

네트워크 관리를 위한 SNMP MIB의 XML 변환 알고리즘

윤정혁⁰ 주홍택 홍원기
포항공대 정보통신학과, 컴퓨터공학과
{infobank, juht, jwkhong}@postech.ac.kr

An algorithm of SNMP MIB to XML translation for NMS

Jeong-Hyuk Yoon⁰, Hong-Taek Ju, James Won-Ki Hong
GSIT, CSE, Pohang University of Technology

요 약

인터넷에서 현재 가장 널리 쓰이는 네트워크 관리 프로토콜은 SNMP로써 거의 모든 네트워크 장치에서 SNMP(Simple Network Management Protocol) agent를 지원하고 있으며 관리정보가 SNMP MIB(Management Information Base)으로 정의 되어 있다. 최근 SNMP 기반 관리 시스템의 확장성과 효율성의 문제점을 해결하려는 시도로 XML(eXtensible Markup Language)을 이용한 네트워크 관리 시스템에 관한 연구가 진행되고 있다. 이와 같은 웹 기반의 네트워크 관리 시스템은 XML로 표현된 관리정보를 사용한다. 기존의 SNMP agent가 장착된 많은 네트워크 장비를 XML 기반의 manager에서 통합관리하기 위해서는 SNMP MIB으로 정의되어 있는 기존의 관리정보를 XML로 변환해서 표현하는 것이 필수적인 요구 사항이다. 본 논문에서는 SNMP MIB을 XML로 변환하는 효율적인 알고리즘을 제시하였다. 제시한 알고리즘의 검증을 위하여 변환기를 구현하였으며, 이를 이용하여 SNMP 메시지를 XML 문서로 상호 변환하는 게이트웨이를 구현하였다.

I. 서론

대부분의 인터넷 네트워크 장치들은 IP 기반의 네트워크 관리를 위한 표준 프로토콜인 SNMP[1]에 의해 관리 되고 있다. SNMP는 간단하면서도 구현이 쉬운 장점을 가지고 있다. 그러나 SNMP 기반 네트워크 관리 시스템은 확장성과 효율성에서 단점을 가지고 있기 때문에 최근 인터넷의 확장으로 거대한 네트워크를 관리하는데 한계점이 있다[2]. SNMP 기반 네트워크 관리 시스템의 단점을 보완하기 위해 XML[4]을 네트워크 관리에 이용하고자 하는 연구들이 활발히 진행되고 있다[2,3]. 웹 기반 네트워크 관리 시스템에서 사용자 인터페이스, 데이터의 교환 및 저장방법으로 XML을 사용한다. 관리정보

를 XML로 표현하면 사용자가 문서 구조를 쉽게 정의할 수 있고, HTTP를 이용한 XML 문서의 전송으로 플랫폼에 관계없이 이기종의 시스템 간에도 쉬운 데이터 교환이 가능하다는 장점이 있다[6].

XML 기반의 네트워크 관리 시스템에서는 관리정보를 XML로 정의하고 manager-agent간에 HTTP 상에서 XML문서를 교환하는 방식으로 데이터의 전달을 한다[3]. 이러한 시스템에서 agent쪽에서는 XML 문서를 처리하기 위해 내장 웹서버(Embedded Web Server)를 사용하여 HTTP 통신을 지원한다.

XML 기반의 manager 시스템에서 기존의 SNMP agent가 장착된 네트워크 장비를 동시에 관리하기 위해서는 MIB으로 정의 된 관리정보를 XML로 변

환하는 것이 필수적인 일이다. 이는 XML기반 manager가 SNMP기반 agent를 관리하기 위해서는 agent에서 사용하는 관리 정보인 MIB을 manager가 알 수 있도록 XML로 표현해 주어야 하기 때문이다. 이러한 변환을 기계적으로 자동 변환이 되도록 하면 다양한 agent의 각기 다른 MIB에 대해서 manager에서는 추가적인 프로그램이 없이 데이터 수집을 할 수 있는 방법을 제공한다.

IP기반의 인터넷 네트워크 관리는 대부분이 SNMP를 기반으로 개발되어 왔기 때문에 SNMP MIB에는 표준화된 관리정보가 잘 구축되어 있다. 본 논문에서는 이러한 SNMP MIB을 XML로 변환하는 알고리즘을 제시하고, 자동변환기를 구현하며, XML기반의 통합 네트워크 관리 시스템에 적용한 사례를 살펴본다. 서론에 이어 2장에서 관련 연구를 살펴보고, 3장에서는 SNMP MIB을 XML로 변환하는 알고리즘을 제시한다. 4장에서는 변환기 구현을 설명하고, XML 기반의 네트워크 관리 시스템에 적용한 시스템 구현을 살펴본 다음, 5장에서 결론을 맺는다.

II. 관련연구

SNMP는 관리정보를 ASN.1[11]으로 기술한 SMI(Structure of Management Information)의 규칙에 따라 정의하고 있으며 정의된 관리정보를 MIB이라고 한다. SMIv1(SMI version 1)은 RFC1155[12]에, SMIv2(SMI version 2)는 RFC2578[13]에 각각 정의되어 있다. SMI에는 MIB 정의를 위해 사용되는 매크로 정의와 데이터형의 정의가 있으며 이 정보를 기반으로 MIB을 기술한다.

SNMP는 간단하고 쉬운 네트워크 관리 방법을 제시하지만 날로 증가하는 복잡한 네트워크에서 생성되는 많은 관리 데이터를 전달하고 처리하는데 한

계점을 가지고 있다[2,3].

XML은 W3C(World Wide Web Consortium)에서 웹상에서 문서교환을 위한 표준으로 1998년에 제정한 메타 언어(언어의 기술을 위한 언어)이다. XML은 구조화된 문서 정의를 위해 태그를 자유롭게 정의할 수 있는 SGML(Standard Generalized Markup Language)[7]의 확장성을 가지고 있으면서도 SGML처럼 복잡하지 않고, HTML처럼 인터넷 상에서 손쉽게 하이퍼미디어 문서를 제공할 수 있는 장점을 가지는 웹 표준 문서 포맷이다.

XML은 문서의 내용과 표현방식을 분리하고 있고 문서의 처리를 위한 표준 방법을 정의하고 있다. XML 문서의 정보 모델 정의는 DTD(Document Type Definition)[21], XML Schema[22]를 사용하여 기술하고, 문서의 표현은 CSS(Cascading Style Sheet)[14]나 XSL(Extensible Stylesheet Language)[15]을 이용하여 나타낸다. 표현법의 분리에 의해 동일한 XML 문서를 가지고 다양한 모양으로 보여줄 수 있고, 필터링이나 정렬을 통해 원하는 데이터만 보여주거나 다른 형식의 XML문서로 변환도 가능하다. 또한 문서의 처리를 위한 표준 인터페이스인 DOM(Document Object Model), SAX(Simple API for XML)등의 API를 정의하고 있어서 문서의 파싱과 처리에 표준을 제공하고 있다. 또한, Xpath(XML Path Language) 표준은 문서에서 특정 부분을 지정하는 방법으로 문서에서 필요한 부분을 찾는 데 유용하다.

이러한 XML의 문서처리 및 표현의 다양한 기능들과 광범위한 데이터 표현능력은 관리정보와 인터페이스 표준으로서 커다란 잠재력을 가지고 있다.

SNMP MIB을 XML로 변환하는 방법에 대한 기존의 연구로는 J.P. Martin-Flatin이 제시한 SNMP MIB

을 XML DTD로 매핑하는 예가 있고[3], Bell 연구소의 CORBA/SNMP 게이트웨이 개발 프로젝트에서 연구한 SNMP MIB의 XML 표현에 관한 연구가 있으며[8], Frank Strauss의 ‘libsmi’라는 MIB을 처리하는 라이브러리가 있다[9]. 그리고, IBM 연구소에서 ASN.1을 XML로 변환하는 연구도 있었다[10]. 그러나, 이전 연구들은 DTD를 사용한 표현을 하고 있어서 데이터형의 지원을 하지 못하고 있으며, 완성된 변환 알고리즘에 대한 언급은 하지 않고 변환결과만 보여주고 있거나 특정 용도를 위한 변환을 하고 있어서 변환 후에 MIB의 정보가 완전히 보존되지 않고 있다.

III. 변환알고리즘

본 연구에서는 SNMP MIB의 모든 정보를 손실이 없이 그대로 XML로 변환하고, 변환 전·후의 문서 구조를 동일하게 유지하여 XML 문서가 가지는 데이터를 처리하기에 효율적인 방법이 되도록 하였다.

1. 문서구조 및 데이터형

MIB을 XML로 변환하기 위해서는 관리정보의 구조와 데이터형에 대한 변환을 고려해야 한다.

문서 구조에서 SNMP MIB은 SMI에 근거하여 트리 구조로 구성이 되어 있고, XML 문서도 트리 구조로 정의 될 수 있으므로 문서의 구조를 바꾸지 않고 그대로 트리 구조로 변환을 하는 것이 manager-agent간에 데이터 전달을 위해 효과적이다. 따라서 MIB의 각 노드를 XML의 엘리먼트(Element)로 변환하고, MIB 노드 내부의 절들은 XML 엘리먼트의 내부에 속성(Attribute)으로 변환한다. 데이터형의 변환에서는 MIB에 정의된 데이터형을 그대로 XML로 정의 하기 위해서 W3C 표준으로 제정된 XML Schema를 이용하였다. 다음 절에서 항목별로 자세한 설명을 한다.

2. 매크로 변환

OBJECT-TYPE 매크로는 RFC1155-SMIv1에 정의된 매크로이며 MIB의 테이블 노드나 데이터 노드를 정의하는 대표적인 매크로 이다. 이 매크로의 변환 방법을 통해 MIB의 노드를 XML의 엘리먼트로 변환하는 방법을 알 수 있다. OBJECT-TYPE 매크로를 이용한 MIB 노드 표현의 일반형은 다음과 같다.

```
nodeName OBJECT-TYPE
  SYNTAX "syntaxType"
  ACCESS "accessType"
  STATUS "statusType"
  DESCRIPTION "descriptionText"
  REFERENCE "referenceType"
  INDEX "indexList"
  DEFVAL "defaultValue"
 ::= { parentNodeName nodeNumber }
```

위의 MIB 노드 정의를 XML Schema를 이용하여 문서구조를 정의하면 다음과 같다.

```
<xsd:element name="nodeName">
<xsd:complexType>
<xsd:simpleContent>
<xsd:restriction base="xsd:string">
  <xsd:attribute name="oid" type="xsd:string"
    use="fixed" value="oidValue"/>
  <xsd:attribute name="access" type="xsd:string"
    use="fixed" value="accessType"/>
  <xsd:attribute name="status" type="xsd:string"
    use="fixed" value="statusType"/>
  <xsd:attribute name="description" type="xsd:string"
    use="fixed" value="descriptionText"/>
  <xsd:attribute name="reference" type="xsd:string"
    use="fixed" value="referenceType"/>
  <xsd:attribute name="index" type="xsd:string"
    use="fixed" value="indexList"/>
  <xsd:attribute name="defval" type="xsd:string"
    use="fixed" value="defaultValue"/>
</xsd:restriction>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
```

노드명에 해당하는 ‘nodeName’을 XML에서는 엘리먼트명으로 사용하고, ‘ACCESS’, ‘STATUS’ 등의 내부 절들은 속성으로 정의한다. 단, ‘SYNTAX’ 절의 내용인 ‘syntaxType’에 따라서 이 노드의 데이터형이 결정되므로 이 절은 속성으로

지정하지않고 데이터형을 나타내는데 사용한다. 그리고 'oid'라는 새로운 속성을 정의하고 이것의 값은 해당 노드의 OID값을 가지도록 하였다. 이 값은 나중에 SNMP메시지와 XML문서간의 데이터 변환에서 키로 사용된다.

3. 데이터형 변환

MIB에서 SYNTAX절의 내용에 해당하는 데이터형을 XML로 변환 시에 표현해 주는 방법은 DTD로는 표현이 불가능하지만 XML Schema를 사용하면 정수형, 문자열형, 특수한 정의형 등 모든 데이터형에 대한 정의가 가능하며, 따라서 데이터 교환에서 생성된 XML 문서의 Validation을 강력하게 지원하게 된다. SMI에 정의된 데이터형 중에서 기본형은 XML Schema의 기본형과 일치하게 변환하고, APPLICATION 형과 MIB에 정의된 데이터형과 같이 특수한 경우는 표 1과 같이 새로운 데이터형을 정의해서 사용할 수 있다.

표 1. 복잡한 데이터형 변환 예

MIB	XML Schema
IpAddress	<pre><xsd:simpleType name="IpAddress"> <xsd:restriction base="xsd:string"> <xsd:pattern value="((([1-9]?[0-9]) ([0-9] ([0-9])?2[0-4] ([0-9] 25[0-5])\.)\.)\.)\{3\}([1-9]?[0-9]) ([0-9] ([0-9])?2[0-4] ([0-9] 25[0-5])\.)\{3\})"/> </xsd:restriction> </xsd:simpleType></pre>
DisplayString (SIZE (0..255))	<pre><xsd:simpleType name="DisplayString_0_255"> <xsd:restriction base="xsd:string"> <xsd:minLength value="0"/> <xsd:maxLength value="255"/> </xsd:restriction></xsd:simpleType></pre>
INTEGER { up(1), down(2), testing(3) }	<pre><xsd:complexType> <xsd:simpleContent> <xsd:restriction base="xsd:int"> <xsd:enumeration value="1"/> <xsd:enumeration value="2"/> <xsd:enumeration value="3"/> </xsd:restriction> </xsd:simpleContent> </xsd:complexType></pre>

표 1에서 첫번째 IpAddress 형은 IP 주소의 표현식을

나타내고 있고, 두 번째 예는 길이를 지정하는 문자열을 나타내고 있으며, 세 번째 예는 열거형의 데이터형을 지정하는 방법을 나타내고 있다. 이러한 표현들은 DTD로 나타낼 수 없다.

IV. 변환기 구현 및 적용

1. 변환기 구현

변환기는 SNMP MIB을 입력으로 받아서 앞 장에서 제시한 알고리즘을 이용하여 XML Schema로 변환하는 자동변환 프로그램이다. 변환기는 JSP(Java Server Page)를 이용하여 웹에서 관리가 가능하도록 자바를 이용하여 구현하였다.

먼저 MIB 파일을 읽어 들여 일련의 토큰으로 나누고 구문을 분석하여 하나의 MIB노드가 발생하면 이것을 클래스로 대응시켜서 클래스 테이블에 저장을 한다. 사용자가 정의한 데이터형 정의는 따로 하나의 테이블에 저장을 한다. IMPORT절에 의해 다른 MIB을 조회하는 경우에는 해당하는 MIB을 먼저 파싱하여 import된 노드나 데이터형을 먼저 해당 테이블에 저장하여 둔다. 모든 MIB의 구문 분석이 끝나고 MIB 노드에 해당하는 클래스 테이블이 완성되면 이것을 이용하여 DOM을 구성하고, XML Schema 출력을 한다. DOM 처리를 위해서는 Apache에서 제공하는 자바 패키지인 Xerces[19]와 Xalan[20]을 이용하였다. 이렇게 하여 구현된 DOM 트리는 데이터의 처리를 위해 사용이 되며, XML Schema는 처리될 문서의 Validation을 위해 사용된다.

2. XML 기반 네트워크 관리 시스템에 적용

데이터의 처리를 XML 기반으로 하는 manager가 SNMP agent를 가지고 있는 장비를 관리하기 위해서는 SNMP를 이용하여 데이터를 요청하고, 수집된

데이터(SNMP 메시지)를 XML 문서로 변환하여 표현하는 모듈이 필요하다. 이러한 기능을 하는 모듈은 그림 1에서 ‘SNMP module’이라고 표시된 부분으로 본 논문에서 구현하고자 하는 부분이다.

먼저 앞절에서 설명한 변환기를 이용하여 agent가 사용하는 MIB을 XML로 변환하고 DOM을 구성한다. XML 기반 manager에서 요청메시지인 HTTP GET 메시지를 받는다.

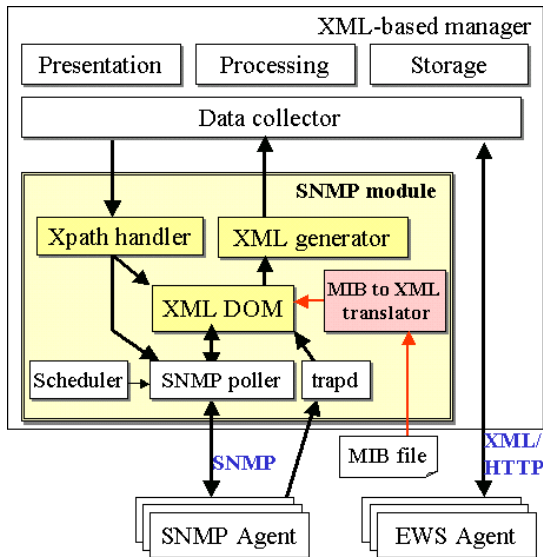


그림 1. XML기반 네트워크 관리 시스템에 적용

Manager의 요청을 받은 ‘Xpath handler’는 snmp-get 메시지로 변환하여 SNMP agent에 요청을 한다.

‘SNMP module’에서 HTTP메시지와 SNMP 메시지를 상호 변환하는 매핑 방법은 표 2에 정리하였다.

표 2. 메시지 변환

SNMP	HTTP
get	http://manager_addr/snmp.jsp?operation=get&community=public&xpath=//system
set	http://manager_addr/snmp.jsp?operation=set&community=public&xpath=//system&value=valueString

Manager가 요청할 때 HTTP 메시지는 표 2에 나타

난 바와 같이 operation, community, xpath, value의 4가지 파라미터로 구성 된다. ‘operation’인자는 동작이 get인지 set인지를 구분하고, ‘community’인자는 SNMP agent에서 사용하는 community값을 전달한다. ‘xpath’인자는 요청하는 SNMP 메시지의 OID에 해당하는 노드명을 XML에서 사용하는 Xpath를 사용하여 표현한다. ‘value’인자는 SNMP set 명령의 경우에 agent에 전달할 값을 저장한다.

‘Xpath handler’에서 요청을 이어받은 ‘SNMP poller’는 SNMP operation으로 SNMP agent에 데이터를 요청한다. 반환되는 SNMP response 메시지는 OID-Value 쌍으로 이루어지므로 미리 작성된 XML DOM 트리의 해당 OID에 대응되는 노드에 값을 입력하게 된다.

데이터 수집이 완료되면 ‘XML generator’는 DOM으로부터 XML 문서를 생성하고 이것을 manager로 전달하게 된다.

SNMP trap 메시지는 전용 데몬인 ‘trapd’에 의해 수신이 되어 DOM의 해당 OID에 해당하는 노드의 값을 변경하고 XML 문서로 표현되어 manager로 전달이 된다.

따라서, XML기반의 manager에 SNMP module을 추가함으로써 기존의 SNMP agent를 agent의 변경이 없이 그대로 두고 사용하면서 관리를 할 수 있게 된다.

V. 결론

본 논문에서는 XML 기반의 네트워크 관리 시스템에서 기존의 SNMP agent를 통합 관리하기 위한 방법으로 SNMP MIB을 XML로 변환하여 관리정보를 재사용하는 방법을 제시하였다. 변환 알고리즘에 따라 변환기를 구현하고, 이를 이용하여 SNMP 메시지와 HTTP 메시지를 변환하는 게이트웨이를

구현하고 적용하였다.

뛰어난 호환성과 사용하기 쉬운 XML의 특성 때문에 XML을 이용한 데이터의 통합은 앞으로 더욱 가속화 될 전망이다. 특히 네트워크 및 시스템 관리를 위한 정보의 전송에서 이기종의 시스템간에 정보 전달을 위해 XML을 미들웨어로 사용할 경우 SNMP MIB을 XML로 변환하는 것은 표준적인 변환 방법으로 연구 가치가 있는 것이다.

향후 과제로는 'SNMP module'을 manager와 분리하여 계층적인 분산구조의 관리 시스템에 적용하는 연구가 필요하다.

참고문헌

- [1] W. Stallings, "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2," Third edition, Addison-Wesley, Reading, MA, USA, 1999.
- [2] J.P. Martin-Flatin. "Web-Based Management of IP Networks and Systems", Ph.D. thesis, Swiss Federal Institute of Technology, Lausanne (EPFL), Oct 2000.
- [3] 주홍택, 홍원기 "내장형 웹 서버 기반의 네트워크 관리 구조 설계", Proc. of KNOM 2000 Conference, Taejeonn, May, 2000, pp. 132-138.
- [4] W3C, Extensible Markup Language (XML) 1.0, W3C Recommendation, October 2000, <http://www.w3.org/TR/REC-xml>.
- [5] W3C, "HTML 4.0 Specification," Internet Draft, HTML Working Group, Apr. 1998.
- [6] N. Bradley, "The XML Companion," Addison-Wesley, Harlow, UK, 1998.
- [7] International Organization for Standardization, "ISO 8879: Standard Generalized Markup Language (SGML)", 1986.
- [8] Subrata Mazumdar, "CORBA/SNMP Gateway", Bell Labs, <http://www.bell-labs.com/project/CorbaSnmp>.
- [9] Frank Strauss, "A Library to Access SMI MIB Information", <http://www.ibr.cs.tu-bs.de/projects/libsmi/>.
- [10] Imamura, T. and Maruyama, H., "Mapping between ASN.1 and XML", Applications and the Internet, 2001. Proceedings, IEEE, 2001, pp. 57 - 64.
- [11] ITU-T, "Specification of Abstract Syntax Notation One(ASN.1)", ITU-T Rec. X.208, 1998.
- [12] M. Rose, K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets," IETF, RFC1155, May 1990.
- [13] K. McCloghrie, et al, "Structure of Management Information Version 2 (SMIv2)", IETF, RFC2578, April 1999.
- [14] W3C, "Cascading Style Sheets, level 1", Recommendation, Dec 1996, revised Jan 1999, <http://www.w3.org/TR/REC-CSS1>.
- [15] W3C, "Extensible Stylesheet Language (XSL) Version 1.0", Recommendation, Oct 2001, <http://www.w3c.org/Style/XSL/>.
- [16] W3C, "Document Object Model (DOM) Level 2 Core Specification", Nov 2000, <http://www.w3.org/TR/DOM-Level-2-Core/>.
- [17] Simple API for XML 2.0, 1999 <http://www.saxproject.org/>.
- [18] W3C, "XML Path Language (XPath) Version 1.0", Recommendation, Nov 1999' <http://www.w3.org/TR/xpath>.
- [19] Apache XML project, "Xerces Java parser", <http://xml.apache.org/xerces-j/>.
- [20] Apache XML project, "Xalan - XSLT processor", <http://xml.apache.org/xalan-j/index.html>.
- [21] W3C, Extensible Markup Language (XML) 1.0 - DTD, W3C Recommendation, October 2000, <http://www.w3.org/TR/REC-xml#dt-doctype>.
- [22] W3C, "XML Schema Part 0: Primer", Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-0/>.