

# 基于目录的资源信息模型<sup>①</sup>

李黎明

(上海警备区司令部 指挥自动化工作站, 上海 200040)

**摘要:** 介绍了资源信息模型的基本情况, 重点介绍了通用信息模型 CIM, LDAP 协议与模型。最后提出了利用 LDAP 实现机群系统资源信息模型的初步方法。

**关键词:** 目录服务; 资源信息模型; CIM; LDAP

## Directory-Based Resource Information Model

LI Li-Ming

(C4ISR Office of Shanghai Garrison, Shanghai 200040, China)

**Abstract:** This article describes the basic conditions of resource information model, focusing on a common information model CIM, LDAP protocols and models. Finally, it describes the use of LDAP cluster system resource information model to achieve a preliminary approach.

**Key words:** directory services; resource information model; CIM; LDAP

## 1 引言

信息模型起源于数据库, 重点是数据结构的模型化。在数据库中, 信息模型的作用是将域中的信息直接映射到模型中的对象上。它也拥有发达的用于定义对象关系和识别冗余数据的技术。传统信息模型的弱点在于它缺少对服务的定义和说明, 同时也缺少对属性、行为和服务封装的显式说明, 以及对关系层次结构的显式说明。但是, 随着传统信息模型与对象模型的结合, 现在的信息模型已经克服了传统的不能描述属性、系统行为和服务的弱点, 能够较为完善的描述系统中的各种元素。资源信息模型是在信息模型和对象模型基础之上建立起来的对资源信息和行为的一种抽象。资源信息模型不仅仅能够表述资源实体对象和实体对象的属性, 还能够通过对各个实体对象之间关系的描述来刻画资源实体的各种行为和服务以及整个资源系统的结构。

## 2 通用信息模型 CIM

DMTF (Distributed Management Task Force) 提出的通用信息模型 CIM (Common Information Model)

是一种用来管理信息和系统的面向对象的方法。它将信息模型的经典概念与对象模型结合起来, 吸收了两者的长处, 从而建立了一个分层的资源信息模型。CIM 在它的模型中定义了三层结构。分别是核心模型、通用模型和扩展模型。

### 2.1 核心模型

核心模型是抽象层次最高的层, 它必须兼容 CIM 规范。核心模型描述信息模型中适用于所有管理区域的概念。它由一组类、属性、方法和关联构成, 它定义了像被管理系统元素 ManagedSystemElement 和依赖关系 Dependency 这样的抽象对象, 所有的知识域都能使用它们。核心模型也定义了某些普通对象 (如: ComputerSystem、PhysicalElement 和 LogicalElement), 用以更详细地说明从本质上来说仍然很普通的行为。比如, CIM 定义了一个 ComputerSystem 的超类, 来模型化计算出结果并作出决策的功能。这样 ComputerSystem 不仅可作为笔记本电脑和桌面电脑的超类, 也可作为任何拥有计算能力的对象的超类。如网络中的路由器、交换机等。它构成了类层次结构和关系层次结构的基础。核心模型没有任何具体的实现

① 收稿时间:2011-03-24;收到修改稿时间:2011-04-23

限制,也就是说它是平台和技术无关的。CIM 核心模型的构成如图 1 所示:

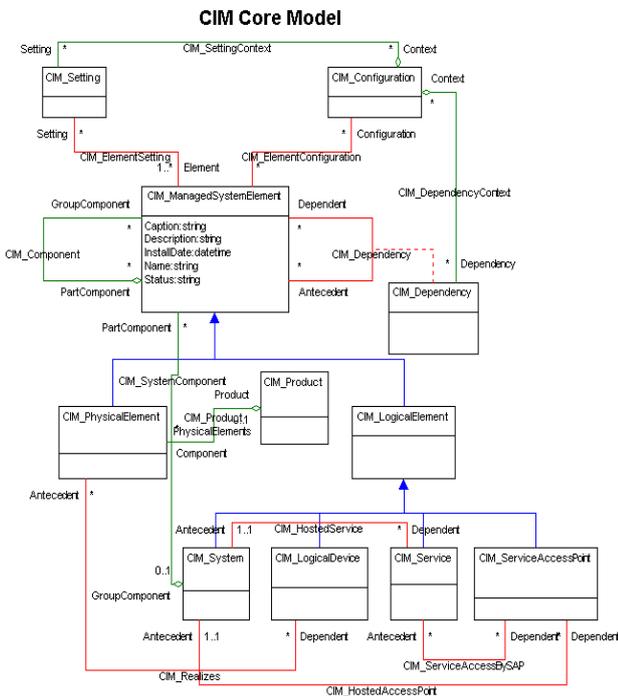


图 1 CIM 核心模型

### 2.2 通用模型

CIM 的第二层是通用模型,它定义了针对某些普通领域的概念。这些领域,尽管从本质上来说仍然是很普通的,但它比那些使用在更普通的核心领域的层拥有更详细的知识。目前,CIM 已经定义了如下这些通用模型:

- **系统模型 (System)** .....它定义系统的组件以及如何组装它们。这些组件包括系统、计算机系统、操作系统、文件和处理过程。
- **设备模型 (Device)** .....它定义用硬件实现物理设备的方法和建立设备连接模型的方法,包括大容量存储设备、媒体、感应器、打印机、电源和其它组件。
- **网络模型 (Networks)** .....它定义建立网络元素和服务模型的物理和逻辑元素类层次结构的规范,包括网络协议模型和网络系统模型。
- **应用程序模型 (Applications)** .....它定义如何在系统上进行软件安装,包括软件特征、元素的概念、监测系统环境是否适合软件安装的能力,以及作为软件执行一部分的动作。

● **物理模型 (Physical)** .....它定义物理组织、包含结构和设备与设备互联的复合体。

● **用户模型 (User)** .....它定义用户、工作组和组织,以及这些对象与其它管理系统组件一起进行相互作用的方式。

● **策略模型 (Policy)** .....它定义一种信息模型和可扩展类层次结构,使应用程序开发人员能够说明控制任何类型的管理实体的访问和分配策略,以达到可以控制任何类型的管理实体的目的。

● **数据库模型 (Database)** .....它建立关系型数据库的操作模型。通用模型为向某一特定领域的扩展提供了基类集。每个通用模型都是具体实现的核心部分。核心模型与一个或者多个通用模型的合成提供了底层的 CIM 规范,这个规范由继承类层次结构和关系类层次结构组成,将一个域的结构和信息模型化。

### 2.3 扩展模型

扩展模型是指针对具体平台、技术及其它方面的考虑而对通用模型所作的扩展。它说明了针对通用模型在技术上的特定的扩展,指明它使用的操作平台和环境(例如,指明用于 Linux 而不是 Windows NT)以及实现对象(比如用于目录而不是关系性数据库)。这些模型都是依赖于具体环境和实现方法。这些扩展仍旧可以共享核心与通用模型的通用基础结构,因此它们也可以进行交互操作。例如,扩展可能针对不同的操作系统(例如,UNIX 和 Microsoft Windows)某些特殊功能的,也有可能是针对具体实现方式的(如,目录)。像我们的机群系统信息模型<sup>[1]</sup>就是针对目录的一种扩展模型。

### 3 目录与目录服务

目录是一个以一定规则排列的对象的属性集合,是一个存储着关于对象各种属性的特殊数据库,这些属性可以供访问和管理对象时使用,类似电话簿和图书馆卡片分类系统。这里,我们所谈的目录服务<sup>[2]</sup>是指网络目录服务。目录服务是指一个存储着用于访问、管理或配置网络资源信息的特殊数据库,它把网络环境中的各种资源都作为目录信息,在目录树结构中分层存储,对这些信息可以存储、访问、管理并使用。网络中的这些资源包括用户、各个应用系统、硬件设备、网络设备、数据、信息等。目录服务是为有效地

集成管理网络目录中的信息提供服务, 是支持网络系统的重要底层基础技术之一。目录服务将分布式系统中的用户、资源和组成分布式系统的其它对象统一的组织起来, 提供一个单一的逻辑视图, 允许用户和应用透明地访问网络上的资源。一个由目录服务支持的网络系统是一个集成的、网络化的、统一的系统, 而不是各个独立功能部分的简单聚合。在目录服务的基础上开发的应用, 易于使用、功能增强和易于管理, 目录信息的共享为应用的开发提供了方便。下一代分布式网络的信息模型和模式将是一种基于目录的, 当我们进入网络时, 是登录到一个基于目录的网络中, 而不是登录到某个机器上。

最早出现的目录服务协议是 1988 年 ITU-T 创建的基于目录服务的 X.500 标准协议, 之后, ITU-T 在 1993 年对 X.500 标准进行了重大修改和补充之后, ISO 接受了这个标准, 并将它作为 ISO/IEC 9594 国际标准。X.500 协议, 它是为了便于建立全局、分布式的目录信息而设计的一套协议。它是一种 C/S (Client/Server) 结构的应用协议, 支持用户通过网络访问和维护资源信息。

X.500 协议中包含了这诸多的强大功能, 可以说在很大的程度上满足了用户和开发者的要求, 但是在用户和开发制的使用过程中却逐渐暴露出其许多的不足之处, 阻碍了它的应用和推广, 主要集中在使用方式和性能开销上这两个方面, 由于 X.500 是在 OSI 协议栈上实现的, 虽然 X.500 提供了强大的信息查询功能, 但是协议所要求的访问格式却十分复杂, 往往难以掌握; 其次在运行 X.500 服务器时对系统的开销要求较高。为了兼顾保留协议的优点和克服这些不尽人意的地方, 推出了由 X.500 目录服务协议延伸发展而来的轻型目录访问协议 (Lightweight Directory Access Protocol, 简称 LDAP)。LDAP 协议<sup>[3]</sup>继承了 X.500 的 90% 左右的功能, 同时兼容所有使用 X.500 协议建立的服务器端数据库, 避免了重复开发的浪费; 在运行开销上却只是 X.500 的 10%。正是由于 LDAP 具有巨大的优越性, 它从一开始仅仅是作为 X.500 客户层的另一种实现方式, 到现在在许多应用中完全替代了 X.500 协议, 成为一个完整实用的应用开发协议。

LDAP 在以下四个方面对 X.500 进行了简化:

传输: LDAP 直接运行在 TCP 上, 避免了 OSI 多层通信的高层开销;

功能: LDAP 简化了 X.500 的功能, 抛弃了较少用到的功能和冗余的操作;

数据表示: X.500 的数据表示结构复杂, LDAP 采用简单的字符串对数据进行表示;

编码: LDAP 用于网络传输的编码规则比 X.500 的编码规则更加简单。

#### 4 LDAP实现机群系统资源信息模型

基于对信息模型的通用性、可扩展性和易于查找信息的考虑, 我们采用了 LDAP 来实现这种面向机群系统的信息模型<sup>[4]</sup>, 并实现了 CIM 信息模型到 LDAP 模型的映射。CIM 信息模型和 LDAP 模型之间的区别主要存在于对象标识、关系和数据类型三个方面, 把 CIM 信息模型映射到 LDAP 模型上主要就是要实现这三个方面的转换。

把 CIM 信息模型的名字空间映射到目录的域名空间需要保证 CIM 信息模型的每个实例都能被唯一标识。CIM 和 LDAP 中关于对象标识的组成和需要标识对象的范围都不同。CIM 对象实例的标识只需要在某个特定的名字空间中依赖键属性和类名, 键属性集合在所有的类实例和子类中都是唯一的。而 LDAP 中每个目录入口项 (Directory Entry) 都由它们的 DN (Distinguished Name) 来唯一标识, 并且 DN 也指明了它们在目录树中的存储位置。目录使用每个入口项的一个或几个属性来形成 RDN (Relative Distinguish Named)。使用 CIM 中的模型路径 (Model Path) 可以直接形成单键属性值的对象类的 RDN。由于 CIM 中存在多键属性值的对象类, 引入 OrderedCimKeys 和 OrderedCimModelPath 类来分别表示单键属性值和多键属性值的模型路径。这样就可以使用 OrderedCimModelPath 或者 OrderedCimKeys 来形成相应对象类实例的 RDN, 从而就解决了对象标识不一致的问题。

在 LDAP 中没有一个显式的机制来表示关系, LDAP 协议本身只能依靠目录的层次关系来表示简单的层次关系。为了在 LDAP 中表示 CIM 模型中的关系模型, 需要引入辅助对象来和关系对象类一起表示 CIM 模型中关系模型的实体对象的位置和相互关系以及关系模型本身的属性。由于关系类可以分为自身有属性的关系和自身没有属性的关系两类。在面向 Cluster 信息模型的关系中没有自身有属性的关系, 因

而相对比较简单。

在 CIM 信息模型中定义的数据类型也有些不能简单的直接映射到 LDAP 中的数据类型,从 CIM 信息模型中定义的数据类型映射到 LDAP 中的数据类型也需要采用某种机制作一定的转换。总共有字符、日期 (datetime) 和实数等三种数据类型需要采取转换措施。在规范中, CIM 使用 USC-2 字符集, LDAP 使用 UTF-8 字符集。一般来说给定的应用环境就提供一致的字符集。虽然也有一些应用需要进行字符转换,但是在这个面向 Cluster 的信息模型中,可以直接选取 ASCII 字符集作为统一的字符集,不进行任何实际的转换动作。CIM 的日期类型用来存储时刻和时间间隔, LDAP 中使用的通用时间 (Generalized Time) 标识也可以用来存储时刻和时间间隔。当用来存储时间间隔时,二者的语义是一致的。并且它们都将日期和时间组合成一个字符串来存储。但是由于表示的精度、开始时间标识和分隔标记等不同,二者之间的转换还是不能直接进行。由于 CIM 的日期时间采用固定长度格式,所以存储时可能在空白的区域填上“\*”,而通用时间格式是变长的,因而不可能出现“\*”。在进行转换时,从左到右如果遇到了“\*”就用“0”来代替,这样就可以完成转换。还有一个需要转换的就是开始时间标志和时间间隔的单位。需要将 CIM 的开始时间标志“0”转换为“Z”,时间间隔由小时/分格式转换为全部用分表示。最后一个需要进行类型转换的是实数。LDAP 是

一个基于字符串的协议,本身不支持实数这种数据类型,在 LDAP 中就需要用字符串来表示实数。CIM 支持 4Byte 的 32 位浮点实数和 8Byte 的 64 位浮点实数,64 位浮点实数的范围是  $1.7976931348623158e+308$  到  $2.2250738585072014e-308$ 。这就需要 3 位十进制数来表示指数,不包括小数点 17 位十进制来表示基数。在 LDAP 中就使用固定的格式来表示实数,从左至右依次是一位标识位 (分别用 1 到 5 来表示指数和基数符号的五种情况的组合)、三位指数位、十七位基数位。每个部分之间用空格隔开,不足的位置右补 0。这样就可以在 LDAP 中唯一的表示 CIM 中的 64 位浮点实数了。32 位浮点实数的情况就可以直接由 64 位浮点实数的表示方法来进行处理。到这里就完成了从 CIM 到 LDAP 的映射。

#### 参考文献

- 1 汤小春,胡正国,卢维扬.基于机群技术的作业管理系统.西北工业大学学报,2001,19(1):6-10.
- 2 赵明,卢文龙,郭军,常桂然.具有目录功能的网络(DEN)架构及其信息模型.小型微型计算机系统,1999,20(12):908-910.
- 3 任剑勇,肖依,黄彪.基于目录服务技术的应用开发.计算机应用研究,2001,18(5):143-148.
- 4 Neumann H. High Performace Computing (HPC) Cluster Scheduler. Linux Technology Cernter (LTC) System Management IBM, 2002.

(上接第 222)

#### 参考文献

- 1 周大伟,何宝民,冯楠.基于预知维修技术的装备维修管理.四川兵工学报,2009,3:105-109.
- 2 王致杰,王耀才,等.现代大型设备故障智能诊断技术的现状与展望.煤矿机械,2003,(7):103.
- 3 张丽莉,储江伟,等.现代汽车故障诊断方法及其应用研究.机械研究与应用,2008,2:1.
- 4 黄景德,王兴贵,王祖光.基于模糊评判的故障预测系统研究.电子机械工程,2000,12:6:42-44.
- 5 吴勉,邵惠鹤.基于时频分析与神经网络的实时智能故障诊

- 断系统的软件设计.总体构架与实时数据采集功能的开发.系统仿真学报,2001(增刊):194-197.
- 6 印签.专家系统原理与编程.北京:机械工业出版社,2000,5:67-109.
- 7 马伦,康建设,赵强.基于 HMM 的设备剩余寿命预测框架及其实现.计算机仿真,2010,5:88-91.
- 8 王三明,蒋军成.BP 神经网络的 C++实现及其在故障诊断中的应用.南京工业大学学报,2009,23(4):372-374.
- 9 李明钊.基于神经网络的电控汽油发动机的智能故障诊断研究.昆明:昆明理工大学,2008.