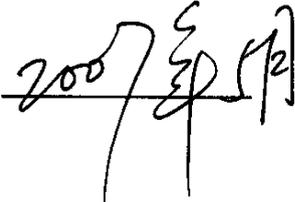


原创性声明

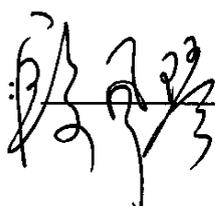
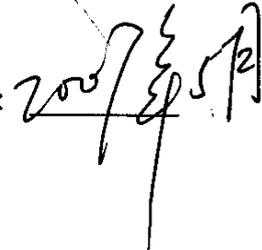
本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究在做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律责任由本人承担。

论文作者签名： 日期：

关于学位论文使用授权的声明

本人完全了解贵州大学有关保留、使用学位论文的规定，同意学校保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权贵州大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

(保密论文在解密后应遵守此规定)

论文作者签名： 导师签名： 日期：

摘 要

随着经济全球化趋势和中国加入 WTO，中国电信市场的政府管制力度将越来越弱，市场更加开放、竞争更加激烈。国内具有经营权的电信运营商，在全国范围内均拥有相当规模的公用电信网，经营多种基础电信业务和增值电信业务，形成本地网、移动、长途、数据、IP 电话、寻呼、增值业务等多种业务并存共同发展的格局，如何发挥各自的综合优势，实现灵活多变的营销策略，提高客户服务水平，成为当前电信企业迫切需要解决的问题。

经过多年的建设和发展，国内电信企业各业务子系统如营业、计费帐务、网络管理、资源管理等系统日趋完善，并支撑着企业日常的业务运营。同时，电信企业越来越意识到客户是企业生存之本，“一切以客户为中心”的经营理念正在改变着电信服务企业的运作方式，实施 CRM、商业智能、大客户/渠道管理系统将通过改善企业和客户的关系，从而达到提高企业利润的目标。

为了提高企业的整体运营效率，国内的电信运营商正在致力于建设企业业务运营支撑管理系统，建设新型的电信运营支撑系统需要参考国际上统一的框架和模型，以使设备供应商、网元提供商、系统集成商、应用开发商等能在一个框架范围内进行合作和协调。

电信综合网络管理系统应运而生，建立了面向整个电信运营网络的统一的数据支撑平台，提供了包括交换、传输、数据、PHS、动力环境等专业在内的跨专业、全网络的综合告警事件和性能报告。通过结合电信运营商的业务流程，以数据支撑平台为基础，系统将实现面向全网络的综合管理。

国内电信运营商采用的设备是多元化的，存在很多种协议，存在多种数据接口。例如 ASCII/TL1、CORBA、SNMP、CMIP/Q3 等；而 CORBA 作为一种总线结构能较好的解决电信应用领域中设备的多样性，平台的异构性。CORBA 可以提供统一的北向和南向接口，为电信运营商综合集成管理不同厂家的设备和网管提供了渠道，无论服务器端是什么操作系统客户端可以采取在同一个操作系统上实现所有服务器的连通。因此被许多家设备供应商采用，在电信运营网络中占有极大的比例。本论文的目的就是讨论 CORBA 在综合告警系统中的研究与应用。

本文首先介绍了当前电信行业的发展现状，提出了综合告警系统的必要性；之后介绍了综合告警系统的体系结构，详细描述了 CORBA 的发展历程、CORBA 原理；继而又阐述了 TMF 组织与 TMF 协议，以及 TMF 协议与 CORBA 的关系，及其共同在综合告警系统中的应用。本文最主要的内容就是详细设计并介绍了 CORBA 在综合告警系统中的应用，最后对综合告警系统的发展进行了展望。

本文研究的关键技术，创新点和所作的工作如下：

1. 探讨了当前电信行业对综合告警系统的需求，以及综合告警系统对电信设备提供商的设备接口的多样性的支持与实现设计。
2. 对 CORBA 协议进行了详细的分析，CORBA 一种异构平台下的语言无关的对象互操作模型。CORBA 规范充分利用了现今软件技术发展的最新成果，在基于网络的分布式应用环境下实现应用程序的集成，使得面向对象的软件在分布、异构环境下实现可重用、可移植和互操作。
3. 对 TMF（电信管理论坛）组织进行了简要的介绍，以及对 TMF 组织提出的 TMF 协议的认真研究，并且对于它与 CORBA 的相互关系进行了研究。在电信设备供应商提供的产品中有很多使用的是基于 CORBA 产品的北向接口，而这些接口又是必须符合 TMF814 协议的，这就增加了问题的复杂度。
4. 基于对 CORBA 规范的研究，对于符合 CORBA 规范的产品进行了广泛的了解，实现多家不同厂商产品之间的顺利连接。
5. 对综合告警系统中的数据采集部分进行了仔细分析与设计，针对 CORBA 接口类型的设备进行了彻底的了解。最终成功实现了对此类设备的数据采集工作。

CETA 电信综合网络管理系统解决方案依靠 TMF 技术与计算机技术相融合的全新思路，根据 TOM 的建议模型，将电信运营商的整个网络，包括交换、传输、数据、接入和 PHS 等专业，都集成到一个系统中，在集中管理运营网络的告警信息、性能信息的同时，提供与资源信息和客户信息互连的接口，为实现面向全网络跨专业的服务质量管理建立了综合统一的数据支撑平台。

本系统的设计开发目标就是要实现通过对综合告警系统的应用，为电信公司业务上达到集中化的规范管理提供良好的技术平台。

本系统已在江苏电信公司实现并实际投入运行，论文工作期间，作者已在《计算机科学》上发表论文一篇。

关键词：综合告警系统、数据采集、tmf、corba、中间件

图书分类号：TP311.52

Summary

With the economical globalization tendency and China join WTO Chinese telecommunication market government regulation dynamics are more and more weak. And the market is more opened, the competition is more intense. The telecommunication operation business who has the operating right has the suitable scale in the national scope the public telecommunication network, manages the many kinds of foundations telecommunication service and the increment telecommunication service. They had formed this network, the migration, long-distance, the data, the IPtelephone, the increment service and so on the many kinds of services coexists the communal development pattern. It becomes question which the current telecommunication enterprise is urgent needs to solve that how display respective comprehensive superiority, realizes the flexible marketing strategy, raises the customer service level.

After many years construction and development domestic telecommunication enterprise various services subsystem are consummates day by day, such as business, costs system, network management , resource management and so on. These subsystems are supporting the enterprise daily service operation. Simultaneously the telecommunication enterprise realizes the customer is the foundation of the enterprise survival gradually. The management idea of "all take customer ascent" is changing the telecommunication to serve enterprise's operation way. This may achieve the goal of enhancing the enterprise profit, through improvement enterprise and customer relations by the implements CRM, the commercial intelligence, the VIP customer/channel management system management system.

In order to mention the enterprise the overall operation efficiency domestic telecommunication operation business is devoting to the construction enterprise service operation strut management system which have provided including specialty and so on exchange, transmission, data, PHS, power environment cross specialized, the entire network synthesis warning event and the performance report. Through the union telecommunication operation business service flow, taking the data as the foundation to support the platform, the systems will realize face the entire network synthesis management.

The equipments used by domestic telecommunication operation business are multifarious. There are lots of protocols and many kinds of data interfaces. For example ASCII/TL1,CORBA,SNMP,CMIP/Q3 and so on. CORBA as one kind of main line structure

can adapt to the equipments' multiformities and the configuration of the different platforms. CORBA may provide the unification the north and the southing connections. It provides the channel for the telecommunication operation business to manage different factory equipment and the network management. Regardless of the server is on any operating system customer end may adopt on the identical operating system to realizes all servers connecting. Therefore it is used by many equipment suppliers and holds in the telecommunication operation network the enormous proportion. The present paper goal is discusses its research and the application in the synthesis warning system.

This article first brief the current telecommunication profession development present situation and propose the Synthesis Warning System necessity. After that it introduces the structure of Synthesis Warning System. Then CORBA development course and CORBA principle are described detailedly. Subsequently it also elaborates the TMF organization and the TMF agreement, as well as TMF agreement and CORBA relations, and its application together in synthesis warning system. The main content of this paper is to introduce and design CORBA's application in Synthesis Warning System detailedly. Finally it is to forecast to the synthesis warning system development.

In this article, the essential technology, innovation, the work has done are as follows:

1. Has discussed the current telecommunication profession to the synthesis warning system demand, and support and realization design to telecommunication equipment provider's equipment connection in Synthesis warning system.
2. Has carried on the detailed analysis to the CORBA agreement. Under the CORBA one kind of isomerism platform language irrelevant object mutually operates the model. The CORBA standard has fully used the nowadays software technological development newest achievement, achieved the application software integration based on under the network distributional application environment, causes object-oriented software in the distribution, under the isomerism environment the realization to be possible to entrust with heavy responsibility, may transplant and mutually to operate.
3. The brief introduction to TMF t, as well as TMF agreement earnest research which proposed to the TMF organization, and has conducted the research regarding it with the CORBA reciprocity. The product from the telecommunication equipment supplier has used north interface based on CORBA. But these connections also are must conform to

the TMF814 agreement, this increased the question's complexity.

4. Based on to the CORBA research, regarding conformed to the CORBA standard product to carry on the widespread understanding, realized between the many different merchants product smooth connection.
5. Has partially carried on the careful analysis and the design to in the synthesis warning system data acquisition, has carried on the thorough understanding in view of the CORBA connection type equipment. Finally succeeded has realized regarding this the kind of equipment data acquisition work.

The CETA telecommunication synthesis network management system solution depends upon the brand-new mentality which the TMF technology and the computer technology fuses, based on the model of TOM, All integrates the telecommunication operation business entire network to a system including the exchange, the transmission, the data, turns on with specialty, PHS and so on. During the centralized management operation network warning information and performance information, it provides the connection which interlocks with the resources information and the customer information. It established the synthesis unified data to support the platform for facing entire network cross specialized grade of service management.

Keywords: CETA、data collection、TMF、CORBA、middle-software

Classification NO.: TP311.52

第一章 绪论

1.1 电信行业的发展现状

近年来在信息技术革命和推动下，中国通信业发生了巨大的变化，成为增长最快市场潜力最大的行业，极大的促进了社会生产力的提高，对人们的工作和生活方式产生了深远的影响。

中国通信业是在发展中改革，在改革中发展，保持了稳定、快速、健康发展的良好势头，综合实力实现历史性跨越，在推进增长和社会进步，提高人民的生活质量上有了很好的推动作用，在技术先进、业务多样化的网络基本形成，首先我们实现了向世界第一大电信网络的跨越，电话用户数已经达到 6 亿 3 千万户，继续保持世界首位的地位，从 1882 年中国开通第一部电话到 1992 年我们达到 1000 万电话用户的时候，我们经历了 110 年的历史，从 1000 万用户到 1 亿用户我们仅仅用了 6 年时间，随后我们从 1998 年到 2000 年又有了增长，目前我们已经达到 6 亿 3 千万，互联网迅猛发展，用户总数居世界第二，电话普及率大幅度提高，业务速度快速增长，首先到目前为止我国综合普及率已经达到 49.3%，移动电话达到 24.8%，固定电话行动性的比例已经高达 90%左右，前四个月的业务收入已经较去年同期增长 13%，保持高于 GDP 的增长速度；电信业务结构日益多样化，各种新业务发展迅猛，目前世界上已有的各种电信业务我国已经开办，满足了多层次的需求。04 年 10 月移动短信的业务总量已经达到了 1760 亿条，比去年同期增长 60%多，宽带产业也得到了迅猛发展；不同规划、不同业务、不同所有制企业相互竞争、共同发展的市场格局基本形成，全国性的企业有 6 家、增值电信领域的企业有 1 万多家，70%以上是民营企业或含有民营成分的企业；初步建立了公平、公正的电信法的环境，国家先后颁布了全国人大常委会关于维护互联网安全的规定，中华人民共和国电信条例，外商投资电信管理规定等有关通信业的法律法规，信息产业部到今年年底为止一共公布了 32 个部长令，出台了一系列配套的规章。

市场经济是法制经济，政府监管必须依法行政，要全面推行国务院依法行政纲要为契机，推动政府职能转变，加快行为规范，运转协调、公正透明高效的监管体制，要进一步做好行政许可法实施后的相关工作，要贯彻纲要的要求，为了能在未来十年建设法制政府而努力，加强贯彻国办 75 号文件的力度，建立有序的市场竞争。一年多来的时间，经过有关部门的努力，两个文件得到了有效、系统的贯彻执行，使两个文件得到继续有效的实施。加强全局性、方向性、战略性政策的研究，政府对经济的管理应该后退一步，同时要站高一步，管宏观、管政策、管环境，促进通信业健康持续发展，要继续深化通信业的改革，促进公平公正

有效的市场竞争秩序的形成。党代会和经济工作会议上进一步提出要进一步完善电信业的改革。电信业是国有资本占主导地位的行业，是为了经济和社会服务的，电信业要始终重视高效的服务，服务于社会的发展，加强企业自律，要引导企业围绕国际一流企业的目标，全面提升创新能力，走质量、速度、效益、规模之路，努力进行发展。

电信网是当今社会中最为庞大和复杂的网络体系，是现代社会的基础设施。“智能化”是未来通信网的发展方向。智能化不仅要求网络有传递和交换信息的能力，而且还有存储和处理信息的能力。例如，要能够自动选择最佳的通信路由，使通信始终保持畅通无阻和高效运行。

现在无线通信已经在全球范围得到了迅猛发展，采用无线手段提供数据业务的应用成为新的通信热点。目前，各运营商、制造商和内容提供商都非常积极地工作，先后推出了 GPRS 技术、CDPD 技术等，这些技术为无线业务的迅速开展发挥了重要的作用。

作为关键业务及服务的承载平台，电信运营商的各专业网络正面临着保证服务可靠性和高效性的巨大压力，对网络基础设施和网络服务设施完全的监控已成为迫在眉睫的需求。所有的这些，要求有一种更先进的网络管理模式来积极主动地防止网络意外，持续而快速地修复网络中的常见问题。

1.2 综合告警系统简介

1.2.1 综合告警系统概念

综合告警系统是样一种基于先进网络管理模式的软件解决方案，它基于全球最先进的事件关联和处理平台，可以实现对不同厂商设备的交换、传输、数据以及电源等专业网络实施集中的故障管理，为运营商提供跨专业、跨厂商的综合网管平台。它通过与专业的告警关联，与故障单管理系统的对接，以及与客户信息的关联，帮助运营商提高网络和业务管理水平。

1.2.2 综合告警系统的功能

综合告警系统主要功能包括：对各种网元管理系统、专业网管系统的告警数据进行采集、处理和呈现以及关联分析的综合告警管理；对各种网元管理系统、专业网管系统的性能数据进行采集、处理和呈现的综合性能管理；提供端到端电路的路由管理和相关参数管理的综合端到端管理；实现自动业务激活的综合操作维护管理；对网络告警进行相关业务影响分析，对大客户业务进行实时监测的业务故障管理；用户和业务的 SLA 管理。系统还可以与故障单

管理系统和网络资源管理系统灵活地集成。

1.2.3 综合告警系统的市场现状

中国电信行业占世界电信极大的比重，中国电信市场被中国移动、中国联通、中国电信、中国网通等几家大的运营商分割。中国移动与中国联通主要经营移动通讯方面的业务，而中国电信、中国网通主要业务以固定电话和网络数据通讯为主。虽然各大运营商经营的业务略有差异，但运营设备的稳定性是他们向客户提供优质服务的前提条件，是他们在激烈竞争中保持不败之地的可靠保证。因此，对于各种各样的运营设备的监控与稳定性要求处于及其重要的地位。推动完善运营商“集中监控、集中维护、集中管理”的集中化运维模式，大大提高了运营商网络维护的效率和网络分析管理的水平是他们统一的追求。

每一种设备都有其自身的网管与监控系统，每一个网管都有其特殊性。监控中心工作人员的任务就是第一时间发现网管提示的告警，定位告警的设备类型具体位置，及时排除障碍，确保设备的正常运行，保证向客户提供完善的服务。进一步加强运行维护改革，提高网络运行监控水平，建立更完善的故障管理体系，加快对网络及用户故障的响应速度。而由于运营商使用的设备种类众多，网管众多，监控人员将大量的时间浪费于循环检查网管以发现问题，而导致监控效率低下，而增加监控人员数量虽然可以提高监控效率，但必然导致运营成本增加，因此，运营商就迫切需要一个可以监控所有或大部分设备告警的系统，也就是综合告警系统。

由于中国电信行业市场份额巨大，运营商众多，且中国地域广大，因此综合告警系统有很大的市场潜力，很多公司都看到了这一个市场需求，纷纷投入到综合告警系统的研发中去。目前中国有数家公司的综合告警系统分别在各省市做了试点及实验：

- 1、大唐电信承揽了杭州电信“全专业网络与客户故障监控系统”。
- 2、上海贝尔阿尔卡特综合化集中告警系统
- 3、亿阳信通移动通信网络管理系统
- 4、上海切点信息科技有限公司的综合告警系统

除了以上公司的综合告警系统，还有广州瑞达等一些公司也都有参与综合告警系统的研发工作。在今后的内容中介绍的主要是上海切点信息科技有限公司的综合告警系统。

1.3 CORBA 与综合告警系统

1.3.1 CORBA 概念

CORBA (Common Object Request Broker Architecture, 公共对象请求代理体系结构, 通用对象请求代理体系结构) 是由 OMG 组织制订的一种标准的面向对象应用程序体系规范。或者说 CORBA 体系结构是对象管理组织 (OMG) 为解决分布式处理环境 (DCE) 中, 硬件和软件系统的互连而提出的一种解决方案。

CORBA 标准由物件管理组织 (OMG) 设立并进行控制, CORBA 定义了一系列 API, 通信协议, 和物件/服务信息模型用于使得异质应用程序能够互相操作, 这些应用程序用不同的程序语言编写, 运行在不同的平台上。CORBA 因此为定义明确的物件提供了平台和位置的透明性, 这些物件是分布式计算平台的基础。

通常来说, CORBA 把用其他语言开发的程序码和关于该程序码能力和如何调用该程序码的资讯包到一个套装 (package) 中, 包成套装的物件则可以在网络上被其他程序 (或 CORBA 物件) 调用。在这个意义上讲, CORBA 可以被看作是一个机器可读的文件档格式, 类似于标头档 (header), 但是具有相当多的资讯。

在 CORBA 体系规范中定义了多种类型的服务 (Service), 如命名 (Naming)、生存期 (LifeCycle)、事件 (Event)、事务 (Transaction)、对象持久化 (Persistent Objects)、查询 (Query)、特征 (Property)、时间 (Time) 等服务功能。

在 CORBA 规范中, 没有明确说明不同厂商的中间件产品要实现所有的服务功能, 并且允许厂商开发自己的服务类型。因此, 不同厂商的 ORB 产品对 CORBA 服务的支持能力不同, 使我们在针对开发系统的功能进行中间件产品选择时, 有更多的选择余地。

1.3.2 CORBA 规范的技术特点

- 引入了代理的概念;
- 所实现的客户方程序与服务器方程序的完全分离;
- 将分布计算同面向对象的概念相互结合;
- 提供了软件总线的机制;
- 分层的设计原则与实现方法。

1.3.3 CORBA 产品一览

- IONA 公司的 Orbix、Orbacus;

- Inprise 公司的 VisoBroker;
- Digital 公司的 Component Broker;
- IBM 公司的 Component Broker;
- Sun Microsystems 公司的 NEO、JOE;
- SunSoft 公司的 DOE;
- 东南大学开发研制的 ORBUS;

在上面介绍的 CORBA 产品中,有一些是收费产品,如 IONA 公司的 Orbix 等,有一些是免费产品。其中付费产品中 IONA 公司的产品市场占有率比较高。另外一些免费产品,如 JacORB 的使用也很多。

1.3.4 CORBA 在综合告警系统中的应用

综合告警系统集成多专业、多厂家的网元设备进行综合管理,在实现故障管理、性能管理的基础上,结合资源信息和客户信息,向电信运营商提供支持。它最主要的作用就是对不同厂商、不同专业的不同设备进行监控,以保证设备的稳定性。因此,综合告警系统的最基础的部分就是对于不同设备的告警数据采集工作。综合网管系统从各专业的网元、网元网管系统、专业网管系统以及其他企业应用系统和数据库中采集所需要的各种数据,而由于告警设备的多样化,设备网管的复杂性,网管接口的多样性,要实现与所有设备的网管的联通就成为一个及其困难的事情。

现有设备的网管接口类型主要有 CMIP/Q3、CORBA、SNMP、各种非标准接口(数据库、厂商 API、日志文件、串口、X.25、TCP 端口等)。CORBA 接口是各种设备的网管接口类型中占有很大的比重,属于很重要的接口类型。目前有很多厂商的设备北向接口采用的是 CORBA 接口,例如 UT 斯达康公司的 NETMAN6000 网管系统、武汉烽火科技的 OTNM2000 网管系统、深圳华为的 T2000_ASON 网管系统、深圳华为的江苏省网波分 1 系统、北京格林威尔公司的 UniView SE、UniView DA 网管。

在上面介绍的网管中深圳华为的 CORBA 接口网管与武汉烽火科技的网管应用最多。

利用 CORBA 系统进行分布对象应用的开发具有下面三个特点:①开发代价小、效率高。系统开发者只需要编写描述服务对象接口的 IDL 语言文件并安装描述文件,实现服务对象的功能即完成了全部的任务。其它的相关代码或者是由 IDL 编译器自动创建,或者是由 ORB 类库提供,应用程序员并不需要编写例如网络通信、数据编码/解码、名址映射和安全管理程序代码,从而可以把工作重点放到服务对象实现的过程中去。②通过 CORBA 系统的支持,一个服务对象可以透明地被分布在本地和网络上的客户所调用,扩大了服务对象的使用范

围，为分布的客户所共享。③CORBA 系统作为“软件总线”，可以为服务对象提供“即插即用”的功能，而且当对象实现改进或升级时，只要接口保持不变，客户代码无需作任何改动。

1.4 拟解决的关键问题

1、客户端程序对不同厂商的 ORB 互联时存在的问题。尽可能减少在连接不同厂商设计的 ORB 产品时应用代码的修改量。虽然不同厂商的 ORB 可以互联，但是每一个厂商的产品都有其自身的特点，在很多细节的部分仍旧不一样，要做到一个客户端连接多个服务器完全不修改代码还不可能。但可以尽可能减少修改量。

2、网管设备是同一厂商的，ORB 也是同一厂商的，但是一台客户端连接多台服务器时，可能会存在同步的问题，怎么尽可能的减少获取各个不同地区的服务器告警时间的延迟。

3、每次获取的只能是当前告警，每隔 60S 获取一次，怎样判断获取的告警之中是否已经存在于之前获得的告警。通过事件请求是否可以申请只获取新告警。

4、轮询多少次之后断开连接，重新建立连接。以保证服务的稳定性。

5、进程异常停止后多长时间内自动重启。

6、出现大批量的数据时是否能保证系统的稳定性。

1.5 技术路线及试验设计

整体设计按照先用 java 自带的 ORB 先测试，在 windows 下调试成功后再移植到 linux 运行。之后考虑使用厂商开发的 ORB，如 TAO、ORBIX、Orbacus 等。

1、首先进行整体策划设计，搭建平台。建立最简单的客户/服务器模式测试。然后按照以下步骤进行：

- 1) 基于要提供给最终用户的功能来设计和编写代码中非 CORBA 部分。
- 2) 选择支持应用程序功能的通信样式（同步、延迟同步、单向）和激发类型（桩类型激发、动态激发或两者都选）。
- 3) 使客户机激发请求所需的接口定义可用，从而支持所选的激发类型。
- 4) 编写实际激发所使用的请求并获取激发所需的信息，如获取正确对象引用和其他参数的客户机应用程序代码。
- 5) 编写得到成功请求的结果或返回的客户机应用程序代码。

6) 编写客户端应用程序代码以处理返回的表示该请求有问题发生的任何自定义或标准异常。

2、在 windows 下测试成功后，进入 linux 系统测试调试。

3、测试成功后考虑程序性能问题，比如多次异常推出之类的情况，是否设定次数不允许客户端重连以节省服务器端的开销。

第二章 综合告警系统的体系结构

2.1 建立综合运维机制的必要性

随着市场竞争的日益激烈以及网络向 NGN 的演进，如何兼顾网络技术发展和投资保护，快速响应大客户个性化的综合业务需求，是目前接入层投资建设的焦点问题。

下一代网络是以分组交换为核心的业务驱动型网络，与现有的网络架构完全不同。对于占网络整体投资 70% 的接入层来讲，要实现接入层的融合，还存在一定的问题和困难。首先，在业务提供上，现有接入层网络因核心网分离而采用叠加建网的方式，导致业务的开展受交换机机型的限制和制约，与 NGN 分层建设的思路相矛盾，业务扩展性差。其次，在综合业务需求上，大客户对宽窄带及数据等综合业务的需求还不明显，现有建网方式基本能满足其打电话和宽带上网的业务需求，接入层综合业务统一平台的优势得不到很好的体现。再次，在维护体制上，传统的维护方式按业务网独立设置维护部门，对于多业务需求的用户响应要协调多个部门，业务开通时间长，故障受理环节多，已经极不适应竞争环境的发展，也阻碍了接入层的融合。

现阶段，许多运营商都已经意识到这些问题。在综合接入网的维护上，运维体制由原有的分散维护体制向统一集中网管和大客户 SLA 服务的方向进行调整。因此，有必要建立一种“以客户为中心、效益为导向”的综合运维机制，面对 NGN 网络设备分层、业务融合的发展趋势，做好维护体制变换的准备；面对大客户综合业务需求的变化，在有效降低维护成本，确保网络安全、高效运行的同时，提高对用户、对市场的响应速度，更好地为用户和市场服务。

现有的网络管理系统是运维部门的 IT 支撑系统的重要组成部分，经过多年的建设，各种网管系统初步满足了每种专业的网络或服务的管理和监控的需要，而这与公司的经营、生产、管理现状与现实的要求还有很大的差距。

现有的管理和监控系统存在以下问题：

1) 监控效率低下：

多厂商设备、多技术种类、多专业网络并存、监控终端众多、告警数量巨大、缺乏足够的高经过律手段，导致监控人员无法把精力集中在重要告警上，造成监控效率低下。

2) 故障定位困难：

专业内厂家监控系统多，彼此没有接口；各专业网管相互独立，缺乏故障关联手

段，对涉及各专业的故障不能自动定位，无法机制找到根告警。

3) 客户故障难以发现：

缺乏故障与客户资料和业务信息的关联，无法保证对大客户的有效支撑。

4) 故障单自动生成复杂：

故障信息分散在各个系统中，很难实现故障信息向故障单的自动传递。

针对接入网的综合化和融合化趋势，为了提高客户的满意度和降低维护成本、达到减员增效的目的，应该建立分层的接入网综合维护机制，其核心思想是“设备层统一维护，业务层分权维护”。

首先是设备层应统一管理。接入网网管中心负责综合接入网的设备层维护，其网络管理系统可以实现窄带语音接口板、宽带 ADSL、LAN 接口板、数据专线接口板等基本硬件的维测、硬件故障告警、环境电源监控和线路测试等，即保障接入层物理网络的安全稳定运行。对于设备层的统一管理，可以根据网络规模的大小，采用不同的网管平台，通过选用不同的服务器配置（可以是台式机、PC 服务器和 workstation）和服务器数量实现全网设备的统一集中网管。该集中网管系统支持全网的拓扑管理（例如全网 OLT、ONU 的组网拓扑）、环境电源监控（包括综合接入设备机柜、机房环境的监控）、集中告警等功能，并可以根据行政区域的划分支持多个客户端的域管理和授权，收集和统计各客户端上报的故障告警、性能统计等信息。各客户端主要负责本区域设备的管理维护，接受统一网管中心的权限管理，向统一网管中心上报告警数据等。

其次是业务层应分权管理。数据维护中心负责多业务的发放，其业务管理系统是根据不同业务网资源情况，进行纵向调配的专业资源管理和用户管理系统。由于各专业业务网络的维护人员对业务层比较熟悉，他们的精力可以集中在如何更好地开展各种业务和如何使网络增值上。而对于设备、线路的维护则可以统一交给设备层的接入网网管中心。业务层的分权管理主要是建立多业务的发放策略，通过业务端口管理功能实现多业务网络（PSTN、DDN 和 ATM/IP 网络）的业务发放。例如，窄带语音业务的开通，就是通过业务端口管理功能中的 PSTN 端口管理模块和 V5 接口管理模块，建立交换机和接入网之间业务发放的渠道，从而保证交换机侧和接入网侧 V5 接口和 E1 中继接口数据配置的一致性，实现语音业务的安全发放。同时，各业务的发放渠道在统一网管平台上有相应的权限控制，维护人员可以根据自身的专业特长维护相应的业务网络。

综上所述，“设备层统一管理，业务层分权管理”的综合运维机制，通过建立统一集中网管平台和多业务的发放渠道，可以充分保障综合业务的顺利开展和网络的安全、可靠运营，是面向下一代网络在接入层建设上实现可维护性设计的有力措施。

2.2 综合告警系统的架构

电信综合告警系统建立了面向整个电信运营网络的统一的数据支撑平台，提供了包括交换、传输、数据、PHS、动力环境等专业在内的跨专业、全网络的综合告警事件和性能报告。通过结合电信运营商的业务流程，以数据支撑平台为基础，系统将实现面向全网络的综合管理。

电信综合告警系统由一系列基本的软件产品组成，包括综合网管平台核心系统 CETA core（包括 CETA Core 核心系统、数据采集分析软件 LODE、分布式数据服务器 TACNODE 等）、告警事件处理软件 AlarmMonitor、可视化应用程序框架 Archon、接口适配软件 Probe、综合网管 WEB 客户端软件 PromptWEB、自由报表软件 CoreVision、网元控制管理软件 SCM 等。各管理模块均支持灵活的定制，可实现跨专业的信息关联，并能够调用相关的资源信息和客户信息。

综合告警系统将提供数据采集分析、数据归并收敛、网络事件可视化表现、告警事件管理、多视角网络管理、故障关联定位、统计分析和报表生成、流程管理等功能。

2.2.1 综合网管平台 CETA

综合网管平台 CETA 由核心系统 CETA core 和其他各种软件组成，可分成多个层次：

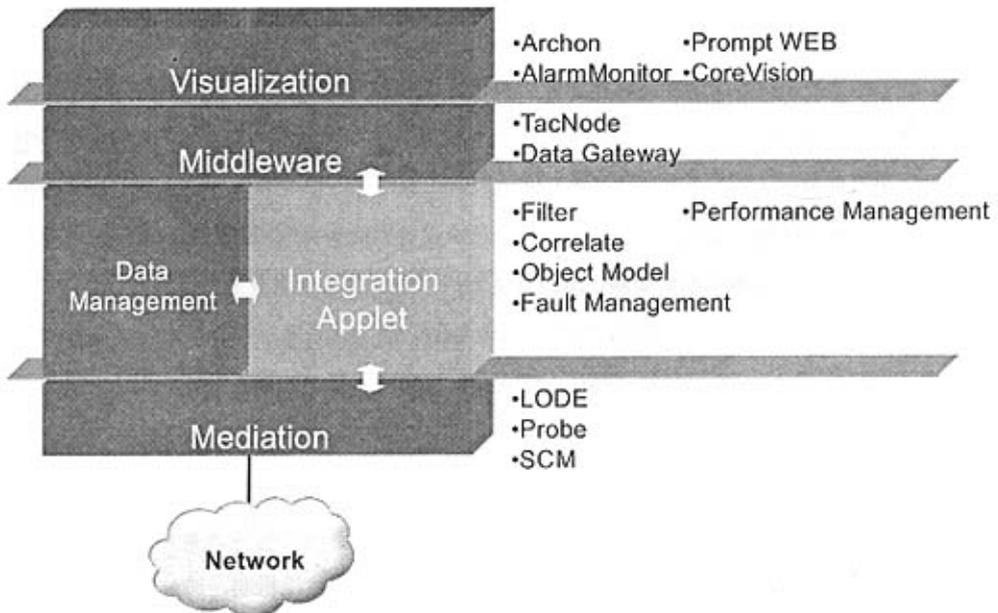


图 2-1 电信综合网络管理系统产品架构

1) 数据采集层 (Mediation) :

系统通过适应本地化需求的接口适配软件 Probe 和数据采集分析软件 LODE 对各专业网络的告警数据进行收集并分析, 实现与其他系统的互联。

2) 数据处理层 (Data Management) :

综合网管平台核心 CETA core 将实现对跨专业、跨厂家的告警信息、资源信息和客户信息的整合, 支持的数据库包括 Oracle 数据库、Sybase 数据库。

3) 数据中间件 (Middleware) :

分布式数据服务器 TACNODE 将进行数据提炼, 对表现层的各个应用分发数据, 高速、低耗地实现数据共享。

4) 应用表现层 (Visualization) :

图形化网络决策支持系统 Archon、告警图表统计分析系统 Actual、基于 Windows 平台的告警监控 AlarmMonitor、综合网管 WEB 客户端 PromptWEB、自由报表软件 CoreVision、网元控制管理系统 SCM 等将实现跨专业、跨厂家的综合管理。

系统采用层次架构, 提供了故障管理、性能管理、可视化系统、报表统计分析系统等在内的完整支持。系统具备丰富的数据接口供进一步的开发和功能扩展, 怡杰公司将在其基础上, 进行配置、开发和集成, 能够实现运营商的各种本地化要求。

2.2.2 接口适配软件 Probe

系统根据网络的实际情况, 从专业网管系统、网元网管系统或直接从网元采集所需要的网管数据及其他信息, 系统对专业网管系统数据、网元网管系统数据和网元数据进行分析处理, 采用统一的格式输入综合数据支撑平台中。

电信运营商的网元种类、网管系统等非常复杂, 为了实现其各种特殊的本地化需求, 怡杰公司开发了接口适配软件 Probe 以实现与网元、网元网管以及其他系统的互连。

在保证系统与现有的网管系统、应用系统的数据一致性, 避免重复开发以及对网元设备的数据采集压力等前提下, 系统可根据需要从专业网管系统、网元网管系统或直接从网元中获得所需要的网管数据。

系统包括一系列接口程序, 用于连接各种网元、网元网管系统和专业网管系统, 实现对包括串口文本数据流、串口二进制数据流、数据库、网元网管系统 API、X. 25、TCP/IP 等多种方式的数据采集。

在保证电信综合网络管理系统与现有的网管系统、应用系统的数据一致性, 避免重复开发以及对网元设备的数据采集压力等前提下, 系统可根据需要从专业网管系统、网元网管系

统或直接从网元中获得所需要的网管数据。

系统包括一系列接口程序，用于连接各种网元、网元网管系统和专业网管系统，实现对包括串口文本数据流、串口二进制数据流、数据库、网元网管系统 API、X.25、TCP/IP 等多种方式的数据采集。

功能及特点：

- 将不同的数据转换为标准格式，连接综合数据支撑平台，对网元网管数据进行分析和处理；
- 通过长期的经验积累，以及与电信设备厂商良好的合作关系，怡杰公司具有丰富的电信运营经验和软件经验，熟悉各种数据接口及协议的详细情况；
- 支持各种网元、网元网管系统的数据接口，并支持对各种复杂系统的数据分析、处理、调用和互连；
- 采用业界流行的开发工具，支持与 Q3/CMIP、CORBA、SNMP 等接口的互连；
- 与系统中其他软件实现紧密的集成，能够配合网管平台对网元网管数据进行分析和处理；

2.2.3 数据采集分析软件 LODE

数据采集分析软件 LODE 将实现对原始告警、性能数据流采集后的分析及入库，并提供脚本编辑、调试工具，以及对自身运行状态的监控和管理。

通过 LODE，用户可实现对告警数据的采集、分析、建模及入库等工作，这些过程通过编辑配置文件实现，可根据需求进行灵活定制；系统支持对告警标准模型的定义、修改，并可根据需要针对不同的应用增加新的模型；系统可结合告警原始信息和资源信息，将告警标准模型中的相关内容进行细化，并定位到网元最小监控单位，以满足告警定位的具体要求；

2.2.4 分布式数据服务器 TACNODE

分布式数据服务器 TACNODE 实现了应用系统对数据库的中间件功能，并提供对自身运行状态的监控和管理。

TACNODE 的目标是实现数据中间件的功能，为多种应用提供数据的综合数据服务系统。

TACNODE 的特点是对数据库中的数据提供缓存，防止应用程序频繁访问数据库，减轻数据库和数据库服务器的压力，使整个系统能够运行的更稳定、效率更高，同时它还提供对实时数据的快速转发，满足各种不同应用的需求。

2.2.5 告警事件管理软件 AlarmMonitor

AlarmMonitor 是怡杰公司开发的基于 Windows 平台的告警监控系统。系统实现了告警的实时表现、告警过滤、告警关联查询、告警内容打印、告警触发（声音、闪烁、电子邮件、短消息）等功能，提供对告警的资源信息、客户信息的查询，并实现面向客户的专题管理。

系统可安装在 PC 或工作站上，通过集成分布式数据服务器实现和综合数据支撑平台的互连，并向网络中所有安装本系统的终端提供告警管理服务。



图 2-2 AlarmMonitor 工作界面

AlarmMonitor 将帮助操作维护人员及时有效地了解到设备和网络出现的非正常运行状态；通过各种图形，图表，颜色和语音方式向操作人员提示有关告警内容和告警统计信息；帮助操作人员确定故障原因和故障位置，以便能及时纠正问题，保证设备和网络的正常运行；系统对告警的处理延迟时间小（从告警发生到告警显示），从而大大提高维护人员的工作效率，同时帮助运营商提高服务质量及市场竞争力。

2.2.6 告警统计分析系统 Actual

告警统计分析系统 Actual 是根据电信运营商的本地化需求开发，基于 Windows 平台，提供统计分析的报表及图表生成、报表保存、打印输出等功能。系统包括单一事件及资源统

计分析、事件及资源分类统计、事件及资源分时统计、二元统计四个基本功能模块及注册管理和窗口两个辅助模块，并采用文字式菜单与图形化工具按钮相结合的人机界面，使系统的使用更加方便和快捷。

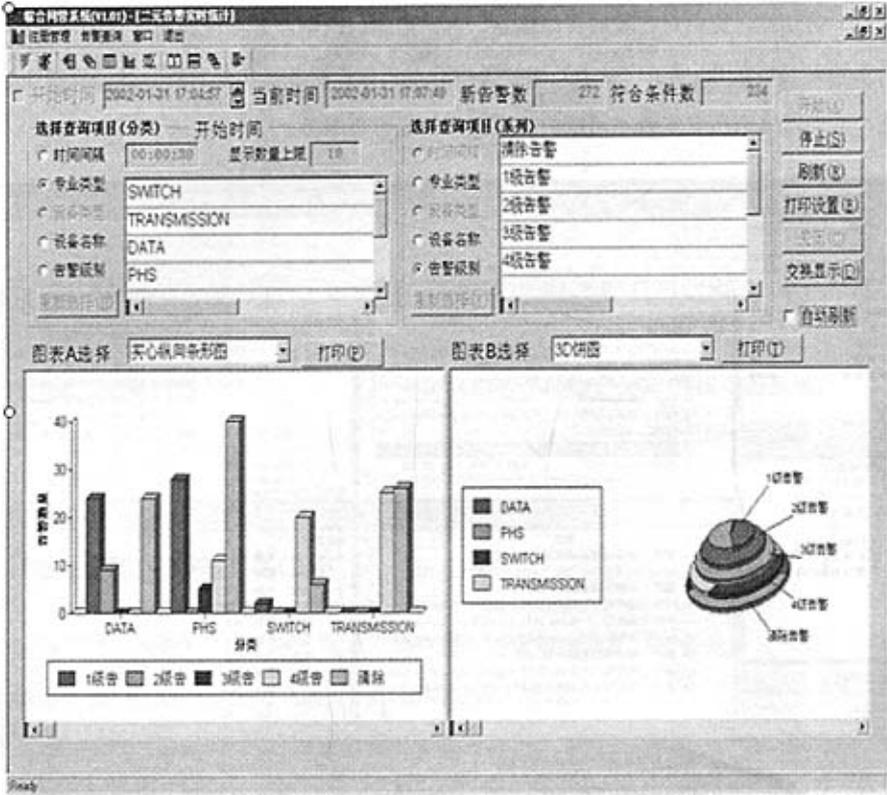


图 2-3 告警统计分析系统工作界面

2.2.7 综合网管 WEB 客户端 Prompt WEB

作为综合网管系统的 WEB 客户端, PromptWEB 提供了基于 WEB 浏览器的表现和管理方式, 实现对系统的各种告警、性能对象模型、相关性对象、过滤规则、资源信息、客户信息进行维护和管理; 观察、分析、处理历史和实时的网络告警及性能数据; 对网络、服务和业务的发展趋势进行判断和预测自身等。

2.2.8 自由报表软件 CoreVision

自由报表软件 CoreVision 可根据用户需要, 自由定制报表格式和报表数据源, 可根据

需要手动、自动或定时查询数据库，并生成相关报表。

CoreVision 的主要功能包括：日常表格处理、报告单位集定义与管理、表集定义与管理、报告周期管理、报表模板管理、汇总条件设定、多维表数据分析、数据备份导入导出

2.3 研究实现的网络环境及硬件环境介绍

2.3.1 网络环境

综合告警系统的应用环境是基于 DCN 网的。DCN 网是 Data Communication Network 的简称。

随着本地电信网络的扩大，业务的增多，网络种类和结构层次日趋复杂。在维护和管理电信网络的工作中，网管系统是必不可少的，是运行管理的关键。但采用不同技术的网络，甚至采用同样的技术而不是相同生产厂家的产品，由于网管互通性差，运营商都必须配备各厂家的管理系统。如此之多的管理系统都需要与被管理的设备或其他相关系统进行连接，需要专用的网管信息传输网络，这就是 DCN。

采用本地网现有的 DDN、FR 网或 SDH/PDH 传输网可以起到 DCN 的作用，如采用专线来组成网管数据传输网。这在网管网络规模较小时，造价低，且工程实现容易。但在网管网规模较大时，随着接口协议种类的增多，需要采用协议转换设备，每增加一套网管系统就需要配置相应的设备和独立的传输电路。这将产生如下问题：①协议转换设备的品种多生产厂家多，该部分设备本身维护管理都成了问题；②独立配置的传输电路在维护管理上易出现职责划分和单位之间的配合问题。

尤其是通信网络是在不断发展扩大的，网络和设备种类在增多，网管系统也在继续增加，DCN 网也必须随之扩容。采用专网专用，分别建设和管理，在运行期间，会遇到很多企业内部组织之间的管理职责冲突。另外，DDN/FR 网络规划一般是根据专线用户市场变化确定的，而且网络容量一般分布较为均匀。节点容量规划时无法考虑到日后运营商自身网管建设的需要。而网管系统传输电路是需要网管中心所在地点的 DDN/FR 节点机集中终结，大量专线集中到一个节点，该节点带宽和端口极易拥塞，造成整个 DDN/FR 网络联通性降低。

因此，在网管网规模大到一定程度时，就应该建设一个统一的综合数据通信网。江苏电信的运维就是建立于内部 DCN 网，同时综合告警系统也是建立于内部 DCN 网络。

2.3.2 硬件环境

综合告警系统从硬件方面分为三部分：采集机、应用服务器、客户端。其中采集机是采用 hp 的服务器，使用 Red Hat 8.0 系统，包括 CETA Core 核心系统、数据采集分析软件 LODE、分布式数据服务器 TACNODE 等在内的几大部分都是在此系统上实现的。

可视化应用程序框架 Archon、综合网管 WEB 客户端软件 PromptWEB、自由报表软件 CoreVision、网元控制管理软件 SCM 等是在应用服务器上实现的，它采用的同样是 hp 的服务器，但使用的是 windows 2003 server 系统。只有告警事件处理软件 AlarmMonitor 是在客户端上，顾名思义，客户端是软件系统直接面向使用人员的平台。在这里，客户端是指提供电信监控人员对设备进行监控，察看告警，统计告警的平台。

第三章 信息传递方式和 CORBA 中间件技术的发展

3.1 电信综合网管信息传递方式的分类

综合网管系统适应国内电信运营商网络多元化的设备现状，支持多种协议，提供丰富的数据接口，能连接并管理包括 ASCII/TL1、CORBA、SNMP、CMIP/Q3 等主要类型的设备；通过设置系统的应用服务网关，系统可连接其他厂商的应用系统、管理系统和数据库，包括 Sybase、MS SQL、Informix 和 Oracle 等。非标准接口则包括串口、X.25、厂商 API、厂商日志、厂商数据库等多种情况。

电信设备点多面广，且生产厂家、网管系统型号各不相同，长期以来一直无法实现真正意义上的综合性网络监控。

3.1.1 ASCII/TL1

TL1 (Transaction Language 1) 是一种在电信领域广泛使用的管理协议，尤其在北美。其得以在世界范围流行，是因为它能管理多种宽带网和接入网，包括 SONET/SDH, ATM。上述这些其中一个关键的原因就是，和其他的协议比较，它很容易实现和维护。TL1 报文有一个很好的，甚至职业用户和操作者都能设计和理解的结构化格式。例如，REPT 报文指示报表和 RTRV 指示检索等等。在早些日子，传统上命令行接口 (CLI) 成为大多数设备 (网络设备或电信设备) 厂商必须提供的完整的接口。管理这些设备的唯一方法就是通过串口连接，发送命令。这使得 TL1 更加流行，并且 CLI 在电信领域成为首选。

TL1 描述了以下四种消息类型：

- 1) 输入命令消息 (Input Command Messages) : 输入命令消息是一种由操作系统或其它资源 (如管理者) 发送给网元 (如代理) 的消息，这些消息请求网元执行某种动作。
- 2) 确认消息 (Acknowledgement Messages) : 确认消息是网元发出的简短应答消息，用来表示一条输入命令消息正被执行或是已经被拒绝。由于需要确保用户能了解在发出的执行命令是否最终被网元所接收到，所以，确认消息类型是有必要存在的。

- 3) 响应消息 (Response Messages) : 响应消息对输入命令消息的详细应答 (或一系列应答) 消息, 包括该命令消息是否成功执行消息和需要返回给操作系统/用户的任何数据。
- 4) 自治消息 (Autonomous Messages) : 自治消息是一种由网元周期性产生的消息, 或者是一种用于报告某种异常发生的消息。

广义上讲, TL1 消息可以逻辑地分为两种类型, 即输入和输出消息类型。其中, 输出消息类型包括确认消息、响应消息和自治消息。所有的消息类型都有自己的消息格式和结构。

3.1.2 简单网络 管理协议 (SNMP)

简单网络管理协议 (SNMP) 是目前 TCP/IP 网络中应用最为广泛的网络管理协议。1990 年 5 月, RFC1157 定义了 SNMP (simple network management protocol) 的第一个版本 SNMPv1。RFC1157 和另一个关于管理信息的文件 RFC1155 一起, 提供了一种监控和管理计算机网络的系统方法。因此, SNMP 得到了广泛应用, 并成为网络管理的事实上的标准。

SNMP 在 90 年代初得到了迅猛发展, 同时也暴露出了明显的不足, 如, 难以实现大量的数据传输, 缺少身份验证 (Authentication) 和加密 (Privacy) 机制。因此, 1993 年发布了 SNMPv2, 具有以下特点:

- 支持分布式网络管理
- 扩展了数据类型
- 可以实现大量数据的同时传输, 提高了效率和性能
- 丰富了故障处理能力
- 增加了集合处理功能
- 加强了数据定义语言

简单网络管理协议 (SNMP) 在体系结构分为被管理的设备 (Managed Device)、SNMP 管理器 (SNMP Manager) 和 SNMP 代理 (SNMP Agent) 三个部分。被管理的设备是网络中的一个节点, 有时被称为网络单元 (Network Elements), 被管理的设备可以是路由器、网管服务器、交换机、网桥、集线器等。每一个支持 SNMP 的网络设备中都运行着一个 SNMP 代理, 它负责随时收集和存储管理信息, 记录网络设备各种情况, 网络管理软件再通过 SNMP 通信协议查询或修改代理所记录的信息。

SNMP 代理是驻留在被管理设备上的网络管理软件模块, 它收集本地计算机的管理信息并将这些信息翻译成兼容 SNMP 协议的形式。

SNMP 管理器通过网络管理软件来进行管理工作。网络管理软件的主要功能之一, 就是

协助网络管理员完成管理整个网络的工作。网络管理软件要求 SNMP 代理定期收集重要的设备信息,收集到的信息将用于确定独立的网络设备、部分网络或整个网络运行的状态是否正常。SNMP 管理器定期查询 SNMP 代理收集到的有关设备运转状态、配置及性能等的信息。

SNMP 使用面向自陷的轮询方法(Trap-directed polling)进行网络设备管理。一般情况下,网络管理工作站通过轮询被管理设备中的代理进行信息收集,在控制台上用数字或图形的表示方式显示这些信息,提供对网络设备工作状态和网络通信量的分析和管理工作。当被管理设备出现异常状态时,管理代理通过 SNMP 自陷立即向网络管理工作站发送出错通知。当一个网络设备产生了一个自陷时,网络管理员可以使用网络管理工作站来查询该设备状态,以获得更多的信息。

3.1.3 CMIP

CMIP 协议是在 OSI 制订的网络管理框架中提出的网络管理协议。与其说它是一个网络管理协议,不如说它是一个网络管理体系。这个体系包含以下组成部分:一套用于描述协议的模型,一组用于描述被管对象的注册、标识和定义的管理信息结构,被管对象的详细说明以及用于远程管理的原语和服务。CMIP 与 SNMP 一样,也是由被管代理和管理者、管理协议与管理信息库组成。在 CMIP 中,被管代理和管理者没有明确的指定,任何一个网络设备既可以是被管代理,也可以是管理者。

CMIP 管理模型可以用三种模型进行描述:组织模型用于描述管理任务如何分配;功能模型描述了各种网络管理功能和它们之间的关系;信息模型提供了描述被管对象和相关管理信息的准则。从组织模型来说,所有 CMIP 的管理者和被管代理者存在于一个或多个域中,域是网络管理的基本单元。从功能模型来说,CMIP 主要实现失效管理、配置管理、性能管理、记帐管理和安全性管理。每种管理均由一个特殊管理功能领域(SMFA, SpecialManagementFunctionalArea)负责完成。从信息模型来说,CMIP 的 MIB 库是面向对象的数据存储结构,每一个功能领域以对象为 MIB 库的存储单元。

CMIP 是一个完全独立于下层平台的应用层协议,它的五个特殊管理功能领域由多个系统管理功能(SMF)加以支持。相对来说,CMIP 是一个相当复杂和详细的网络管理协议。它的设计宗旨与 SNMP 相同,但用于监视网络的协议数据报文要相对多一些。CMIP 共定义了 11 类 PDU。在 CMIP 中,变量以非常复杂和高级的对象形式出现,每一个变量包含变量属性、变量行为和通知。CMIP 中的变量体现了 CMIPMIB 库的特征,并且这种特征表现了 CMIP 的管理思想,即基于事件而不是基于轮询。每个代理独立完成一定的管理工作。

CMIP 的优点在于:

1) 它的每个变量不仅传递信息,而且还完成一定的网络管理任务。这是 CMIP 协议的最大特点,在 SNMP 中是不可能的。这样可减少管理者的负担并减少网络负载。

2) 完全安全性。它拥有验证、访问控制和安全日志等一整套安全管理方法。

但是,CMIP 的缺点也同样明显:

1) 它是一个大而全的协议,所以使用时,其资源占用量是 SNMP 的数十倍。它对硬件设备的要求比人们所能提供的要高得多。

2) 由于它在网络代理上要运行相当数量的进程,所以大大增加了网络代理的负担。

3) 它的 MIB 库过分复杂,难于实现。迄今为止,还没有任何一个符合 CMIP 的网络管理系统。

3.1.4 Q3

国际电信联盟 (ITU) 在 M. 3010 建议中指出,电信管理网的基本概念是提供一个有组织的网络结构,以取得各种类型的操作系统(OSs)之间,操作系统与电信设备之间的互连。它是采用商定的具有标准协议和信息的接口进行管理信息交换的体系结构。提出 TMN 体系结构的目的是支撑电信网和电信业务的规划,配置,安装,操作及组织。TMN 及电信网的总体关系如图 1-1 所示。图中的操作系统 (OS) 代表实现各种管理功能的处理系统,工作站代表实现人机界面的装置,数据通信网提供管理系统与被管理网元之间的数据通信能力。

在 TMN 中共有四种接口,即 Qx, Q3, X, F。目前的标准化主要集中在 Q3 接口上, Q3 接口与我们通常谈到的接口很不同,比如一个 RS232 接口等,都是比较单一的通信接口,而 Q3 接口是一个集合,而且是跨越了整个 OSI 七层模型的协议的集合。从第一层到第三层的 Q3 接口协议标准是 Q. 811,称之为低层协议栈。从第四层到第七层的 Q3 接口协议标准是 Q. 812,称之为高层协议。Q. 811/Q. 812 适用于任何一种 Q3 接口。Q. 812 中最上层的两个协议是 CMIP 与 FTAM,前者用于面向事物处理的管理应用,后者用于面向文件传输的文件传送,接入与管理。在这里还要特别指出, Q3 接口不仅包括在第七层中用到的管理信息和管理信息模型 (MIB),在通信协议 Q. 811/812 之上,还要有 G. 774 和 M. 3100。M. 3100 是面向网元的通用信息模型。G. 774 是 SDH 的管理信息模型。Q. 821, Q. 822 是 Q3 接口中关于告警和性能管理的支持对象定义。Q3 接口是一个集合。

3.2 CORBA 中间件技术的发展历程

1991 年 10 月,OMG 推出 CORBA 1.0。该版本的主要内容包括:CORBA 对象模型、界面定义语言 (IDL, Interface Definition Language)、用于动态请求管理和动态调用的 API 集

合和界面仓库。

1992年2月,OMG推出CORBA1.1,这是第一个广泛发行的版本。它解决了CORBA1.0中的许多二义性问题,澄清了对象模型和界面仓库中的一些模糊概念。最重要的是,它引入了对象适配器(Object Adapter)的概念,并提供了基本对象适配器(BOA, Basic Object Adapter)的界面。对象适配器是ORB与对象实现之间的连系纽带,它的功能主要包括:创建CORBA对象和对象应用;识别客户向CORBA对象发送的请求;将请求分发给服务方的对象实现;激活CORBA对象。

OMG于1993年12月推出的CORBA1.2没有新的技术突破,其主要贡献是解决了以前版本中的一些二义性问题。

1996年8月,OMG推出了CORBA2.0。该版本在解决互操作问题方面迈上了一个新台阶。其中一个突出的贡献是引入了GIOP/IIOP(General Inter-ORB Protocol/Internet Inter-ORB Protocol)协议,它能用于解决不同供应商的CORBA平台之间的互操作问题。有人不禁要问,既然CORBA的主要目标是解决互操作问题,OMG为什么不在CORBA的早期版本中提供ORB之间的通信协议规范?其原因如下:使用早期CORBA版本的应用,大多数规模较小,单个CORBA平台所提供的服务已经可以满足这些应用的需求;但随着应用规模的不断扩大,它们最终需要与建立在其它CORBA平台上的应用相互协同工作,于是CORBA之间的互操作问题便摆在了OMG的面前。除解决了互操作问题外,CORBA2.0还增加了以下内容:动态框架界面、对界面仓库的扩展、对多层次的安全和事务处理服务的支持、与OLE2/COM之间的协同工作、IDL与C++和Smalltalk之间的映射。

1997年8月,OMG推出CORBA2.1。这一版本增加了两个语言映射(Cobol和Ada),并修订了与互操作有关的内容,扩充了IDL类型。

1998年2月,OMG推出CORBA2.2。在该版本中,最引人注目的是可移植对象适配器(POA, Portable Object Adapter)。从CORBA的第一个版本到CORBA2.1,OMG只定义了BOA,它提供了用于创建和实现CORBA对象的基本服务。不幸的是,ORB供应商在实现BOA时遇到了OMG未曾注意到的问题,并在BOA的规范中发现了不少有二义性的内容,于是,各个供应商便使用自己的专用技术去独自解决这些问题。由此导致的严重的后果是,不同CORBA平台上的应用的可移植性极差。POA的目标就是实现应用的可移植性。CORBA2.2的另外两个新规范是,IDL/Java之间的映射和CORBA/DCOM(Distributed Common Object Model)之间的协同工作。

OMG于1998年夏季推出的CORBA2.3是一个改动内容较少的版本,它主要对以下内容进行了较小的修订:IDL/Java语言之间的映射;CORBA/DCOM之间的协作;ORB之间的互操作和它的安全性。

OMG 即将推出的 CORBA3.0 是一个非常重要的版本，它将实现 CORBA 与 Internet 的彻底集成。它引入的新技术主要包括：CORBA 构件模型 (CORBA Component)、CORBA 的消息服务 (CORBA Messaging) 和通过值传递对象 (Objects by Value)。以下将对它们进行详细的论述和分析。

3.3 CORBA 原理

CORBA (通用对象请求代理体系结构) 是在当今快速发展的软件与硬件资源的情况下发展出的一种新技术。它可以让分布的应用程序完成通信，无论这种应用程序是什么厂商生产的，只要符合 CORBA 标准就可以相互通信。(OMG, 2004)

为了创建一个遵从 CORBA 规范的应用程序，ORB 是 CORBA 必须提供的。没有 ORB，CORBA 应用程序就无法工作。CORBA ORB 最显著的功能，是对应用程序或是其他 ORB 的请求予以响应。实现细节对软件开发者的透明性，是 ORB 的一个杰出的特性。

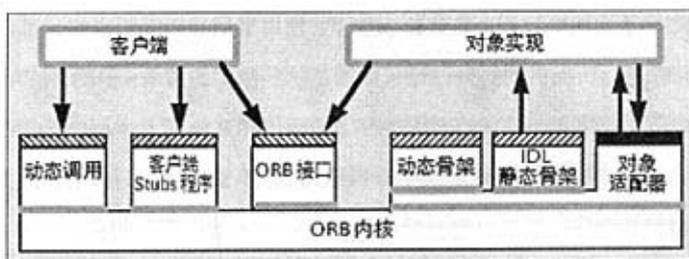


图 3-1 ORB 结构图

上图所示为一个独立的 ORB 的结构，箭头说明 ORB 的调用关系。为了提出一个请求，客户端可以使用动态调用接口 (Dynamic Invocation) 或者客户端的 Stub 程序。客户端也可以直接和 ORB 交互。

对象的实现 (Object Implementation) 通过 IDL 静态骨架 (IDL Static Skeleton) 或动态骨架 (Dynamic Skeleton) 的调用来接受请求。

下图 (如图 5.2) 所示为一个远程调用过程。为了调用远程的对象实例，客户端首先获取它的对象参数。客户端在进行远程请求时使用与本地请求相同的代码，只不过用对象参数代替了远程实例。当 ORB 检测对象参数并发现目标是远程对象的时候，它会调度这些参数，把请求转移到网络上，从而连接到远程对象的 ORB 上。ORB 可以从对象参数中判断出目标对象是远程的，客户端却做不到这一点。当客户端发出请求时，对象参数不用去分辨目标对象的

位置。这保证了对象位置的透明性——CORBA 的这些基本原理简化了分布式对象的计算机应用的设计。

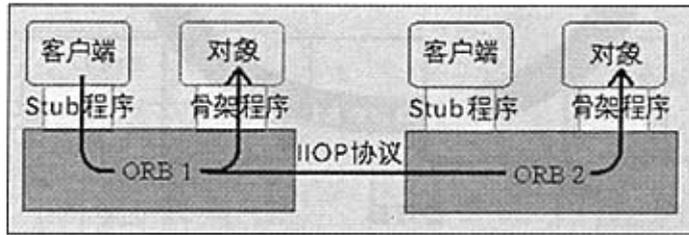


图 3-2 ORB 工作原理图

ORB 是一个在对象间建立客户/服务器联系的中间件。使用 ORB，客户可以调用服务器的对象或对象中的应用，被调用的对象不要求在同一台机器上。由 ORB 负责进行通信，同时 ORB 也负责寻找适于完成这一工作的对象，并在服务器对象完成后返回结果。客户对象完全可以不关心服务器对象的位置，实现它所采用的具体技术和工作的硬件平台，甚至不必关心服务器对象的与服务无关的接口信息，这就大大简化了客户程序的工作。既然能够这么方便，那 ORB 就需要提供在不同机器间应用程序间的通信，数据转换，并提供多对象系统的无缝连接。

通常编制客户/服务器程序时，常常需要自己定义通信协议，而协议的制定往往与硬件和实现的方法有关，而 ORB 能够简化这一过程。在 ORB 下，协议通过 IDL 语言进行定义，保证了一致性，为了照顾到灵活性，ORB 允许程序员选择相应的操作系统，执行环境和编程语言。更重要的是它可以使原来的代码通过一定的方式重用。CORBA 是面向对象标准的第一步，有了这个标准，软件的实现与工作环境对用户和开发者不再重要，可以把精力更多地放在本地系统的实现与优化上。CORBA 被设计用来对不同对象系统进行集成，提供灵活的的对象调用与功能实现。

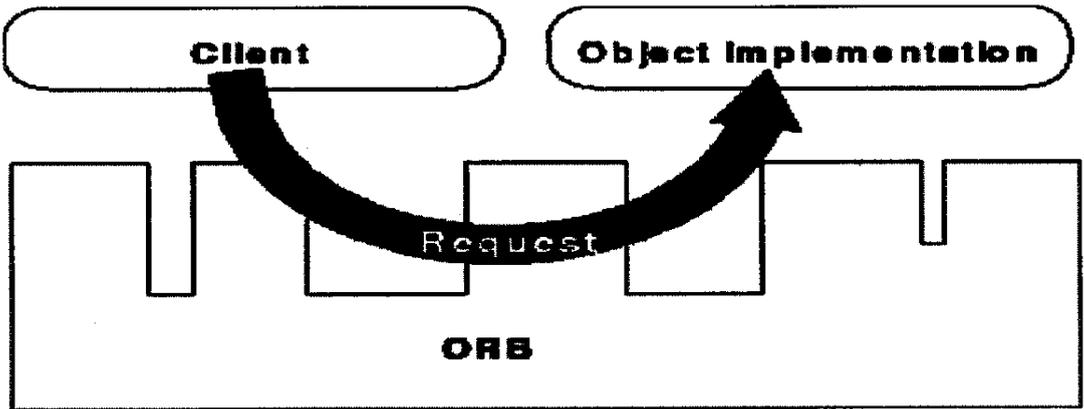


图 3-3 ORB 调用服务器对象

对象请求代理结构的大体工作过程就象上面的工作过程一样。客户将需要完成的工作交给 ORB，由 ORB 决定由哪一个对象实例完成这个请求，然后激活这个对象，将完成请求所需要的参数传送给这个激活的对象。除了客户传送参数的接口外，客户不需要了解其它任何信息，这就大大节省了用户的开发精力。

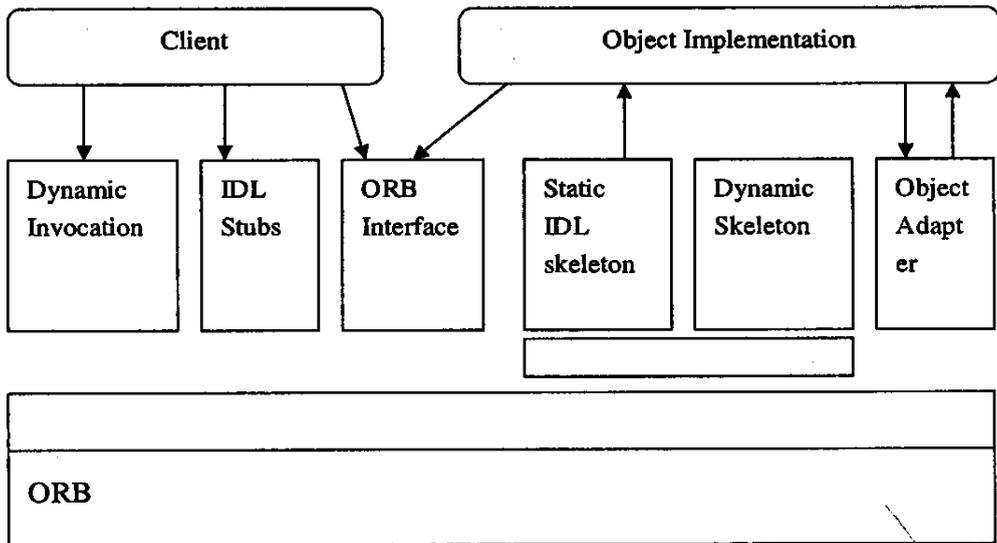


图 3-4 ORB 的接口结构

在提出请求时，客户可以使用动态调用接口或者 OMG IDL 句柄。当然用户也可以直接调用一些 ORB 内部的功能。对象实现通过 OMG IDL 产生的框架或通过动态框架接收到调用请求，在处理这些请求时，对象实现可以调用对象适配器和 ORB。对象的接口有两种定义方式，可以使用接口定义语言（称为 OMG 接口定义语言，OMG IDL）进行静态定义，这种语言根据进行的操作和传送的参数定义对象。另一种方法，可以将接口加入接口库服务中，这种服务代

表作为对象的接口的组件，允许在运行时对这些成为组件的接口进行访问，这两种方法是等效的。

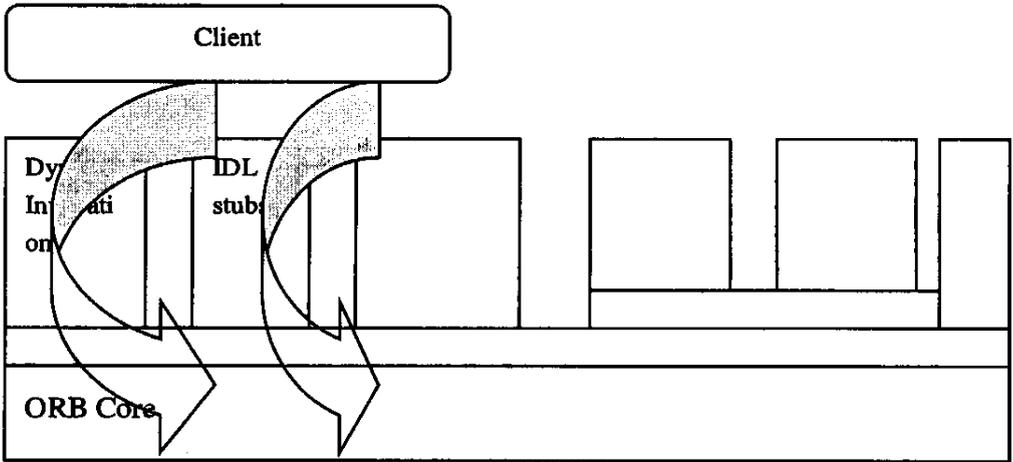
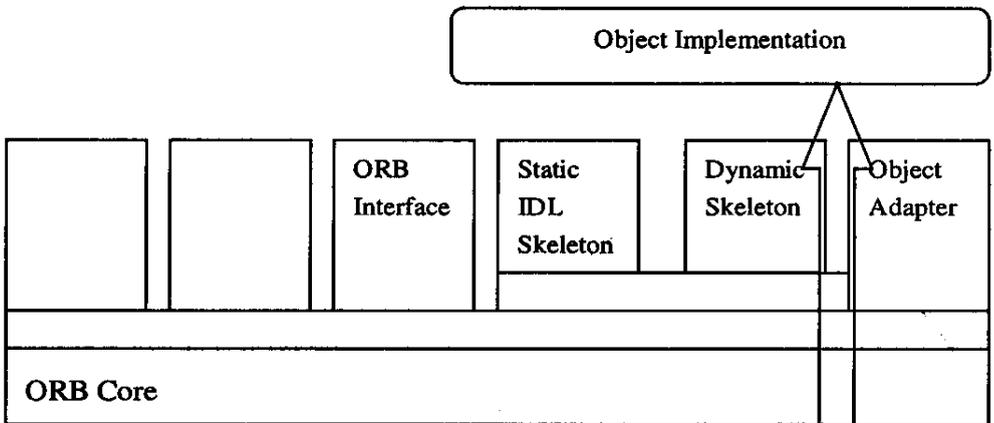


图 3-5 客户使用句柄或动态调用接口进行访问

客户知道对象的类型和希望进行的操作，客户可以通过访问一个对象的对象参考提出请求。客户可以通过调用句柄函数初始化调用，也可以动态提出请求。动态发出的请求和通过句柄接口发出的静态请求两者在格式是一样的，请求的接收者不可能知道这种请求是动态发出的还是静态发出的。



如图 3-6 对象实现接收请求的示意图

ORB 定位合适的可以实现这个功能的代码，通过 IDL 框架或动态框架传向对象实现传送参数，并将控制权交给对象实现。框架是指定于接和对象适配器的，在实现请求的过程中，对象实现可以通过对象适配器获取一些 ORB 服务。在完成请求时，将控制权和输出数据返回

给客户。对象实现可以根据自己的需要选择需要的对象适配器使用。

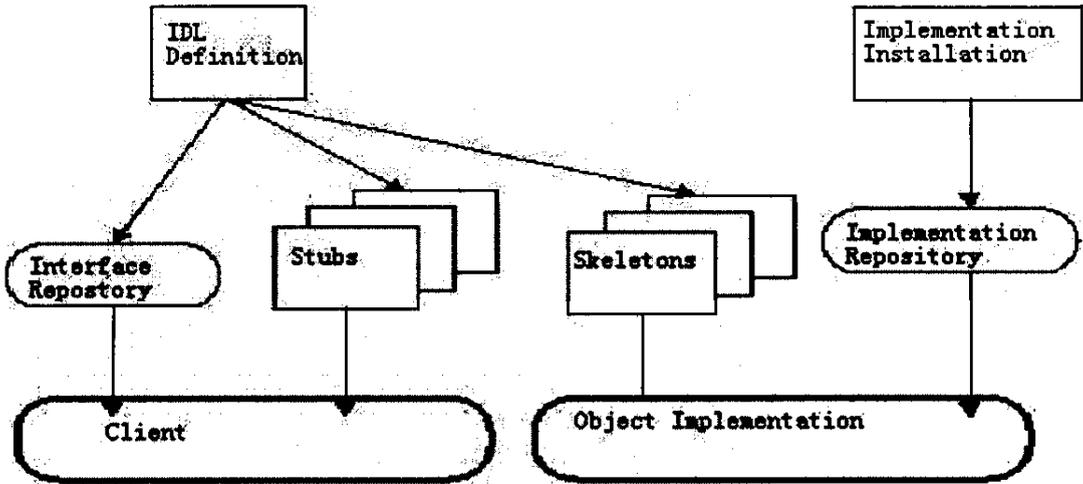


图 3-7 接口和实现库的结构示意图

上图显示了接口和实现住处如何对客户和对象实现是可用的，接口可以在 OMG IDL 或在接口库中实现，这种对接口的定义用于产生客户句柄和对象实现框架。对象实现信息在安装时提供，保存于实现库中，在传送请求时可以使用这个信息库中的内容。

对象请求代理这个结构在上图中并不需要作为组件单独实现，它由接口定义。任何提供正确接口的 ORB 实现都是可被接受的。接口可分为以下几大类：

1. 对于所有 ORB 实现均相同的接口；
2. 指定于特定对象类型的操作；
3. 指定于对象实现的特定形式的操作；

不同的 ORB 可以采用不同的实现策略，加上 IDL 编译器，库和不同的对象适配器，这一切提供了一系列对客户的服务和对具有不同属性对象的实现。可以存在多个 ORB 实现，它们有不同的名称和不同的实现方法与调用方法，对于客户而言，客户可以同时访问由不同 ORB 实现管理的对象，当几个 ORB 共同工作时，它们必须能够区别它们各自的对象名（也就是对象参考），客户不管区别只管使用。ORB 内核是 ORB 的一部分，它提供了对象的基本命名和请求通信机制。CORBA 设计得可以支持不同的对象机制，它是通过在 ORB 内核上建立 ORB 来完成这一点的。

一个对象的客户可以访问此对象参考，并对对象进行操作。客户不清楚对象的内部结构，它只知道对象的接口和执行操作所需要的时间和空间等资源。虽然我们可以把客户想象为一个调用对象的进程，但是我们也不要忘记了对象也可以调用另外对象的服务。客户看到的

ORB 接口和人观念中的接口有差不多，这就为编程提供了帮助。客户不需要对代码进行改变就可以通过 ORB 实现功能，对象适配器只能由 ORB 或对象实现调用。

对象实现提供了对象的表现形式。通常实现由另一对象提供或由相应的软件提供，当然也可以自己编程实现。在某些情况下，对象的主要功能是非对象实体产生作用。在 CORBA 中可以支持对象的不同实现。通常，对象实现不依赖于 ORB 或客户请求，对象实现可以通过选择对象适配器选择和 ORB 相关服务来选择接口。

对象参考是需要由 ORB 内指定的信息，客户和对象实现相应于语言映射有对象参考的一个透明定义，这样就实现的表示与参考隔离开了。两个 ORB 实现可能在选择对象参考表示时是不同的。所有的 ORB 必须提供相对于对象参考一致的语言映射，这使得程序能够独立于 ORB 对对象参考进行访问。

OMG 接口定义语言 (OMG IDL) 通过对象的接口定义了对象的类型。一个接口由一些命名的操作和与这些操作相关的参数组成。请注意，虽然 IDL 提供概念框架用于描述对象，但不需要有 IDL 源代码供 ORB 工作。只要相同的信息以句柄函数或运行接口库的形式提供，特定的 ORB 就可以正常工作。IDL 是一种方法，它使对象实现能够告诉潜在的客户，什么样的操作可以执行。从 IDL 的定义上可以将 CORBA 对象映射为特定的编程语言或对象系统。

不同的面向对象语言和非面向对象语言可以以不同的方式访问 CORBA 对象。对于面向对象语言而言，它希望看到的是对象的形式，即使对非面向对象语言来说，它所希望看到的也不包括具体的内部实现。将 OMG IDL 映射为编程语言的方法对于所有的 ORB 实现应该是一致的。这些映射可能包括数据类型的映射和调用 ORB 的过程（或函数）接口的映射。语言映射还定义了对象调用和客户（或实现）中的控制线程之间的相互作用。最普通的映射提供了同步调用，结果可以在过程完成时返回。其它的映射可以用来初始化调用并将控制权返回给程序，在这些情况下，附加的函数必须相应的同步功能。

为了映射非面向对象语言，将有一个对每个接口类型的程序接口。通常，句柄将提供访问 OMG IDL 定义的操作的机制。句柄调用对于 ORB 核心是私有的那部分 ORB。如果有多于一个 ORB，将会有对应于不同 ORB 的接口。在这种情况下，需要 ORB 和语言映射相互协调以访问正确的对象参考句柄。面向对象语言不需要句柄接口。接口允许对象动态调用，用户不必调用一个特定对象上的操作，他可以指定调用特定的对象。客户程序提供关于操作和参数类型的信息就可以了。

允许动态处理对象调用的接口是非常有用的，不是由与特殊操作相关的框架来访问对象实现，而是由一个提供访问操作名和参数的界面用一种类似于动态调用接口的方式来访问对象实现。动态框架界面可以由客户句柄或动态调用接口来调用，它们向动态框架接口发出对

象请求。动态框架接口的基本思想是让所有的对象请求通过调用同一组例程来达到调用对象实现中方法的目的，这组例程便叫做动态调用例程 DIR。

3.3.1 OMG（对象管理组）接口定义语言（IDL）

OMG IDL 是 CORBA 的基本抽象机理，它从实现中分离对象接口，OMG IDL 在客户机和服务器之间建立一个契约，用它来描述在应用程序中需要用到的类型和对象接口。这些描述在应用程序中需要用到的类型和对象接口。这些描述与实现的语言无关，所以不用考虑客户端程序的编程语言是否与服务器程序的编程语言一致。（OMG，2004）

IDL 定义由一个 IDL 编译器编译一个具体的实现的语言。编译器将这些与语言无关的定义翻译成特定语言的类型定义和 API，开发者使用这些类型的 API 来提供应用程序的功能和与 ORB 的交互。各种实现语言的转换算法是由 CORBA 来确定的，目前 CORBA 定义了 C、C++、Smalltalk、COBOL、Ada 和 Java 的语言映射，目前还在考虑提供更多的语言映射，其中的一些可能将来成为标准。

因为 IDL 只描述接口，不描述实现，它是一个纯说明性的语言。

IDL 定义把焦点集中在对象接口，其他接口所支持的操作和操作时可能引发的异常上，实际上，它只有很少一部分涉及到机器的支持，IDL 的大部分内容只涉及到数据类型的定义。这是因为只有当数据类型用 IDL 定义时，这些数据才能在客户程序和服务器程序之间交换。

一个 IDL 编译器生成源文件，源文件必须与应用程序代码一起生成客户机和服务器的可执行文件。CORBA 并没有标准化的开发环境，这就是说各家厂商提供的 IDL 编译器和 ORB，以及生成的源文件和数目是不一样的，但是 CORBA 的概念对于所有的 ORB 和实现的语言都是一样的。

开发过程的结果是生成客户机和服务器的可执行程序，这些可执行程序可以在任何地方被调度，无论他们是用同一个 ORB 开发的，还是不同 ORB 开发的，也无论它们用同一种语言还是不同语言实现的。

IDL 规范是由编译器翻译成特定语言的存根和框架。存根和框架提供客户端和服务器端 API 用来支持具体语言的实现。

IDL 的词法规则，除了标识符略有区别外，其他的与 C++ 没有区别。

IDL 提供一系列内置的类型，可以很容易地将它们翻译成多数编程语言，内置类型集可以由用户定义类型来扩展，比如结构和序列。IDL 通过接口继承提供对象方位，而接口继承创立了类型的兼容性和多态性。

具体的 IDL 语法可以查阅相关手册。

3.3.2 CORBA 的可移植对象适配器 POA (Portable Object Adapter) 规范

在 CORBA 中, 对象适配器(Object Adapter)可作为伺服程序和 ORB 之间的纽带, 一个对象适配器是一个对象, 它将一个对象接口配置给调用程序所需要的不同借口, 也就是说它是一个代理, 它可以让调用程序在不知道对象实际接口情况下调用一个对象的请求。(O. Otte, P. Patrick, M. Roy, 2004)

CORBA 规范中要求系统实现时必须有一种对象适配器。对象适配器完成如下功能:

- 1) 生成并解释对象的引用, 把客户端的对象引用映射到服务对象的功能中;
- 2) 激活或撤消对象的实现;
- 3) 注册服务功能的实现;
- 4) 确保对象引用的安全性;
- 5) 完成对服务对象方法的调用。

但是在 CORBA2.0 及以前的版本 CORBA 中基本对象适配器 (BOA) 存在一些局限。

1) 可移植问题

BOA 的定义过于宽泛, 不同厂商的 ORB 实现对 BOA 的解释也不尽相同, 由此导致服务器端的对象实现不能在不同的 ORB 产品进行移植。

2) 对象标示的永久性

在 BOA 中, 只要简单的调用 `impl_is_ready()` 或 `object_is_ready()` 就可以使服务器端的对象为客户端所用, 过程固然简单, 但无法控制对象引用的生成, 也没有给一个永久的标识, 这给实现跨服务器生命周期的对象 (永久对象) 带来了困难。

3) 对象实现粒度

BOA 缺乏对服务器端对象实现粒度的支持。所谓服务器端的对象实现粒度, 就是指客户端看到的一批服务器对象, 其中每个对象都用单独的实现对象来实现, 还是只用一个实现对象来实现。这实际上是指客户端看到的 CORBA 对象与实现对象之间的映射关系, 这种关系可以是 1 对 1, 也可以是 1 对多的。BOA 中没有区分客户端看到的对象和实际的实现对象, 因此也无法体现这种映射关系, 实现力度难以控制。

这一事实决定了 BOA 只可能向客户提供数来能够有限的服务器对象。随着服务器端对象数量的增多, 它的实现对象也增多, 导致服务器端系统资源 (主要是内存) 更大的消耗。

3.2.2.1 POA（可移植对象适配器）支持的抽象模型

POA（可移植对象适配器）支持的抽象模型是 OMA 对象模型的特殊形式。因此，大多数的 CORBA 对象模型中的元素都会在 POA 抽象模型中出现，只是定义的更为详细。根据 CORBA2.3.1 规范的描述，抽象模型包括如下的组件：

- 1) 客户：客户是一个计算环境，通过它的对象引用对指定对象中的操作发出请求。
- 2) 服务器：服务器是一个包含对象实现的计算环境。通常一个服务器对应一个进程。
- 3) 对象：在 POA 抽象模型中，他表示抽象意义上 CORBA 的对象，即一个具有标识、接口和实现的程序体。
- 4) 对象标识：在 POA 抽象模型中，他表示抽象意义上的 CORBA 对象的一个值。从客户的角度看，对象的标识封装在对象引用中。从服务器角度看，对象标识是通过 POA 接口有对象实现直接管理的。
- 5) 对象引用：他在 CORBA 对象模型中的含义相同，具体的封装了一个 ObjectId 和一个 POA 标识。在某些特定的 ORB 实现中，对象引用还可能包括更多的信息，例如 POA 的全名。ORB 用这些信息来识别并寻找服务器和与对象相关的 POA。
- 6) 仆从：仆从实现一个或多个对象请求的编程语言对象或实体，通常存在服务器进程的环境中。ORB 将客户端根据对象引用发出的操作请求转换为对仆从的调用。一个对象可能与多个仆从相联系。
- 7) POA：它是在服务器环境中的一个可识别的尸体。每个 POA 都为 ObjectId 以及其他嵌套的或子 POA 提供一个名字空间。
- 8) POA 管理器：他是封装了一个或多个 POA 处理的对象。应用开发人员可以通过 POA 管理器的操作对 poa 的请求进行排队或者丢弃，也可以终止一个 POA。
- 9) 仆从管理器：ORB 利用仆从管理器的操作来激活和种植一个仆从。仆从管理器用来管理已 ObjectId 来表征的对象与特定仆从之间的联系，并负责确定一个对象是否存在。有 2 种类型的仆从管理器：仆从激活器和仆从定位器，可以根据 POA 中的策略来选去使用的类型。
- 10) 适配器激活器：当 ORB 收到一个对当前不存在的子 POA 的请求时，调用适配器激活器重的操作，创建所需要的 POA。

从 CORBA2.2 开始，CORBA 规范中用 POA 取代了 BOA。下图给出了 CORBA2.3.1 规范中的 POA 抽象模型。

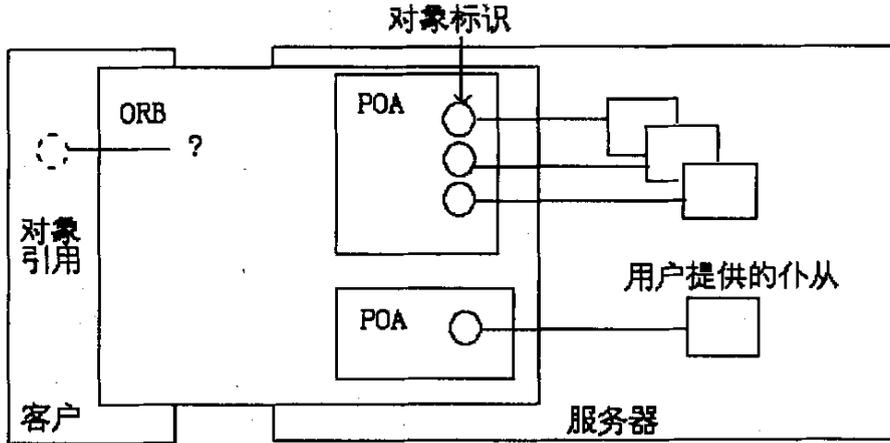


图 3-8 POA 抽象模型

3.3.2.2 POA 的体系结构

每一个 POA 都可以创建新的 POA，前者是父 POA，或者是子 POA。POA 之间通过这种船舰关系次女过程一个 POA 体系结构，如图所示。体系结构的根是根 POA (root POA)。

POA 要处理的是对象标识 ObjectId (图中用圆圈表示) 和关联的活跃仆从 (图中用方框表示)。所谓活跃仆从是指已经在内存中，并通过相关的对象表示提供给 POA 的程序对象。如果没有活跃对象映射，它将对象标识和活跃仆从关联起来，每个关联构成一个活跃对象。如果没有活跃对象以 ingshe，或者受到的请求中的目标对象没有出现在活跃对象映射表中，POA 就会通过缺省仆从能够来执行请求 (见图中的 POA A)，也可以调用仆从管理器来获得一个仆从 (见图中 POA B)，然后由这个仆从执行请求。

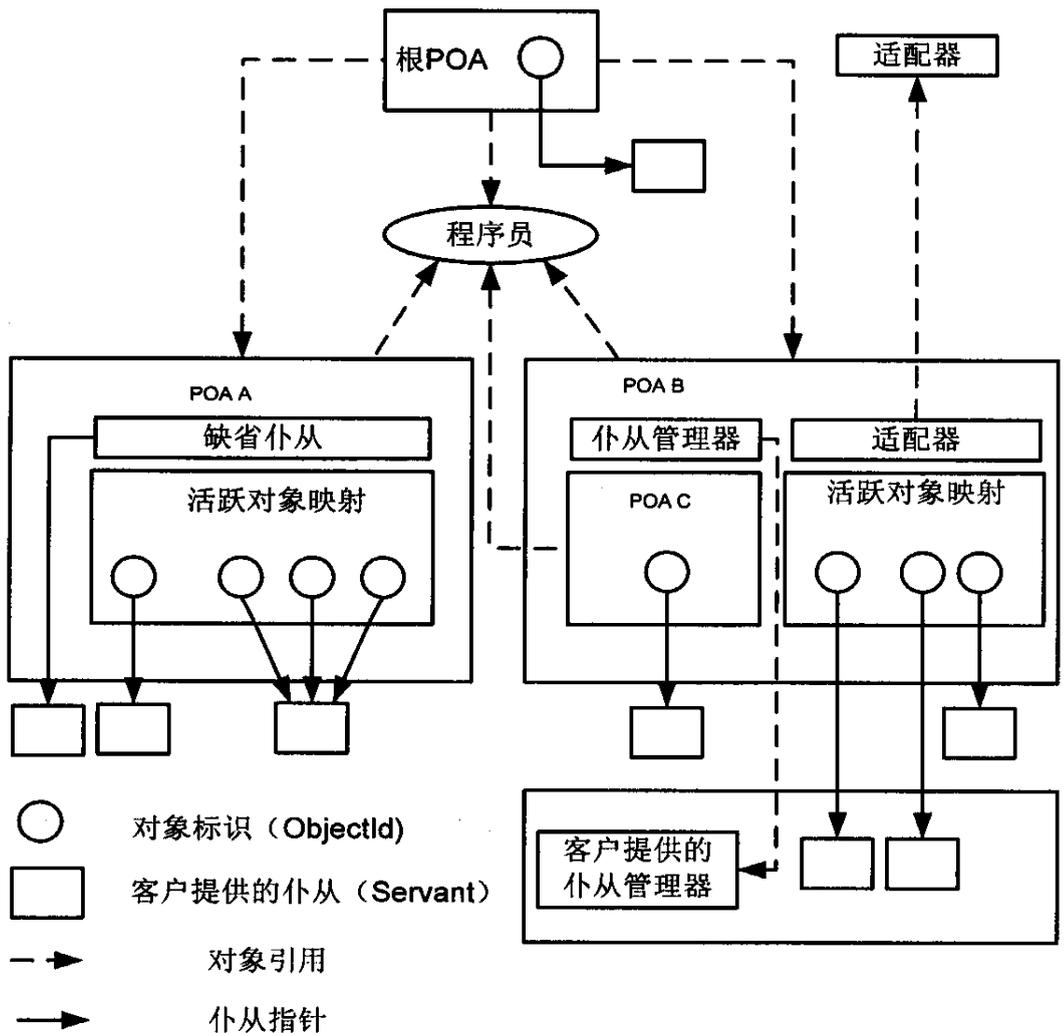


图 3-9 POA 的体系结构

3.3.3 CORBA 机理，GIOP 和 IIOP 协议简介

尽管 CORBA 想要尽量的屏蔽开应用程序和网络细节，但是了解 CORBA ORB 内部是怎样实现网络信息传送的对程序的调试和编写还是非常有好处的。

首先解释两个名词：

GIOP: General Inter-ORB Protocol

IIOP: Internet Inter ORB Protocol

CORBA 规范将 GIOP 定义为它的基本的互用性框架。GIOP 并不是一个可以直接用于 ORB 之间进行通信的具体协议，它描述了特定的协议如何创建以适用于 GIOP 框架。IIOP 是 GIOP

的一种具体实现。(Michi Henning, 2000)

GIOP 对传递消息的底层传输作了一些假设:

1. 传输是面向连接的
2. 连接是全双工的
3. 连接是对称的
4. 传输是可靠的
5. 传输提供了一个字节流抽象
6. 传输混乱表示失去连接

以上的假设完全与 TCP/IP 提供的保证相匹配, 同时其他的一些传输协议也满足这些要求, 比如 SNA, ATM 等。

GIOP 定义了一个公共数据类型 (CDR), 为了传输, 还确定了 IDL 类型的二进制格式。CDR 的主要特点是:

1. CDR 同时支持长字节和短字节
2. CDR 按照自然边界对齐原始的类型
3. CDR 编码的数据是非自识别的

CDR 的编码规则适用于所有可能的 IDL 类型, 比如联合, 序列, 数据, 异常等等, 所有的 IDL 类型都有严格定义的编码方式, 并且它确保了 ORB 之间的互用性。CDR 编码的数据不是自描述的, 并且始发方和接受方强制遵守 IDL 定义的接口规范。

GIOP 具有八种消息类型, 而 Request 和 Reply 是主要的, 因为它们实现了基本的 RPC 机制。下表列出了消息类型和始发方。

消息类型	始发方
Request	客户机
Reply	服务器
CancelRequest	客户机
LocateRequest	客户机
LocateReply	服务器
CloseConnection	服务器
MessageError	客户机或服务器
Fragment	客户机或服务器

表 3-1 消息类型和始发方

为了通过线路传输一个 GIOP 消息，发送端发送一个消息头，接着是一个消息体（消息体的内容与消息头所表示的准确消息有关）

GIOP 消息头共有 12 个字节，如下表：

G	I	O	P	1	1	0	0	4-Byte Message Size
---	---	---	---	---	---	---	---	---------------------

表 3-2 GIOP 消息头

消息头的前四个字节总是字符 GIOP，然后下面两位是版本号，再下一位是标志位，第 7 个（从 0 开始）表示消息类型，最后四位是无符号值，表示消息大小。

每种具体的消息由 GIOP 消息头加上具体消息头和具体消息体组成。

例如 Request 消息格式如下表：其他的消息具体格式可以参考 CORBA 规范。

12-byte GIOP head	Variable-length GIOP Request Head	Variable-length GIOP Request Body
-------------------	--------------------------------------	--------------------------------------

表 3-3 Request 消息格式

CORBA 客户机和服务器程序所见到的交互模型是无连接的，但是 GIOP 请求在一个面向连接的传输上被调用。

GIOP 指定了客户机与服务器之间进行通信所必需的大部分协议细节，GIOP 与具体的传输无关，它是一个抽象的协议，而 IIOP 明确为 TCP/IP，它是 GIOP 的一个具体实现（或映射）。IIOP 添加了 ORB 通过 TCP/IP 进行互用所需要的特定的信息。

为了将 GIOP 转化为一个具体的协议，IIOP 只要指定 IOR 的编码。IOR 包含一个对象的接口类型和一个或更多的协议配置文件，每个配置文件包含客户机使用一个特定协议发送一个请求所需要的信息。

第四章 国际级电信通讯协议 TMF

4.1 TMF 组织

电信管理论坛 (TMF) 是一个为电信行业开发多种管理和 OSS 标准的全球标准化组织。TMF (Tele Management Forum) 组织成立于 1988 年。TMF (电信管理论坛) 也就是过去的 NMF, 是一个为电信运营和管理提供策略建议和实施方案的世界性的非赢利性组织, 是专注于通信行业 Operations Support Systems (OSS) 和管理问题的全球性的非赢利性社团联盟。TM Forum 领先的信息资源、知识和技术方案被业界广泛认同; 它还为会员提供了一个协同工作的环境, 在这个环境里成员可以探讨电信服务提供商最重要的业务和技术需求。

目前该论坛的会员来自全球 36 个国家的包括制造商、软件商、专业 OSS 解决方案提供商, 以及 OSS 领域的战略咨询公司等 384 家公司会员, 包括亚信、中国移动和中国电信等中国会员, 其中 90% 会员是世界领先的电信服务提供商, 其中一些活跃的成员正在积极帮助驱动和把握行业的发展方向。它不仅以领先的信息资源、知识和技术方案被业界广泛认同, 同时还为会员提供了一个协同工作的环境, 在这个环境里成员可以探讨电信服务提供商最重要的业务和技术需求。目前这些成员已经就 OSS/BSS 的发展方向建起了公开的、全行业范围的战略性设想。其对电信行业的影响和作用可以从下面三个标准略见一斑:

- 1、 电信运营图 (TOM): 电信业务过程自动化事实上的行业标准;
- 2、 SLA (Service Level Agreement) 手册: 支持双方以对基本问题的实际观点来开发的 SLA ;
- 3、 多技术网络管理协议: 提供了一个独立的公共解决方案, 使得多种网络元素的管理成为可能。

TMF 每年举办两次全球性的会议, TMF 提出的 NGOSS 功能模型, 包括 TOM (电信运营规划) 和 eTOM, 被国际电信运营商和设备制造商以及电信运营支撑系统开发商广泛接受, 成为事实上的国际标准。

4.2 国际电信通用协议 TMF

TMF MTNM (也就是 TMF 513) 是一个电信网络集成管理的 TMF 标准, 是一个针对多技术、多供应商的网络管理的标准, 它为简化端到端的管理和多供应商网络的预配提供了新的契机。对于电信设备提供商来说, 提供一个兼容 TMF MTNM 的 EMS 应用程序来支持每个网络元

素越来越必要，因为这样才能更加容易地把网络元素集成到运营网络中去。该规格的 2.5 版本将在下列 TMF 文档中予以描述。

- TMF.513, BA (Business Agreement) ,
- TMF.608, IA (Information Agreement) ,
- TMF.814, IDL (Interface Definition Language) 解决方案集合
- TMF.814A, Implementation Agreement.

目前 TMF 协议已经成为国际电信的通用规范。TMF 标准支持在多供应商和多重技术环境中的互通性。在管理底层机构中，基于不同标准和协议的 NE 和 EMS/NMS 系统需要在它们各自系统之中以及北行 OSS 系统中相互作用。一些大型的服务提供商如 AT&T, Worldcom 和设备供应商如朗讯和富士通都已经成功地使用了 TMF 标准来促进轻松地互用和操作。

TMF513 按照对象和操作定义了接口的需求，这些接口支持标准接口的所有行为。TMF513 包含了到 TMF806 的映射，这个映射链接了在 UML 信息模型中定制的类和方法。TMF608 描述了类之间的关系和操作行为的细节，从而支持 NML-EML 接口需求，并且强化了 TMF513 和 TMF608 完全定义 NML-EML 接口。

TMF814 详细说明了 NML-EML 接口基于 CORBA IDL 的规范行为。并且为了详细定义与 TMF513 和 TMF608 中规定的行为规范共同存在时的行为，对于诸如效率和兼容性的之类的问题进行了详细的描述。下面将详细介绍 TMF814 协议的方案设置。

4.2.1 TMF814 接口目标

TMF814 接口提供了 NMS 与 EMS 之间的接口操作，这里的 NMS 是指客户段，EMS 是指接口的服务器端。TMF814 包含的接口允许 NMS 有如下的操作：

- 1、运用接口去管理几种技术之间的组成方式。例如 SDH、SONET、WDM、ATM 等技术。
- 2、发现那些不止是 EMS 启动，同样还包括 EMS 正常运作时管理的资源。
- 3、配置程序终止点。
- 4、确定资源用法和 EMS 管理的网元内部的子网连接。
- 5、请求 EMS 建立子网连接和移除子网连接。
- 6、寻找当前网络中的物理资源。
- 7、配置和重新获得网络的表现方法。

4.2.2 对象模块

网络中的对象具体展现了接口分层的概念和层次可分解性，这些性质都在 ITU G . 805 有定义。对象模块正是基于这种由 EMS 在使用中呈现的概念。需要重点标明的是 EMS 需要的不是每个具体模块的特殊性而是以 map 的形式详细描述内部数据到可以重现对象特性的接口结构。

具体详细准确的对象模块描述不属于本部分的内容，在别处有更好的正面的详细介绍。下面简单扼要的列出了本模块包含的对象：

1、TerminationPoint_T:

- CTP (连接终止点): 是 ITU G . 805/G . 803 连接点 (CP) 和 ITU G . 805/G . 803 终止点 (TCP) 的集合。TCP 不是这个规范的特殊模块。
- PTP (物理终点): 是部分 ITU G . 805/G . 803 CPs 和 ITU G . 805/G . 803 个别的物理端口 TCPs 的集合。
- TPPool (终点池): 以管理目标分组的 CTP 逻辑终点池。

2、MultiLayerSubnetwork_T: 重现 G . 805 单层子网的集合。

3、EMS_T: 区分 EMS 的标识。

4、ManagedElement_T: 重现一个单网元素。

5、SubnetworkConnection_T: 重现一个单独的交叉链接或一些交叉链接。

6、Equipment_T: 重现 EMS 管理的物理资源。

7、EquipmentHolder_T: 重现一个可以控制其它物理元素的网元的物理资源。

8、ProtectionGroup_T: 用于发现和请求保护信息。

9、TopologicalLink_T: EMS 管理的网络拓扑图。

10、TrafficDescriptor_T: 定义带宽和 TP 的服务质量特征。

4.2.3 对象管理器

“谷粒”实现要求通过适当的对象管理器实现的端口到达对象模块。

作为一个推论，一部分对象管理器被定义为拥有取得数据的服务，这些数据是关于由 EMS 管理的网元对象的。拥有上面那些服务的网元管理器有：

1、EMSMgr_I

2、MultiLayerSubnetworkMgr_I

3、ManagedElementMgr_I

- 4、EquipmentInventoryMgr_I
- 5、GuiCutThroughMgr_I
- 6、PerformanceManagementMgr_I
- 7、ProtectionMgr_I
- 8、TrafficDescriptorMgr_I
- 9、MaintenanceOperationsMgr_I

另还有一个 Common_I 接口，它继承了上面提到的接口。

4.2.4 CORBA 服务的使用

CORBA 服务计划同下列下列规范共同使用：

- COS: NamingService
- COS: NotificationService

4.3 TMF 与 CORBA

事实上，TMF 协议包含的第三部分即 TMF814 协议就是专门针对 CORBA 协议的。TMF814 协议是完全实现了对 CORBA 协议的支持。TMF81 协议的解决方案设置分为已经打包好的几个部分，包含 IDL 文件、Html 文件、支持文档。刚才在上面对 TMF 协议的介绍中已经介绍了关于 CORBA 的相关内容，如 4.2.4 节。

4.4 TMF 协议在综合告警系统中的应用

综合网管系统适应国内电信运营商网络多元化的设备现状，支持多种协议，提供丰富的数据接口，能连接并管理包括 ASCII/TL1、CORBA、SNMP、CMIP/Q3 等主要类型的设备。综合告警系统完全实现了对于接口类型为 CORBA 类型的网元设备的接入。当前在各大公司的 CORBA 接口中，大部分都是使用的 CORBA2.3 以上的版本，而 CORBA2.3 以上的规范中均完全兼容并实现了对 TMF814 协议。

第五章 CORBA 在综合告警系统中的应用

CORBA 作为一种总线结构能较好的解决电信应用领域中设备的多样性，平台的异构性。CORBA 可以提供统一的北向和南向接口，为电信运营商综合集成管理不同厂家的设备和网管提供了渠道，无论服务器端是什么操作系统客户端可以采取在同一个操作系统上实现所有服务器的连通。

5.1 基于 CORBA 应用程序的开发过程

开发基于 CORBA 的程序，首先要选择某种中间件产品做为基础来开发，在实际开发的过程中，我们选择 IONA 公司的 Orbacus, Orbacus 是支持 CORBA2.5 的 ORB, 它支持 C++和 JAVA, 并且提供实现源代码，速度较快，且有很好的灵活性，而且 ORBACUS 很小，非常适合嵌入式开发。（IONA，2003）

首先需要用 IDL 语言写出描述接口的 idl 文件，并使用 Orbacus 提供的 IDL 编译器，IDL 编译器将会生成四个文件，两个存根文件（其中包括头文件），两个框架文件（其中包括头文件）。

存根文件，包含在 IDL 中定义的相应类型，它将要被包括在客户机和服务器源代码中，以确保客户机和服务器使用同样的类型和接口，存根文件还提供客户程序发送消息给远端对象所需要的 API，由客户端程序开发者所编写的客户端文件将包含应用程序逻辑，存根和客户机代码被编译和链接为客户端可执行文件。

框架文件中所包含的源代码用来提供一个从这个 ORB 到由开发人员所编写的服务器源码的上端调用接口，并且提供在 ORB 的网络层和应用程序之间的连接。另外服务器端需要编写实现文件，实现文件包含服务器端的应用程序逻辑（对象实现也成为伺服程序）。框架和存根源代码已经实现的源代码被编译，链接为服务器可执行文件。

最后客户端和服务器还必须与 ORB 的库连接，这个库提供必要的运行时支持。并没有限定客户端和服务器只能有一个实现，例如，可以建立多个服务器程序，每一个实现相同的接口，但是实现不同的实现，这种多个服务器实现可以共同存在同一个系统中。

客户机和服务器程序共享一个开发环境时的开发过程图示（如下图）：

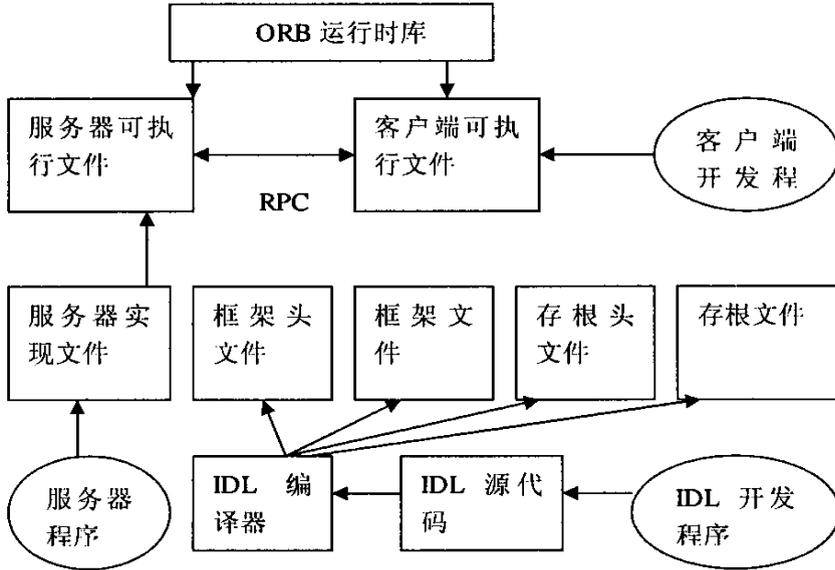


图 5-1 开发过程图

5.2 基于 CORBA 技术的数据采集机制

在这里，我选择 java 作为我的开发语言，这主要是因为 java 的跨平台性。简单地说 Java 的跨平台性就是指，编译后的 Java 程序可直接在不同的平台上运行而不用重新编译。这样我就完全可以简单的在 windows 系统下编程，测试，成功后再移植到 linux 系统下，而不需要重新编译。

这里的 CORBA 产品主要选用 jdk1.4 自带的 ORB，这样的方便之处在于不需要另外再安装其他的 ORB 产品，减少了工作量。同时，为了适应各 ORB 之间的连接区别，也使用了 JacORB。

5.2.1 告警信息采集

通过 connect 方法连接对方服务器：

```
public synchronized boolean connect (
    String args[], String ems, String user, String pwd)
```

```

{
    boolean coreStarted;
    core = new Core();
    coreStarted = core.startCore(args, null);
    if (!coreStarted)
        return false;
    try {
        //获取命名服务
        NamingContextExt ncRef = core.getNamingContext();
        //初始化 POA
        POA rootpoa = core.getRootPOA();
        NameComponent ncs[] = getNameComponentsByName(ems);
        NMSSession nmsSImpl = new NMSSession();
        org.omg.CORBA.Object oRef = rootpoa.servant_to_reference(nmsSImpl);
        nmsSession = NmsSession_IHelper.narrow(oRef);
        org.omg.CORBA.Object objBind = ncRef.resolve(ncs);
        _EmsSessionFactory_IStub emsSF_stub = (_EmsSessionFactory_IStub)
            EmsSessionFactory_IHelper.narrow(objBind);
        NmsSession_IHolder nmsS_IHolder = new NmsSession_IHolder();
        nmsS_IHolder.value = nmsSession;
        EmsSession_IHolder emsS_IHolder = new EmsSession_IHolder();
        emsSF_stub.getEmsSession(user, pwd, nmsS_IHolder.value,
            emsS_IHolder);
        emsSession = emsS_IHolder.value;
        isConnected = true;
        fireConnectStatusChanged();
        //TileWindowsAction action = new TileWindowsAction();
        //action.actionPerformed(null);
    }
    catch (Exception e) {
        Logger.debug(this, e);
        return false;
    }
    return true;
}

```

```
}
```

连接成功后获取当前告警信息:

```
public ArrayList getAllEMSAndMEActiveAlarms(  
    String excludeProbCauseList[],  
    PerceivedSeverity_T excludeSeverityList[],  
    int how_many) throws Exception  
{  
    ArrayList objList = new ArrayList(0);  
    //获取告警信息, 以 ArrayList 类型返回  
    EMSConnector conn = EMSConnector.getInstance();  
    EMSMgr_I emsMgr_I = conn.getEMSManager();  
    EventList_THolder eventList_THolder = new EventList_THolder();  
    EventIterator_IHolder eventIt_IHolder = new EventIterator_IHolder();  
    emsMgr_I.getAllEMSAndMEActiveAlarms(excludeProbCauseList,  
                                        excludeSeverityList,  
                                        how_many,  
                                        eventList_THolder,  
                                        eventIt_IHolder);  
    org.omg.CosNotification.StructuredEvent events[] =  
        eventList_THolder.value;  
    if (null != events && 0 < events.length) {  
        for (int i = 0; i < events.length; i++) {  
            org.omg.CosNotification.StructuredEvent event = events[i];  
            AlarmData data = TMP814Util.getAlarmData(event);  
            objList.add(data);  
        }  
    }  
    EventIterator_I eventIt_I = eventIt_IHolder.value;
```

```

EventList_THolder tempEventList_THolder = new EventList_THolder();
boolean moreData;
do {
    if (null == eventIt_I)
        break;
    moreData = eventIt_I.next_n(100, tempEventList_THolder);
    org.omg.CosNotification.StructuredEvent tempEvents[] =
        tempEventList_THolder.value;
    if (null == tempEvents || 0 == tempEvents.length)
        break;
    for (int i = 0; i < tempEvents.length; i++) {
        org.omg.CosNotification.StructuredEvent event = tempEvents[i];
        AlarmData data = TMF814Util.getAlarmData(event);
        objList.add(data);
    }
} while (moreData);
return objList;
}

```

程序中对于当前连接轮询次数为 100 次时断开连接，重新连接后再获取相应的信息，解决了多次轮询后可能导致连接状态为无效连接的问题。

虽然 CORBA 的设计原理是基于异构平台的，理论上在不同的 ORB 之间的连接是畅通的。但是在实际应用中，仍旧出现了不兼容的现象。例如程序中连接方式对于对方是 IONA 公司的 ORBIX6.1 的连接是正常的，但对于 ORBIX5.1 却不能连接成功。经过与 IONA 公司工程师的多次沟通，服务器端修改了配置参数，客户端使用 JacORB 进行连接解决了问题。

5.2.2 性能数据采集

一、参数层次关系

获取当前所有指定的性能数据：

```

void getAllCurrentPMData(
    in PMTPSelectList_T pmTPSelectList,
    in PMPParameterNameList_T pmParameters,
    in unsigned long how_many,
    out PMDataList_T pmDataList,
    out PMDataIterator_I pmIt)
    raises(globaldefs::ProcessingFailureException);

```

输入参数:

```

PMTPSelectList_T pmTPSelectList //性能终结点列表
typedef sequence<PMTPSelect_T> PMTPSelectList_T;
struct PMTPSelect_T {
    globaldefs::NamingAttributes_T name; //名称属性列表
    transmissionParameters::LayerRateList_T layerRateList; //速率层列表
    PMLocationList_T pMLocationList; //位置列表
    GranularityList_T granularityList; //粒度列表
};
typedef NVSList_T NamingAttributes_T; //名称属性列表
typedef sequence<NameAndStringValue_T> NVSList_T;
struct NameAndStringValue_T
{
    string name;
    string value;
};
typedef sequence<LayerRate_T> LayerRateList_T; //速率层列表
typedef short LayerRate_T;
typedef sequence<PMLocation_T> PMLocationList_T; //位置列表
typedef string PMLocation_T;
typedef sequence<Granularity_T> GranularityList_T; //粒度列表
typedef string Granularity_T;
PMPParameterNameList_T pmParameters //性能参数列表
typedef sequence<PMPParameterName_T> PMPParameterNameList_T;

```

```
typedef string PMPParameterName_T;  
unsigned long how_many //时长
```

5.3 基于 CORBA 技术的数据采集机制在综合告警系统中的地位

基于 CORBA 的告警采集在综合告警系统中是属于数据采集分析软件模块。数据采集分析模块是实现对原始告警、性能数据采集后的分析及入库，并提供脚本编辑、测试工具，以及对自身运行状态的监控和管理。

通过数据采集分析软件，用户可实现对告警数据的采集、分析、建模及入库等工作，这些过程通过编辑配置文件实现，可根据需求进行灵活定制；系支持对告警标准模型的定义、修改，并可根椐要针对不同的应用增加新的模型；系统可结合告警原始信息和资源信息，将告警标准模型中的相关内容进行细化，并定为到王元最小监控单位，以满足告警定位的具体要求。

综合告警系统的其他模块的所有操作对象都是告警数据，由此可见，数据采集分析模块在整个综合告警系统中的地位，它是整个综合告警系统最基础的部分，没有这个模块的数据采集实现，就没有其他模块对告警的处理、归并与展现，没有报表部分对数据库中数据的统计。而基于 CORBA 技术的告警采集机制又是数据采集分析软件模块一个至关重要的模块。

现代电信运营商网络多元化的设备现状决定了数据采集分析软件模块的复杂性，而各种多样性的协议中 CORBA 作为一种总线结构能较好的解决电信应用领域中设备的多样性，平台的异构性。CORBA 可以提供统一的北向和南向接口，为电信运营商综合集成管理不同厂家的设备和网管提供了渠道，无论服务器端是什么操作系统客户端可以采取在同一个操作系统上实现所有服务器的连通。

第六章 综合告警系统的发展

6.1 综合告警系统发展

中国的电信行业发展到今天,已经由当初的规模竞争发展为服务竞争。而网络管理系统是运维部门的 IT 支撑系统的重要组成部分,也是服务质量得到保证的基本条件之一。现有的网管系统经过多年的建设,初步满足了每种专业的网络或业务的管理和监控的需要。但随着各种通信业务的不断发展与客户服务要求的不断提高,现有的网管系统与经营、生产、管理的现实要求还有很大的差距,因此所有运营商都把企业流程的重组、运营支撑系统的改造作为首要解决的问题。从对网络系统的故障监控来说,建设一套集中的、综合的告警监控系统已经越来越被各运营商所认同且在各个分公司逐渐进行试点建设。

正是在这种背景下,上海切点信息科技有限公司集推出了 OSS 领域一个重要的产品:全专业综合告警系统。本系统在应用上具有“各专业告警统一”、“告警过滤灵活”、“告警相关性分析”、“业务告警生成”、“业务质量监测”、“客户信息管理”等特点。而在系统设计上采用了多层架构体系,实现界面表现与业务逻辑分离,采集系统独立的构件方式,通过 CORBA 中间件作为数据总线,连接所有的应用服务程序,以提供安全可靠的数据传输通道。此外,系统还具有“丰富报表工具”、“自动化的系统维护”、“灵活的配置能力和伸缩性”等特点。

综合告警系统将江苏电信公司的“交换”、“传输”、“动力”、“数据”等六个专业的将近 15 套系统的告警监控数据接入了综合监控平台。经过将近一年的建设,系统已经基本建设完成且正式上线运行。系统把各个专业的物理故障和用户业务数据进行统一和关联,结合业务层面来分析和保障网的可靠、高效、优质运行,达到了“集中监控,快速响应,完善维护”这一建设目标。目前系统运行稳定,已经成为了运维部门进行网络系统日常监控维护的一个重要应用平台,并得到了江苏电信公司的认可与验收。

江苏电信综合告警系统的成功建设,是全专业综合告警系统在电信集团的一个典型应用。在系统建设的过程中,积累了大量的建设与使用经验,系统也得到了进一步的锤炼和完善,也为本产品的大面积应用与推广打下了一个良好的基础。随着“新运营时代”的来临,对全专业进行综合监控是一种不可避免的趋势,综合告警系统也必将得到越来越广泛的应用。

迄今为止,系统在国内已经成功地获得多个省的成功案例。在湖北、云南、江西,综合告警系统也成功地管理了交换、传输、数据、动力环境、主机监控等多个专业,多个厂商的

系统把这些分散成功的融入了综合告警系统，使得用户的运维实现了实时性和便利性，极大地提高了运维效率和对大客户的保障能力。

致 谢

衷心感谢导师李祥教授！从论文的选题、可行性研究、文献的收集、到研究工作的开展，特别是论文的撰写，导师给予了无微不至的关怀，提出了许多富有建设性的意见。

导师学识渊博，学术功底深厚，治学态度严谨，对当今前沿学科敏锐的洞察分析力和处理问题的高效率令我敬佩，对学生的严格要求和爱护使我深深感激，导师伟大的人格力量将激励我今后在科学的道路上克服困难，奋勇向前。导师认真、严谨、踏实的科研作风和渊博的学识，对学生的严格要求和爱护，使我受益非浅。今后唯有加倍努力，以报答导师的培育之情。

另外还要感谢上海切点信息科技有限公司的同事们在我对论文的整理中给的宝贵意见。

感谢所有关心和帮助过我的人们！

谢谢！

段 凤 琴

参 考 文 献

- [1] [美] Robert Orfali, Dan Harkey 著, 亢勇等译. Java 与 CORBA 客户/服务器编程(第二版). 北京: 电子工业出版社. 2004.
- [2] [美] R.Otte, P.Patrick, M.Roy 著, 李师贤译. CORBA 教程-公共对象请求代理体系结构. 北京: 清华大学出版社. 1999.
- [3] 朱其亮, 郑斌. CORBA 原理及应用. 北京: 北京邮电大学出版社. 2001.
- [4] [美] OMG 著. CORBA 系统结构、原理与规范. 北京: 电子工业出版社. 2000.
- [5] Jason Pritchard. COM 与 CORBA 本质与互用-体系结构·策略·实现. 北京: 北京科海电子出版社. 2002.
- [6] 李华飏, 郭英奎等编著. Java 中间件开发技术. 北京: 中国水利水电出版社. 2005.
- [7] Michi Henning, Steve Vinoski 著, 徐金栢等译. 基于 C++CORBA 高级编程. 北京: 清华大学出版社. 2000.
- [8] [美] 莱瑞 (Leroy, N.M.) 等著, 武欣等译. OCPOracle8!Java 组件编程—EJB、CORBA 和 JSP. 北京: 机械工业出版社. 2002.
- [9] 徐新华等. C++ Builder 5 高级编程技术—COM、CORBA 与 Internet 编程. 北京: 人民邮电出版社. 2000.
- [10] [美] Charles Petzold. 北京博彦科技发展有限责任公司译. Windows 程序设计 (第 5 版) (上、下册). 北京: 北京大学出版社. 2004.
- [11] [美] Dirk Slama, Jason Garbis, Perry Russell 著, 李师贤等译. CORBA 企业解决方案. 北京: 机械工业出版社. 2001.
- [12] [美] OMG 编著, 韦乐平等译. CORBA 服务. 北京: 电子工业出版社. 2002.
- [13] [美] OMG 编著, 韦乐平等译. CORBA 语言映射-CORBA Language Mapping. 北京: 电子工业出版社. 2001.
- [14] 龙湘明编. CORBA 程序设计指南 (入门). 北京: 人民邮电大学出版社. 2001.
- [15] 严蔚敏, 吴伟民编著. 数据结构. 北京: 清华大学出版社. 2002. 7.
- [16] 金永华, 曲俊生等编著. java 网络高级编程. 北京: 人民邮电大学出版社. 2001. 4.
- [17] 美国对象管理组织(OMG). CORBA 系统结构、原理与规范. 北京: 电子工业出版社. 2000.
- [18] 李明, 袁晓君编著. JAVA 计算机语言与函数应用. 北京: 科学出版社. 2000.
- [19] 孙冬梅, 陈小安, 罗天洪, 方义平. 基于 CORBA 和 Java 的协同设计通信平台. 重庆:

重庆大学学报(自然科学版). 2006年02期.

- [20] 侯捷译. java 编程思想(第二版). 北京: 机械工业出版社. 2005.
- [21] 孙卫琴编著. JAVA 面向对象编程. 北京: 电子工业出版社. 2006.
- [22] 周巍松等. Linux 系统分析与高级编程技术. 北京: 机械工业出版社. 1999.
- [23] Olaf Kirch. Linux 系统管理白皮书. 北京: 机械工业出版社. 2000.
- [24] 祁伟, 林美蓉. Java 和 CORBA 集成技术的研究. 电子质量. 2004 年 12 期.
- [25] 庞辽军, 刘坚, 李慧贤. CORBA 和 Java 的结合使用. 计算机应用研究. 2003 年 02 期.
- [26] 胡维华, 杨柏林, 童孟军. 基于 Corba 技术 SDH 网管 EML-NML 接口设计实现. 微电子与计算机. 2004, 21(1).
- [27] 刘兴伟, 贾年. 基于 CORBA 的统一网络管理系统. 计算机工程. 2004, 30(3):123~125
- [28] 刘文平, 侯晓林. 基于 CORBA 标准的企业集成信息平台技术. 制造技术与技术. 2002 年 11 期
- [29] 王建新, 陆炜妮. 基于 CORBA 技术的网上虚拟实验室系统架构设计. 小型微型计算机系统. 2004 年 10 期.
- [30] 孙凯, 周跃进. 基于 CORBA 框架及 JAVA 模式的 PDM 系统研究. 机械制造与自动化. 2004 年 06 期.
- [31] 欧阳云, 史殿习, 王怀民等. 一种组通信模型在 CORBA 平台上的设计和实现. 计算机工程与应用. 1999 年 11.
- [32] OMG 网站 www.corba.org
- [33] 灰狐论坛网站 <http://www.huihoo.com/corba/>
- [34] JAVA 公司网站 java.sun.com/developer/onlineTraining/corba/corba.html
- [35] www.mico.org/
- [36] www.cs.wustl.edu/~schmidt/corba.html
- [37] linas.org/linux/corba.html
- [38] en.wikipedia.org/wiki/CORBA
- [39] OMG Organize CORBA Resource Page. www.omg.org/gettingstarted/corbafaq.htm
- [40] IONA Technologies Inc. ORBacus for C++ and Java. 2003.5.
- [41] Gerald Brose. Java Programming with CORBA-Advanced Techniques for Building Distributed Applications.
- [42] MSDN Q244651. HOWTO: Creating a Shell Notification Icon from a Windows NT Service. 2001.

- [43] TeleManagement Forum. MTNM NML- EML Interface, Business Agreement(S). TMF513 v3.0. 2003.
- [44] TeleManagement Forum. MTNM NML- EML Interface, CORBA IDL Solution Set (S).TMF814 v3.0. 2003.
- [45] Gerald Brose, Andreas Vogel, Keith Duddy. Java programming with CORBA (Third Edition). 2004.
- [46] <http://www.jsping.com/jsp-learn/jsp%BF%AA%B7%A2318.html>

附 录

发表论文:

1. 《petri 网性质的线性时序逻辑描述与 spin 建模》段风琴 李详 计算机科学 2006.5

参加科研项目:

1. CORBA 协议的应用研究
2. 电信综合告警系统的研究与应用
3. 基于 CORBA 协议的综合告警系统数据采集