

# XML 在 SNMP 管理者协议实现中的应用

吴黎兵<sup>1</sup>, 崔建群<sup>2</sup>, 吴产乐<sup>1</sup>, 孟学军<sup>1</sup>, 邢建兵<sup>1</sup>

(1. 武汉大学计算机学院, 武汉 430072; 2. 华中师范大学计算机科学系)

**摘要:** 利用 XML 在网络管理中的应用优势, 将 XML 应用于 SNMP 管理者协议的实现过程中, 不但降低了协议实现的难度, 而且使该协议实现具有更好的通用性和可扩展性, 并为新一代网管提供了 XML 管理接口。

**关键词:** XML; SNMP; 管理者; 协议实现

## Application of XML in Protocol Implementation of SNMP Manager

WU Libing<sup>1</sup>, CUI Jianqun<sup>2</sup>, WU Chanle<sup>1</sup>, MENG Xuejun<sup>1</sup>, XING Jianbing<sup>1</sup>

(1. Computer Science Institute, Wuhan University, Wuhan 430072; 2. Department of Computer Science, Central China Normal University)

**【Abstract】** The paper discusses the advantage of using XML in network management. Using XML in the implementation of SNMP manager does not only make the implementation easier, but also make the implementation more universal and scalable. The implementation also provides the XML management interface for the new generation of network management.

**【Key words】** XML; SNMP; Manager; Protocol implementation

基于 XML 的新一代互联网网管已经成为当今网络管理发展的新趋势<sup>[1]</sup>, 越来越多的设备、服务及平台都宣称支持 XML 技术。本文所讨论的 SNMP 管理者在研究过程中, 将 XML 技术用于实现的多个方面, 既利用了 XML 的优势, 使系统具有良好的扩展性, 同时也为基于 XML 的网管模式提供了管理接口。

### 1 XML 技术简介

XML(eXtensible Markup Language, 可扩展标记语言)是由 W3C(World Wide Web Consortium) 于 1998 年 2 月发布的一种标准, 它是一种数据交换格式, 允许在不同的系统或应用程序之间交换数据, 通过一种网络化的处理机制来遍历数据, 每个网络节点存储或处理数据并且将结果传输给相邻的节点。它是一组用于设计数据格式和结构的规则和方法, 易于生成便于不同的计算机和应用程序读取的数据文件。

XML 在网络管理软件开发中的优势主要表现在如下几个方面<sup>[2]</sup>:

- 复杂数据处理优势;
- 使底层数据更具可读性和标准性;
- 增强了基于 Web 的网络管理方式。

### 2 XML 在 SNMP 管理者协议实现中的应用

由于 XML 具有一些其它网络管理技术所不能提供的特性, 将 XML 应用于网络管理是网络管理领域的必然发展趋势, 因此本文所讨论的 SNMP 管理者的实现, 在这方面做了大胆的尝试, 主要在以下几个方面运用了 XML 技术:

- 可解析采用 XML 来描述管理对象的 MIB 文件。
- 从 GUI/API 中接收到的输入数据一律采用统一的 XML 接口, 使系统可以十分方便地采用不同模式实现用户数据与系统的交换。
- 数据在系统内部的处理以 XML 数据流为主, 一方面通过成熟的 XML 解析器, 可以减少数据处理的复杂度; 另一方面, 因为只在最后向传统 SNMP Agent 发送 BER 编码时才进行格式转换, 所

以如果 Agent 支持 XML 格式报文管理, 去掉转换层就可以达到 XML 管理的目标。

· 通过 XML 配置文件对系统进行初始化配置。

以下将就实现上述功能所需要解决的关键技术作进一步的讨论。

#### 2.1 ASN.1 与 XML 的映射与转换

ASN.1 是一种用于描述结构化客体的结构和内容的语言。它有两部分: 一部分描述信息内数据、数据类型及序列格式; 另一部分描述如何将各部分组成消息。ASN.1 广泛应用于通信协议(如 SNMP、LDAP)、安全协议(如 X.509) 以及数据格式(如 PKCS#7) 等领域。

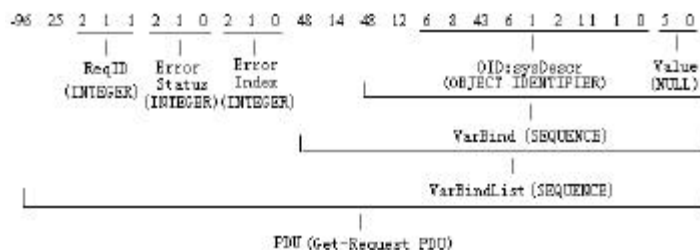


图 1 报文的 BER 编码

在本文研究的 SNMP 管理者的实现过程中, SNMP 报文、PDU 采用的是 ASN.1 语法。例如要向某一代理发送一个 SNMP v1 版本的 Get-Request 请求, 请求获得 sysDescr 对象实例(OID:1.3.6.1.2.1.1.0) 的值, 则此请求 PDU 的 BER 编码如图 1 所示(以字节为单位, 为便于阅读, 每一字节以十进制表示)。

基金项目: 国家“863”计划引导基金资助项目(2003AA001032)

作者简介: 吴黎兵(1972—), 男, 博士生, 研究方向: 网络管理, 网络协议, 服务质量; 崔建群, 硕士; 吴产乐, 教授、博导; 孟学军, 工程师; 邢建兵, 博士生

定稿日期: 2004-04-09 E-mail: wufox@sohu.com

我们将 XML 引入 SNMP PDU 编解码过程,通过 ASN.1 和 XML 的互相转换,可以简化 SNMP PDU 的编解码难度、提高正确性并使得通过 Web 方式进行 SNMP 报文的传递和接收成为可能。

#### (1) 映射规则

为了能将给定的 ASN.1 语法描述通过程序自动转换为 XML DTD 或 Schema,可定义如下映射规则<sup>[3]</sup>:

1) 采用 ASN.1 的标识(以小写字母开头的标识)作为相应实例的元素标识名称。

2) 同一个 ASN.1 描述中可能不同的位置出现多个相同的标识,因此采用在标识前面加上父标识,中间用“.”进行分隔(“.”在 ASN.1 中不能出现在标识中,但在 XML 中可以作为元素名的一部分),最上层没有父标识时,就加上 root 作为父标识。

3) 每一个元素缺省的属性用来描述该元素的类型,以便以后将 XML 文档转换为 ASN.1 数据时使用。用两个属性 typeRef 和 type 分别代表类型名称和基本类型,由于这些属性是每个元素都固有且确定的,在 DTD 或 Schema 中已给出具体的值,因此不需要在 XML 文档中指出,具有正确性验证的 XML 处理器会自动将其加入 XML 文档中。

图 1 报文可转换成下面的 XML 文档描述:

```
<GetRequest-PDU_>
<PDU.request-id>1</PDU.request-id>
<PDU.error-status>0</PDU.error-status>
<PDU.error-index>0</PDU.error-index>
<PDU.variable-bindings>
  <VarBindList_>
    <VarBind.name>1.3.6.1.2.1.1.1.0</VarBind.name>
    <VarBind.value>
      <ObjectSyntax.simple>
        <SimpleSyntax.empty/>
      </ObjectSyntax.simple>
    </VarBind.value>
  </VarBindList_>
</PDU.variable-bindings>
</GetRequest-PDU_>
```

#### (2) ASN.1 与 XML 转换的实现

根据以上讨论的结果,要完成 ASN.1 与 XML 之间的转换,需要经过以下几个步骤(以 ASN.1 转换为 XML 文档为例):

- 1) 将需处理的 ASN.1 数据的语法映射为 XML Schema。
- 2) 将输入的 ASN.1 数据的 BER 编码转换为 XML 编码,该 XML 编码用 JDOM 表示。
- 3) 将 2) 中产生的 XML 编码与 1) 中产生的 XML Schema 进行比对,得到具有语义标识的 XML 描述文档。
- 4) 输出该 XML 描述文档。

XML 文档向 ASN.1 数据的转换与上述转换过程相反。转换的实现框架如图 2 所示。

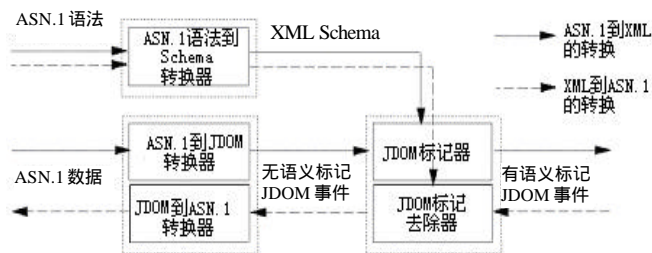


图 2 XML/ASN.1 转换

## 2.2 MIB 文件的 XML 格式描述

SNMP 管理者的实现过程中需要从管理端获取对被管设

备管理信息的描述,才能完整地构造/解析出请求报文的请求管理对象,这需要通过 MIB 描述文件的解析来实现。传统的 MIB 文件采用 ASN.1 格式描述,可借助已有的 LL(1)文法解析器进行基础分析。但是这种方法需要花费不少时间和精力来构造大量的规则库,而且随着 SNMP 的发展,某些 MIB 文法很可能不再符合 LL(1) 标准,将无法采用上述方法进行 MIB 文件的解析。

如果采用 XML 文档来描述 MIB 变量的定义,不但具有良好的可读性,而且只要编写一个 SMI 的 DTD 或 XML Schema,就可以借助带有验证功能的 XML 解析器对 XML 格式的 MIB 文件进行语法解析,而且无论 MIB 文法是否符合 LL(1) 文法,都可以采用这一方法来实现。对于 SNMP Agent,采用 XML 描述的 MIB 既可以进行动态修改,又可以用不同的形式进行发布供管理者调用<sup>[4]</sup>。因此通过 XML 格式来描述 MIB 文件,必将成为网络管理的新发展和新趋势。

根据 SMI 规范,可以编写如图 3 所示的 XML Schema 来验证 MIB 的 XML 文档是否正确(由于页面所限,仅列出 Schema 的主要框架)。

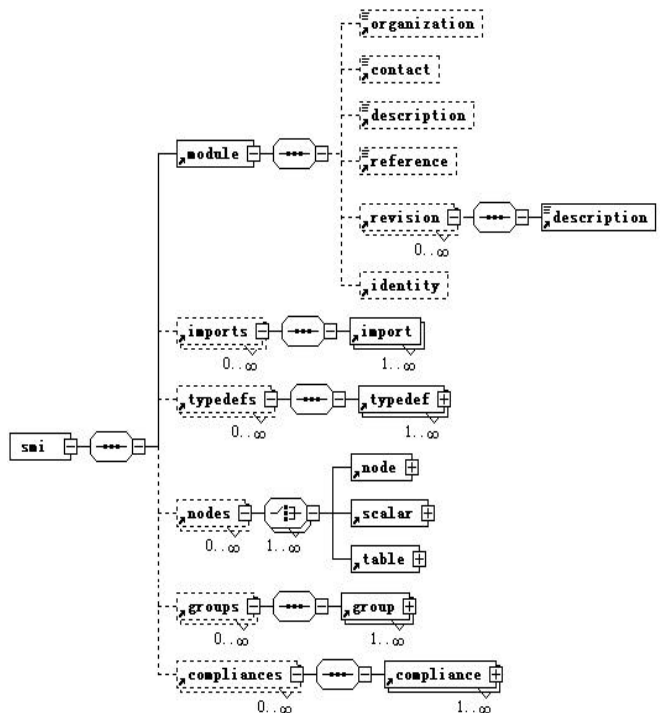


图 3 MIB 文件的 XML Schema 结构

其中实线代表 XML 文档必须包含的元素,虚线代表可选的元素,带“+”号的元素表示有下级元素或属性。元素下面的数字则表示 XML 文档中可以包含该元素的个数。

有了图 3 定义的 XML Schema,就可以用 XML 的格式来描述 MIB 文件,以 RFC-1155 为例,其 XML 文档如下:

```
<?xml version="1.0"?>
<smi xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="d:\xml\schema\smi.xsd">
  <module name="RFC1155-SMI" language="SMIV1"/>
  <typedefs>
    <typedef name="ObjectName" basetype="ObjectIdentifier"/>
    <typedef name="NetworkAddress" basetype="OctetString">
      <parent module="RFC1155-SMI" name="IpAddress"/>
    </typedef>
```

```

<typedef name="IpAddress" basetype="OctetString">
  <range min="4" max="4"/></typedef>
<typedef name="Counter" basetype="Unsigned32">
  <range min="0" max="4294967295"/></typedef>
<typedef name="Gauge" basetype="Unsigned32">
  <range min="0" max="4294967295"/></typedef>
<typedef name="TimeTicks" basetype="Unsigned32">
  <range min="0" max="4294967295"/></typedef>
<typedef name="Opaque" basetype="OctetString"/></typedefs>
<nodes>
  <node name="org" oid="1.3"/>
  <node name="dod" oid="1.3.6"/>
  <node name="internet" oid="1.3.6.1"/>
  <node name="directory" oid="1.3.6.1.1"/>
  <node name="mgmt" oid="1.3.6.1.2"/>
  <node name="experimental" oid="1.3.6.1.3"/>
  <node name="private" oid="1.3.6.1.4"/>
  <node name="enterprises" oid="1.3.6.1.4.1"/>
</nodes>
</smi>

```

### 3 系统原型实现

图 4 描述了 SNMP 管理者系统原型。

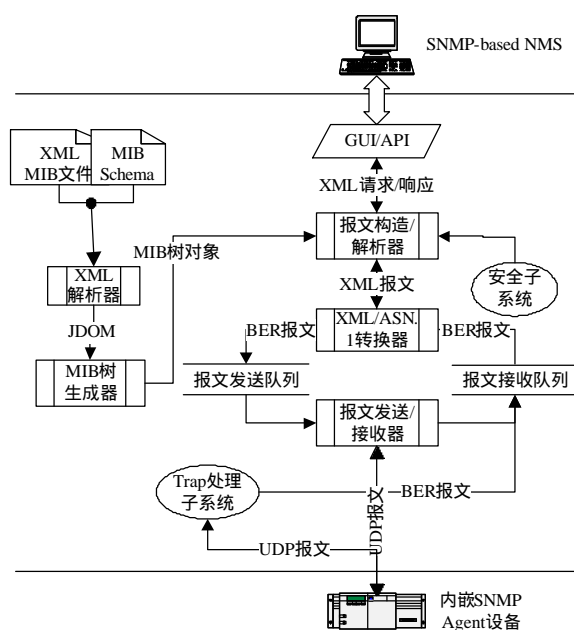


图 4 系统原型

基于 SNMP 协议的 NMS 管理工作站，通过本系统提供

的 GUI/API 发出 SNMP 管理请求，以 XML 数据格式传递给报文构造/解析器进行报文的 XML 格式构造，在构造过程中，与 MIB 对象相关的所有信息都从 MIB 树的 MIB Node 中获得，如果是 v3 版本的报文，则还需要通过安全子系统进行安全变量初始化和安全信息的加载。

一旦 XML 报文构造成功，将其发往 XML/ASN.1 转换器(如果 SNMP Agent 支持 XML 格式管理，则无须此步骤)，转换为 BER 编码，放入报文发送队列准备发送。在系统启动时，会同时启动另一个守护线程来实现报文发送/接收器。当报文发送队列中增加新的报文时，报文发送/接收器就会立即将其发送至目的地址和端口，而同时也在发送端口等待响应报文的到来，只要检测到有响应报文到来，就将其放入报文接收队列。接收到的报文按照刚才发送的逆过程，经过 XML/ASN.1 转换器、报文构造/解析器，最后通过 GUI/API 接口反馈给 NMS 管理工作站。

除此之外，如果管理工作站希望监听某设备的 Trap 信息，则需启动系统提供的 Trap 处理子系统，在特定的 UDP 端口(通常是 162 端口)监听被管对象的 Trap 报文，一旦收到 Trap 报文，同样将其放入报文接收队列，等候报文构造/解析器的最终解析。

### 4 总结

根据本文讨论的方法，我们已经用 Java 语言实现了 SNMP v1、v2、v3 版本管理者的功能，目前正在进一步测试中。在 SNMP 协议实现中引入 XML 是为了提高系统的可扩展性和灵活性，但是这往往会影响系统的性能。XML 数据的解析速度是协议实现中可能出现的瓶颈，因此在这方面还需要做更多的优化研究工作，才能保证系统在灵活性和高效性之间找到合适的平衡点。

### 参考文献

- 1 荣波, 周旗. 基于 XML 的新一代互联网网管方案. 中国数据通信, 2002, (3)
- 2 Ensel C, Keller A. Managing Application Service Dependencies with XML and the Resource Description Framework. 2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings, Seattle, 2001-05: 661-674
- 3 Imamura T, Maruyama H. Mapping Between ASN.1 and XML. IEEE/Applications and the Internet, 2001
- 4 Youn K S, Hong C S. An XML-based Dynamic Network Management System Using Web Technology. Proceedings of 22nd International Conference on Distributed Computing Systems Workshops, 2002: 83-88

(上接第 53 页)

包的部署提供了必要的环境。在进一步的研究中，需要定义 RANDP 协议及包的结构，并研究 RANDP 包的部署策略；结合 RSVP 协议的工作特点，研究灵活的协议部署机制；针对不同的传输控制机制，研究更具适应性的策略。

### 参考文献

- 1 Jackson A W, Sterbenz J P G, Condell M N, et al. SENCOMM Architecture. <http://www.ir.bbn.com/projects/sencomm/>, 2000-04
- 2 Wetherall D, Legedza U, Gutttag J. Introducing New Internet

Services: Why and How. IEEE NETWORK, 1998,(3): 12-19

- 3 Calvert.ed K L. Architecture Framework for Active Networks Version 1.0. <http://www.dcs.uky.edu/~calvert/arch-docs.html>, 1999-07
- 4 Peterson L. Node OS Interface Specification. <http://www.dcs.uky.edu/~calvert/arch-docs.html>, 2001-01
- 5 Alexander D, Braden B, Gunter C, et al. Active Network Encapsulation Protocol(ANEP). <http://www.cis.upenn.edu/~switchware/ANEP/docs/ANEP.txt>, 1997-07