

文章编号: 1005-8451 (2007) 02-0039-03

## 一种电信网管系统配置模块的 CORBA 接口实现

朱菊花, 楼新远

(西南交通大学 信息科学与技术学院, 成都 610031)

**摘要:** CORBA 提供面向对象的互操作, 作为实现分布管理对象访问的处理方式, 是很有应用前景的系统集成标准。它能够使其他的专业子网以最小的代价进行平滑接入, 并最终实现意义上的综合网管系统。提出一种电信网管系统 CORBA 接口的实现框架, 讨论接口的组成成分和功能, 在此基础上详细介绍 CORBA 接口的基础—配置模块的设计和应用。

**关键词:** CORBA; 接口; TMN; 网管系统

**中国分类号:** TP39 **文献标识码:** A

### Implementing of CORBA interface in Telecommunication Management System

ZHU Ju-hua, LOU Xin-yuan

(School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China)

**Abstract:** The CORBA, as a means of implementation of distributed object management, was a promising standard for systems integration, which provided an interoperation standard for object-oriented application. The CORBA could connect other professional subnets smoothly with the minimum price, and implement Integrated Network Management Systems in real sense. In this paper, an implemening frame of Telecommunication Management System's CORBA interface was presented. The components and function of the interface was discussed. At the same time, it was introduced the base of the CORBA interface configure module's design and application in detail.

**Key words:** CORBA; interface; TMN; Telecommunication Management System

TMN 是一个有组织的网络, 是收集、处理、传送和存储有关电信网维护、操作和管理信息的一种综合手段, 它全面采用 OSI 系统管理体系结构, 具有信息模型结构和管理者/代理模式两点关键内容<sup>[1]</sup>。

CORBA 是对象管理组织 OMG 制定的一种分布对象技术的规范, 技术特点主要有: (1) 分布计算和面向对象编程思想融合; (2) 代理概念的引入; (3) 客户方与服务器方完全分离; (4) “软件总线”功能的实现; (5) 设计原则和设计方式的层次化。

本文设计了一种电信网管系统配置模块的 CORBA 接口的方案, 并在具体的电信网管系统设计中加以实现。

### 1 CORBA 接口

#### 1.1 CORBA 接口在系统中的位置

本文设计的 CORBA 接口就是指图 1 中的 IF2 接口。IF2 一般被安置在 OMC 端 (因此本文讨论的重点

是 OMC 侧) 该接口的作用是为 NMC 提供统一的管理接口。依据 TMN 对网管系统功能的划分, OMC 在功能上分为配置、告警、性能和计费 4 个部分。因为计费比较特殊, 所以通常与其他几部分独立。基于上述原因 CORBA 接口对应的分为配置、性能和告警 3 个组成部分。

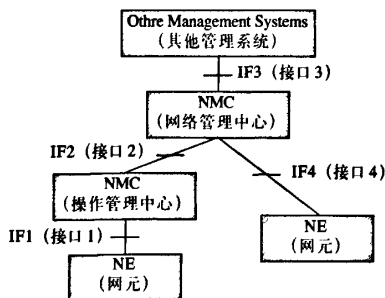


图 1 CORBA 接口的位置

#### 2.2 CORBA 接口各个组成部分及功能

整个 CORBA 接口分为配置、性能和告警 3 个部分, 分别对应着图中 CmIRP, FmIRP 和 PmIRP。如图 2。

收稿日期: 2007-08-09

作者简介: 朱菊花, 在读硕士研究生; 楼新远, 教授。

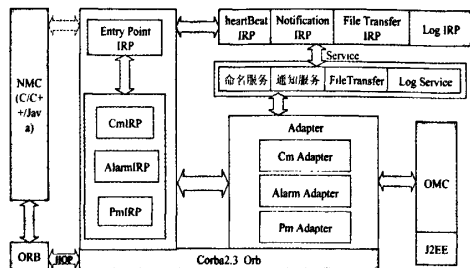


图2 CORBA 接口总体结构图

NMC (Network Management Centre, 网络管理中心), IRP (Integration Reference Point, 集成参考点, 一类对象), CmIRP (Config Management IRP, 配置管理参考点), AlarmIRP (告警管理参考点), PmIRP (Performance Management IRP, 性能管理参考点)。

HeartBeatIRP (心跳 IRP, 定时检查链路状况), NotificationIRP (通知 IRP), File Transfer IRP (文件传输 IRP), LogIRP (日志管理 IRP)。Adapter 部分的 Cm Adapter (配置管理适配, 是 CmIRP 调用接口的实现, 获取配置数据), Alarm Adapter (告警管理适配, 是 AlarmIRP 调用接口的实现, 上报告警数据), Pm Adapter (性能管理适配, 是 PmIRP 调用接口的实现, 用于获取性能数据)。

除上述配置、告警和性能模块外, CORBA 接口还有公用模块, HeartBeatIRP, 称为心跳 IRP, 用于 NMC 和 OMC 之间的链路定期检测。FileTransferIRP, 文件传输服务, 用于 NMC 和 OMC 之间的大数据量传输, 适合于 OMC 刚启动时向 NMC 同步信息, 另外该 IRP 也用于性能数据的上报。LogIRP, 用于记录日志, 记录 NMC 对 OMC 的操作, 同时也可以记录异常, 方便问题定位。

CORBA 接口配置模块功能可以分为: 基本配置功能、配置信息改变通知上报以及配置信息同步功能。基本配置功能包括对配置信息的创建、删除、查询和修改功能。配置信息改变通知上报是指当网元的配置信息发生改变时, OMC 应向 NMC 上报相应的配置信息改变通知。配置信息同步功能是指 NMC 应根据网络配置信息变化, 在需要情况下进行配置信息的同步。

性能管理功能包括性能采集管理功能、性能门限管理功能和性能历史数据管理功能。

性能采集活动是指从 NMC 的角度来看, OMC 从物理设备或逻辑功能中定期获取性能数据并上报

给 NMC 的活动。性能采集管理是指 NMC 对性能采集的相应参数进行管理, 通过对性能采集的管理, NMC 可以要求 OMC 按照要求上报所需的性能数据, 性能门限管理功能是指 NMC 可以对 OMC 的性能数据门限进行设置。

告警信息同步功能可以分为两类, (1) NMC 查询指定时间段内 OMC 中产生的所有告警历史信息; (2) NMC 查询 OMC 中的当前告警信息列表, 可以对告警信息设置过滤条件。告警处理功能项目较多, 主要包括: 告警确认/反确认, 告警清除, 告警说明等。

### 3 CORBA 配置模块的设计

#### 3.1 配置模块

图3是CORBA配置模块的Adapter结构图。



图3 CORBA 接口配置模块 Adapter 结构

CORBA 配置模块是整个 CORBA 接口的基础。因为 CORBA 配置模块初始化管理设备的组织结构, 目的是使 NMC 通过 CORBA 接口能方便的定位到管理设备, 以便能进行配置管理, 告警管理和性能管理等操作。同时 CORBA 配置模块需要完成对配置数据按需查询, 能处理管理设备上报的消息, 以及处理配置信息同步等功能。关于管理设备的组织结构, 一种常用的组织结构是树型结构。每个设备有一个唯一的 DN (iIdentifier Name)。例如: 可以这样标识一个bts. province=0, company=1, bsc=2, bts=3, 这种标识可以有效地定位到具体的bts。CORBA 客户端可以通过迭代的方式获取 OMC 的配置数据。

CORBA 配置模块的设计要素包括:

(1) 需要屏蔽不同产品、版本 OMC 接入方式的差别; (2) MO (管理设备对象, 一类管理对象对应一个 MO) 需要持久化; (3) MOC (管理对象类) 的包容关系集中维护, 采用 XML 文件方式保存; (4) 需要支持 scope (支持管理对象的筛选操作); (5)

采用迭代开发方式

### 3.2 一些可选择的设计方案及比较

方案1: (1) 在CORBA接口处保存所有的MO数据; (2) 对scope操作采用深度优先的搜索算法, 每一个压入queue中的记录为一个三元组(baseMOC, baseDn, submoc), baseMOC为搜索的基MOC; baseDn为搜索的基MOC的DN; submoc为在基MOC下面的子MOC; (3) CORBA接口和各个OMC采用定时同步的方法来同步CORBA的数据, 由各个adapter负责采集数据; (4) 北向接口只在从OMC收到Function级别的事件上报的时候, 才更新NI4C中的MO。

方案1的优点: 获取管理对象属性时, 不需要进行remote操作, 也不需要转换操作。缺点: 同步的实现非常困难, 系统比较复杂。

方案2: (1) 在CORBA不保存MO数据, 数据在OMC中的MO中保存; (2) 对scope操作采用深度优先的搜索算法, 每一个压入queue中的记录为一个三元组(baseMOC, baseDn, submoc), baseMOC为搜索的基MOC; baseDn为搜索的基MOC的DN; submoc为在基MOC下面的子MOC; (3) CORBA使用Adapter获取OMC中的MOI(DN)属性、获取MOI(DN)的包容信息。

本方案的优缺点正好与设计方案1相反, 另外一些CORBA北向接口中公共的MOC在OMC中没有对应的MOC, 设计带来了一些不一致性。

方案3: 设计方案3是对设计方案2的修改, 如果CORBA北向接口中的MOC在OMC系统中没有对应的MOC, 则MO的数据在CORBA北向接口内部保存, 并在启动的时候加载到内存中。在OMC中有对应的MOC同设计方案2。这个方案的MO存储位置更加自然, 略微增加了复杂性。

3种设计方案都各有优缺点, 适合不同情况的网络系统。

### 3.3 配置模块的实现

综合分析3.2中的方案, 最后选择了方案3做为CORBA配置模块的实现方案。下面是一些重要的实现类代码。

CORBA配置模块IDL的定义举例:

```
module BasicCmIRPSystem
{
    typedef string MOCClass;
    typedef string DN;
    typedef string MOAttributeName;
```

```
typedef any MOAttributeValue;
module AttributeTypes
{
    struct MOReference
    {
        DN otherMO;
    }
    enum ScopeType
    {
        BASE_ONLY,
        BASE_NTH_LEVEL,
        BASE_SUBTREE,
        BASE_ALL
    }
}
.....
}
```

IDL定义管理对象类, DN, 以及管理对象的属性结构和scope对管理设备属性和操作的描述, DN用于定位管理设备, scope用于筛选管理设备。

配置模块的启动流程如下:

(1) 创建用户POA; (2) 创建迭代器对象(用户获取配置数据的适配类)和访问策略(该对象只在服务端使用。因此, 不用注册到命名服务对象); (3) 创建Corba配置模块服务对象(new CmlrpServer()), 该对象是真正提供服务的对象, 并将其绑定到用户POA; (4) 创建CORBA配置模块的初始化对象(该对象主要是完成服务对象的初始化), 并绑定到用户POA上; (5) 注册配置模块服务对象和初始化对象到命名服务对象; (6) 激活POA。

## 4 结束语

本文基于TMN框架, 分析了配置模块的CORBA接口在其中的应用, 提出了一种CORBA接口的实现方案, 最后重点描述了CORBA配置模块的设计和实现方法。并给出示例。本文的设计思路已成功运用到电信网管系统设计实现中, 并取得了很好的效果。

参考文献:

- [1] Dirk Slama. CORBA 企业解决方案[M]. 李师贤. 北京: 机械工业出版社, 2001.
- [2] Robert C. Martin. 敏捷软件开发: 原则、模式与实践[M]. 邓辉. 北京: 清华大学出版社, 2003.