

基于 SNMP 及构件组装技术的设备仿真模型

宋国正 吴亚锋

(咸宁供电公司 咸宁 437100)

摘要 基于 SNMP 协议及构件组装技术,构建网络设备的仿真模型来反映真实设备的运行状况。运用 SNMP 协议可以获得或设置设备的运行状态,比如获取设备 CPU、内存及电源的运行状态;运用构件组装技术可以把网络设备的构件组装成一个整体的可视化仿真模型。网络设备的构件包括引擎板构件、业务板构件、电源构件及风扇构件等,这些构件的运行依赖于 SNMP 协议。网络设备仿真模型的建立提高了网络管理人员的工作效率,网络管理人员对仿真模型的操作相当于对真实设备进行操作。

关键词 SNMP 协议,构件,构件组装,网络设备,仿真模型,可视化

Equipment Simulation Model Based on SNMP and Component Assembly Technology

SONG Guo-zheng WU Ya-feng

(Xianning Power Company, Xianning 437100, China)

Abstract Based on SNMP protocol and component assembly technology, building network equipment simulation model to reflect the true operational status of equipment. We can get or set the device operating status with the SNMP protocol, such as to get running status of CPU, memory and power supply. We can get an overall visual simulation model by the use of component assembly technology to assemble the network devices components, network equipment components including the engine board, business board, power and fan components, etc. The operation of these components depends on SNMP protocol. The simulation model of network equipment can improve the network management efficiency. The operation on simulation model is equivalent to operate the true equipment.

Keywords SNMP protocol, Component, Component assembly, Network equipment, Simulation model, Visual

1 概述

一个大型的内网网络包括信息广域网、信息城域网及信息骨干网,覆盖整个地级城市(包括二级县市),有近 30 台三层核心交换机、近 350 台接入交换机及与省级城市互联的 2 台核心路由器,这些网络设备承载着大量的业务。目前,网络管理者对网络设备的管理基本依赖远程登录工具登录网络设备,以命令的方式对网络设备进行查看及维护。这种非可视化图形界面的方式在维护的过程中操作起来比较麻烦,且需要对网络协议及各种厂家的网络命令相当熟悉,一般的运行人员很难达到此技术水平。本文建立一种网络设备的仿真模型,以统一的可视化图形界面来查看及维护网络设备,不需要知道不同设备的命令控制,让不同层次的网络运行人员都可以方便地操作网络设备,一目了然,便于查看及维护。仿真模型通过 SNMP 协议及构件组装技术来实现。

2 SNMP 协议^[1,2]

现在的网络主要基于 SNMP 集中式的网络管理。SNMP 基于传输层的 UDP 数据报协议,在管理者和被管设备代理之间传输管理信息。其监管过程为:在被监控端,嵌入网络设备或主机设备中的 SNMP Agent 收集设备的各种信息和统计数据,并不断地把这些数据记录到 MIB 库(管理信

息库)中;在管理端,通过 SNMP 协议向 SNMP Agent 发出 SNMP 报文,SNMP Agent 收到报文后向 SNMP 管理端响应应答包。

2.1 管理信息库 MIB

SNMP 模型的最大部分就是定义谁应该记录什么信息以及该信息如何进行通信。总之,每个设备都具有一个或多个变量来描述其状态。在 SNMP 文字中,这些变量叫做对象(Object)。网络的所有对象的存放位置就叫做 MIB,它是一个树形的数据结构。MIB 的每个节点都有一个唯一的对象标识符,对象标识符的命名有专门的机构负责。每一个“分支”都被标记中所示的管理组织所管理。

2.2 SNMP 的操作

SNMP 支持轮询和事件驱动 2 种访问方法,使用 5 种通信原语(如表 1 所列),以实现 SNMP 的 GET、GET-NEXT、SET、TRAP 等操作,从而得到管理信息。其中 GetRequest 和 GetResponse 实现了请求响应机制,SNMP 使用轮询(polling)方法实现这一机制,Trap 消息的使用基于事件驱动(event driven)机制的方法实现。网管工作站可以把感兴趣的变量值提取到其应用程序中,只要发出 get-request 或 Get-next-request 报文即可。前者是指定对象的读操作,后者则提供了 1 个树遍历操作符,以便于确定 1 个代理进程支持哪些对象。网管工作站可以修改代理进程中的变量值,只要发出

set-request 报文即可,通信模型如图 1 所示。

表 1 SNMP 5 种通信原语

通信原语	说明
GetRequest	由 NMS 发给 Agent 的请求命令,请求 1 个 MIB 变量值
GetNextRequest	由 NMS 发给 Agent 的请求命令,要求将被说明目标的下一个目标的 MIB
GetResponse	是 Agent 对于收到的请求的 1 个请求应答,此请求时要求将制定数据送到 NMS
SetRequest	由 NMS 发出,命令 Agent 去改变个 MIB 变量值
Trap	Agent 检测到某种预先说明了的状态时,向 NMS 发送的 1 个非请求信息

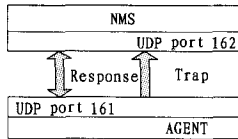


图 1

3 构件组装技术

一个独立的软件构件本身并不具有价值,只有将构件按照一定的规则组装成应用系统,才能实现其本身的价值。构件组装,顾名思义,就是将构件关联起来从而满足某种功能需求。每个构件都会向外界提供一定的服务,并且使用其他构件的服务,构件之间这种服务的请求和提供就是通过接口的形式来实现的。构件组装就是使用构件组装语言,根据构件间服务的请求与提供关系即构件接口,在构件模型的基础上将构件组装到一起,构成复合构件,乃至构成复杂的应用系统。本文的目的是把一些网络构件模型组装成一个复合构件供网络管理系统(以下简称 NMS)使用,即把一些能够提供网络服务的构件组装成可视化的网络设备仿真模型。在 NMS 中,对网络设备仿真模型的操作等同于在真实物理设备上的操作。

3.1 构件组装的一般方法^[3]

如图 2 所示,构件组装首先根据目标系统确定软件体系结构框架,选定构件模型。软件体系结构框架和构件模型不仅定义构件的基本属性,还要定义构件之间及构件与框架之间的交互机制,为之后的构件组装奠定基础;再对目标系统进行功能分解,将目标系统分解为强内聚、松耦合的功能构件,并确定构件之间的服务请求和提供关系;根据需要在构件库提取符合要求的可复用构件,或开发新的构件;将获取的构件进行适当的修改,使其标准化,为下一步组装做好准备;将标准化后的构件在选定的软件体系结构框架下按照接口组装起来,构成新的复合构件^[4,5]或完整的目标应用系统。

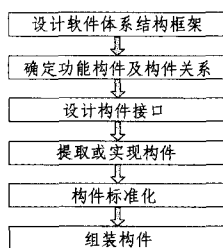


图 2

3.2 构件组装方案

本文采用常用的灰盒组装方法。每个构件的接口不但定

义了其对外提供的服务,而且定义了对外请求的服务。构件在实现时,不是直接调用其他构件提供的服务,而是使用它在接口处定义的对外请求的服务。每个构件都采用 XML 语言^[6]来描述构件的接口信息,如图 3 所示的对 SNMP.dll 构件的 XML 部分描述文档描述 SNMP 通信原语 GetRequest 操作接口,接口名为“GetRequest”,接口的第一个参数“oidText”要求传入一个 MIB 的某个对象标识符 OID 的值,比如 OID 为“1.3.6.1.2.1.1.1”是用来描述某网络设备内置系统软件的说明;第二个参数“snmpVersionText”要求传入使用 SNMP 协议的版本;第三个参数“snmpCommunityText”要求传入使用 SNMP 协议的共同体名(相当于密码);第四个参数 IPAddress 要求传入被管理的网络设备的 IP 地址。另外其还描述了构件的存储位置、类名及方法名。通过这个 XML 文件对 SNMP.dll 构件的描述,很容易知道构件的功能及调用方法。

```

<?xml version="1.0" encoding="utf-8" ?>
<Component xmlns:rt="urn:schemas">
  <ProvidedInterface>
    <Interface ID="GetRequest" Name="GetRequest" Type="Required"
      Function="执行无参数返回命令" Remark="">
      <Declare>
        <Param ID="1" Name="oidText" Type="String"/>
        <Param ID="2" Name="snmpVersionText" Type="String"/>
        <Param ID="3" Name="snmpCommunityText" Type="String"/>
        <Param ID="4" Name="IPAddress" Type="String"/>
      </Declare>
      <Assembly Name="SNMP" Path="SNMP.dll">
        <Class Name="NMS.SNMP">
          <Method Name="GetRequest" />
        </Class>
      </Assembly>
    </ProvidedInterface>
  </Component>

```

图 3

4 网络设备构件模型

一个可视化的网络设备仿真模型需要用来实现网管功能的 SNMP.dll 构件,更需要一些物理构件,比如“引擎板构件”、“业务板构件”等,这些物理构件和 SNMP.dll 构件组装在一起,网络设备仿真模型才能真正“运行”起来。以一款中高档的华三(H3C)交换机 H3C7506E 为例,组装一台 H3C7506E 交换机除了裸机外还需要配置引擎板、业务板、电源、风扇。怎么选择这些“物理构件”是由客户决定的。

因此,本文的构件除了实现网管操作功能的构件、实现数据库操作功能的构件,大部分是不同型号的网络设备物理构件,比如同样是业务板,型号不同其功能就不同。本文的物理构件都采用 .Net 平台开发,以“类库”为输出类型(即构件名以.dll 结尾)。以华三业务板型号为例,业务板构件呈现的组装视图模仿真实设备图形,如图 4 所示,此业务板有 24 端口千兆以太网口。业务板构件的描述采用 XML 描述文档,业务板构件或引擎板与 SNMP.dll 构件通过 XML 文件描述的接口^[7,8]关联起来生成一个完整的 H3C7506E 复合构件,视图结构如图 5 所示。

图 4

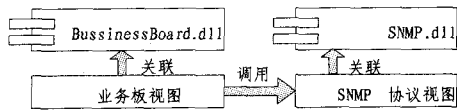


图 5

5 网络设备构件组装

5.1 .NET 反射技术

构件关联从现象上看是两个 XML 视图文档进行关联,本质上是文件名后缀为 .dll 或 .exe 的构件之间的相互调用关系。

反射是 .NET 中的重要机制。通过反射,可以在运行时获得 .NET 中程序集、模块、类型、字段、属性、方法和事件的相关信息,即使在编译时不知道相关类型信息,也可在运行时根据反射机制动态地创建类型实例,调用类型的方法或访问其字段和属性,反射机制调用过程如图 6 所示。。NET 反射机制是本文在构件组装过程运用的核心技术。

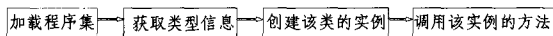


图 6 反射机制调用过程

5.2 组装方案

前面已经提到,本文把网络构件组装成一个仿真的复合构件以便在开发 NMS 时供 NMS 调用。组装方案如图 7 所示,分 3 个步骤:

- 1) 根据需求,通过构件视图的可视化编辑工具创建构件视图,生成构件的连接子描述信息和构件视图描述信息。
- 2) 根据构件视图在构件库中查找构件。
- 3) 通过构件组装工具,根据构件视图描述文档和连接子描述文档将构件组装起来,生成新的复合构件。

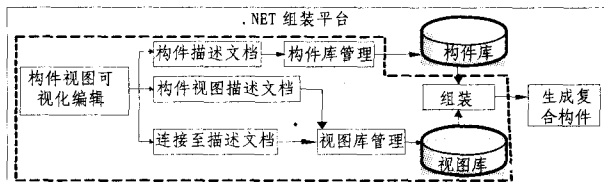


图 7 组装方案

6 网络设备仿真模型实例

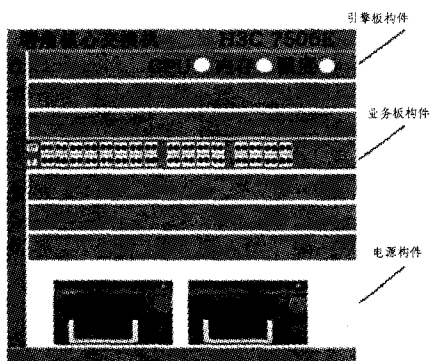


图 8 设备仿真模型视图

前面已经讲述,生成网络设备仿真模型是本文构件组装的目的,这个模型本质上也是一个构件,是一个具有操作真实

物理设备功能的复合构件,如图 8 所示,这是物理构件及功能构件组装后生成的设备仿真模型视图,业务板构件、引擎板构件各个端口的运行状态通过调用 SNMP.dll 构件,用不同的颜色来显示,端口的运行状态有 3 种,分别为禁用状态(红灯)、掉线状态(黄色)、正常状态(绿色);当然还有许多其它功能,比如给端口划分业务,限制流量等网络监控策略。总之,在仿真模型上可以进行各种网络监控策略的操作,这些操作不需要敲任何命令,即仿真模型能实时反映真实物理设备的运行状态,也能改变设备的运行状态。改变某变量的运行状态的核心代码如下所示,不需要知道设备所属厂家,只需要调用各个厂家都支持的通用的 SNMP 协议来对设备进行监控和操作。

```

SnmvV1Packet packet=new SnmvV1Packet();
// Set the community name
packet.Community.Set("public");
// Set the Pdu type
packet.Pdu.Type=PduType.Get;
// Set request id
packet.Pdu.RequestID=100;
// Add an Oid for the SNMP-Get operation
packet.Pdu.VbList.Add(". 1. 3. 6. 1. 2. 1. 1. 0");
// Set the Pdu type
packet.Pdu.Type=PduType.Set;
// Add an Oid and Value for the SNMP-Set operation
packet.Pdu.VbList.Add(new Oid(". 1. 3. 6. 1. 2. 1. 1. 0"),new OctetString("New sysDescr. 0 value"));
byte[] outBuffer=packet.encode();
SnmvV1Packet result=new SnmvV1Packet();
result.decode(inBuffer,inLength);

```

结束语 本文结合构件组装技术及 SNMP 协议来组装一个可视化的网络设备仿真模型,并提出了网络设备的构件模型及仿真设备模型实例。在 NMS 系统中,网络管理者更加方便快捷地监控网络设备各项参数,比如端口的运行状态、端口的业务、端口的流量、路由表、ARP 表等等,不需要远程登录到网络设备以输入命令的方式去访问设备,而是直接通过仿真模型监控及操作网络设备,大大提高了工作效率。

参考文献

- [1] 李明江. SNMP 简单网络管理协议[M]. 北京:电子工业出版社, 2007:127-131
- [2] 武孟军. 精通 SNMP[M]. 北京:人民邮电出版社,2010
- [3] 夏榆滨. 软件构件技术[M]. 北京:清华大学出版社,北京交通大学出版社,2010
- [4] 李超. 过程线构件库及复合构件组装研究与应用[D]. 南京: 淮海大学,2007
- [5] 张世珉,张文娟. 基于软件体系结构的可复用构件制作和组装[J]. 软件学报,2001,12(9):1351-1359
- [6] XML 中国论坛. XML 实用进阶教程[M]. 北京:清华大学出版社,2001
- [7] 谢晓芹,李涓子,王沛,等. 基于 XML 的构件自动组装系统的设计与实现[J]. 计算机工程与应用,2005,41(4)
- [8] 刘元勇. 基于 XML 构件模型的组装技术研究[D]. 重庆:重庆大学,2007