基于 XML 网络管理的 SNMP/XML 翻译网关的开发*)

钱柱中　谢　立

(南京大学软件新技术国家重点实验室　南京 210093)　(南京大学计算机系　南京 210093)

摘　要　作为弥补基于 SNMP 的网络管理缺陷的一种变通方案,最近已提出了基于 XML 的网络管理,但现存的很多网络设备都有 SNMP 代理。为了集成 XML 和 SNMP 网络管理,必须提出 XML/SNMP 网关以进行 SNMP 到 XML 的信息翻译。本文首先介绍了使用 XML 进行网络管理的优点,然后讨论了规范翻译和交互翻译,也给出了这种网关的体系结构。

关键词　网络管理,SNMP,MIB,XML,SOAP

Development of SNMP/XML Gateway for XML-Based Network Management

QIAN Zhu-Zhong　XIE Li

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

(Department of Computer Science & Technology, Nanjing University, Nanjing 210093)

Abstract　Recently XML-based network management has been proposed as an alternative to complement the shortcomings of SNMP-based network management. However, most existing network devices are already embedded with SNMP agents and managed by SNMP managers. In order to integrate network management with XML and SNMP, an XML/SNMP gateway is proposed to translate SNMP to XML. This paper firstly presents the advantages of using XML for the management of network. Then, specification translation and interaction translation are both discussed, the architecture of the gateway is also presented in this paper.

Keywords　Network management, SNMP, MIB, XML, SOAP

1　引言

自 1988 年 IETF 公布 SNMP[1]和 SMI 以及第一个被管理对象的定义以来,SNMP 协议作为网络管理的标准取得了长足的发展,IP 网络的管理基本上都依赖于 SNMP 协议。SNMP 被广泛地应用于管理互连网中的设备,它的简单性使它很容易地在小平台中实现,因此,很多网络设备都有 SNMP 代理。然而,基于 SNMP 网络管理系统(SNMP-NMS),由于其可扩展性(Scalability)和效率(Efficiency)的问题[2],已经不再适应于目前高速发展的 IP 网络的管理。文[3]提出了一个新的基于 XML 的网络管理体系结构 XNM(XML-based Network Management),在这种结构下,XML 将取代 SNMP 被用于管理信息建模和管理者与代理之间的通信。

为了能够继续使用大量存在的被管理网络设备的 SNMP 代理,同时也要采用 XML 先进的网络管理技术,以克服 SNMP 的缺点,我们需要一个 XML 和 SNMP 之间进行规范和交互信息转换的翻译网关,从而使得 XML 管理端能够管理 SNMP 代理。

本文首先简单介绍了基于 XML 的网络管理,阐明了使用 XML 进行网络管理的优势,并介绍了几种 XML 和 SNMP 集成的方法;然后重点介绍了 XML/SNMP 翻译网关,包括规范的翻译、交互的翻译和翻译网关的结构。

2　基于 XML 的网络管理

2.1　XML 简介

XML[4],或称为可扩展标记语言(Extensible Markup Language),是由 W3C 创建,作为在 Web 中交换文件的标准。XML 是基于 SGML——标准通用标记语言(Standard Generalized Markup Language)的,因此 XML 可以像 HTML 一样方便地使用 HTTP 来传送,并且具有 SGML 一样的可扩展性。XML 的语法是基于文本的,对于计算机和人都是可读的。它很灵活,能够扩展,能够在不改变现有文件结构的基础上,增加新的标签。通过 Unicode,XML 也提供各种语言的支持。同时,它能够在不同的平台和设备之间,使得数据方便地重用。如此广泛的文件和数据的表达能力,XML 将有可能成为用于信息管理的标准接口。

2.2　XML 网络管理的优点

1、管理信息可以用 XML 文档来表示　SNMP 的框架集中于简单的管理协议和少量管理数据用于管理,所以在管理端和代理之间传送大块数据是很困难的。而且还没有一个处理模型能通过一系列协议操作来确保数据的完整性。当完整的一系列管理数据用一个 XML 文档表示之后,XML Schemas[5]能够对数据格式进行检查,可以使用 HTTP 协议传送数据,能够支持在所有的软件和硬件平台间的数据转换。另外,通过使用 HTTPS 可以达到增强安全性的效果。

2、有现成的 DOM[6]、XPATH[7]等 API 能够用于访问管理数据　使用这些标准的 API 有很多好处。用 DOM 解析器解析管理信息能够生成一个 DOM 树。基于 XML 的管理端能够通过 XPATH 来访问 DOM 树中选择的节点。通过 DOM,我们可以访问 XML 文档中任意一个元素,因此我们可以通过使用 DOM 操作 XML 文档的方式,来执行像增加新设备和设置管理服务器的配置信息等的操作。能够很方便地抽

*)本文得到国家"863"高技术(NO:2003AA144010)经费资助。钱柱中　博士生,研究方向:信息安全。谢立　教授,博导,研究方向:分布式计算,信息安全。

取出必要的管理数据和分析数据。因此，可以通过 DOM 来分析管理信息。

3. 通过 XSL[8] 和 XSLT[9] 能够方便地提供用户管理界面　一个标准的处理管理信息的模型是，使用 XSL 和 XSLT 将有效的 XML 数据转换为 HTML 或其他 XML 文档。使用软件支持将 XML 转换为 HTML 或者其他显示格式，这样能够很容易地提供基于 Web 的用户管理界面。

4. 高级管理操作能够用 WSDL[11] 来定义，并通过 SOAP[10] 来调用　SNMP 的表格操作中，通过 RowStatus 对象产生或删除一行都是很复杂的，但是如果用更高层的操作，就会很方便了。这些操作必须由 WSDL 很好地为 SOAP 函数进行定义。

2.3　基于 XML 的网络管理的研究

基于 XML 的网络管理系统采用了一种嵌入式的 Web 服务(EWS)——许多设备采用嵌入式的 Web 服务来管理网络设备。被管理设备配备一个具有 EWS 的基于 Web 的管理代理，管理端通过与这个代理的交互来管理设备。管理端和代理之间的交互如图 1 所示[12]。图中(A)描述了管理端用 HTTP GET 请求向代理请求信息，代理将相应的信息回应给管理端。图中(B)管理端发送一个 HTTP POST 消息来控制代理，代理做出相应的动作之后对此响应。图中(C)显示了代理向管理端发送一些突发事件的报告。以上三种方式执行的功能类似于 SNMP 的三种基本工作方式：get、set 和 trap。这个图说明了建立在 HTTP 传输协议上的基于 XML 的网络管理方式。

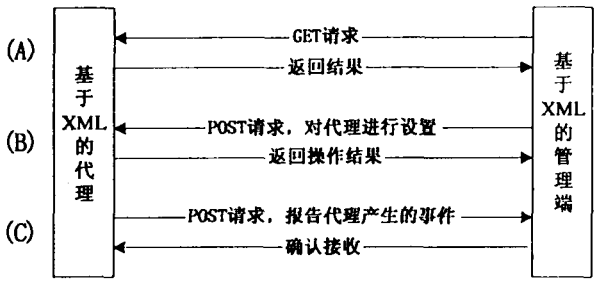

图 1　基于 Web 的管理端和代理之间的交互

为了将 XML 网络管理应用到目前的网络管理，可以有 4 种方式[13]：(1)保留 SNMP 的管理端和代理，只是在管理端，将管理信息转换为 XML 格式，并通过 XSL 将 XML 转换为 HTML，由 HTML 提供基于 Web 的用户管理界面；(2)基于 XML 的代理和基于 SNMP 的管理端；(3)基于 XML 的管理端和基于 SNMP 的代理；(4)基于 XML 的管理端和代理。图 2 表示了这四种方式的组织结构。

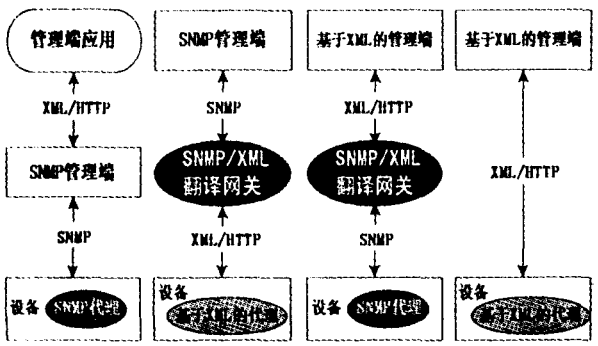

图 2　XML 和 SNMP 集成网络管理

由于 SNMP 在网络管理的广泛使用，每一个网络设备基本都配备有基于 SNMP 代理，因此为了利用 XML 网络管理的优点，同时又不必修改大量的基于 SNMP 的代理，我们一般采用第 3 种方式实现 XML 的集成网络管理。而最后一种方式应该是今后网络管理的趋势，最终 SNMP 将会完全被 XML 所取代。

为了能够使得 XML 管理端和 SNMP 代理之间正常的通信，中间需要一个 XML/SNMP 的翻译网关，实现 SNMP 管理数据到 XML 格式文件的映射(规范翻译)，和将基于 SNMP 格式消息转换为基于 XML 的消息传送到管理端(交互翻译)，反之亦然。

3　SNMP MIB 翻译为 XML 格式

3.1　规范翻译算法

使用 SNMP 协议进行管理系统的基础是被管理元素信息数据库，即 MIB。每个被管理资源由一个对象来表示，MIB 就是这些对象的结构化集合，MIB 的本质是一个树型结构的数据库，通过读取 MIB 中对象的值，网络管理实体可以监视系统中的资源，也能够修改这些值来控制系统中的资源。本节，我们将解释如何把 SNMP MIB 翻译为 XML Schema 的转换算法用于 SNMP/XML 翻译网关，这个翻译过程是规格说明的翻译，以此来生成同 MIB 的树状结构对应的 DOM 树。

为了进行网络管理，我们所关心的是 MIB 中的每个对象，以及这个 MIB 的结构。因此，在 MIB 到 XML 的转换中，将对原有 MIB 进行相应的处理，提取有用的信息而忽略某些不用的信息。J. P. Martin-Flatin 提出了 SNMP MIB 到 XML 转换的模型[14]，分为模型级映射(Model-level mapping)和元模型级映射(Metamodel-level mapping)。模型层映射是为每个特定的 SNMP MIB 建立 DTD 或 XML Schema，而元模型层映射是为所有的 SNMP MIB 建立通用的 DTD 或 XML Schema。这里，我们采用的方式是将 MIB 文件中的节点进行分类，分别定义这几类节点的 XML Schema，最终生成 XML 文件存储整个 MIB 文件库的信息。这样，XML Schema 的定义不会过于复杂，同时节点的分类也为交互翻译提供了很大的方便。

SNMP MIB 文件是以模块的形式定义的，在 MIB 树中这些都是以中间节点的形式存在。相应地，通过 XML 生成的 DOM 树中这类节点也还是以中间节点存在，我们把它归为 node 类型，翻译后的 XML 只记录 OID 这个属性。表 1 先给出了 MIB 库中模块对应的 XML Schema。

表 1　MIB 模块定义和相应的 XML Schema

MIB	XML Schema
ModuleName DEFINITIONS ::= BEGIN IMPORTS… FROM othermodule Name (模块具体内容) END	⟨xsd:element name = "node"⟩ ⟨xsd:attribute name = "name" type = "xsd:string" use = "fixed" value = "ParentNodeName"⟩ ⟨xsd:complexType⟩ ⟨xsd:sequence⟩ (模块具体内容) ⟨/ xsd:sequence⟩ ⟨xsd:attribute name = "oid" type = "xsd:string" use = "fixed" value = "Parent_oid" /⟩ ⟨/ xsd:complexType⟩ ⟨/ xsd:attribute⟩ ⟨/ xsd:element⟩

在 XML Schema 中定义 ParentNodeName 元素时，引用了它的子元素 NodeName，如果 NodeName 也是模块定义，则定义它时也要引用它的子元素的定义。这样按 MIB 树的树状结构，从上层节点一层一层向下引用，一直引用到被管理对

象。由此生成了与 MIB 树相对应的 DOM 树结构。

MIB 中被管理对象分为 4 类：表、行对象（表入口）、列对象、标量，我将节点分别定义为：table、row、column 和 scalar，这样就可以标识 MIB 树中的每个节点。加上模块定义，实际上我们一共定义了 5 类节点。限于篇幅，表 2 只列出了标量相对应的 XML Schema。

表 2　MIB 树中标量的定义和相应的 XML Schema

MIB	XML Schema
NodeName OBJECT-TYPE SYNTAX "syntax Type" ACCESS "access-Type" STATUS "status-Type" DESCRIPTION "descriptionText" REFERENCE "referenceType" INDEX "index-List" DEFVAL "defaultValue" ::= { parentNodeName nodeNumber}	〈xsd:element name = "scalar"〉 〈xsd:complexType〉 〈xsd:simpleContent〉 〈xsd:restriction type = "SyntaxType"〉 〈xsd:attribute name = "name" type = "xsd:string" use = "fixed" value = "NodeName"〉 〈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 = "Description-Text" /〉 〈xsd:attribute name = "Reference" type = "xsd:string" use = "fixed" value = "ReferenceType" /〉 〈/ xsd:restriction〉 〈/ xsd:simpleContent〉 〈/ xsd:complexType〉 〈/ xsd:element〉

许多对象中的 Syntax 表示中会用到一下数据类型。这些数据类型同样的可以转换为 XML Schema。表 3 列举了 SMIv2data 中的 IpAddress 类型和用户自己定义类型 Enumerated INTEGERs 的 XML Schema 的定义。

表 3　MIB 中数据类型和 XML Schema 的转换

MIB definition	XML Schema definition
IpAddress::= OCTET STRING (SIZE (4))	〈xsd:simpleType name = "IpAddress"〉 〈xsd:restriction base = "xsd:string"〉 〈xsd:pattern value = "([0-255].){3}[0-255]" /〉 〈/ xsd:restriction〉 〈/ xsd:simpleType〉

Enumerated INTEGERs
SYNAX
INTEGER {
Up (1)
Down (2)
Testing (3)
}
	〈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〉

以上介绍了将 SNMP MIB 文件转换为 XML 格式的方法，MIB 中的每个节点将对应转换成 XML 的节点，节点的类型将作为元素的名字，其他所有的项作为元素的属性，最后加入 OID 属性用以标识 MIB 节点。

3.2　规范翻译的设计和实现

MIB 翻译模块使用上节描述的翻译算法将把 SNMP MIB 自动转换为 XML Schema。本模块的输入是 SNMP MIB 库文件，输出为 XML DOM 树结构和 XML Schema 文件。图 3 为翻译模块结构。

MIB 读取程序读取 SNMP MIB 文件库，使用 SMI 和 ASN.1 的语法规则检查 MIB 库的正确性。MIB 节点树状结构生成器将读取的 MIB 文件信息组成树状的数据结构，即生成了 MIB 树。DOM 树生成器，根据上节的翻译算法，将 MIB 树的节点信息转换为 XML DOM 树存储。XML Schema 生成器则直接使用算法将 MIB 树翻译为 XML Schema 文件输出。

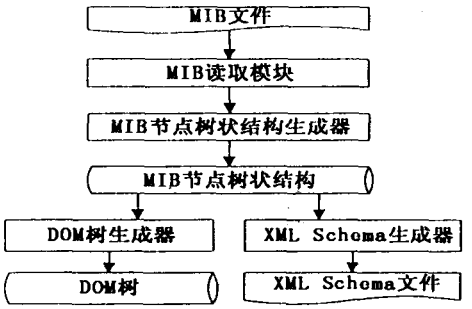

图 3　MIB 到 XML 转换模块结构

4　交互翻译

对于翻译网关的设计，我们需要考虑交互信息的翻译和规范的翻译。上一节中讨论了 MIB 文件到 XML 格式文件的规范翻译，这里要定义交互信息的翻译。交互翻译的实质是：将管理端发送给翻译网关的 XML 格式的消息转换为 SNMP 协议格式消息转发给 SNMP 代理；同时，将 SNMP 代理发送给翻译网关的 SNMP 协议格式消息转换为 XML 格式消息发送给管理端。

交互信息翻译根据翻译网关和管理端的通信方式，可以分为 3 个层次[15]：(1)操作层翻译；(2)消息层翻译；(3)协议层翻译。这三者是逐层递进的。

对于第一种方式的翻译，管理端直接使用 DOM 提供的接口来访问被管理设备的信息，是一种比较简单的翻译方式，适用于管理端和网关集成的应用。在第二种方式中，管理端使用 XPath 和 XQuery[16] 查询语言来查找目标节点，为管理端和代理提供了一个有效和方便的通信方式，从而大大减少了数据的传输量。通过这种方式管理端和翻译网关实现了分离，它们之间通过建立在 HTTP 基础之上的 XML 进行通信。最后一种是在管理端和网关之间的通信使用 SOAP 协议，通过 RPC 调用的一种翻译方式。一个定义于 SOAP 消息中的结构由网关应用服务程序翻译为一个特定的服务调用。这是对第二方式的扩展，SOAP 是基于 HTTP 的，SOAP 消息可以包含 XPath 和 XQuery 表达式，因为 SOAP 是一个公开的标准，并且 SOAP 信息很容易封装，所以这是一个更好的方法。SOAP 是一个内嵌在 XML 的标准协议，用于唤醒远程主机上的方法、对象和服务等，它提供了一个标准的 XML 关键词用于表示输入参数，返回值和错误。使用 SOAP 之后，网关和管理端之间能够通过标准的方式传送数据，同时它能够使用远程网关的服务而不需要知道如何封装 XML 消息。任何类型的管理端都能同网关通信，只要知道 XML Schema 的接口定义，并从定义产生一个 SNMP 的 RPC 消息。

为了能够支持大型网络的管理，更好的可扩展性，我们采用 SOAP 协议作为网关和管理端信息传输方式，网络管理系统为分布式的结构，如图 4 所示。管理端可以是异构的，只要遵循 SOAP 调用的标准并知道我们定义的 XML Schema 元素。

在 SNMP 协议中，操作分为 3 大类：get、set 和 trap。get 操作是管理端像被管理对象的 SNMP 代理取得数据；set 操作是管理端发送数据给被管理对象的代理，代理根据接收到的这个信息做出相应的动作，并返回操作结果；trap 操作则是在被管理对象的代理端发生了特殊事件，由代理主动向管理端发送数据。

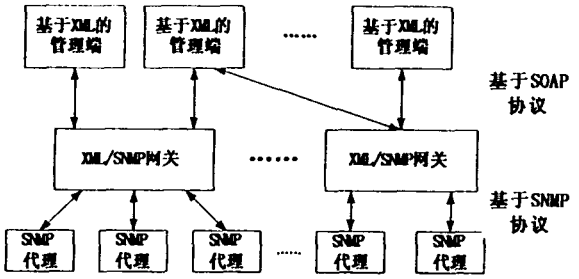

图 4　XML 管理端和 SNMP 代理的分布式结构

我们相应地定义 getRequest、setRequest、getResponse 这 3 种基本类型的 XML 元素(图 5),用于作为基于 XML 的管理端和 XML/SNMP 网关之间通信的 SOAP RPC 消息。getRequest 应当包含 5 个子元素:version、community、path、query。version 元素是 SNMP 的版本号;community 描述了在代理中定义的 community 名;path 元素指定管理端想要操作数据的位置,可以直接使用 oid 用于对象标识或者是使用 xpath 表达式的"path"元素;query 元素定义一个包含 XQuery 表达式的复杂的查询。对于 setRequest 而言,则还要增加一个 value 元素,此参数用于指定要设置对象的值。getResponse 则包含 path 和 result 元素。

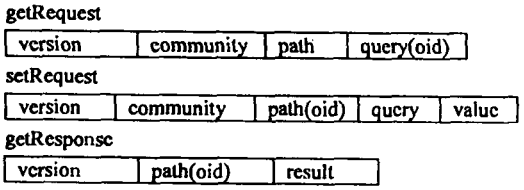

图 5　3 种基本类型 XML 元素

5　翻译网关的结构

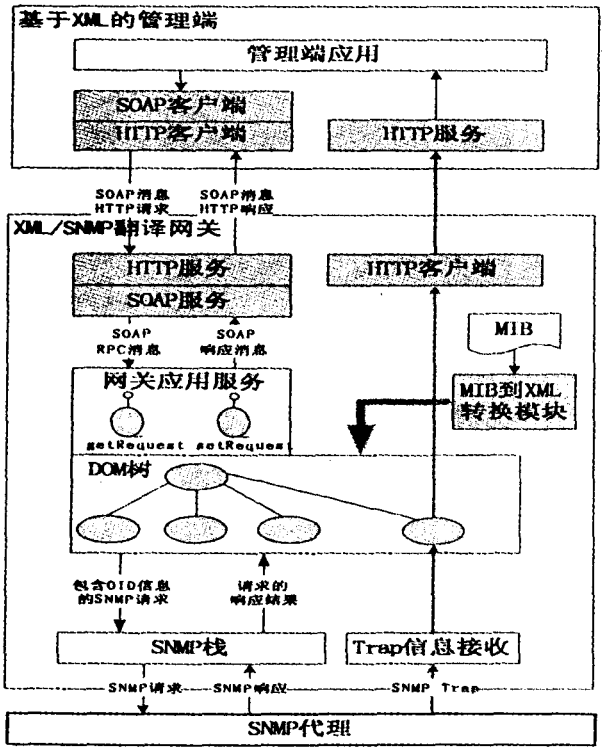

图 6　XML/SNMP 翻译网关结构

图 6 所示为 SNMP/XML 翻译网关的结构。在这种结构下,一个 SOAP 客户端产生一个嵌入 XML 的 SOAP 消息,相反地,一个 SOAP 服务端解析消息并调用翻译网关所提供的适当的应用服务。SOAP 客户端先从管理端应用程序中产生 RPC 消息,然后生成 SOAP 消息。这个 SOAP 消息传送到 HTTP 客户端,并在这里向网关发送 HTTP POST 请求。网关的 HTTP 服务器接收到 POST 请求后,将其转换为 HTTP 消息,并传送给 SOAP 服务器。于是 SOAP 服务器将 HTTP 消息翻译为标准格式的 RPC 并调用由翻译网关提供的合适的对象服务程序。

这里,我们提供了 getRequest 和 setRequest 两个对象调用(分别对应于 SNMP 协议中的 get 和 set 操作),这两个处理过程再查询 DOM 树定位相应节点,通过节点类型可以判断对象操作的类型,封装相应的 PDU 发送给 SNMP 代理。这里的 DOM 树由 MIB 到 XML 转换模块在系统初始化时生成。SOAP 服务器接收到响应之后,产生一个 SOAP 响应消息并返回 SOAP 客户端,最终管理端应用接收到结果。当代理端发生一个事件之后,它通过 TRAP 信息发送到翻译网关,TRAP 信息模块接收到包含 OID、值和时间等的 TRAP 信息,由此更新 DOM 树,然后由翻译网关的 HTTP 客户端产生一个 XML 消息发送给管理端。

结论和展望　本文着重描述了 SNMP/XML 翻译网关的设计方案,提出了 SNMP MIB 到 XML Schema 的翻译算法,用于将 MIB 转换为 DOM 树,同时我们定义了基于 SOAP 协议的交互翻译方式,定义了相应的 XML 元素,制订了翻译网关的结构。在不改变原有设备 SNMP 代理的基础上,为基于 XML 的网络管理系统提供了桥梁。

随着网络的飞速发展,SNMP 在可扩展性、效率、安全等问题上的缺陷,使得基于 XML 的网络管理技术得到了很大发展契机,我们将致力于开发基于 XML 的网络管理端和基于 XML 的管理代理。借助于 XML 通用标准方面的优势,可以收集网络上的各类设备包括防火墙、路由器、入侵检测系统等各类管理信息,由基于 XML 的管理端进行智能分析,并及时反馈分析结果,使得各个设备协同工作,更好地保障网络的安全和性能。

参 考 文 献

1　Case J, Fedor M, Schoffstall M, et al. A Simple Network Management Protocal(SNMP). RFC 1157, IETF, May 1990
2　Ju H T, Choi M-J, Han S, et al. An embedded Web server architecture for XML-based network management. In: Proc. of the IEEE/IFIP Network Operations and Management Symposium (NOM 2002); Florence, Italy, April, 2002
3　Ju Hong-Taek, et al. An Embedded Web Server Architecture for XML-Based Network Management. Network Operations and Management Symposium, 2002. 5～18
4　W3C. Extensible Markup Language (XML) 1. 0. W3C Recommendation, October 2000
5　W3C. XML Schema Part 0, 1, 2. W3 Consortium Recommendation, May 2001
6　W3C. Document Object Model (DOM) Level 1 Specification. W3C Recommendation, Oct. 1998
7　W3C. XML Path Language (XPath) Version 2. 0. W3C Working Draft, Apr. 2002
8　W3C. Extensible Stylesheet Language (XSL) Version 1. 0. W3C Candidate Recommendation, November 2000
9　W3C. XSL Transformations Version 1. 0. W3C Recommendation, November 1999
10　W3C. SOAP Version 1. 2 Part 0: Primer. W3C Working Draft, Dec. 2001
11　W3C. Web Services Description Language (WSDL) 1. 1. W3C Note, Microsoft, IBM, March 2001
12　夏海涛, 詹志强. 新一代网络管理技术. 北京邮电大学出版社, 2003
13　Choi Mi-Jung, Ju Hong-Taek, Hong. J W. Towards XML and SNMP Integrated Network Management. In: Proc. of the Asia-Pacific Network Operations and Management Symposium, Jeju Korea, Sep. 2002. 507～508
14　Martin-Flatin J P. Web-Based Management of IP Networks and Systems: [Ph. D. Thesis]. Swiss Federal Institute of Technology, Lausanne(EPFL), Oct. 2000
15　Oh Y J, Ju H T, Choi M J, et al. Interaction Translation Methods for SML/SNMP Gateway. DSOM 2002, Montreal Canada, Oct. 2002
16　W3C. XQuery 1. 0: An XML Query Language. W3C Working Draft Ap. 2002