

文章编号: 2096-1472(2016)-06-31-03

基于CORBA的网管接口的设计与实现

陈欣

(中国地震局第一监测中心, 天津 300180)

摘要: 本文基于不同行业对网络大型综合管理系统的应用需求, 通过不同行业各类网络管理系统、网络设备之间通用接口的设计, 建立综合网络管理模块, 实现对各类网络设备的统一管理, 提高网管工作的便利性和通用性。对综合网络管理系统功能模块所需要的接口的详细设计与实现进行了相应的阐述, 设计并实现了各个功能接口、这些接口的实现类以及继承这些类的子类, 通过这些接口和类的实现, 再结合消息机制实现综合网络管理系统的各个功能。

关键词: 综合网管; 适配器; CORBA

中图分类号: TP309 **文献标识码:** A

Design and Implementation of CORBA-Based Interface

CHEN Xin

(First Crust Deformation Monitoring and Application Center,CEA,Tianjin 300180,China)

Abstract:Based on the application requirements of large-scale integrated network management systems in different industries,the paper sets up an integrated network management module through the design of universal interfaces of network management systems and devices in different industries.It achieves unified management to different types of network equipment and improves the convenience and universality of the network management.The paper elaborates on the design and implementation of the interfaces required by the functional modules in integrated network management systems,the design and implementation of each functional interface,class and its subclass.Combined the implementation of these interfaces and classes with the message mechanism,all the functions in the integrated network management system are implemented.

Keywords:integrated network management;adapter CORBA

1 引言(Introduction)

随着互联网的持续高速发展, 网络技术的应用遍及了各行各业, 各行业对于网络管理系统(以下简称网管)尤其对于大型网络综合管理需求日益明显。由于行业特点不同, 不同行业对于网络管理系统的应用需求存在着很大的差异。同时, 全球网络设备、网管系统提供商数量庞大, 来自于不同技术来源、技术习惯的各类软硬件设备千差万别, 给网络管理和数据共享带来了不便。为了实现各种网络设备的统一管理和综合运用, 以提高网络管理的便利性和通用性, 对综合网络管理模块在各个数据层对应的数据接口进行设计^[1-5]。利用基于JAVA语言的CORBA技术^[6-9], 编写华为SDH网管适配器。主要接口的实现分为: 适配层接口、管理层接口、数据库接口、上报接口、服务接口。

2 接口设计与实现(Design and implementation of interface)

2.1 适配层接口的设计与实现

适配层接口的功能是与厂家设备接口进行连接, 从而获

取所需信息, 完成相关功能, 它被适配层基类BaseAdapter类所实现, 并继而由XXAdapter类继承。如图1所示。

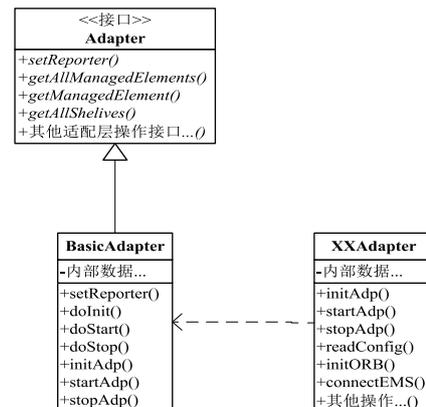


图1 适配层接口类图

Fig.1 Adapter layer interface diagram

这个接口的设计目的是提供给管理层接口, 供管理层管理适配层、通过适配层获取厂家数据。所以, 这个接口中所定义的方法都与获取信息有关, 同步资源时, 在获取指定网元信息getManagedElement和其框、槽、盘、

端口、交叉，以及拓扑信息之前，会先获取所有网元信息getAllManagedElements，同样，在获取指定设备的当前告警之前也要获取所有当前告警getAllActiveAlarms，而这些方法，被定义在这个类中。

将所有获取到的信息以List的形式存进相应的表中，保存到数据库中以备在配置管理功能、故障管理功能等功能模块运作时使用。需要说明的是，在上述方法中，有关获取指定网元的信息中，以所要采集的网元标识meId来判别所要获取哪个网元的信息，网元标识meId是每个不同网元的唯一标识，通过它的唯一性，可以准确的获取它的信息，而不会进行错误的信息采集。

在它的具体实现类中，除了对获取网元信息的方法的重写，还定义了初始化、启动、停止适配器，连接网管、读取配置文件，探测线程的运行、转化信息和保存信息等具体的方法。通过主函数调用这些相应的方法，就可以实现适配器的启动运行，为系统提供登陆北向接口、获取事件通道等系统接口。除了XXAdapter这个类之外，还有一些相应的辅助性质的类，比如对事件类型进行定义的EventType类、进行数据格式化的Format类、帮助类Helper类、层速率定义类LayerRate类、与日志相关的日志类Logger类、管理器名称类ManagerName类以及定义对象名称的ObjectName类，这些类都是给Adapter类提供各种相关定义支持的辅助类。

2.2 管理层接口和上报接口的设计与实现

作为管理层的两个接口，它们的作用是与上层进行通信，通过管理接口，可以对适配层进行统一的管理，而上报接口则是为了适配层能够主动上报数据而提供的。它们都由一个管理层基类BasicManager来实现。它们之间的关系如图2所示。

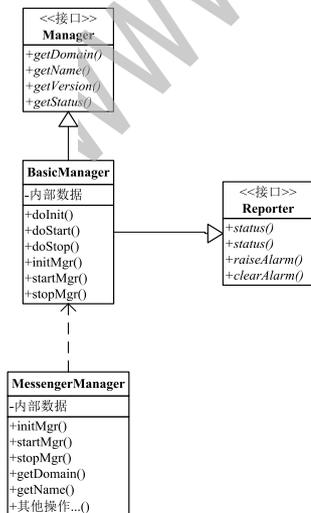


图2 管理层接口类图

Fig.2 Management layer interface diagram

通过manager接口可以获取适配器的相关信息，其中包括适配器的标识、名称、版本号，以及适配器的状态。而在reporter接口中，可以获取实体的状态、产生告警以及清除告警。

在它们的实现类BasicManager中，首先连接适配层接口，然后定义了初始化服务调用函数、启动服务调用函数和停止服务调用函数，以及初始化管理层、启动管理层和停止管理层三个方法。

MessengerManager类是基于消息传输机制的管理层具体类，它是实现资源同步和纠错、告警同步和比较等等功能的方法提供者，在这个类里，具体定义的方法有：初始化管理层并读取相应的数据，如config和common文件、domain值、name值、version值、localhost值、system值、resource值、alarm值、route值和interval值。启动、停止管理层，获取适配器资料，获取消息类型(资源同步，告警同步或者未知消息类型)。同步网元、主域等相关资源。数据纠错，包括网元，框、槽、盘、端口、交叉等数据，纠错的原则是比较采集上来的数据与录入库中的数据是否一致，如果采集上来的某数据在数据库能找到，并且所有属性都一致则表示无需纠错，如果采集的数据在录入库中无法找到，说明是设备侧新增添的数据，那么就要把采集上来的数据录入到数据库中，以完成纠错任务。如果采集上来的数据在录入库中可以找到，但是有某个或某些属性不相符，那么说明设备侧的设备数据与录入库中的数据不一致，发生了属性错误，那么根据设备侧采集上来的“正确”数据修改录入库中此数据的错误属性。还有一种情况就是在录入库中有而在设备侧没有的数据，可以认为是相关资源已经删除，所以根据正确的信息，将此类数据从录入库中进行删除。这样，关于资源数据的纠错，就先分种类(框、槽、盘等)采集再分以上三种情况完成了。

与此同理，这个类中还提供了对告警的相关处理，当相关的网元标识、适配器标识、对象标识被验证通过后，就可以开始同步告警了，首先要采集设备侧的告警，然后与资源一样与录入库中的告警进行比较，也是分为上述三类，这里为了避免重复不再赘述，当告警为新增告警时，就要去产生一个告警以便和设备侧同步，要产生一条告警，首先要确认告警信息、告警对象类型、告警对象名称、告警原因、告警时间是否为空，在都不为空的情况下进行初始化并将相应的信息进行赋值，例如告警类型、原因、内容、时间，以及相对应的网元设备数据等等。然后进行上报，提交到数据库中。另外一种情况是告警的删除，这里就要提供一个清除告警的方法，根据告警ID，将数据库中多余的告警进行删除就

可以了。

资源数据的纠错流程图如图3所示。

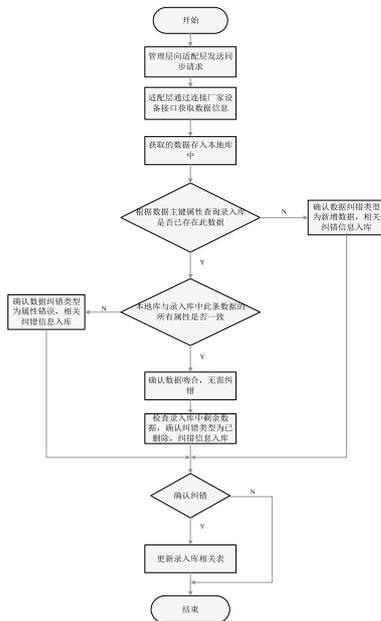


图3 数据同步纠错流程图

Fig.3 Data synchronous error correction flow

2.3 本地库接口的设计与实现

数据库系统采用Oracle 10g，而作为操作数据库的客户端软件，采用PL/sql。为了更直观的展示本地库接口以及其实现类之间的关系，绘制类图如图4所示。

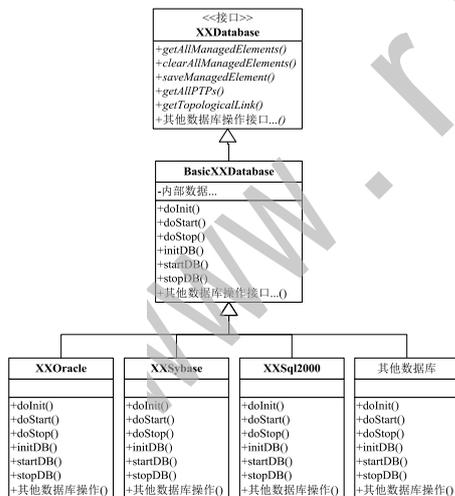


图4 本地库接口类图

Fig.4 Local data base interface diagram

本地库是为了方便数据的临时存储所存在的，从位置上来说，本地库是后台适配层的数据库，而录入库是GUI和管理后台所使用的库，当资源或者告警数据被采集上来的时候，首先要存放在本地库中，再将这些数据以消息的形式上报给GUI。在本地库的接口设计中，定义了保存和清除相关信息，比如保存网元信息，清除所有网元信息，清除指定网元信

息，保存端口信息，保存机盘信息，清除端口和机盘信息等。

本地库基类BasicLocalDatabase类是本地库接口的实现类，它定义了初始化、启动和停止相关的服务调用函数和初始化、启动、停止本地库。

基于Oracle的本地库类OracleLocalDatabase类。因为本设计的数据库设计是基于Oracle数据库设计的，所以必须要用一个具体的类来继承本地库基类，以实现针对Oracle的数据库设计，从这里也能看出，本设计结构有着良好的可复用性与可扩展性，通过继承基类，重写基类的方法，可以实现针对其他数据库的设计，比如可以编写针对sql server、Sybase等数据库的设计。这个类中主要定义了连接和关闭数据库，通过设置sql语句对数据库进行操作，主要是数据的获取(查询)、保存和清除。

2.4 服务接口与服务管理接口的设计与实现

通过这两个接口可以对服务对象进行匹配和创建，为适配器功能上的实现提供服务调用支持，相关类图如图5所示。

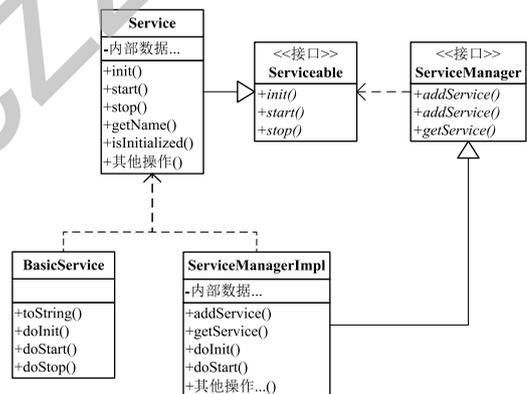


图5 服务接口以及服务管理接口类图

Fig.5 Service interface and service management interface diagram

服务类Service类是服务接口Serviceable的实现类，他定义了服务的初始化、启动和关闭，以及将对象字符串化的几个方法，和初始化、启动与关闭时的服务调用函数。BasicService类继承了Service类，是对Service类方法的重写。而服务管理接口的实现类ServiceManagereImpl类继承了服务类Service类，它通过创建服务映射表和顺序表来实现对服务的管理，它通过将服务类型进行添加，可以根据服务类型来获取相关的服务对象，在获取服务对象时，如果该对象不存在则创建一个新的服务对象，而创建服务对象、添加服务对象、获取服务对象、检查该服务类型的服务对象是否存在等方法在这个类中有着明确的定义。除

(下转第21页)