

객체지향기술을 도입한 망 관리 시스템 설계 및 구현

김중훈^o, 손진곤, 강상원

한국방송통신대학교 평생대학원 정보과학과, 고려대학교 컴퓨터학과 분산시스템 연구실
subkjh@sypol.com^o, jgshon@mail.knou.ac.kr, swkang@disys.korea.ac.kr

Design and Implementation of NMS With Object-Oriented Technology

Jong Hoon Kim^o, Jin Gon Shon

Dept. of Computer Science, Graduate School, Korea National Open University
Sang-Won Kang

Distributed System Lab. Dept. of Computer Science & Engineering, Korea University

요 약

망 운용에 있어서 망 관리 시스템은 다양한 네트워크 장비의 증가로 인하여 점점 더 중요한 위치를 차지하고 있다. 과거 망을 구성하는 요소는 네트워크 장비로 국한되었지만 현재는 서비스를 지원하는 데몬(Daemon) 등 장비 외적인 요소까지 속하게 되어 더욱 복잡해 지고 있다. 기존의 망 관리 시스템은 SNMP(Simple Network Management Protocol)나 ICMP(Internet Control Message Protocol) 프로토콜을 사용하였으나 이 기술만으로는 데몬과 같은 장비 외적인 요소들까지 제어, 관리하지 못했다. 반면, TELNET등과 같은 기능을 추가적으로 이용할 수 있다면, 보다 쉽게 망 정보를 얻을 수 있는 장점이 있어 기존의 망 관리 시스템에서 지원하지 못했던 기능의 한계점을 극복할 수 있다. 이에 본 논문은 객체지향기술(OOT : Object-Oriented Technology)을 도입하였다. 클래스(Class)는 고유한 특성을 가지는 관리 대상 종류를 선언한 것이고, 관리 대상은 네트워크 장비 뿐만 아니라 데몬이나 지역, 부서등과 같은 추상적인 대상이 포함되며 이는 객체(Object)를 의미한다. 또한, 상속(Inheritance) 개념을 도입하여 중복요소를 없앴으며, 클래스의 메소드(Method)에 관리 대상을 처리할 수 있는 기능을 정의하였다. 기능 정의에는 SNMP와 TELNET, ICMP등을 내부적으로 이용하여 구현된 스크립트(Script)를 이용하였다.

1. 서 론

망 관리 시스템(NMS : Network Management System)은 망의 비정상적인 운영 상태를 감시하고 필요에 따라 관리 대상의 설정 값을 조정하여 효율적인 네트워크 환경을 제공한다. TCP/IP 기반의 네트워크 환경에서 망 관리에 가장 많이 사용되는 관리 기법으로는 SNMP와 ICMP 프로토콜을 이용한 방법이 있다. SNMP나 ICMP를 이용한 망 관리 시스템에서는 ICMP를 이용하여 장비의 On/Off 상태를 체크하고 SNMP를 이용하여 관리 대상으로부터 정보를 조회하거나 설정 작업을 하여 SNMP Trap이나 Inform을 이용하여 관리대상 스스로 보낸 정보를 적용하기도 한다[1].

기존의 망 관리 시스템에서는 관리 대상이 개발 시점에서 이미 정해진 상태이므로, 개발 완료 후 새로운 관리 대상의 출현 시에 망 관리 시스템에 추가 작업이 필요하게 된다. 현재 IPv6의 출현으로 기존의 네트워크 장비 뿐만 아니라 가전제품 등에도 IP를 할당하는 Home Networking 기술이 연구되고 있다. 이러한 기술 연구에 발맞추어 다양한 관리대상을 기존의 망 관리 시스템 개발 입장에서 접근하게 된다면 새로운 제품 출시 때마다 제품에 맞는 새로운 NMS를 개발해야 하는 추가 작업이 예상된다.

또한, SNMP와 ICMP만을 이용하여 개발되어지기 때문에 TELNET을 이용할 때 얻을 수 있는 장점을 효과적으로 적용할 수 없다. 예를 들면, 광동축혼합망(HFC : Hybrid Fiber Coaxial Cable)을 이용하는 인터넷 서비스에서, 운영되는 장비 중 케이블모뎀종단시스템(CMTS : Cable Modem Termination System)이라는 장비에는 케이블모뎀(CM:Cable Modem)이 연결되어 있다. 이 장비를 사용할 경우, 연결된 CM 목록을 조회하고 싶다면 SNMP를 이용하여 SNMP get을 CM 수 만큼 수행해야 한다. 그러나, TELNET을 이용할 경우에는 "show

cable modem"이라는 명령어 하나만으로 해결할 수 있다[2,3].

본 논문은 이처럼 다양해지는 관리 대상을 쉽게 추가, 관리할 수 있도록 객체지향기술을 도입하여 관리대상의 범위성(Scalability)을 꾀하였고 관리대상 처리를 위해 SNMP와 TELNET, ICMP등을 내부적으로 사용할 수 있도록 작성된 스크립트를 이용하여 클래스의 메소드를 정의 하였다. 본 논문의 구성은 2장에서 객체 지향 기술에 대해서 간단히 언급하고, 3장에서는 제안 기법을 명세하며, 4장에서는 결론 및 향후 연구 과제에 대해서 논하고 있다.

2. 객체 지향 기술

객체 지향 기술은 소프트웨어 위기 문제를 해결하기 위한 대안으로 개발되었다. 추상화(Abstraction), 캡슐화(Encapsulation), 상속성(Inheritance) 등의 개념을 기반으로 하고 있으며, 확장성과 재사용을 높이는 핵심 기술로 부상했다. 객체 지향 기술에서는 모든 것을 객체(Object)로 표현한다. 객체란 정수, 실수, 복소수, 학생, 자동차 등 사람이 하나의 단위로 인지할 수 있는 대상을 총칭한다. 클래스(Class)는 동일한 특성의 자료 구조와 연산을 갖는 객체들의 집합이다[4].

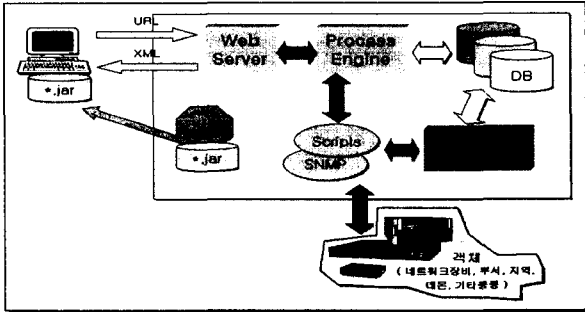
본 논문은 기존의 객체 지향 기술 관련 연구를 적용하여 관리대상을 하나의 객체로 인식하고 네트워크 장비 뿐만 아니라 장비 외적인 요소까지 관리대상으로 표현하며 관리대상에 대한 필요한 정보나 처리 절차를 클래스의 속성과 메소드의 기능을 이용하여 구현하였다.

3. 제안 기법

3.1 시스템 모델

본 논문에서 제안하는 시스템 모델은 클라이언트 / 서버 방식을 따르고 있으며 운영자 인터페이스는 JWS(Java Web Start)를 이용하여

JAR(Java Archive)를 내려 받아 실행한다. 내려 받은 JAR들의 통신은 URL(Uniform Resource Locator)로 호출하고 결과를 XML형태로 받는다. 프로세스 엔진(Process Engine)과 컬렉션 엔진(Collection Engine)은 SNMP를 이용하거나 이미 작성된 스크립트를 이용하여 관리대상으로부터 자료를 수집하거나 설정한다. <그림 1>은 본 논문의 시스템 모델을 나타낸다.



<그림 1> 시스템 모델

3.2 클래스 (Class)

본 논문은 유사한 성질을 가지고 있는 관리 대상들을 총괄적으로 관리하기 위하여 각각의 망 구조와 망 관리대상에서 사용될 수 있는 객체의 기능을 정의하고 이 객체를 미리 생성해 놓아 망 관리에 적용하고자 한다.

클래스는 관리 대상의 모델을 의미한다. 클래스의 특성은 곧 관리 대상의 특성을 규정한다. 네트워크 장비 모델, 인터페이스, 서버, 지역, 부서 등을 관리대상으로 설정하고자 할 때 이들 특성을 클래스에 정의해야 한다. 클래스의 속성(Attribute)은 명칭, 태그(Tag), 인수정보(Parameter), 그리고 클래스가 장비를 나타내는 클래스인지를 알 수 있는 값(Device) 등으로 구성되며, 처리 기능을 담당하는 메소드(Method), 상속 받은 부모 클래스 정보(Parent) 등을 포함한다. <그림 2>는 클래스를 정의한 DTD (Document Type Definition)[5]이다.

```
<?xml version="1.0" encoding="KS_C_5601-1987" ?>
<ELEMENT tag ( #PCDATA ) >
<!ATTLIST tag name NMTOKEN #REQUIRED >
<ELEMENT parent ( #PCDATA ) >
<!ATTLIST parent priority NMTOKEN #REQUIRED >
<ELEMENT class ( parameter+, tag+, parent+, method+ ) >
<!ATTLIST class name NMTOKEN #REQUIRED >
<!ATTLIST class device NMTOKEN #REQUIRED >
<ELEMENT method EMPTY >
<ELEMENT parameter ( #PCDATA ) >
<!ATTLIST parameter name NMTOKEN #REQUIRED >
<!ATTLIST parameter security NMTOKEN #REQUIRED >
```

<그림 2> 클래스 정의의 DTD

3.2.1 메소드 (Method)

클래스의 메소드는 관리대상으로부터 정보를 얻거나 설정하는 기능들을 정의한다. 본 논문에서는 SNMP에서 이용한 기능 구현과 스크립트를 이용하여 기능 구현이 가능하도록 메소드를 설계하였다. <그림 3>과 <표 1>은 메소드에 대한 DTD이다.

3.2.1.1 SNMP를 이용한 메소드 정의

SNMP를 이용할 때에는 일반적인 SNMP get, set 명령어를 이용한다. 그 정의의 내용을 보면 Object명, OID(Object Identifier), SYNTAX, ACCESS를 이용하여 작성된다[1].

```
<?xml version="1.0" encoding="KS_C_5601-1987" ?>
<ELEMENT somes ( method ) >
<ELEMENT method ( value, access, syntax, parameter*, description, control?, expression? ) >
<!ATTLIST method name NMTOKEN #REQUIRED >
<!ATTLIST method class NMTOKEN #REQUIRED >
<ELEMENT value ( #PCDATA ) >
<!ATTLIST value type ( snmp | script ) "snmp" >
<ELEMENT access ( #PCDATA ) >
<ELEMENT syntax ( #PCDATA ) >
<ELEMENT parameter ( #PCDATA ) >
<!ATTLIST parameter name NMTOKEN #REQUIRED >
<ELEMENT description ( #PCDATA ) EMPTY >
<ELEMENT control ( action+ ) >
<ELEMENT action ( #PCDATA ) >
<!ATTLIST action name NMTOKEN #REQUIRED >
<ELEMENT expression ( fire+ ) >
<ELEMENT fire ( getValue, express?, state?, condition?, alarm?, precondition? ) >
<ELEMENT getValue ( #PCDATA ) >
<ELEMENT express ( #PCDATA ) >
<ELEMENT state ( #PCDATA ) >
<ELEMENT condition ( #PCDATA ) >
<ELEMENT alarm ( #PCDATA ) >
<ELEMENT precondition ( #PCDATA ) >
```

<그림 3> 메소드 정의의 DTD

<표 1> 메소드 정의의 DTD 설명

요소	의미
method	속성 class : 소속 클래스명
	속성 name : 메소드명
value	속성 type : SNMP, 스크립트 구분
	속성 autorun : 관리대상을 등록 시 자동으로 메소드를 실행할 것인지 여부
	"snmp"이면 OID, "script"이면 스크립트 파일명
access	접근 권한 읽기, 쓰기, 읽기쓰기
syntax	리턴되는 값의 형
parameter	속성 name : 메소드 인수 명
	메소드 인수명의 기본값
description	메소드에 대한 설명을 나타낸다.
control	access가 write인 경우에 사용되며 값의 별칭
	속성 name : 값에 대한 별칭
action	위 name을 선택하면 SNMP Set 또는 스크립트에 의해 관리대상이 보내지는 값
expression	관리대상으로부터 값을 얻어 왔을 때 운영자가 인식하기 쉬운 형태로 표현하거나 알람 설정
fire	expression의 하위 element
getValue	실제 관리대상으로부터 얻은 값과 비교할 값
express	관리대상으로부터 얻은 값을 운영자가 인식하기 쉬게 표현할 값
state	관리대상으로부터 값에 대한 상태를 나타낸다.
condition	getValue와 관리대상으로부터 얻은 값과 비교 조건
alarm	발생할 알람을 나타낸다.
precondition	alarm을 발생시킬 때 이전에 조회된 값 비교 조건

3.2.1.2 스크립트를 이용한 메소드 정의

스크립트를 이용한 기능의 실행은 SNMP보다 효율적인 방식이다. 스크립트 내의 버그 존재 등을 감안하여 특정 형식에 따라 값을 되돌

러 주도록 설계되었다. 또한 스크립트로 작성된 기능에 대해서는 필요한 인수들을 입력 받을 필요성이 있다. 입력 값은 각 객체의 속성 값을 이용한다. 만일 스크립트에 의한 실행 결과가 <그림 4>의 DTD 형식을 따르지 않는다면 오류로 판단한다. <그림 4>와 <표 2>는 스크립트에 의한 기능을 정의했을 때의 리턴(Return) 형식을 나타낸다.

```
<?xml version="1.0" encoding="KS_C_5601-1987" ?>
<ELEMENT status ( #PCDATA ) >
<ELEMENT sonos ( sonosScriptReturn ) >
<ELEMENT filename ( #PCDATA ) >
<ELEMENT value ( #PCDATA ) >
<ELEMENT sonosScriptReturn ( status, value, filename, msg, xmlMsg ) >
<ELEMENT xmlMsg ( #PCDATA ) >
<ELEMENT msg ( #PCDATA ) >
```

<그림 4> 스크립트 리턴 형식 DTD

<표 2> 스크립트 리턴 형식 DTD 설명

요소	의미
status	스크립트 실행 결과를 나타낸다. "OK" "ERROR" "RETURN" 3가지 중 하나의 값을 준다.
value	관리 대상으로부터 얻은 결과를 나타낸다.
filename	스크립트에 의한 처리로 결과를 특정 파일에 기록했음을 의미하고 이때의 파일명을 나타낸다.
xmlMsg	XML 형태의 값으로 리턴됨을 나타내고 그 값을 가지고 있다.
msg	"ERROR" 리턴일 때 오류가 발생 이유이다.

3.2.2 속성 (Attribute)

클래스의 속성 중 태그는 관리 대상이 갖는 값으로 클래스를 자동으로 설정해 주기 위해 사용된다. 관리대상에 클래스가 설정되지 않는다면 그 관리 대상은 어떠한 자료의 조회나 설정도 이루어 지지 않는다. 클래스마다 태그의 값은 여러 값들의 조합으로 이루어질 수 있게 구현 되었다.

인수정보 속성은 관리 대상의 고유 정보를 표시하기 위해 사용된다. 고유 정보라 함은 SNMP를 이용할 때 사용되는 Get/Set 커뮤니티(Community)나[1], TELNET을 내부적으로 이용한 스크립트인 경우 필요한 유저(User)나 패스워드(Password) 등이 이에 해당된다.

3.3 상속 (Inheritance)

상속은 클래스간 중복되는 요소를 없애기 위해 사용되었다. 상속 시 메소드와 속성 중에서 인수정보만 상속을 받으며 다중 상속이 가능하다. 다중 상속은 중복되는 요소를 최소화 하기 위해 사용되었다.

3.4 다형성 (Polymorphism)

상속 받은 클래스 내에 동일한 메소드나 속성값이 존재하면 자신의 메소드나 속성값이 적용된다. 만약 자신의 것이 없다면 부모 클래스의 것을 적용하는데 각각에 우선순위를 두어 높은 우선 순위를 갖는 클래스의 속성 또는 메소드가 적용되게 구현되었다..

3.5 객체 (Object)

3.5.1 관리대상

관리대상은 객체(Object)이다. 객체는 실제계에 존재하는 어떤 것을 추상화 한 것이다. 망 관리 시스템상의 관리대상도 네트워크 장비, 각종 서버, 데온, 지역, 부서 등등 다양한 형태를 포함하고 있다. 관리 대상은 하나의 클래스에 속하게 되어 클래스에 정의된 메소드를 수행한다. 어떠한 클래스에도 속하지 않은 관리대상은 어떠한 기능도 할 수 없게 되어있다.

3.5.2 범위

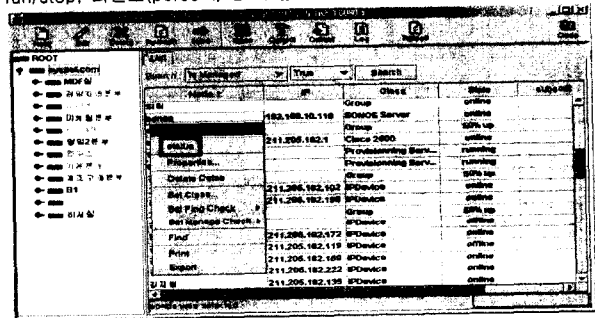
관리대상이 곧 객체이고 객체가 곧 실제계에 존재하는 모든 것을 추상화 한 것이므로 본 논문에서 관리되는 대상은 제한이 없다. 클래스의 정의가 곧 관리대상의 종류를 정의한다. 즉 클래스의 종류에 따라 관리대상의 종류가 결정된다.

3.5.3 확장성

새로운 관리 대상 종류의 추가는 클래스의 추가로 이루어지도록 구현되었다. 새로운 관리 대상 종류의 추가로 인하여 시스템에 대한 추가 작업이 이루어 지지 않고 클래스 정의의 관련 부분만 추가 작업한다.

3.6 구현 (Implementation)

구현 환경에서 UI는 Java를 이용하였고 서버의 운영체제는 Linux, 데이터베이스로 MySQL, 웹 서버로 Apache를 사용하였다. 구현된 UI는 jar로 묶여져서 서버에 존재하고 운영자는 Java Web Start[6]를 이용하여 내려 받아 실행한다. <그림 5>는 관리대상 목록을 보이는 화면으로 클래스 종류가 달라도 이름이 공통적인 메소드 "status"를 보이고 있다. "status"를 실행할 때 각각의 클래스에서 정의한 메소드를 실행하게 된다. "status"의 결과는 클래스에 따라서 on/off, up/down, run/stop, 퍼센트(percent) 값 등을 현재 보이고 있다.



<그림 5> 관리대상 목록 화면

4. 결론 및 향후 연구 과제

본 논문은 객체지향기술을 도입하여 망 관리 시스템을 설계, 구현하였다. 객체지향기술의 도입은 관리 대상의 다양성과 범위를 충족시켰다. 클래스의 메소드 기능에 SNMP와 TELNET, ICMP등을 수행할 수 있도록 내부적으로 적용한 스크립트 또는 프로그램을 통하여 보다 효율적인 관리가 가능하도록 구현하였다. 또 새로운 관리대상이 출현했을 때 클래스만을 추가하여 해결할 수 있도록 구현되었다.

클래스를 구성하는 이미지 또한 망 관리에서 중요한 위치를 차지하고 있어 향후 연구 과제로 남겨두었다.

5. 참고문헌

- [1] W.Stallings, "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2," Third edition, Addison-Wesley, Reading, MA, USA, 1999.
- [2] CMTS MIB & CLI. <http://www.cisco.com>
- [3] DOCSIS Specification, <http://www.cablemodem.com/specifications/specifications10.html>
- [4] 이경환, 황선명 등, 소프트웨어 공학, 청문각, 1992.
- [5] W3C, Extensible Markup Language (XML) 1.0, W3C Recommendation, October 2000, <http://www.w3.org/TR/REC-xml>.
- [6] Java Web Start Home, <http://java.sun.com/products/javawebstart>