다.

• CIMMethod

모든 단일 CIM Operation 요청 메시지에 나타나야 하며, 호출되는 CIM Operation 메소드의 이름을 의미한다.

• CIMObject

모든 단일 CIM Operation 요청 메시지에 나타나야 하며, 메소드가 호출된 CIM 객체의 경로로써 Intrinsic 메소드의 경우 이름공간의 경로를, Extrinsic 메소드의 경우 클래스나 인스턴스의 경로를 의미한다.

• CIMExportMethod

모든 단일 CIM Export 요청 메시지에 나타나야 하며, 호출되는 CIM Export 메소드의 이름을 의미한다.

• CIMBatch

모든 다중 CIM Operation 요청 메시지에 나타나

야 하며, 여러 개의 메소드 호출을 포함한 Operation 요청 메시지임을 의미한다.

• CIMExportBatch

모든 다중 CIM Export 요청 메시지에 나타나야 하며, 여러 개의 메소드 호출을 포함한 Export 요청 메시지임을 의미한다.

• CIMError

CIM 응답 메시지가 아닌 모든 HTTP 응답 메시지에 나타날 수 있으며, CIM Server 또는 Listener에서 CIM Operation 요청을 처리하는 동안 발생한 CIM과 관련된 오류 정보를 나타낸다.

이 중에서 CIMMethod, CIMObject, CIMExportMethod, CIMBatch, CIMExportBatch 헤더는 방화벽과 프록시가 라우팅 및 포워딩을 결정하기 위해 이용하기도 한다.

그리고 새로이 개정될 규격[2]에서는 CIMError가 헤더뿐만 아니라 트레일러(trailer)에서도 사용될 수 있도록 정의하고 있으며, 또한 다음과 같은 확장 헤더를 추가적으로 정의하고 있다.

• CIMRoleAuthenticate

CIM Server가 '401 Unauthorized' 응답의 일부로 WWW-Authenticate 헤더와 함께 전송할 수 있으며, CIM Server가 채택하고 있는 인증 방식과 관련된 방침을 의미한다.

• CIMRoleAuthorization

CIM Client가 사용자 인증을 수행하고자 할 때 일반적인 Authorization 헤더와 함께 전송된다.

• CIMStatusCode

CIM Error를 의미하는 상태 코드를 나타내며, 헤더가 아닌 트레일러로 사용된다.

• CIMStatusCodeDescription

CIM Error에 대한 설명으로써, CIMStatusCode 트레일러와 함께 나타날 수 있다.

• WBEMServerResponseTime

모든 CIM 응답 메시지에 나타날 수 있으며, CIM Server가 요청 메시지를 처리하고 응답 메시지를 생성하는 데 소요된 시간을 나타낸다.

다음은 단일 CIM Operation 요청에 대한 HTTP 메시지의 예이다. 이 예는 M-POST 요청 메시지로서 CIMOperation, CIMMethod, CIMObject 등의 확장 헤더를 사용하고 있다.

```
M-POST /cimom HTTP/1.1
HOST: http://www.myhost.com/
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx
Man: http://www.dmtf.org/cim/http/v1.0 ; ns=73
73-CIMOperation: MethodCall
73-CIMMethod: GetClass
73-CIMObject: root/cimv2
<?xml version="1.0" encoding="utf-8" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
 <MESSAGE ID="87872" PROTOCOLVERSION="1.0">
 <SIMPLEREQ>
  <IMETHODCALL NAME="GetClass">
   <LOCALNAMESPACEPATH>
    <NAMESPACE NAME="root"/>
    <NAMESPACE NAME="cimv2"/>
   </LOCALNAMESPACEPATH>
       <IPARAMVALUE NAME="ClassName"><CLASS NAME
NAME= "CIM_VideoBIOS"/></IPARAMVALUE>
       <IPARAMVALUE NAME= "LocalOnly"><VALUE>FALSE
</VALUE></IPARAMVALUE>
  </IMETHODCALL>
 </SIMPLEREQ>
 </MESSAGE>
</CIM>
```

4. 기타 고려사항

본 절에서는 HTTP 메시지 생성과 관련된 기타 고려사항을 소개한다.

가. HTTP의 버전

WBEM 표준에서는 CIM Client, CIM Server 그리고 CIM Listener가 HTTP 1.1을 지원하도록 권고하고 있지만 HTTP 1.0의 지원도 허용하고 있다. 그러나, M-POST와 같이 확장 헤더를 사용하고자

할 때에는 반드시 HTTP 확장 프레임워크[9]에서 정의한 요구사항을 충족해야 한다.

나. 표준 HTTP 헤더의 사용

CIM 시스템은 반드시 표준 HTTP 헤더의 사용과 관련된 요구사항은 물론 다음과 같은 헤더에 대해 표준에서 정의하고 있는 추가 요구사항을 충족해야 한다.

• Accept

CIM 요청 메시지에 이 헤더를 포함할 수 있으며 "text/xml"과 "application/xml" 중 응답 메시지의 형식으로 지정된 값을 의미한다. 만일 CIM Server나 CIM Listener가 이 두 형식 모두를 응답 메시지의 형식으로 수락할 수 없다면 '406 Not Acceptable'을 반환한다.

• Accept-Charset

CIM 요청 메시지에 이 헤더를 포함할 수 있으며 응답 메시지의 본문이 "UTF-8"을 사용하여 인코딩하도록 명시한다.

• Accept-Encoding

CIM 요청 메시지에 이 헤더를 포함할 수 있으며 CIM Server나 CIM Listener가 "Identity" 인코딩을 사용하도록 명시한다.

• Accept-Language

CIM 요청 메시지에 이 헤더를 포함할 수 있으며 서버가 응답 메시지에서 사용할 언어를 명시한다.

• Accept-Range

CIM Client는 요청 메시지에 이 헤더를 포함해서는 안되며, 이 헤더가 포함된 요청 메시지를 받은 CIM Server나 CIM Listener는 반드시 "406 Not Acceptable"을 반환해야 한다.

• Allow

CIM Server나 CIM Listener가 "405 Method Not Allowed" 응답 메시지를 반환하는 경우 이 헤더에는 반드시 M-POST나 POST를 포함해야 한다.

• Authorization

CIM Client는 이 헤더의 값으로 "Basic" 또는 "Digest" 인증 방식과 사용자 계정 정보를 포함한다.

• Cache-Control

CIM Server나 CIM Listener는 CIM 응답 메시지에 이 헤더를 사용해서는 안된다. 단, 요청 메시지가 M-POST인 경우에는 이 헤더를 사용하고 그 값으로 "no-cache"를 반환한다.

• Connection

CIM Client는 해당 연결을 통해 전송되는 메시지가 유일한 요청 메시지라고 사전에 알려지지 않는 한 "Connection: close" 헤더를 사용해서는 안되며, CIM Server와 CIM Listener는 가능한한 persistent 연결을 지원하며 pipelining도 지원해야 한다.

• Content-Encoding

CIM 요청 메시지에 이 헤더를 포함할 수 있으며 "identity"의 값을 명시한다.

• Content-Language

CIM 메시지는 이 헤더를 포함할 수 있으며, CIM 응답 메시지에 있는 이 헤더의 값은 대응되는 요청 메시지의 "Accept-Language" 헤더 값과 일치해야 한다.

• Content-Type

모든 CIM 시스템은 이 헤더의 값으로 "text/xml" 또는 "application/xml" 값을 명시해야 한다.

• Expires

"Cache-Control" 헤더와 마찬가지로 이 헤더는 응답 메시지에 포함되어서는 안된다.

• Proxy-Authenticate

"407 Proxy Authenticate Required" 응답 메시지와 함께 전송되며 이 헤더의 값으로 "Basic" 또는 "Digest" 값을 포함해야 한다.

• WWW-Authenticate

"401 Unauthorized" 응답 메시지와 함께 전송되며 이 헤더의 값으로 "Basic" 또는 "Digest" 값을 포함해야 한다.

이 밖에 Content-Range, If-Range, Range 헤더는 모든 CIM 시스템에서 사용되어서는 안되며, 새로이 개정될 규격[2]에서는 헤더뿐만 아니라 트레일러도 사용할 수 있도록 변경됨으로써 다음의 헤더에 대해 추가 요구사항을 정의하고 있다.

• TE

CIM Client는 "TE: trailers" 헤더를 포함할 수 있으며, 이는 chunked transfer 인코딩이 사용되는 경우 CIM Error를 HTTP 트레일러로 처리할 것임을 의미한다.

• Trailer

CIM 시스템은 어떤 정보가 트레일러에 포함되어 있음을 알리기 위해 이 헤더를 사용할 수 있다.

• Transfer-Encoding

모든 HTTP 1.1 응용 환경에서는 "chunked transfer" 인코딩을 수신하고 디코딩 할 수 있어야 한다.

다. HTTP 오류 코드

CIM Server나 Listener가 CIM 요청 메시지를 처리할 때 각종 오류가 발생할 수 있다. 일반적으로 HTTP Request-Line이나 표준 HTTP 헤더를 처리할 때 오류가 발생하는 경우에는 HTTP 표준에 정의된 조치를 취해야 한다.

만일 HTTP 확장 프레임워크[9]를 CIM Server가 지원하지 않는 경우에는 "510 Not Extended" 상태 코드를 반환해야 하며, 이 외의 경우에는 다음의 규칙을 적용한다.

먼저 "501 Not Implemented" 오류 코드는 CIMProtocolVersion 헤더가 CIM Server나 Listener가 지원하지 않는 버전 값을 명시했거나, CIM Client가 다중 요청 메시지를 전송했지만 CIM Server나 Listener가 이를 지원하는 경우 또는 요청 메시지의 CIMVERSION 속성이나 DTDVERSION 속성이 정확한 값으로 설정되지 않은 경우 반환된다.

또 CIM Client가 요청 메시지를 전송하기 전에 자신을 인증하도록 CIM Server나 Listener에서 설정한 경우 "401 Unauthorized" 상태 코드가 전송되며, CIM Client가 요청 메시지를 전송하도록 허락되지 않은 경우에는 "403 Forbidden" 상태 코드를, 그리고 CIM Client를 대신하여 프록시가 요청 메시지를 전송하기 전에 자신을 인증하도록 CIM Server나 Listener에서 설정된 경우 "407 Proxy Authentication Required" 상태 코드가 반환된다.

만일 CIM 확장 헤더가 잘못되지 않았다면, CIM Server 또는 Listener는 CIM 요청 메시지를 본문을 처리할 때 "400 Bad Request" 상태 코드를 사용한다. 이 상태 코드는 메시지 본문이 CIM XML DTD 형식에 따라 유효하지 않은 경우에 반환된다.

라. 보안 고려사항

앞에서 언급한 바와 같이 HTTP에서는 보안 기능으로서 Basic Authentication, Digest Authentication과 같은 Web Authentication과 SSL, TLS, 그리고 SHTTP를 지원한다. 이 중에서 가장 일반적으로 사용되는 방법이 SSL 프로토콜을 이용한 방법이다.

WBEM 표준에서는 HTTP에서 제공하는 보안 기능 중에서 일반적으로 Basic Authentication 방법과 Digest Authentication 방법이 사용된다. 물론 보안이 중요하지 않은 환경에서는 인증 절차를 사용하지 않을 수도 있다. 그러나, Basic Authentication 방법이 널리 알려진 바와 같이 사용자 정보가 암호화되지 않은 채 전송되는 취약점을 갖고 있기 때문에 이 방법보다는 오히려 Digest Authentication 방법이 장려된다.

Digest Authentication 방법은 간단한 암호화(cryptography) 원리를 사용하는데, 클라이언트가

서버로 실제 암호를 전송하지 않으면서도 계정에 대한 암호를 알고 있음을 입증하기 위해 서버로부터 제공된 특정 값과 암호를 이용하여 digest를 생성한 다음 서버에게 전송하고 서버는 이를 검증함으로써 사용자를 인증한다. 만일 Web Authentication 방법보다 더욱 안전한 보호가 필요한 경우에는 SSL이나 SHTTP를 사용해야 한다.

5. 문제점 및 주요 이슈

지금까지 살펴본 바와 같이 WBEM 표준에서는 CIM 데이터를 XML로 인코딩한 후 이를 HTTP 프로토콜을 이용하여 전송한다. 따라서 ASCII 기반의 XML 속성과 CIM 정보의 의미를 구체화하기 위해 추가로 정의한 HTTP 확장 헤더 때문에 네트워크 상에 전송되는 데이터 트래픽의 양이 커지는 단점을 갖고 있다. K. Sandlund[10]는 WBEM 데이터 메시지를 오늘날 가장 지배적인 네트워크 관리 프로토콜인 SNMP 메시지와 비교하였는데, 그에 따르면 WBEM 데이터 메시지가 약 10배 정도 큰 대역폭을 사용한다고 한다.

이러한 문제점을 해결하고자 일부 WBEM 구현물에서는 XML 인코딩 대신 바이너리 인코딩을 사용하여 성능을 높이기도 한다. 물론 바이너리 인코딩을 채택하는 경우에는 WBEM 표준에서 정의한 데이터 인코딩 방식이 아니기 때문에 상호호환성이 결여되는 문제점을 갖고 있다. 결국 XML이 갖고 있는 폭넓은 확장성 및 호환성과 데이터 전송 성능 사이의 절충(trade-off)이 필요하다.

Ⅳ. 결론

기업의 IT 환경에 대한 관리는 점차 복잡해지고 어려워지고 있으며 사업의 핵심 요소가 IT 환경에 점차 더 의존해가는 상황에서 상호호환성과 유연성이 결여된 시스템은 막대한 시스템 운용비용을 지출하는 기업 자체에 큰 위험 요소로 존재할 수 밖에 없다. 결국 이기종 시스템간의 상호호환성을 제고하기 위한 시스템 관리 표준화는 점차 필수불가결한 요소로 자리매김할 것이다.

본 고에서는 분산 네트워크 환경에서의 효율적인 시스템 관리 프레임워크로 자리매김하고 있는 WBEM 표준과 이 표준에서 채택한 데이터 전송방식에 대해 소개하였다. WBEM 표준에서는 데이터의 인코딩과 전송방식으로 이미 검증되어 널리 활용되고 있는 XML과 HTTP를 각각 사용하고 있으며, CIM 데이터를 더욱 효과적으로 표현하고 전송하기 위해 필요한 태그와 헤더 그리고 오류 코드 등을 추가적으로 정의하고 있다.

그러나 앞에서 언급한 바와 같이 현재의 WBEM 시스템은 상호호환성과 데이터 전송 성능 사이의 절충이 요구된다. 따라서 앞으로 WBEM 표준이 시스템 관리 표준으로서 더욱 확고한 자리매김을 하기 위해서는 상호호환성과 데이터 전송 성능을 동시에 증대시킬 수 있는 방안을 모색하여 WBEM 표준이 갖고 있는 문제점을 해결해 나가야 할 것이다.

약어 정리

ASF	Alert Standard Format
CIM	Common Information Model
CIMOM	CIM Object Manager
DEN	Directory Enabled Network
DMI	Desktop Management Interface
DMTF	Distributed Management Task Force, Inc.
DTD	Document Type Definition
HMMP	HyperMedia Management Protocol
HTML	HyperText Markup Language
HTTP	Hypertext Transport Protocol
LDAP	Lightweight Directory Access Protocol
SGML	Standard Generalized Markup Language
SHTTP	Secure-HTTP
SMBIOS	System Management BIOS
SNMP	Simple Network Management Protocol
SSL	Secure Sockets Layer
TLS	Transport Layer Security
W3C	World Wide Web Consortium
WBEM	Web-Based Enterprise Management
XML	eXtensible Markup Language

참 고 문 헌

[1] DMTF, "Specification for CIM Operations over HTTP," Ver. 1.1, DSP0200, 2003. 6.

[2] DMTF, "Specification for CIM Operations over HTTP," Ver. 1.2 Preliminary, DSP0200, 2004. 12.

[3] DMTF, "DMTF Organization Backgrounder," http://www.dmtf.org/newsroom/presskit/DMTF_backgrounder.pdf

[4] DMTF, "DMTF Standards and Terminology," DMTF Technical Note, 2003. 6., http://www.dmtf.org/education/technote_Standards.pdf

[5] 안창원, 김영호, 김지연, 조희남, 정성인, "분산 시스템 관리 표준화 동향," 주간기술동향, 통권 1164호, 2004. 9., pp.1-15.

[6] DMTF, "The Value of the Common Information Model," DMTF Technical Note, 2003. 6., http://www.dmtf.org/education/technote_WhyCIM.pdf

[7] 조희남, 안창원, 정성인, 김영호, 김지연, "오픈 소스 프로젝트 WBEM 구현물에 대한 분석," KNOM Review, Vol.7, No.1, 2004, pp.9-19.

[8] Chris Hobbs, A Practical Approach to WBEM/CIM Management, Auerbach Publications, 2004.

[9] IETF, "HTTP Extension Framework," IETF Internet Draft, 1999. 6., http://www.ieft.org/rfc/rfc2774.txt

[10] Kristofer Sandlund, "Network Management Using WBEM," Master's thesis, Lulea Tekniska Universitet, 2001.