

# Web-Based Graphic Network Computing System in Power Enterprises

Major: Electric Power System and Automation  
Graduate: Li Gang      Advisor: Liu Junyong

The professional computing software of power system is playing a vital role in power system. The operators in power enterprises often purchase this kind of software individually. Due to the complexity of using the software, a considerable number of well-trained professionals are compulsory for power enterprises. What's more, because of different computing software and hardware systems, it's very tough for power enterprises to share data of electric network and computing software. Therefore, traditional computing software is incapable of meeting the demand of present power enterprises.

Taking the above factors into account, we designed the Web-based graphic network computing system, which adopted the Web-based multi-tier distributed frame. On one hand, the web browser was adopted as the client side of the computing system, which is able to avoid the users' inconvenience by not installing any client application. Furthermore, the software has a very friendly GUI (Graphic User Interface) and is easy to use by applying the powerful functions of Java in graphics display. On the other hand, the computing hub was established on the side of server. It's very easy to exchange information and system maintenance because the hub is in charge of the interconnection and intercommunication.

By using a computer connected to Internet, users are able to make professional computations when logging in the specified server. With the development of the new computing system, the traditional computing software will be replaced to a great extent.

The new computing system just needs to be updated on the server side. The users can always obtain the latest edition. The workload of system maintenance can be reduced tremendously. And meanwhile the Web-based online help documents are available so that the cost of training is also reduced enormously. For those who are not professionals in power enterprises, they can log in the computing server and receive professional training. So the use rate of the professional computing software is eventually increased.

Key words: Computing in Power System; Network Computing; Distributed System;  
Java

## 第一章 绪论

### 1.1 课题的背景和目的

随着电力系统朝着大电网、大机组、超高压、远距离发展,电力生产经营的管理者们越来越迫切的要求准确及时的得到电网的运营情况。电力系统中的专业计算在其中扮演着非常重要的角色。它为电力系统的决策者们提供了强有力的决策依据。

电力系统的专业计算通常包括潮流、故障分析、优化计算、暂态稳定、开关控制、边际成本分析等。通常对于电力企业的经营者们需要单独购买这样的专业计算软件,由于这样的计算软件使用复杂,需要大量的专业人员和专业培训,并且由于各个省电力局购买的专业计算软件不同,而且需要维护自己的硬件系统,其数据格式也各不相同,使得相互的信息共享和维护都十分困难。

如何使得各个电力局使用的专业计算系统能够互相利用和被更多的人使用并且减低系统的维护工作量?随着 Web 技术和分布式计算技术的飞速发展,基于 Web 的电力系统计算系统的出现解决了以上的问题。它在全省甚至全国使用,在各大省电力局维护多个分布式的计算中心。它在设计上采用了基于 Web 的 n 层模式的分布式系统实现方法,客户端应用基于 Web 服务的方式,并采用浏览器担当网络应用系统的前端。它彻底的解放了客户端,用户只需要一台接入 Internet 的计算机,通过浏览器便可以完成电力系统的专业计算。由于利用了 Web 强大的图形功能<sup>[26]</sup>,使得使用界面非常友好和易于使用。很大程度上替代了以前客户端软件的功能。

这种基于 Web 的电力系统专业计算软件的每一次升级只需要在服务器端进行,用户端每次访问的都是最新版本,真正做到了“一次安装、到处运行”。节省了大量维护工作。并且同时可以在网站上提供基于 Web 的在线帮助,大大减少了培训费用。对于非电力系统的用户,也可以登录网站进行自学和接受专业培训,大大提高了电力系统专业软件的利用率。

传统的集中式应用软件和基于传统的 C/S 结构的系统在电力系统中应用中显示出了越来越多的缺点。如这些软件和系统采用的是集中式的软件结构,模块之间多采用紧耦合方式,如共享全局变量进行通信,企业逻辑在客户端程

序中实现等。这样不利于系统的可靠性和可维护性,也不利于代码的重用。所以提出了面向对象和组件的电力系统软件<sup>[24,25,27,28,31,33]</sup>的设计方法。用对象和组件技术独立实现各个模块的功能,克服了集中式结构的特点,大大改善了系统的可重用性、可靠性、可维护性和开发效率。本文采用了 Java<sup>[4,10,11,12]</sup>这种完全面向对象的语言进行系统设计和开发,并发挥了它是网络开发语言的特性,很好的实现了系统功能。在第四章详细介绍了 Java 的特点。

由于电力系统规模日益扩大,其中的专业运算的计算量增加很快,加上电力系统运算对实时性的要求,使得对系统的运算能力提出了更高的要求。分布式系统<sup>[14,17,18,41]</sup>的出现有效的平衡了运算负荷,使得多台计算机可以共同协作来完成特定的运算任务,有效的提高了运算速度。其中 CORBA<sup>[20,40]</sup>(公共对象请求代理体系结构)是分布式结构中最常用的一种实现方式,得到了绝大多数计算机系统的支持。本文采用了这种分布式系统,在第四章详细的讲解了 CORBA 系统的结构和开发过程。

随着分布式技术和 Web 技术在电力系统中的深入应用,基于能量管理系统(EMS)和监视控制和数据采集系统(SCADA)的网络在线发布系统<sup>[1,7]</sup>得以应用。这种系统利用了 Internet 接入的便捷性,使得随时、随地都可以了解电力系统的网络信息。随着电力市场<sup>[61]</sup>在中国的开始普及和推广,这种模式的发布系统必然越来越多的被使用。

电力系统软件与网络技术,尤其是分布式计算技术和 Web 技术的结合一直是电力系统软件发展的方向,并取得了很大的进展。下面首先介绍了计算机技术发展的情况,然后具体介绍了这些技术在电力系统专业计算软件中的应用情况,并指出目前存在的问题和今后发展的方向。最后给出了本文所做的工作。

## 1. 2 计算机技术在电力系统中应用的历史和现状

现在计算机已经被广泛的应用到电力系统领域中,大大提高人们的工作效率。计算机的每一次硬件和软件上的革新都对电力系统的信息化带来了新的气息。尤其是现在 Web 技术和分布式计算技术的飞速发展,给电力系统行业的信息化带来了革命性的变化。下面简要介绍一下各种计算机技术的演变及对电力系统软件的影响。

### 1. 2. 1 计算机计算模式的历史沿革

随着计算机硬件和软件技术的飞速发展以及人们对高性能计算的需求, 计算机计算模式先后经历了以下几个阶段。

1. 单机计算模式。软件在独立分隔的计算机上运行, 不能进行互相的数据交换和调用。由于单独的计算机的计算能力有限, 不能完成如大规模数据查询、高性能的数据处理等任务。这时候电力系统领域中的软件主要完成一些数据整理和功能有限的计算功能。

2. 客户/服务器网络计算模式。网络兴起以后, 高性能的服务器开始出现, 数据开始向服务器集中, 服务器可以完成大规模的数据检索和存储, 并把不同的客户机连接起来, 从而实现了网络的互联和数据的共享。这种模式开始把客户端解放出来, 使得客户机之间能够协同工作, 完成以前不能够完成的工作。这时候电力系统领域中的软件开始丰富和成熟起来, 如电网调度自动化软件<sup>[45]</sup>、继电保护自动化软件<sup>[46,53]</sup>等。这种模式下虽然大多数的工作由服务器完成, 但仍然不能脱离对客户机的依赖。它需要客户端有相应的软件支持。

3. 基于 Web 的 n 层分布式计算模式<sup>[60]</sup>。随着基于 Web 的网络浏览普及以后, 网络浏览器(如 Internet Explorer, Netscape 等)成为了每个客户机上的必备软件, HTTP 协议成为了应用最广泛的网络协议之一。这种模式在设计上采用了 n 层的分布式结构, 客户端采用浏览器担当网络应用系统的前端, 它彻底的解放了客户端, 客户端不需要安装任何的客户端软件。任何的软件维护和升级都在服务器端进行。除此之外, 各种高性能的分布式计算协议的出现, 如 CORBA、RMI、DCOM 等, 使得计算机之间的协作更加高效和快捷。在电力系统专业计算领域中, 也有不少结合了分布式技术和 Web 技术的软件出现, 基于在线的电力系统计算模拟系统<sup>[2]</sup>就是这样的软件系统。

4. 端到端 (Peer to Peer) 的计算模式。这是最近才兴起的一种计算模式。以往的计算模式是以服务器为中心的, 服务器负责核心事务的处理, 久而久之, 它造成了客户机过度的依赖服务器, 如果服务器崩溃, 客户机便不能正常工作。而且由于数据的过度集中, 还会造成数据“垄断”。与网络的开放和透明原则相违背。随着 Web 服务和网格计算的提出, 使得打破这种垄断成为可能。客户机与客户机之间互相通信, 不再完全依赖服务器。在此结构中, 客户机可以同时做客户机和服务器。电力系统中这种模式的软件刚开始出现, 如基于 Web 服务

的负荷预测<sup>[61]</sup>在美国开始出现。

### 1. 2. 2 计算机语言的变迁

计算机语言的发展决定了计算机应用的普及,也决定了计算机技术的生命力。只有高效可靠的计算机语言才能开发出值得信赖的应用软件。从计算机发明开始的机器语言到 Fortran、C 等过程语言到功能强大但不易使用的面向对象语言 C++,都对计算机软件起到了很大的推动作用。下面简要介绍一下随着网络兴起流行起来的网络编程语言 Java,它从很大程度上推动了网络应用程序的开发和使用。

Java 语言<sup>[4]</sup>是一种随着网络发展起来的语言,它的基本结构与 C++极为类似,但却简单的多。Java 语言集成了一些语言的特点和优势,有避开了它们的不足之处。Java 与 C++相比,不再支持运算符的重载、多级继承、容易出错的指针操作以及一些容易混淆和较少使用的一些特性,而增加了内存空间的自动垃圾收集功能。复杂特性的省略和使用功能的增加使得软件开发简单而可靠。

平台的独立性也是 Java 的另一大特性。由于它采用先编译成中间码(字节码),然后装载与校验,再解释成不同的机器码执行,即“Java 虚拟机”的思想,它“屏蔽”了具体的“平台环境”的要求,使只要能支持 Java 虚拟机,就能运行各种 Java 程序。

面向对象技术具有继承性、封装性、多态性等众多的特点,Java 在保留这些特点的基础上,有具有动态联编的特性,更能发挥出面向对象的优点。

Java 的良好多线程机制使得程序能并行执行,通过使用多线程,程序开发者能用不同的线程完成特定的行为,这样就很容易实现网络上实时的交互行为。

### 1. 2. 3 计算机图形和动态显示技术

在电力系统中的图形应用<sup>[58]</sup>主要是将枯燥的电网数据转化成可视化的图形进行输入、输出、仿真计算等。使用户更直观的了解电网的运营情况。

以前的计算机图形的显示多采用基于点阵的位图格式,它记录了每一个象素点的信息。但这样的图形对于扭曲、放大、缩小都会失真,并且对于要求色彩丰富的画面更是需要很大的存储空间。对于图形的动态效果,只能靠更换位图图片,这样效率很低。矢量图形<sup>[23]</sup>(Vector Graphics)的出现改变了这一现状,

它记录了描述构成该图形的所有直线、圆、圆弧、矩形、曲线等图元的位置、维数和形状。矢量图的优点也就在于它在任何解析度下输出时都同样清晰。

即使对于经验丰富的计算人员来说,从电力系统的动态过程的在线检测数据或者仿真计算生成的大量中间数据和结果数据中获取对系统动态过程全景的完整了解也并非易事。借助科学计算可视化<sup>[22,71]</sup> (Visualization in Scientific Computing)对计算结果进行形象化的描述,即进行动态效果的显示便显得尤为重要。

在电力系统专业计算的软件中引入一些动态效果的显示,如有功无功在节点之间的流动效果、节点之间的电压高低分布等,会使用户对整个系统有着更清晰的了解。

#### 1. 2. 4 计算机安全技术

网络的飞速发展使得安全显得格外重要。在电力系统的软件系统中确保数据安全也是必须要考虑的一个因素。两个方面的问题需要考虑:

1. 传输过程中的安全<sup>[15]</sup>。对数据传输的起点用加密算法进行加密,在数据传输的终点用对应的解密算法解密。

2. 防止非法用户侵入系统<sup>[88]</sup>。加强对用户的身份效验。在用户登陆系统进行身份效验过程中使用一些必要的安全措施。如必须使用 8 位以上的密码,密码必须是数字和字母的混合,以及定期更换密码等措施。

#### 1. 3 目前存在的问题和发展的方向

目前国内的电力系统方面的软件在 Web 方向有了一定的发展,并且发展势头良好。已经出现了不少较为成熟的 Web 电力系统软件,可是这些系统基本上局限于企业管理系统<sup>[5,6,8,9]</sup> (MIS),如电网实时信息系统、电网管理系统等。

并且在网络图形化的应用方面进展缓慢,大多采用枯燥的表格形式发布,即使有图形,也是静态的网络图形,使得界面的友好性和可操作性较差。并且在电力系统的专业计算软件向 Web 方向的发展涉及很少。主要有两个方面的难度:

1. 电力系统专业计算所使用的算法虽然有很多书介绍<sup>[3,51,52]</sup>,可是要开发出真正实用的算法程序需要花费大量的人力物力。

2. 普通的单机版专业计算软件<sup>[7,50]</sup>与基于 Web 的计算软件开发流程和方式差别很大。从单机版的软件向基于 Web 的网络软件移植需要大量的工作。

国外尤其是以美国为代表的发达国家对网络的分布式计算的研究处于领先地位, 在 Web 上实现了电力系统领域的专业计算<sup>[1,2]</sup>, 可是在 Web 的客户端基本上还是简单的数据输入和显示。网络系统基本上没有实现图形化, 不能实现电网系统的图形输入和动态显示<sup>[23,26]</sup>, 还不能替代传统的单机板的专业计算系统。

随着 Web 服务<sup>[13]</sup>(Web Service)这种新的基于 Web 的分布式技术在 Internet 上的普及, 分布式技术将开始得到越来越多的使用。基于 Web 的分布式平台的电力系统软件系统将成为主流。

### 1.3 本文的主要工作

本文根据以上的分析, 提出了基于 Web 的电力系统专业计算软件的解决方案, 能很好的解决以上现有的计算系统所面临的问题。这种系统集成了 Web、Java、分布式系统、数据库、矢量图形、网络安全等技术。系统高度模块化, 做到了真正的跨平台和完全免维护。

本文系统的阐述了基于 Web 的电力系统分布式网络图形化计算系统的整个设计过程, 包括 Web 终端的设计、数据库系统的设计、算法服务器的设计以及对于系统设计的安全。系统采用了 JBuilder7.0 开发, 分布式组件采用 VisiBroker 的 CORBA<sup>[20]</sup>组件, 数据库系统采用 Borland 公司的 Data Store。最后还对系统向 Web 服务和网格计算 (Grid Computing) 方向扩展提出了构想。

对以上系统本文使用了 IEEE6 和 IEEE30 节点系统进行了模拟, 均取得了较好的效果。

## 第二章 电力系统专业计算

### 2.1 前言

电力系统中的专业计算<sup>[28,51,52]</sup>主要有潮流计算、故障分析、开关控制、优化求解、边际成本分析。

### 2.2 潮流计算

潮流计算是根据给定的电网结构、参数和发电机、负荷等元件的运行条件，确定电力系统各部分稳态运行状态参数的计算。通常给定的运行条件有系统中各电源和负荷点的功率、枢纽点电压、平衡点的电压和相位角。待求的运行状态参量包括电网各母线节点的电压幅值和相角，以及各支路的功率分布、网络的功率损耗等。

潮流计算是电力系统是电力系统分析中最基本的计算，用以研究系统规划和运行中提出的各种问题。对规划中的电力系统，通过潮流计算可以检验所提出的电力系统规划方案能否满足各种运行方式的要求；对运行中的电力系统，通过潮流计算可以预知各种负荷变化和网络结构的改变会不会危及系统的安全，系统中所有母线的电压是否在允许的范围以内，系统中各种元件（线路、变压器等）是否会出现过负荷，以及可能出现过负荷时应事先采取哪些预防措施等。

通常潮流计算的流程和结构如下：

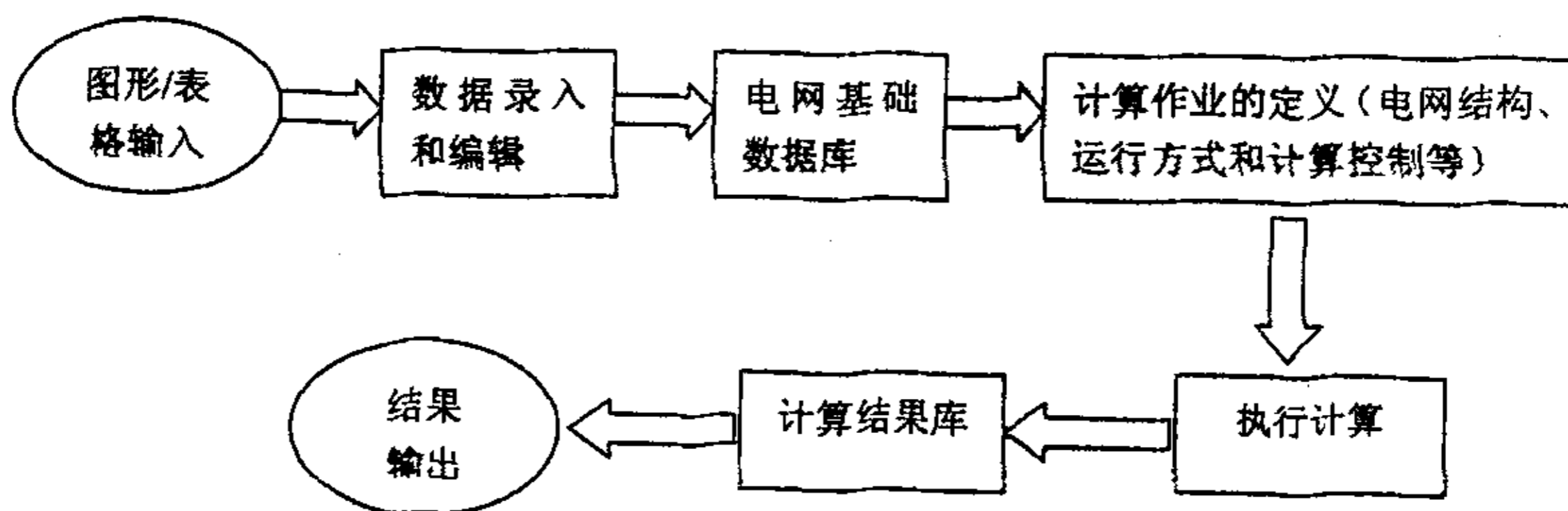


图 2-1 电力系统潮流计算流程图

Fig. 2-1 The flow chart of computing of power system



由于潮流计算是否收敛, 不仅与被计算的系统有关, 而且和所选用的计算方法也紧密相关。下面介绍两种通常的计算方法。

### 2. 2. 1 牛顿潮流

牛顿潮流是一种解非线性方程组的方法, 它是当前广泛采用计算潮流的方法。

它的基本原理是: 是在解的某一邻域内的某一初始点出发, 沿着该点的一阶偏导数——雅可比矩阵, 朝减小方程残差的方向前进一步, 在新的点上在计算残差和雅可比矩阵继续前进, 重复这一过程直到残差达到收敛标准, 即得到了非线性方程组的解。

因为越靠近解, 偏导数的方向越准, 收敛速度也就越快, 所以牛顿潮流有收敛性好的特点。由于在每次迭代后都需重新计算 Jacobi 矩阵的元素, 所以影响了该算法的解算快速性。

### 2. 2. 2 标准快速解耦法

这一算法采用了两种近似条件:

1. 节点有功注入功率的变化, 主要与有关节点的电压相位角变化相关;
2. 节点无功注入功率的变化, 主要与有关节点的电压量值变化相关。

标准快速解耦法对具有  $r > x$  的绝大部分电力系统有着很好的收敛特性, 只有少数例外。

快速分解法潮流是一种试验算法, 实际上仅仅“快速”化, 即将雅可比矩阵常数化, 收敛性很差; 而仅仅“分解”即对 P 和 Q 分别修正, 收敛性也很差。只有在两者结合的条件下才出现了优势, 一般比牛顿法收敛多几次, 但总计算量下降数倍。

### 2. 3 故障分析

事故分析可以自动模拟和求解一组用户在事故列表中定义的交流潮流的意外事故, 分析事故情况, 输出事故状态下的潮流, 并且根据系统安全系数指标(负荷、电压)对各种事故的严重情况进行排序。

根据需要, 用户可以自行定义一系列假想的事故情况, 例如: 某一发电机的退出运行、某一支路的退出运行等等。事故分析可以根据用户设定的事故进

行事故的分析,按照系统安全指标自动排列出事故严重的情况。事故分析的结果可以反映出现存的电力网络存在的一些潜在的危险,用户可以进行网络的修正,使得整个电力网络实现对电能的安全、稳定、优质地传输。

事故分析一般都采用收敛性较好的潮流计算方法进行分析计算,即采用牛顿法或快速解耦法。

## 2.4 开关控制

开关控制是在一系列事故运行条件下(事故运行条件由用户在事故列表中自行定义),进行系统性搜索,找出特定约束下(又称触发约束)对减轻越限具有最好调节作用的一系列开关,人为地进行开关控制。在修正模式下,开关控制的目的是减轻正常系统运行中的触发约束越限;在事故约束模式中,开关控制的目标是预防事故后系统运行时的触发约束越限。

开关控制分析功能可以在某一设定的事故运行条件下,自行搜寻出一组开关,对这组开关进行控制,可以满足电力网络电能传输的安全与稳定约束。

开关控制可以作为独立功能应用,也可以作为优化求解(OPF)方法的一部分。开关事件列表是根据电力系统工程知识来定义的,每种事故可以包含开断、投入以及开关位置的改变,特别是意外事故。用户必须指定与每一开关事件相关的一系列触发约束,通过开关控制可以减轻这些越限。

## 2.5 优化求解

优化求解就是解决最优有功和/或无功分配(包括开关控制)的问题。在保证正常工作约束条件或是有选择地防止边界限制扰动的情况下,以设定的某一目标函数为目标进行优化求解,从而实现电力网络有功或无功分配最为合理的目的。

优化求解的性能包括:

1. 优化有功或无功控制,包括一系列优化目标选项的开关控制。
2. 考虑大量电力系统运行的限制条件,并提供大量的节点、约束条件和点对点边际成本信息。
3. 可按标准调度、校正(补救措施)和预防安全(意外事故约束条件)的模式运行。补救措施可以在电力系统正常工作下进行,也可以在事故状态后进行。

4. 自动执行预先给定的控制和极限的优先级。
5. 当问题不可解时获得最好的工程解法。
6. 适用于闭环控制系统, 包括限制性经济调度。

对电力网络进行优化, 合理分配电网中各个节点有功或无功。进行优化后的电力网络可以降低传输电能所需的成本, 实现其特殊的经济利益。

## 2. 6 边际成本分析

边际成本分析是指用户自行设定一系列费用曲线, 程序以这些费用曲线为系统运行的约束边界条件, 对系统的运行进行控制计算。该程序提供了一个短期边际最优费用的全面分析, 而且只能在以某一费用目标函数为目标, 进行耦合优化求解 (OPF) 运算后方可进行。

边际成本包括:

1. 针对目标的每个捆绑限制的边际费用。
2. 通过每根母线或一组母线 (区域) 的有功和无功传输的边际费用。
3. 母线或区域的崩溃造成的对发电、损耗、阻塞的边际费用。
4. 损耗和阻塞崩溃下的点对点边际传输费用。

每个捆绑限制对每个边际费用的相关分量的作用。

### 第三章 网络计算的历史和发展

#### 3.1 传统的客户/服务器机构

几年前个人电脑迅速在全世界普及,大量的系统采取了客户/服务器<sup>[44]</sup>这种结构来设计,电力系统中的系统也大多如此。在客户/服务器结构中通常有客户端来执行应用程序,然后连接到后端的数据库服务器中存取应用系统需要的数据。图 3-1 说明了它的结构。

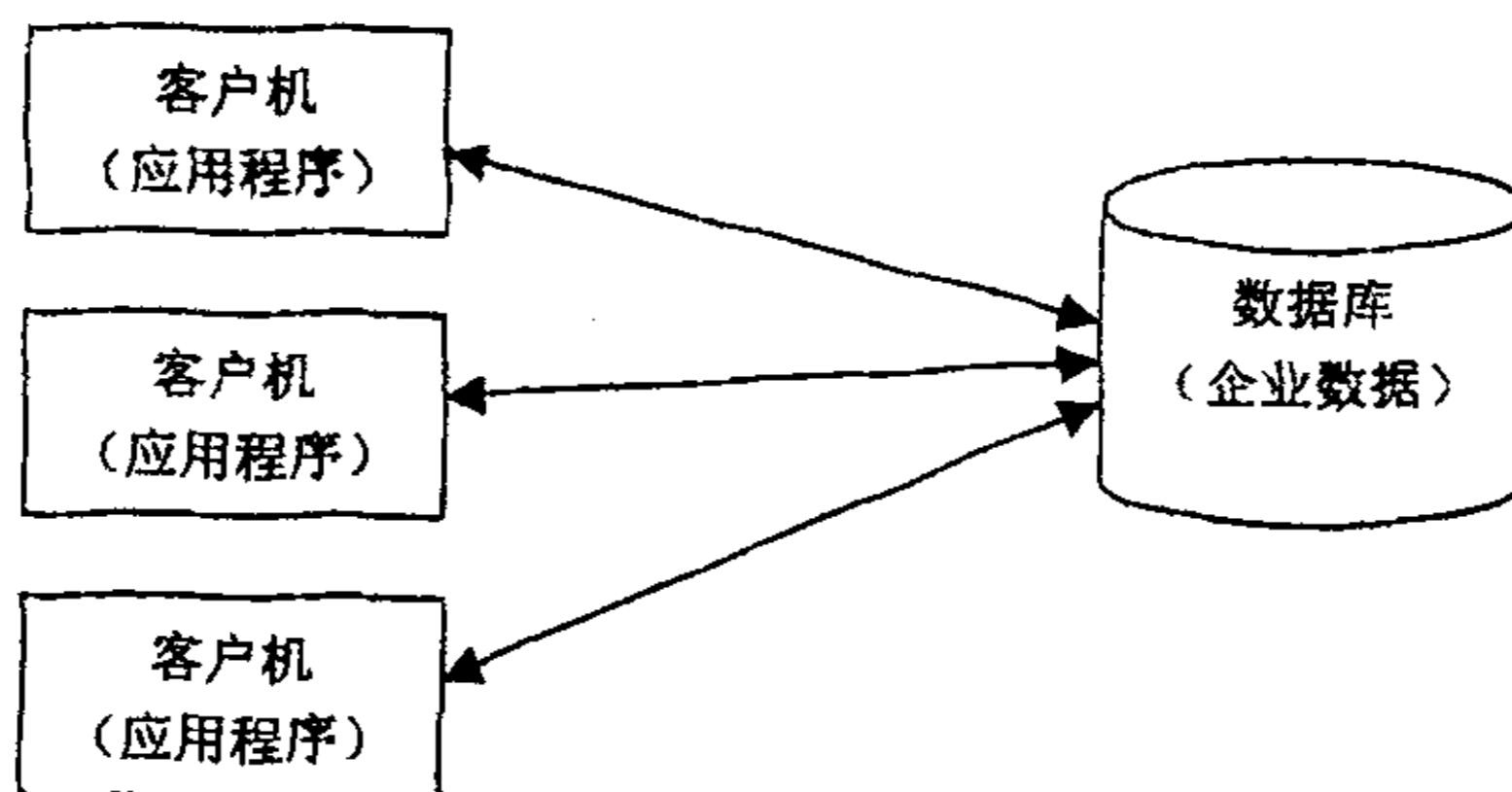


图 3-1 传统的客户/服务器结构示意图

Fig. 3-1 The traditional client/server structure

近几年来,由于 Internet/Intranet 的兴起,电力企业需要公开各种信息提供给网上的用户,使其信息能够得到大范围的共享。因此企业要求开发的系统必须能够让客户使用浏览器来查询信息。所以在企业内部架设了一台 Web 服务器<sup>[45,46,47]</sup>,然后使用 CGI/ASP/Servlet<sup>[74]</sup>把客户端应用程序重新编写一次,并分发到 Web 服务器上。当用户通过浏览器访问服务器时,程序在服务器端执行,然后在把执行的结果发回浏览器上显示。系统结构图见图 3-2 所示。

这种结构我们发现了很多的问题。它经常把应用系统的企业逻辑编写在客户端的应用程序中,因此当应用程序需要改变时,所有在客户端的应用程序都要改写,使系统的维护成本提高。虽然有一些系统把应用逻辑和数据库绑定,但这样会有更大的问题,因为如此一来不但应用系统绑定在特定的数据库上,

而且许多应用逻辑代码并不适合在数据库中执行，并会严重的拖累数据库的执行效率。

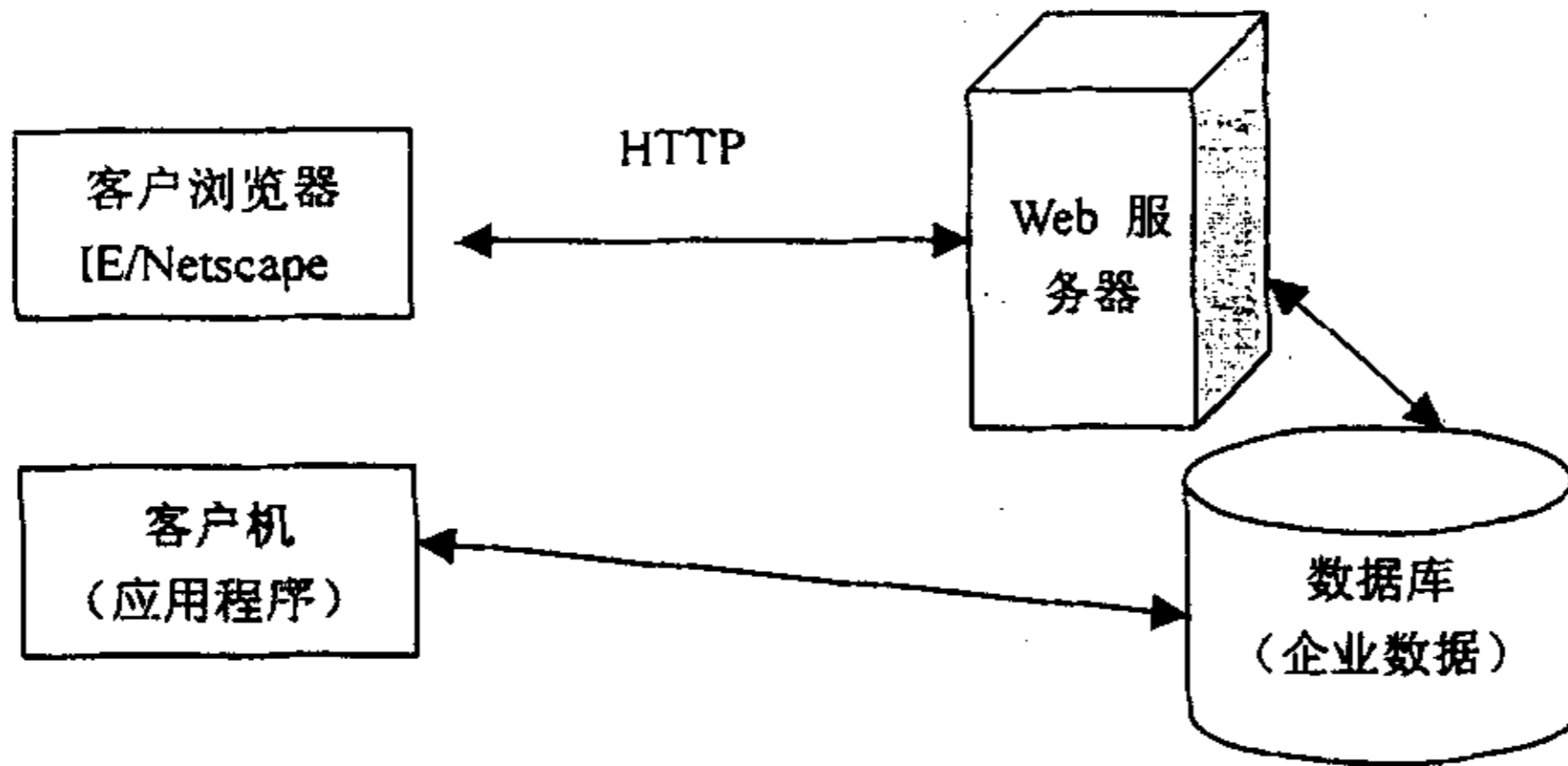


图 3-2 Internet/Intranet 加上传统的客户/服务器

Fig.3-2 Internet/Intranet and traditional client/server structure

### 3. 2 分布式计算结构

为了解决传统的两层 C/S 结构在维护使的高成本同时为了增强系统的重复利用率，发挥面向对象设计的功能，因此提出了分布式的三层或多层体系结构，也就是应用了所谓的应用程序服务器。应用程序服务器简单的说就是一个包含企业逻辑的应用程序，软件开发人员以一种特定的组件形态，例如 COM/DCOM、CORBA 对象、RMI 等，封装企业的逻辑代码。这种经过封装，能够执行特定企业功能的对象便称之为“企业对象”，然后它这些企业对象分发到应用程序服务器中。当应用程序服务器包含很多企业对象后，就可以在开发企业应用程序时使用这些在应用程序服务器中的企业对象提供的服务。同时也能够使用 CGI/Servlet<sup>[12]</sup>等技术存取应用程序服务器中的企业对象，那么普通的客户端和使用浏览器的客户端就可以使用到最新的企业对象。采用这种包含企业应用服务器的结构有很多的好处，除了可以增加企业对象的重用性外，整个系统的开发和维护成本也立刻降低下来。由于分布式结构可以把系统的不同运作部署在不同的服务器上，所以整个应用系统的灵活性和性能也有了大幅度的提升。图 3-3 显示了分布式系统<sup>[59,60,90]</sup>的结构。

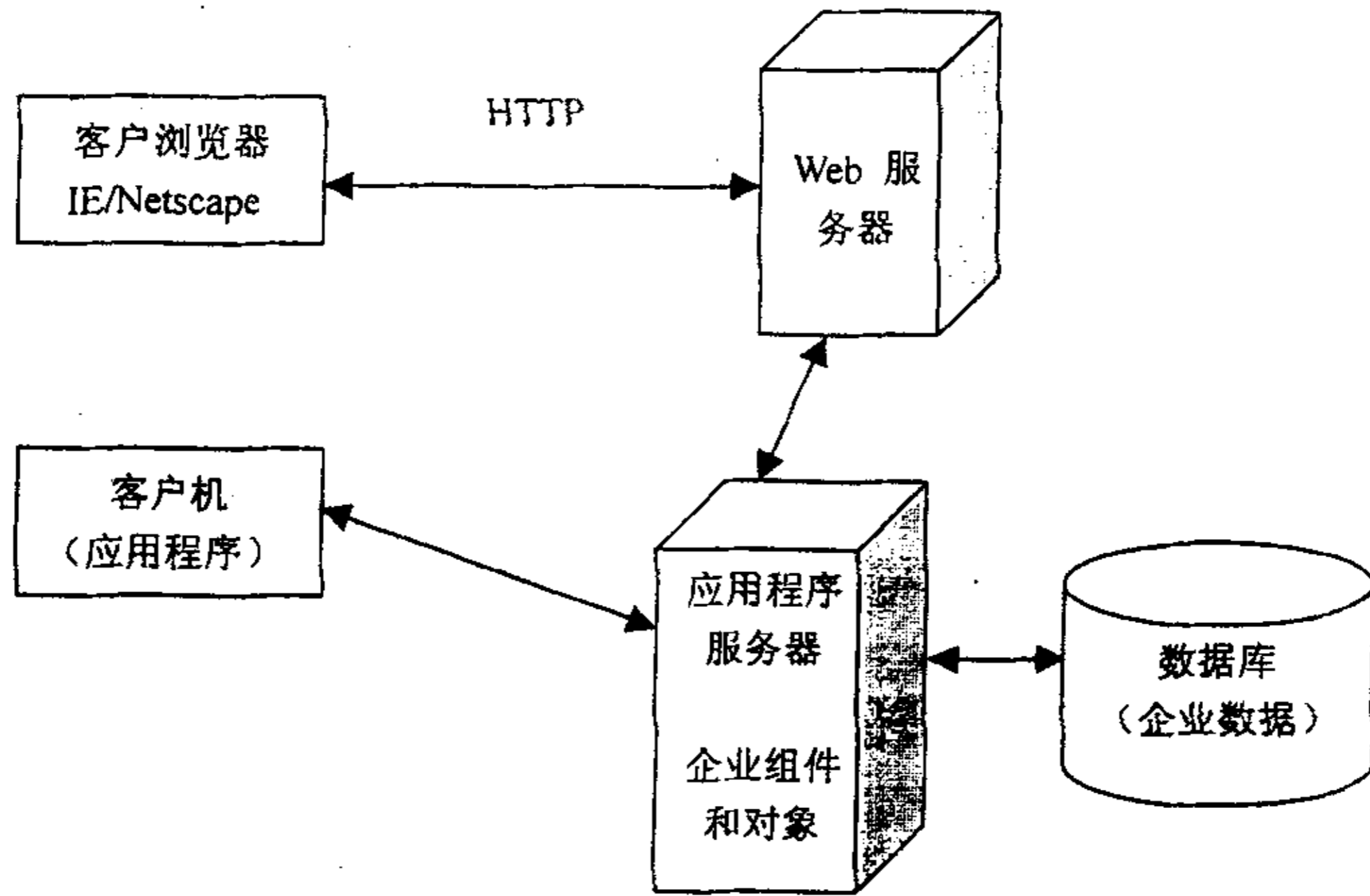


图 3-3 三层分布式系统结构示意图

Fig.3-3 The three-tier distributing system infrastructure

### 3. 3 Web 服务和网格计算

由于计算机的平台（如 Windows, Unix, Linux, Mac 等）都是由计算机商业公司（如 Microsoft, SUN, Apple 等）提供的，平台之间的数据交换通常都是通过通用的网络协议来进行（如 HTTP, FTP 等）。但是随着 Web 在全世界的广泛使用，Web 已经开始逐渐从仅仅提供信息浏览向逻辑运算的平台转变。这样在 Web 上需要交换大量的数据，就使得现有的协议在基于 Web 的数据交换上变得很不方便。扩展性标记语言 XML<sup>[66]</sup> (Extensible Markup Language) 的出现便解决了以上问题。它是一种平台独立的数据交换语言。它把数据的内容和显示分开，使得它成为了各种平台上在 Web 上的数据交换语言

Web 服务<sup>[61,67]</sup>是最近开始蓬勃发展起来的一种应用。它是一种可以接收从 Internet 或者 Intranet 上的其它系统中传递过来的请求，轻量级的独立的通讯技术。这种技术允许网络上的所有系统进行交互。Web 服务的概念就是使用松散

耦合的消息(而且越来越多的是依靠 XML 界面)发送的内容和软件过程来为一些特定的用户需要进行"服务"。WEB 服务是现今开放网络环境下的最新发展趋势。

Web 服务和 CORBA 的功能基本相同,但是适用领域不一样。Web 服务实现的就是基于 Web 的远程方法调用。服务器端利用 Web 向外界提供一些 API (Application Programming Interface) 供客户端调用。Web 服务的传输是基于 HTTP 协议的。它也有接口定义和服务定位的问题,接口的定义采用 WSDL(Web Service Description Language), 服务定位采用 UDDI (Universal Description, Discovery, and Integration)。这两者都是基于 XML 规范实现的。而 CORBA 由于其客户端配置的复杂性,以及穿越防火墙的问题不适合互联网传输。但是具有更高的可靠性和处理能力。因而适合于客户端数目比较少,通信数据量比较大的应用。所以 CORBA, Web 服务实现的都是远程方法调用。CORBA 适用于后台分布式服务器之间的互联。Web 服务是广域网内客户端和服务端一种比较好的交互方式。

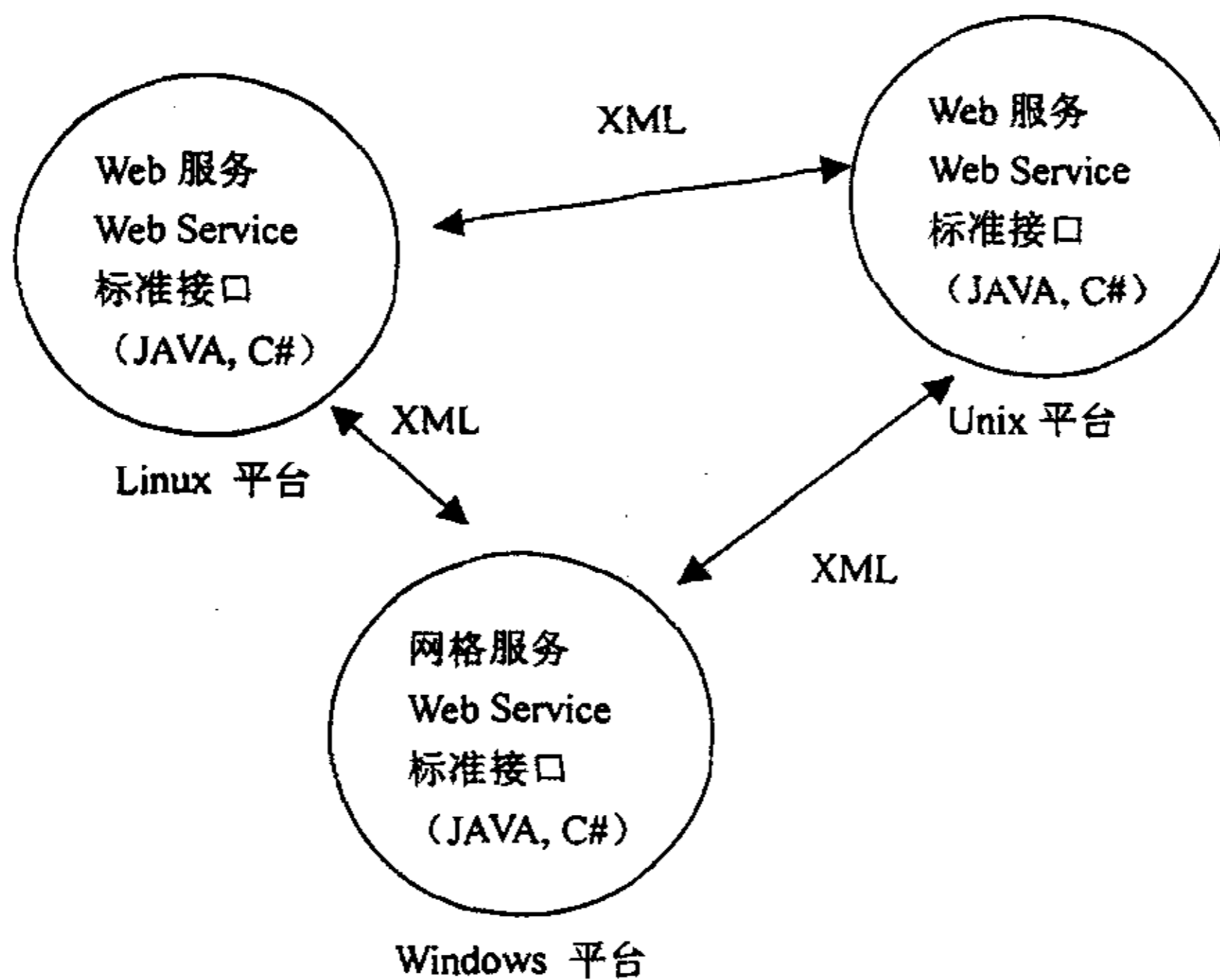


图 3-4 Web 服务结构示意图

Fig. 3-4 Web Service infrastructure

网格计算 (Grid Computing) 的出现, 目的是将互联网应用变成一种智能、高效、简单的基础设施, 就像我们正在使用的电力、水利设施一样。网格计算是多年高性能计算或者分布式计算后的一个更新的发展。建立网格计算的意图在于协调或共享各种资源, 这些资源可以是硬件的、软件的、通讯的、网络的以及信息的。可以说在更高的层次、更多的层面上把信息更好的运用起来, 使得这个过程变得更加的优化。所以说它可以把很多很多的信息孤岛、或者分布的资源联系起来, 因为在 Internet 上面, 信息本身是一个一个的孤岛, 现在由于有了网格计算, 可以把物理层、网页、信息层都连接起来。

美国《福布斯》杂志的科技版《Forbes ASAP》2001年9月10日发表一组文章, 预测信息技术的下一波大浪潮将在2004、2005年度出现, 这一波浪潮的本质特征就是万维网 (World Wide Web) 升华为网格 (Great Global Grid), 2020年, 由此产生的互联网将成长为一个20万亿美元产值的大产业。

从本质上讲, 网格计算的概念非常大, 它出于一个初衷: 充分使用全世界的计算资源。举个例子来讲, 很多大型公司的计算能力非常强, 但是实际上这些计算能力不是每天24小时都在使用。那么如此强大的计算能力在闲置时是否可以应用到其它需要强大计算能力的工作中呢? 一方面可以使闲置资源得到合理利用, 另一方面又可以为公司提供额外的收入。网格计算的出现, 为这种应用提供了可能。

从以上的对网络计算的分析来看, 网络计算的模式已经开始从客户机/服务器、多层的客户机/服务器结构开始向服务器对服务器、客户机对客户机的计算模式转变。传统意义上的客户机/服务器体系开始被打破, 端对端 (Peer to Peer) 的计算模式开始普及。网络中的任意一台终端可以是客户机, 也可以是服务器。



## 第四章 JAVA 和分布式系统

### 4. 1 Java 的特点

Java<sup>[4,,10,11]</sup>是由美国太阳电脑公司(Sun Microsystems, Inc.)所发展出的第一种能在国际互联网络(Internet)上面具有(硬件、软件中立性)交互能力的程序语言。自从它被开发以来,在互联网上被普遍使用。总的说来,Java是一种简单、面向对象<sup>[23,24]</sup>、分布式、解释、健壮、安全、结构的中立、可移植、性能很优异、多线程、动态的语言。

#### 4. 1. 1 简单

Java 提供了一个功能强大语言的所有功能,但几乎没有一点含混特征。C++安全性不好,但 C 和 C++被大家接受,所以 Java 设计成 C++形式,让大家很容易学习。Java 去掉了 C++语言的许多功能,让 Java 的语言功能很精炼,并增加了一些很有用的功能,如自动收集碎片。

Java 去掉了以下几个 C 和 C++功能:

- 指针运算
- 结构
- typedef
- #Define
- 需要释放内存

2. 这将减少了平常出错的 50%。而且,Java 很小,整个解释器只需 215K 的 RAM。

#### 4. 1. 2 面向对象

Java 实现了 C++的基本面向对象技术并有一些增强,(为了语言简单,删除了一些功能)。Java 处理数据方式和用对象接口处理对象数据方式一样。

#### 4. 1. 3 分布式

Java 建立在扩展 TCP/IP 网络平台上。库函数提供了用 HTTP 和 FTP 协议传送和接受信息的方法。这使得程序员使用网络上的文件和使用本机文件一样容易。

#### 4. 1. 4 健壮

Java 致力于检查程序在编译和运行时的错误。类型检查帮助检查出许多开发早期出现的错误。Java 自己操纵内存减少了内存出错的可能性。Java 还实现了真数组，避免了覆盖数据的可能。这些功能特征大大提高了开发 Java 应用程序的周期。Java 提供：

- Null 指针检测
- 数组边界检测
- 异常出口
- Byte code 校验

#### 4. 1. 5 结构中立

另外，为了建立 Java 作为网络的一个整体，Java 将它的程序编译成一种结构中立的中间文件格式。只要有 Java 运行系统的机器都能执行这种中间代码。现在，Java 运行系统有 Solaris2.4(SPARC), Win32 系统 (Windows95 和 Windows NT). Java 源程序被编译成一种高层次的与机器无关的 byte-code 格式语言，这种语言被设计在虚拟机上运行，由机器相关的运行调试器实现执行。

#### 4. 1. 6 安全

Java 的安全是两方面的。一方面，在 Java 语言里，象指针和释放内存等 C++ 功能被删除，避免了非法内存操作。二方面，当 Java 用来创建浏览器时，语言功能和一些浏览器本身提供的功能结合起来，使它更安全。Java 语言在你的机器上执行前，要经过很多次的测试。它经过代码校验，检查代码段的格式，检测指针操作，对象操作是否过分以及试图改变一个对象的类型。

#### 4. 1. 7 Byte-code 校验

如果 byte-code 通过代码校验，没有返回错误，我们可知道：

- 代码没有堆栈上溢出和下溢出
- 所有操作代码参数类型都是正确的
- 没有发生非法数据转换，如将整数转换成指针。
- 访问对象操作是合法的

#### 4. 1. 8 类装载

Class Loader 通过将本机类与网络资源类的名称分开，来保持安全性。因为

调入类时总要经过检查, 这样避免了 Trojan 马程序的出现。从网络上下载的被调进一个与源相关的私有的名字域。当一个私有类访问另一个类时, build-in(本机类)首先被检查, 然后检查相关的类。这样就避免了破坏本机类情况的出现。

#### 4. 1. 9 可移植

除了结构中立的可移植性外, Java 还实现了各平台上移植, 以方便程序的开发。整形数在 Java 里永远是整数, 32 位整数。而且, Java 建立在一个虚拟的 Windows 系统上, 所以很容易地在 UNIX, PC 和 Mac 环境下实现。

#### 4. 1. 10 解释性

Java 解释器(运行系统)能直接运行目标代码指令。链接程序通常比编译程序所需资源少, 所以程序员可以在创建源程序上花上更多的时间。

#### 4. 1. 11 高性能

如果解释器速度不慢, Java 可以在运行时直接将目标代码翻译成机器指令。Sun 用直接解释器一秒钟内可调用 300,000 个过程。翻译目标代码的速度与 C/C++ 的性能没什么区别。

#### 4. 1. 12 多线程

Java 提供的多线程功能使得在一个程序里可同时执行多个小任务。线程有时也称小进程, 它是一个大进程里分出来的小的独立的进程。因为 Java 实现的多线程技术, 所以比 C 和 C++ 更健壮。多线程带来的更大的好处是更好的交互性能和实时控制性能。当然实时控制性能还取决于系统本身(UNIX, Windows, Macintosh 等), 在开发难易程度和性能上都比单线程要好。任何用过当前浏览器的人, 都感觉为调一副图片而等待是一件很烦恼的事情。在 Java 里, 你可用一个单线程来调用一副图片, 而你可以访问 HTML 里的其它信息而不必等它。

#### 4. 1. 13 动态

Java 尽可能地利用了面向对象技术的优点。Java 并不试图在运行前将所有模块全部连接起来。一个新的或升级的库函数并不需要更改源程序就能正确运行。Java 还简化了使用一个升级的或全新的协议的方法。如果你的系统运行 Java 程序时遇到了不知怎样处理的程序, 没关系, Java 能自动下载你所需要的功能程序。

## 4. 2 分布式计算系统

目前有三种比较流行的分布式系统设计方案, 以下比较了它们的优劣。

CORBA (Common Object Request Broker Architecture) 有几个基本的优点: 与开发语言无关的独立性, 与开发者无关的独立性和与操作系统无关的独立性。CORBA 在当前每一种主流操作系统上均有实现 (仅就 Microsoft 的各种操作系统来说, CORBA 获得的支持甚至超越了 DCOM)。

DCOM (Distributed Common Component Model) 是 Microsoft 独立开发, 仅运行于 Windows 平台上。它和 CORBA 一样, 虽然是独立于语言的, 但有三个重大缺点。首先, 它由单一开发者 (微软) 定义并控制, 这大大限制了 DCOM 使用者的选择范围 (比方说开发工具和风格)。其次, DCOM 缺乏众多的平台支持, 这极大地制约了代码的可重用性和 DCOM 应用的可扩展性。最后, 与 CORBA 相比, DCOM 是一种非常不成熟的技术。尽管微软目前正为 DCOM 加入消息和事务支持, 但这些功能在 1994 年的 CORBA 2.0 就已经实现了, 并且正由几家不同的 CORBA 软件开发商所发行。

RMI (Remote Method Invocation) 是 Sun 的 Java-Only 解决方案。RMI 使 Java 客户能够访问远地的服务对象。与 CORBA 类似, 但两者并不一样。其关键在于服务器端的应用程序也必须用 Java 编写, 且只能使用 JDK 开发包中提供的工具。你根本无法把过去编制的代码加到新程序中去, 除此之外, RMI 没有服务这一概念。另外, 根据 RMI 写出的 Java 服务器对象往往性能低劣, 这个缺点源于 Java 虚拟机的效率低下。但 RMI 开发要比 CORBA 容易, 故对于用纯 Java 书写的小规模的应用程序来说, 是一种可行方案。

## 4. 3 CORBA 与 JAVA 的相互关系

CORBA 是非常全面的分布式对象平台。CORBA 使 JAVA 应用可以跨越网络、语言以及操作系统, 并为 JAVA 提供了一组分布服务, 如分布式自我观察、动态发现、事务、关系、安全和命名等。

JAVA 不仅是一种语言, 它还是一个动态代码系统。它对运行对象来说是一个可移植的虚拟机 (JVM)。JAVA 为开发、管理、发布 Client/Server 应用提供了更简单的方式。人们可以通过将应用放在一个 Web 服务器上, 将这一应用发布给成千上万用户, 而不必关心它的安装和升级。JAVA 还非常适合服务器的开

发, 它可以动态地将服务代码移向最需要它们的地方。

JAVA 将会使 CORBA 对象能够运行在从主机、网络计算机到蜂窝电话等可编程的各种机器上, 并简化了大型 CORBA 系统的代码发布。对客户和服务对象来说 JAVA 是很理想的编程语言, JAVA 内置的多线程、垃圾收集和错误处理使编写健壮的网络对象变得很容易。

这两种对象模型可以很好地相互补充, CORBA 处理网络的透明性, JAVA 处理实现的透明性, CORBA 为 JAVA 可移植应用环境提供了一个分布式的结构。

## 4. 4 CORBA 系统具体描述

### 4. 4. 1 对象管理体系结构

对象管理组织 OMG (Object Management Group) 负责制订与发布 CORBA 规范。由 OMG 发布的对象管理体系结构 OMA (Object Management Architecture) 是包括 CORBA 规范在内的所有 OMG 规范的概念模型基础。

对象管理组织是一个成立于 1989 年的非盈利性联盟, 其目标是促进在分布式系统开发中面向对象技术的理论与实践的发展。该组织现有成员 800 多个, 包括信息系统产品供应商、软件开发商以及最终用户。OMG 负责制订并维护一套规范, 以支持分布式、异类 (heterogeneous) 环境的软件开发项目, 覆盖了从分析、设计到编码、部署、运行和管理的整个软件开发过程。这些规范是一种工业或行业的标准, 并不是由 ISO、ANSI 或 IEEE 接纳并发布的正式标准。

OMG 已发布的规范主要是统一建模语言 UML (Unified Modeling Language) 和公共对象请求代理体系结构 CORBA (Common Object Request Broker Architecture)。遵循 OMG 规范可开发出具有标准面向对象接口的、可互操作的、可重用的、可移植的软件组件。

UML 是关于面向对象分析与设计阶段的表示技术的规范, 其底层支持为元对象设施 MOF (Meta-Object Facility)、XML 元数据交换 XMI (XML Metadata Interchange) 和公共库元模型 CWM (Common Warehouse Metamodel), 这些基础设施规定了在不同设计工具之间交换 UML 设计模型的标准。由美国瑞理公司开发的 Rational Rose 是当前比较流行的 UML 软件工具。

CORBA 是 OMG 最具影响力的规范集, 它保证了应用程序的可互操作性以

及对于硬件平台、操作系统、编程语言以及网络与通信协议的无关性。CORBA 规范中包含一系列单独的规范,例如 OMG 接口定义语言 IDL、网络通信协议 GIOP 和 IIOP、可移植对象适配器 POA、CORBA 组件模型 CCM 等。

CORBA 所基于的概念框架是对象管理体系结构 OMA, OMA 描述了一个较高抽象层次的分布式计算环境。

由 OMG 发布的 Object Management Architecture Guide 是关于 OMA 的正式规范,该指南描述了 OMG 的技术目标与相关术语,并为所有 CORBA 规范提供了概念性的基础设施。指南的核心内容是对象模型与参考模型,其中对象模型定义了对象外部可见特征的、独立于具体实现的语义,参考模型则标识与刻划了组成 OMA 的组件、接口与协议。

OMA 体系结构的核心组件是对象请求代理 ORB (Object Request Broker),它支持客户程序与对象实现在一个分布式环境中通信。ORB 仅提供了最基础的通信枢纽,在开发不同的实际应用程序时,对象还有许多共性可以提取与重用,OMG 将对象的这些共性依其基本性分别标准化为对象服务、公共设施与领域接口,应用程序中的对象接口则因其差异性较大而无法标准化。

OMA 体系结构中 ORB 组件与其他 4 类对象接口之间的关系如图 4-1 所示。对象服务是基于分布式对象的所有应用程序都可能用到的通用服务的接口。公共设施是可用于大多数应用领域的面向终端用户的工具接口,对象服务与公共设施的主要区别在于对象服务比公共设施更加基本。领域接口是与具体领域有关的接口,应用接口则是与应用领域有关的非标准化接口。

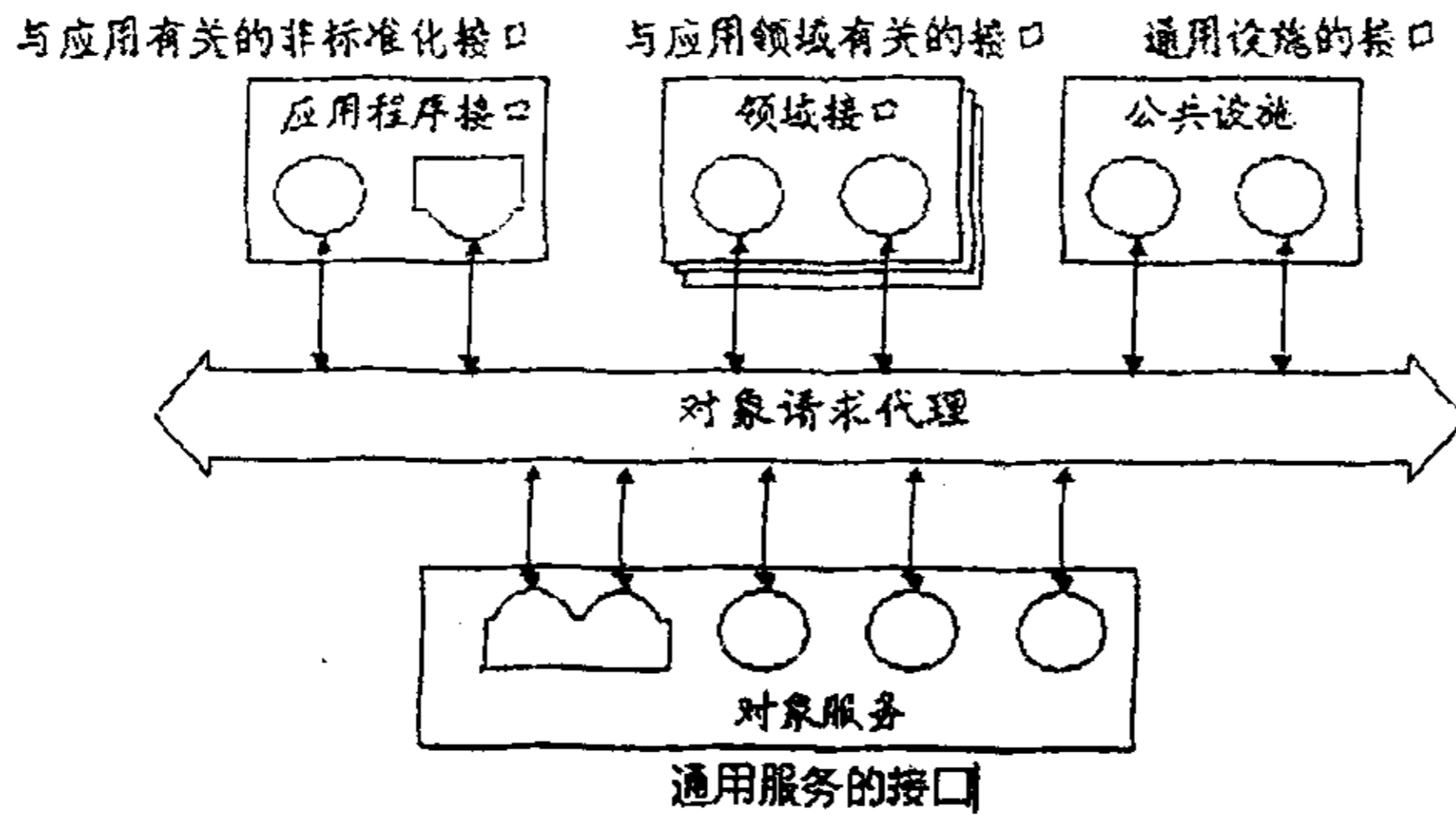


图 4-1 OMA 参考模型

Fig.4-1 The OMA reference model

从软件体系结构的角度看，OMA 的各个组件形成一种层次设计风格，位于最上层的是应用程序接口，往下依次为领域接口、公共设施和对象服务，最底层是对象请求代理组件。上层组件可跨层调用底层的组件，例如在应用程序中可直接调用公共设施和对象服务提供的功能。

### 1. 对象请求代理

对象请求代理是 OMA 参考模型的核心，它提供了分布式对象之间透明地发送请求或接收响应的基本机制，独立于实现对象的特定平台与技术。客户程序无需知道如何与对象通讯、如何激活对象、对象如何实现、如何查找对象等。ORB 是基于分布式对象构建应用程序的基础，并保证了在异类网络中对象的可移植性与可互操作性。

OMG 的接口定义语言 IDL (Interface Definition Language) 为定义 CORBA 对象的接口提供了一种统一标准，用 IDL 定义的对象接口是对象实现与客户程序之间的合约。IDL 是一种强类型的说明性语言，独立于任何程序设计语言。IDL 到程序设计语言的映射支持开发者选择自己的程序设计语言来实现对象和

发送请求。

OMG 发布的 CORBA: Common Object Request Broker Architecture and Specification 是关于 ORB 体系结构的规范, 定义了 ORB 组件的程序设计接口。OMG 发布的 CORBA languages 是一系列独立的语言映射规范, 包括 Java、C++、Smalltalk、Ada、C、COBOL 等语言。

## 2. 对象服务

对象服务是一些通用的服务, 这些服务要么是利用分布式对象开发基于 CORBA 的应用程序的基础, 要么为应用程序的可互操作性提供与具体应用领域无关的基础。对象服务是分布式应用程序的基本组成构件, 开发者可以用不同方式组合对象, 让它们在应用程序中发挥不同的作用。对象服务可用于构造具有可互操作的高层设施和应用程序的对象框架。

对象服务的本质是将覆盖对象整个生存期的对象管理任务标准化, 例如对象服务提供的功能包括了创建对象、对象访问控制、查找对象、维持对象间关系等。这种标准化可导致不同应用程序的一致性, 并提高软件开发者的生产率。

OMG 的对象服务统称 CORBA Services, OMG 发布的 CORBA Services: Common Object Services Specifications (简称 COSS) 是关于对象服务的一组规范, 其中包括对象命名、事件、生存期、持久对象、事务、并发控制、关系、外表化、许可机制、查询、属性、安全性、时间、对象收集、交易对象等服务的规范。

## 3. 公共设施

公共设施是可用于大多数应用领域的、面向终端用户的设施, 包括分布式文档设施、打印设施、数据库设施、电子邮件设施等。公共设施提供的一系列通用的应用程序功能可配置为特定的应用需求, 公共设施标准化使得通用操作具有统一性, 并且终端用户可方便地选择自己的配置。

OMG 的公共设施统称 CORBA Facilities, OMG 发布的 CORBA Facilities: Common Facilities Architectures 是描述公共设施体系结构的规范。与对象服务规范一样, 公共设施规范包括了一系列用 OMG IDL 表达的接口定义。



#### 4. 领域接口

领域接口是与应用领域有关的接口, 例如金融、医疗、制造业、电信、电子商务、运输等应用领域。图2-1中的领域接口表示为一组领域的接口, 暗示了开发者可按照不同的应用领域来组织领域接口。

OMG正是按不同应用领域组织与发布一系列领域接口规范, 例如OMG已发布了制造、医疗、金融、电信等行业的规范集CORBA Manufacturing、CORBA Med、CORBA Finance、CORBA Telecom 等。OMG正在进一步完善并将陆续推出新的领域接口规范。

#### 5. 应用程序接口

应用程序对象为终端用户执行特定的任务, 它不是OMG标准化的内容, 而是构成整个OMA参考模型的最上层元素。一个典型的应用程序由大量基本的对象类构建而成的, 其中部分对象与具体应用有关, 部分对象则来自领域接口、公共设施与对象服务。应用程序对象可通过继承机制重用现有的对象。

应用程序只需支持或使用与OMG一致的接口即可加入到OMA中, 这些程序本身未必要用面向对象风格来实现。图2-1的对象服务与应用程序接口展示了现有的非面向对象软件可以嵌入在一些对象包装器中, 从而融入OMA体系结构。

### 4. 4. 2 CORBA 体系结构

CORBA建立在OMG的对象模型基础之上, 主要有三个关键部分: 接口定义语言IDL、对象请求代理ORB和标准通信协议IIOP。

#### 1. OMG 的对象模型

OMG在Object Management Architecture Guide中定义的对象模型描述了对对象外部特征的标准语义, 其中对象、类型、操作、属性、对象实现等语义与Java、C++、Eiffel等面向对象程序设计语言十分相近。在该模型中, 客户程序通过一个由IDL书写的接口向服务对象提出服务请求。

请求是一个在特定时刻发生的事件, 它携带的信息包括操作、提供服务的目标对象引用、0个或多个实际参数以及一个可选的请求上下文 (request context) 等。对象引用是可以有效地指称一个对象的对象名字。请求上下文提供了可能影响请求执行的额外信息, 这些信息通常与操作有关。请求表 (request

form) 用于发送请求, 可多次求值与执行。请求表由IDL与特定语言的绑定来定义, 另一种形式的请求表通过调用动态调用接口DII创建一个调用结构, 往调用结构中添加参数后可发出调用。

在OMG的对象模型中, 对象可以被创建或撤销。但从客户程序的角度看, 并没有什么特别的机制用于创建或撤销对象, 对象创建与撤销只是发出请求的结果。客户程序通过对象引用指称新创建的对象。

## 2. 对象请求代理的结构

对象请求代理(ORB)是OMA的核心基础设施, CORBA规范规定了ORB的标准体系结构。ORB负责完成查找请求的对象实现、让对象实现准备好接收请求、传递构成请求的数据等任务所需的全部机制。客户程序所看到的对象接口完全独立于对象所处的位置、实现对象的程序设计语言以及对象接口中未反映的其他特性。

为调用远程对象实现的一个实例, 客户程序必须首先获取一个对象引用(以后我们将知道有多种方式可获取对象引用)。客户程序发出远程调用的方式与本地调用相似, 只不过调用的是远程对象实例的对象引用。ORB检查对象引用, 如何发现目标对象是远程的, 就将参数打包并通过网络传递给远程对象所在的ORB。

ORB提供的最基本功能是从客户程序向对象实现传递请求。在逻辑上ORB可理解为一个由ORB接口定义的服务集合, 但在物理上ORB通常不必实现为一个单独的组件(例如进程或程序库)。ORB内核(ORB Core)是ORB最关键的部分, 负责请求的通信设施, 每一个ORB产品供应商都有一个自己特有的ORB内核。图4-2展示了ORB体系结构的主要组成部分以及它们之间的关系。

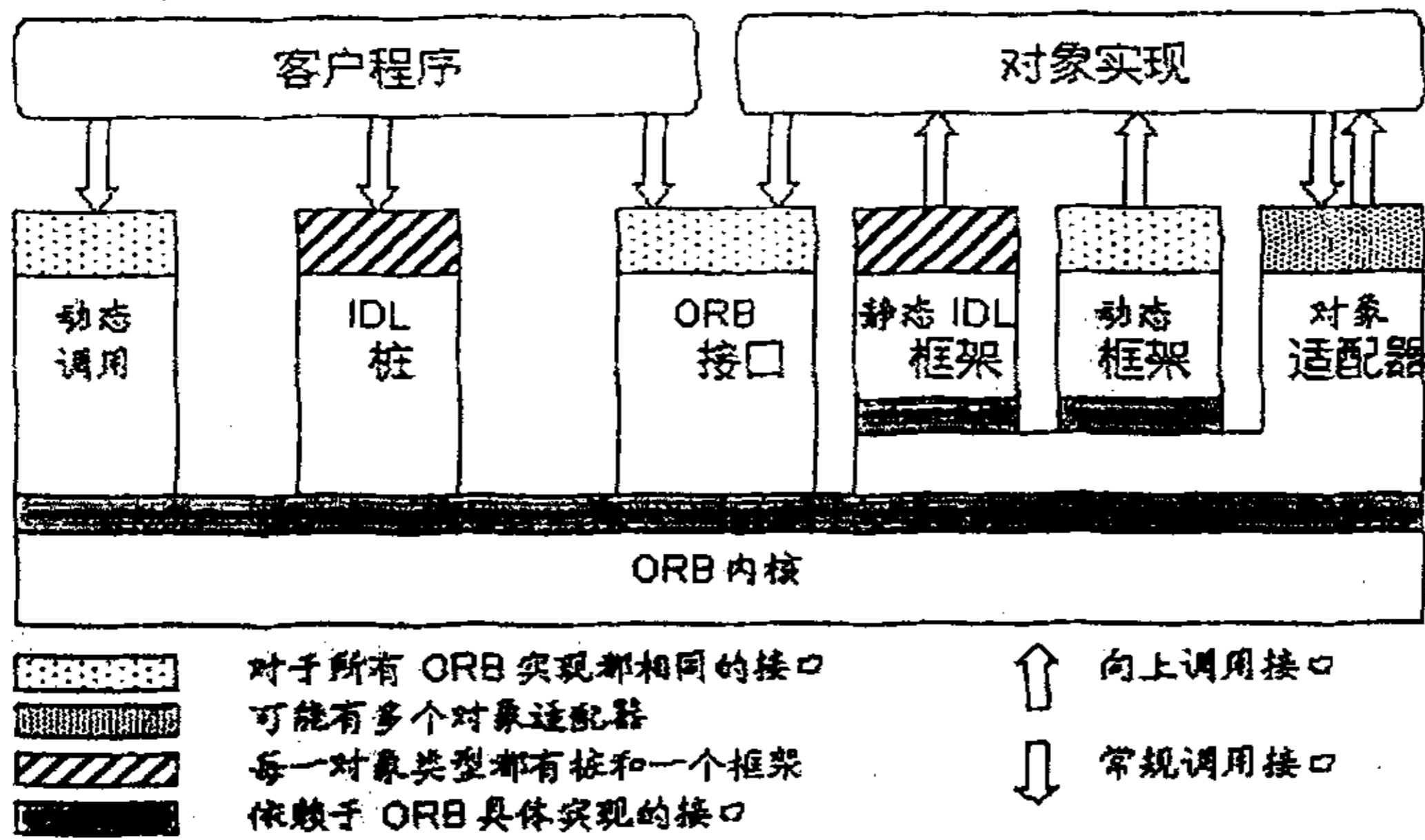


图 4-2 对象请求代理的体系结构

Fig.4-2 The infrastructure of ORB

图4-3展示了客户程序与对象实现如何有效地组织和利用对象接口与对象实现的有关信息。

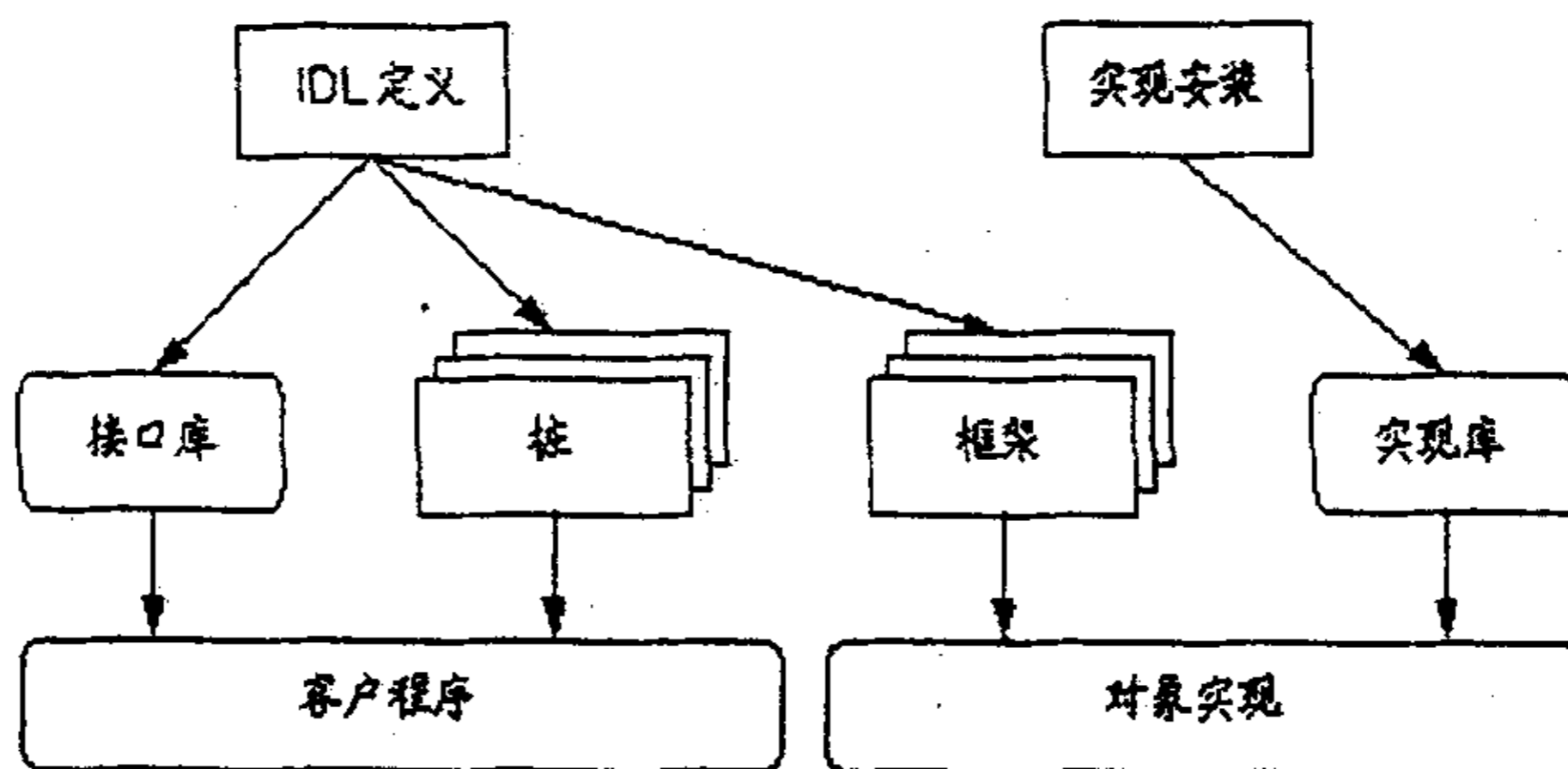


图 4-3 接口库与实现库

Fig. 4-3 The interface repository and implementation repository

### 3. 对象接口定义

客户程序是想要执行对象上的操作的实体，对象实现则是真正实现该对象的代码与数据。客户程序与对象实现之间的界面是对象的接口定义。

对象接口采用接口定义语言IDL (Interface Definition Language) 定义，IDL根据可执行的操作以及操作的参数来定义对象的类型，并可映射到特定的编程语言或对象系统。为能在运行时充分利用对象接口定义的有关信息，还可将对象接口定义添加到接口库 (Interface Repository) 服务中 (参见图4-3)，该服务将接口组件表示为对象，允许运行时动态访问这些组件。IDL与接口库具有同等的表达能力。

客户程序只能通过对象的接口定义掌握对象的逻辑结构，并通过发送请求来影响对象的行为与状态。客户程序不必了解对象实现的具体实现方式，也不必知道该对象实现采用哪个对象适配器以及需要用哪个ORB访问该对象实现。CORBA通过对象接口进一步延伸了传统程序设计语言的封装与信息隐藏概念。

对象实现可以用多种方式实现，例如独立的服务程序、程序库、每个方法的程序代码、被封装的应用程序、面向对象数据库等。通过使用附加的对象适

配器, ORB事实上可支持所有风格的对象实现。通常对象实现不依赖于ORB或客户程序调用对象的方式, 如果对象实现需要使用依赖于ORB的服务, 则需要通过选择合适的对象适配器获得服务的接口。对象实现信息在安装时提供, 并被存储在实现库 (Implementation Repository) 中 (参见图4-3), 供传送请求时使用。

#### 4. 客户程序发送请求

客户程序通过发送请求去调用由对象实现提供的服务。客户程序可通过静态调用或动态调用方式将请求发送给ORB内核, 然后由ORB内核将请求转发给对象实现。静态调用方式借助于客户程序桩完成, 动态调用方式则使用动态调用接口DII (Dynamic Invocation Interface)。从发送请求的功能上看, 这两种调用方式具有完全相同的能力 (即两者的调用语义相同), 对象实现并不知道请求从客户端是如何发出的。

通常ORB产品会提供一个IDL编译器, 在编译IDL文件时会创建客户程序桩与对象实现框架 (参见图4-3)。如果IDL文件进行了修改, 则必须重新用IDL编译器创建新的桩与框架。由于桩与框架在编译时创建并且在运行时不再改变, 因而这些接口相对于动态调用接口又称作静态调用接口SII (Static Invocation Interface)。静态调用方式与Java RMI类似。IDL桩负责客户程序的实现语言与ORB内核之间的映射, 因而只要ORB支持某种语言的映射, 客户程序就可选择该语言作为实现语言。使用SII的客户程序开发者必须在程序编译之前就知道操作的名字和所有参数与返回值的类型, 实际的操作名字、参数值和返回值是编写在应用程序的源代码中。

SII是一种最简单的调用机制, 所有使用SII的应用程序也可用DII实现, 但使用DII的应用程序则未必可用SII实现。选择这两种调用方式的基本原则是: 如果应用程序可以用SII实现, 那么就应使用SII, 这是因为DII需要更多的编码 (本质上是要由程序员自己完成桩的所有任务) 并且运行效率更低, 并且编译器无法帮助检查类型和优化代码。

DII允许客户程序调用在编译客户程序时尚未确定对象接口的对象实现。客户程序使用DII时必须生成一个请求, 其中包括对象引用、操作以及参数表。DII的这种动态特性使得DII在某些应用场合更优于SII, 例如编写CORBA服务的

浏览器、应用程序浏览器、转换协议的桥接、访问大量不同接口、应用程序的监控、通用对象测试程序等等。使用DII的应用程序访问对象实现提供的服务时，不必包含由IDL编译器生成的桩，只需在运行时访问ANY对象。当然，使用DII会比SII更麻烦，程序员必须用DII接口指定操作和每个参数的类型与值，并且由程序员自己利用CORBA定义的类型码 (type code) 作类型检查。DII类似于Java语言的自省 (introspection) 机制。

DII还提供了一种延迟同步调用，使得客户程序提交请求后不必等候答复。延迟同步调用与单向 (one-way) 操作相似，但延迟同步调用支持返回值和输出型参数 (这时必须进行轮询)。CORBA 3.0将通过异步消息服务 (Asynchronous Messaging Service) 同时支持静态的和动态的延迟同步调用。

### 5. 对象实现接收请求

ORB将请求分派给对象实现也有两种方式：静态方式通过由IDL生成的框架静，动态方式使用动态框架接口DSI (Dynamic Skeleton Interface)。ORB通过IDL框架或DSI查找合适的实现代码、传送参数，并将控制传给对象实现，对象实现执行请求时可通过对象适配器OA (Object Adapter) 获取ORB的某些服务，请求完成后控制与结果返回给客户程序。

对象实现与ORB内核之间的通信由对象适配器完成，对象适配器负责对象引用的生成与解释、方法调用、交互的安全性、对象实现的激活与冻结、将对象引用映射到相应的对象实现、对象实现的注册等。为满足特定系统的需要，供应商会提供不同的专用对象适配器。对象实现可选择使用哪种对象适配器，这取决于对象实现所需的服务。

ORB屏蔽了客户端发送请求与服务端接收请求的不同方式。从客户程序的角度看，使用DSI的对象实现与使用IDL框架的对象实现行为相同，客户程序不必提供特别的处理去与使用DSI的对象实现通信。对象实现框架的存在并不意味着一定要有客户程序桩，客户程序也可通过DII发送请求。

虽然DSI比较复杂且性能不高，但DSI的动态特性使得DSI在某些应用场合更优于静态分派请求，例如开发服务端的桥接 (协议转换器)、监控应用程序、解释性服务或由脚本驱动的服务等，下节我们还将看到DSI在实现对象的可互操作性时发挥的作用。使用DSI的应用程序提供服务时不必包含由IDL编译创建

的框架,因而这些服务支持更通用的请求,但需要更多的手工编程并且必须检查类型的安全性。

#### 4. 4. 3 对象的可互操作性

##### 1. 可互操作性

可互操作性 (interoperability) 是指在一个系统中用不同工具或不同供应商的产品开发出来的两个组件是否可以协调工作。当前有许多商品化和免费的 ORB 产品,每一种产品都试图满足它们的操作环境的特定需求,因而产生了不同 ORB 之间可互操作的需要。此外,有些分布式应用系统并不是 CORBA 兼容的,例如 DCE、DCOM 等,这些系统与 CORBA 的可互操作需求也在不断增长。

影响对象之间可互操作性的因素不仅仅是实现方面的差异,还包括安全性方面的严格限制或需要提供正在开发产品的受保护测试环境等原因。为了提供一个完全可互操作的环境,必须将这些差别都考虑进去。

这就是为什么 CORBA 2.0 引入了域 (domain) 这一高层概念。域支持开发人员根据自己的实现因素或管理原因将对象划分为不同的集合,不同域的对象之间需要桥接机制才可彼此交互。此外,桥接机制必须足够灵活,既可适用于协议翻译量少而传输性能比较重要的情况 (例如位于同一个 ORB 上两个不同的域),也可适用于性能不太重要但需要访问不同 ORB 的情况。

##### 2. ORB 域和桥接

实现可互操作性的方法通常可分为直接桥接和间接桥接两种方式。采用直接桥接方式时,需交互的元素直接转换为两个域的内部表示,这种桥接方式具有较快的传输速度,但在分布式计算中缺乏通用性。

采用间接桥接方式时,需交互的元素在域的内部表示形式与各个域一致认可的另一种表示形式之间相互转换,这种一致认可的中间表示形式要么是一种标准 (例如 OMG 的 IIOP),要么是双方的私下协议。如果中间表示是某一运行环境的内部表示 (如 TCP/IP),则称为全桥 (full bridge);否则如果一个 ORB 运行环境不同于公共协议,则称该 ORB 为半桥 (half bridge)。

桥接既可在一个 ORB 的内部实现 (例如只是连接两个管理领域的边界),也可在更高层次实现。在 ORB 内部实现的桥接称为内联桥 (in-line bridge),否则

称为请求层桥 (request-level bridge)。实现内联桥时,既可要求ORB提供某种附加的服务,也可引入额外的桩和框架代码。

请求层桥的工作方式大致如下:客户程序的ORB将桥与服务程序的ORB看作对象实现的一部分,并通过DSI向该对象发送请求(注意DSI无需在编译时知道对象的规格说明);DSI与桥协作,将请求转换为服务程序的ORB能够理解的形式,并通过服务程序的ORB的DII调用被转换的结果;如果请求有返回结果,也是通过类似的路径返回。实际上桥为了完成其功能不得不了解对象的有关信息,因而要么必须访问接口库,要么只能是一个与特定接口有关的桥。

### 3. GIOP、IIOP 与 ESIOP

为了让桥能正常工作,还有必要制订传输请求的统一标准,规定传输底层的数据表示方法与消息格式,由OMG定义的通用ORB间协议GIOP (General Inter-ORB Protocol)负责完成这一功能。GIOP专门用于满足ORB与ORB之间交互的需要,并设计成可在任何传输协议的上层工作,只要这些协议满足最小的假设集。当然用不同传输协议实现的GIOP版本不必直接兼容,但它们能够交互将更加有效。

除了定义通用的传输语法外,OMG还规定了如何以TCP/IP协议为基础实现GIOP协议,这一更具体的标准称为因特网ORB间协议IIOP (Internet Inter-ORB Protocol)。GIOP与IIOP之间的关系就好象IDL与Java、C++等具体语言之间的映射关系。由于TCP/IP是独立于供应商的最流行传输协议,IIOP为ORB提供了开放式的(OMG术语为out of the box)可互操作性。此外,IIOP还可作为半桥的中间层,除了提供可互操作性的功能之外,供应商还可用于ORB间的消息传递。

IIOP是ORB之间的通信协议,虽然也可用于实现ORB内部的消息传递,但并不是CORBA的硬性规定。一个特写的ORB产品可能支持多种通信协议,但声称与CORBA 2.0兼容的ORB产品至少必须支持IIOP。

CORBA规范还提供了一套特定环境ORB间协议ESIOP (Environment-Specific Inter-ORB Protocols),这些协议可用于特定环境(诸如DCE、DCOM、无线网络等系统)的可互操作性。与IIOP相比,ESIOP可针对于特定环境进行优化。



#### 4. CORBA 对可互操作性的支持

CORBA 的目标是支持多个层次的可互操作性, CORBA 规范经过多次改进与发展才达到这一目标。早期的 CORBA 版本强调不同平台与语言之间的可互操作性, 包括了 IDL 标准以及 IDL 到程序设计语言的映射。使用同一供应商的 ORB 产品开发的客户程序与服务程序之间可以交互, 但使用不同供应商的 ORB 产品开发的客户程序与服务程序则未必是可互操作的。

CORBA 2.0 版引入 GIOP 和 IIOP, 从而实现了不同供应商的 ORB 产品之间的可互操作性, 所有供应商的 ORB 产品如果与 CORBA 2.0 兼容则彼此之间可互操作。

更完善的可互操作性还应包括不同服务之间是可互操作的, 例如一个 CORBA 对象可通过协议桥接操作一个 DCOM 对象。但 CORBA 的事务服务能否与 Microsoft 的事务服务进行交互, 从而为不同系统提供一个无缝的事务? 这类在服务层次的更广义可互操作性是 CORBA 规范尚未解决的问题之一。

#### 4. 4. 4 CORBA 规范与 CORBA 产品

OMG 本身不生产任何软件或实现任何规范, 它只是将 OMG 成员的信息需求 (RFI) 与建议需求 (RFP) 汇集为规范。CORBA 规范是一套开放式的规范, OMG 的成员或非成员公司均可免费实现符合 CORBA 规范的 ORB 产品。

##### 1. CORBA 规范

CORBA 这一名词既用于专指关于 ORB 体系结构的规范, 也泛指 OMG 基于 OMA 参考模型发布的一系列规范集。由于 OMG 需要不断改进与完善 CORBA 体系结构, 因而为 CORBA 规范编制了完整的版本号。仅当对体系结构作重大改变时, OMG 才增加 CORBA 规范的主版本号, 所以有时也以主版本号代指 CORBA 的新特征, 例如 CORBA 2 指 ORB 之间的可互操作性和基于 TCP/IP 协议的 IIOP, 而即将正式发布的 CORBA 3 则指 CORBA 的组件模型。

##### 2. CORBA 产品概述

尽管 OMG 不断改进与完善 CORBA 规范, 但每一版本保持了较好的向后兼容性, 因而 CORBA 规范相当成熟与稳定, 并且拥有大量产品, 在企业计算与因特网计算领域拥有庞大的市场。基于 CORBA 的软件适用于因特网应用与

企业计算, 特殊版本的 CORBA 还可运行在实时系统、嵌入式系统与容错系统。

CORBA 规范并未约束 ORB 产品的实现方式, 因而 ORB 可驻留在客户程序和对象实现中, 也可将 ORB 作为一个服务程序, 或者将 ORB 作为操作系统提供的基本服务。一个特定的 ORB 产品可能支持上述多种方式, 并可能支持多种通信协议。

由于 CORBA 规范独立于软件供应商, 在不同供应商的 ORB 产品上运行的程序之间具有较好的可互操作性。当前市场上有许多 CORBA 产品可供用户选择, 其中一些是商品化的, 一些是免费的, 许多商品化商品也提供了免费的试用期。但这些产品在对 CORBA 规范的支持程度、对 CORBA 对象服务的支持程度、其他附加特征等方面有很大差异。

下面简要介绍以下本系统所选用的 CORBA 产品 VisiBroker。

由 Visigenic 和 Borland 合并而成的 Inprise 公司是目前商品化 ORB 产品的主要供应商之一, 其产品 VisiBroker 支持 Java 和 C++ 语言, 可在 Windows 95/98/NT/2000/XP、Solaris、RedHat Linux、HP-UX 和 AIX 平台运行。最新推出的 4.0 版完全遵循 CORBA 2.3 规范, 支持可移植对象适配器、传值对象、IIOP 上的 RMI、具有容错特性的命名服务、属性管理、服务质量、拦截器等, 并扩充了位置服务、对象包装器等 VisiBroker 专有特性。

为迎合企业开发大规模、高性能的基于 Web 分布式应用系统的需要, 许多软件供应商在 ORB 产品的基础上为开发者提供了更高层次的开发平台——应用服务器。应用服务器支持企业的业务逻辑与表示逻辑分离, 这种分离可显著提高企业应用系统的容错性、可伸缩性与性能。

业务逻辑通常由 EJB 开发, 并且运行在一个 EJB 容器中。EJB 容器负责安全性、事务策略、对象生存期、线程以及许多其他问题, 这些问题实际上就是 CORBA 中的对象服务和公共设施, 解决这类问题可让开发者将精力集中在开发业务逻辑自身。

表示逻辑通常由 Servlet 或 JSP (Java Server Pages) 执行。Servlet 采用 Java 语言编码, 负责在 Web 服务器中将业务逻辑转换为表示逻辑, 返回具有动态内容的 HTML 文档。JSP 与 Servlet 类似, 可用于返回具有动态内容的 HTML 文档, 但 JSP 是由嵌入在 HTML 文档中的 Java 代码片段产生, 并且自动被编译

为 Servlet。由于大多数动态文档的核心内容是 HTML, Java 代码只是其中的一小部分, 因而 JSP 比 Servlet 更具吸引力。

Inprise 公司的应用服务器 IAS(Inprise Application Server)建立在 VisiBroker for Java 之上, 提供了 VisiBroker 的所有机制。IAS 的主要组成部分有: 支持静态 HTML 文档和表格的 Web 服务器、运行 Servlet 的引擎、JSP 编译器和引擎、EJB 容器、基于控制台或浏览器的配置与管理工具、以及用于存储应用程序数据的纯 Java 关系数据库 JDataStore。

VisiBroker 因其短小精悍的特点还嵌入在其他产品之中, 例如 Netscape Communicator 浏览器嵌入了 VisiBroker 产品后, Applet 可向 CORBA 对象发出请求, 这时不必再下载相关的 ORB 类。

#### 4. 4. 5 CORBA 小结

##### 1. CORBA 为我们带来什么

OMG 致力于解决分布式应用程序的复杂性与高成本问题。CORBA 顺应软件技术发展的潮流, 成功地融合了两种技术: 一是基于消息传递的分布式客户机 / 服务器技术, 一是面向对象软件开发技术。CORBA 采用面向对象方法创建在应用程序之间可重用与可共享的软件组件, 每一个对象对外隐藏了它内部工作的细节, 并提供一个定义良好的外部接口, 从而降低应用程序的复杂性。一旦实现并测试一个对象后, 它可以在新应用环境中多次被重用, 因而减少了开发应用程序的成本。

CORBA 的平台无关性很适合用于集成企业内的异类计算机系统, 包括不同供应商的不同硬件平台、操作系统、网络系统、程序设计语言或其他特性。CORBA 应用程序客户可运行在小至到手持无线设备或嵌入式系统, 大至大型计算机的平台。CORBA 支持多种现有的程序设计语言, 并且支持在单个分布式应用程序中混合使用这些语言。

CORBA 的最大特点是提供了在异类分布式环境中对象之间高度的可互操作性, 从而保证了建立在不同 CORBA 产品之上的分布式对象可互相通信。大型企业不必强制规定所有开发项目使用单一的 CORBA 产品。

CORBA 提供了一个工业标准而不是一个软件产品, 不同供应商的竞争保证了有高质量的、完全遵循 CORBA 的产品。使用 CORBA 标准也为开发者的实

现提供了一定程度的可移植性 (注意应用程序在不同 CORBA 产品之间并不是 100%可移植)。

CORBA 的一个典型应用是开发需要有效处理大量客户请求的服务程序, CORBA 的可伸缩性与容错性使得许多大型网站后台都应用了 CORBA 技术。

虽然 CORBA 规范只支持详细设计与编码阶段的任务, 在编码的前端 CORBA 可得到 OMG 的 UML 规范的支持。

此外, 超过 800 家企业支持 CORBA, 包括硬件公司、软件公司、电报公司、电话公司、大型银行等, 这使得 CORBA 具有良好的市场前景。

## 2. 未解决的问题

影响分布式计算复杂性的因素有些是必然的, 例如必须考虑延迟时间带来的影响, 必须检测与恢复网络与主机的部分失败, 需要处理负载均衡和服务划分, 需要考虑分布式事件时序的随机性, 要求有高度安全性等等, 这此必然因素分布式计算领域中最基本的问题。此外, 有些因工具或技术手段等原因造成的限制增加了分布式计算的复杂性, 这些偶然因素可能包括缺乏类型安全的、可移植的、可扩展的系统调用接口或组件库, 缺少测试工具的支持, 大量使用基于功能或算法的分解, 没有统一的规范标准等等。

显然 CORBA 帮我们解决了不少问题, 如问题分解、容错性、安全性等方面, 但在标准 CORBA 中仍有许多问题尚未解决, 例如时间延迟、随机时序、死锁、负载均衡、资源控制等。CORBA 规范不断发展就是为了更好地解决这些问题, 但在此之前我们必须在实际应用中自己考虑并动手解决这些问题。

## 4. 5 CORBA 程序的开发

虽然 OMG 为 CORBA 制订了统一的规范, 但规范中也赋予了软件供应商实现 ORB 产品时自由选择各自不同的实现途径的权利, 例如 ORB 可以是一个独立运行的守护进程, 也可以嵌入到客户程序和对象实现中, 有些 ORB 产品则选择了几种实现方式的组合。所以不同供应商提供的 ORB 产品在具体使用方法上可能存在较大差异。

尽管使用不同 ORB 产品的具体操作差异较大, 但程序员开发一个 CORBA 应用程序通常会遵循一定的框架, 即首先通过面向对象分析与设计过程标识应

用程序所需的对象,包括对象的属性、行为与约束等特性,然后遵循本小节所述的几个开发步骤完成应用程序的开发、部署与运行,如图 4.1 所示,图中的箭头表示了任务之间的先后次序。

图 4-4 仅示意了一个典型的 CORBA 应用程序开发过程,具体实施时各个步骤会由于不同项目的各自特点而有所区别。例如将现有的管理信息系统(MIS)利用 CORBA 技术集成时,是先有对象实现后再重新用 IDL 编写对象的规格说明(即对象接口定义),这一过程有时还可利用 java2idl 这类开发工具自动完成。

#### 4. 5. 1 编写对象接口

对象接口实际上是服务对象的规格说明,更严格地说,是关于服务对象语法的规格说明,服务对象的语义则借助于自然语言描述或隐含在对象实现的代码之中。对象接口定义是客户服务与对象实现之间的一种合约,即双方都必须严格遵守接口定义中的约定,包括提供或使用的服务的名字、参数、返回值、异常、上下文环境等,ORB 正是基于这些合约才能够保证客户方与服务方的顺利通信。

每一个提供服务的对象都需要利用 OMG 统一制订的接口定义语言(IDL)编写规格说明。根据面向对象分析和设计的结果,例如 OMG UML 所表达的设计模型,很容易写出相应的 IDL 接口定义。在 IDL 文件中指明了每一个对象将提供的操作,以及客户程序如何调用这些操作或处理操作的返回结果。本书第四章详细介绍了 IDL 的语法与语义。

#### 4. 5. 2 编译 IDL 文件

IDL 是一种独立于具体程序设计语言的说明性语言,IDL 编译器的作用是将 IDL 映射到具体程序设计语言,产生客户程序使用的桩代码以及编写对象实现所需的框架代码。由 OMG 制订的语言映射规范允许将 IDL 语言映射到 Java、C++、Ada、C、COBOL 等多种程序设计语言,这通常是由软件供应商提供的不同编译器分别完成的。

VisiBroker for Java 4.0 提供的 idl2java 编译器(不同 ORB 产品的 IDL 编译器名字可能不同,例如 Java 2 SDK 提供的 IDL 编译器名为 idltojava)将 IDL 映射到 Java 语言,生成 Java 语言的客户端桩代码以及服务端框架代码,桩和框

架可看作分别是服务对象在客户端和服务端的代理。IDL 文件严格地定义了客户程序与对象实现之间的接口,因而客户端的桩可以与服务端的框架协调工作,即使这两端是用不同的程序设计语言编译,或运行在不同供应商的 ORB 产品之上。

每一个对象实例都有一个用于唯一标识自身的对象引用。客户程序利用对象引用指明调用的方向,即向 ORB 表明真正想调用的对象实例。表面上看,客户程序调用的是对象实现上的操作,但实际上调用的是作为代理的 IDL 桩。调用经过客户端的桩传递给 ORB,然后转发到服务端的框架,最后到达真正要执行的对象。

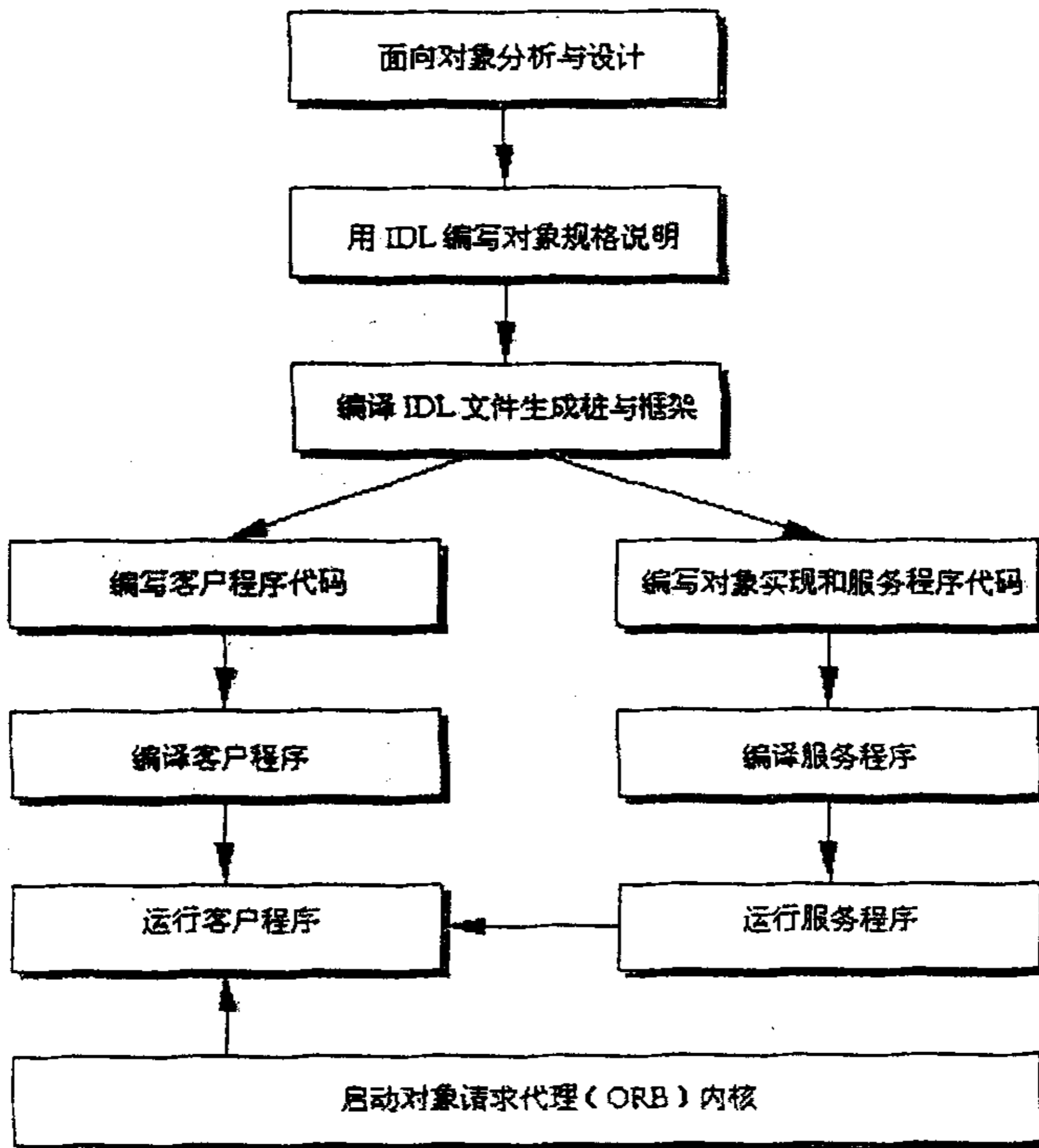


图 4-4 典型的 CORBA 应用程序开发过程

Fig. 4-4 The typical development process of CORBA application

#### 4. 5. 3 编写客户程序

初始化 ORB 是应用程序进入 CORBA 世界的起点, 无论是编写客户程序还是编写服务程序, 都必须在利用 ORB 进行通信之前先将 ORB 初始化。初始化 ORB 的作用一方面是让 ORB 了解有新的成员加入, 另一方面是获取 ORB 伪对象的引用, 以备将来调用 ORB 内核提供的操作。

所谓伪对象专指在 CORBA 基础设施中的一个对象 (例如 ORB 本身可看作一个伪对象), 这同时意味着伪对象是本地的, 从而将远程访问的 CORBA 对象区别开来。引入伪对象的概念后, 我们可以用一种统一的观点看待所有对象。在 CORBA 规范中, 伪对象也使用 IDL 定义, 不过这时我们称之为伪 IDL (Pseudo IDL)。

ORB 内核提供了一些不依赖于任何对象适配器的操作, 这些操作可由客户程序或对象实现调用, 包括获取初始引用的操作、动态调用相关的操作、生成类型码的操作、线程和策略相关的操作等。初始化 ORB 即是其中的一个操作。

然后客户程序需要绑定到服务对象, 利用这些对象提供的服务完成程序的功能。在典型的企业应用中, 对象实现执行企业的业务逻辑, 而客户程序则在业务逻辑的基础实现表示逻辑。

#### 4. 5. 4 编写对象实现和服务程序代码

IDL 文件只定义了服务对象的规格说明, 程序员必须另外编写服务对象的具体实现。CORBA 规定所有对象接口定义必须统一用 IDL 书写, 但对象实现则有很多选择的余地, 例如可使用 Java、C++、C、Smalltalk 等程序设计语言, 并且选择这些语言与客户程序所选用的语言无关, 只要 ORB 产品供应商支持 IDL 到这些语言的映射即可。

选用任何一门程序设计语言的程序员应该熟悉 IDL 到该语言的映射规则。例如 VisiBroker for Java 的 IDL 编译器将自动生成一些对象适配器的 Java 类和各种辅助性的 Java 类, 编写对象实现的代码时必须继承其中的一些类或使用某些类提供的方法。

编写对象实现代码后, 还必须实现服务程序的 main 例程, 由服务程序利用可移植对象适配器 (POA) 激活伺服对象供客户程序使用。服务程序通常是一个循环执行的进程, 不断监听客户程序请求并为之服务。



#### 4. 5. 5 创建并部署应用程序

创建客户程序时, 应将程序员编写的客户程序代码与 IDL 编译器自动生成的客户程序桩代码一起编译; 创建服务程序时, 应将程序员编写的对象实现代码与 IDL 编译器自动生成的服务程序框架代码一起编译。一些 ORB 产品提供了专门的编译器以简化这一过程, 例如 VisiBroker for Java 提供的编译器 vbjc 会自动调用 JDK 中的 Java 编译器 javac, 指示 javac 在编译客户程序的同时编译相关的客户程序桩文件, 在编译服务程序的同时编译相关的服务程序框架文件。

程序员创建的客户程序和服务应用程序已通过测试并准备投入运行后, 进入应用程序的部署 (deployment) 阶段。在该阶段由系统管理员规划如何在终端用户的桌面系统安装客户程序, 或在服务器一类的机器上安装服务程序。本书第七章介绍了在部署阶段需要考虑的若干问题。

#### 4. 5. 6 运行应用程序

运行 CORBA 应用程序时, 必须首先启动服务程序, 然后才可运行客户程序。其他步骤可能与具体 ORB 产品有关, 例如 VisiBroker for Java 的 ORB 内核是一个名为 osagent 的独立运行进程 (又称智能代理, smart agent), 可以在启动服务程序之后才启动 osagent, 但必须在运行客户程序之前让 osagent 启动完毕。

## 第五章 计算机矢量图像技术

### 5.1 矢量图形

矢量图形(Vector Graphics)的特点是: 绘画程序中物体定位、形体构造建立在以数学方式记录构件(图形元素)的几何性质上, 例如直线、曲线、圆形、方形的形状和大小。它不是记录象素的数量, 在任何解析度下输出时都同样清晰。矢量格式更适合于以线条物体定位为主的绘制, 通常用于计算机辅助设计(CAD)和工艺美术设计、插图等。使用物体定位绘画程序可以把特定物体作为一组、单独改变线条的长度、放大或缩小原形、移动和重叠。但是在屏幕上显示的时候, 由于监视器的特点, 矢量图也是以象素方式来显示的。

矢量图形是用一组指令集合来描述图形的内容, 这些指令用来描述构成该图形的所有直线、圆、圆弧、矩形、曲线等图元的位置、维数和形状。在屏幕上显示矢量图形要有专门软件将描述图形的指令转换成在屏幕上显示的形状和颜色。用于产生和编辑矢量图形的程序通常称为 Draw 程序。这种程序可以产生和操作矢量图形的各个成分, 并对矢量图形进行移动、缩放、旋转和扭曲等变换; 使用矢量图形的一个很大的优点就是容易进行这类变换。但是, 用矢量图形格式表示复杂图像(如人物或风景照片)的开销太大, 因此矢量图形主要用于表示线框型的图画、工程制图、美术字等。绝大多数 CAD 和 3D 造型软件使用矢量图形作为基本的图形存储格式。

矢量图的优点也就在于它在任何解析度下输出时都同样清晰。

### 5.2 本系统矢量图形的设计模式

根据电网元器件的特点, 设计了一种实用价值很高并容易实现的平面矢量图形模式。下面以电阻为例, 见图 5-1 所示。

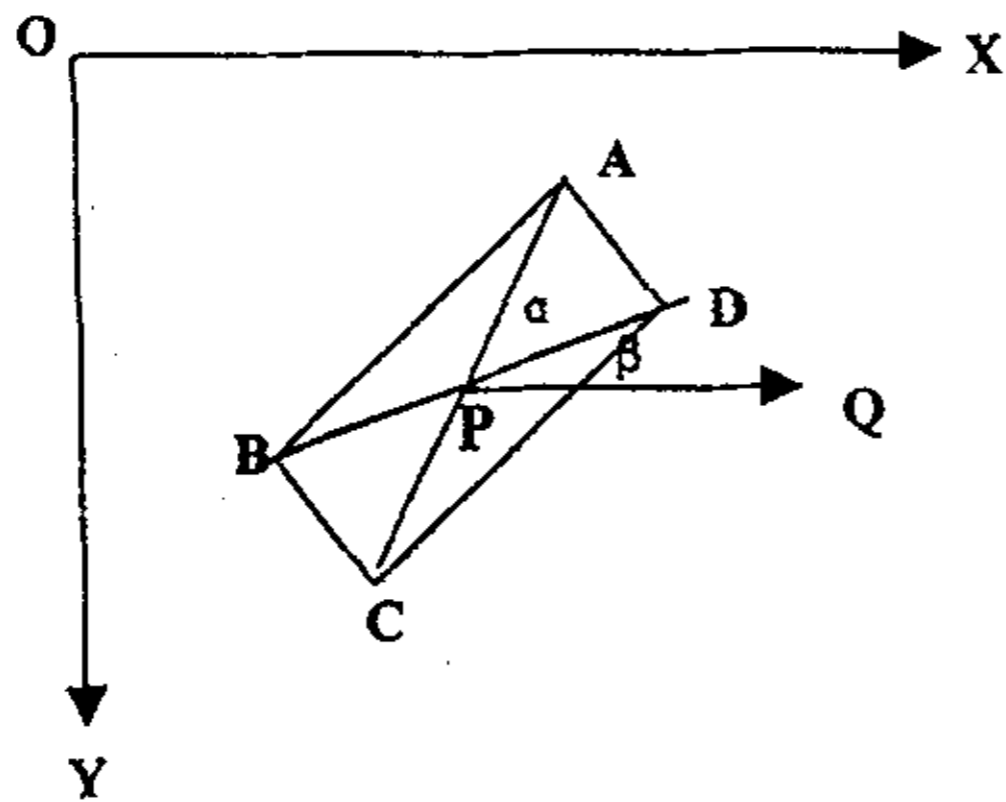


图 5-1 矢量图形设计图

Fig.5-1 The design diagram of vector graphics

矩形框 ABCD 是所有的元器件的最小外接矩形, P 是矩形的中点, P 点的横坐标为  $P_x$ , P 点的纵坐标为  $P_y$ , 已知 P 点坐标和矩形的长 AB 和高 BC 以及围绕中心点 P 旋转的角度  $\beta = \angle DPQ$ , 就可以对此矩形进行任意的缩放和旋转了。以下是得到 ABCD 的直角坐标的方法:

令  $AB = W, BC = H, PD = r$

$$r = \frac{\sqrt{W^2 + H^2}}{2} \quad (5-1)$$

$$\alpha = 2\arctg(H/W) \quad (5-2)$$

在 P 点为中心, PQ 为轴的极坐标中, A 的极坐标为  $(r, \alpha + \beta)$  所以 A 点的直角坐标为:

$$\begin{aligned} A_x &= P_x + r \cos(\alpha + \beta) \\ A_y &= P_y - r \sin(\alpha + \beta) \end{aligned} \quad (5-3)$$

其余 B、C、D 同理可推得。然后使用标准的绘图函数即可。

## 第六章 电力系统图形化计算系统设计

### 6.1 系统的体系结构

我们对系统的需求做了比较完备的分析，最后提出了基于 Web 的分布式图形化计算系统是电力系统专业计算系统目前最好的结构，主要是从以下几个方面来考虑的。图 6-1 显示了系统的体系结构。

首先，根据系统要求更多的人使用的特点，我们使用浏览器实现客户端，这样不仅满足了多用户的要求，而且实现了客户端的“瘦客户”型，便于软件的发布。用户可以通过 IE(Internet Explore)或者 Netscape 直接在浏览器中对电网系统进行编辑和浏览。

其次，要求对各个省局之间的计算系统进行互联。由于各个省局维护着不同的系统（如 Windows NT, Unix, Linux 等），所以我们采用了跨平台的 JAVA 和分布式计算系统 CORBA 来构架我们的系统。我们建立一个 CORBA 的计算中心，它负责与各个计算中心进行通信，包括发送数据给不同的计算中心，然后启动计算，最后得到结果。每个计算中心都可以部署自己的计算模块，它们通过 JAVA 程序进行外包，随时可以被 CORBA 计算中心唤醒进行计算。

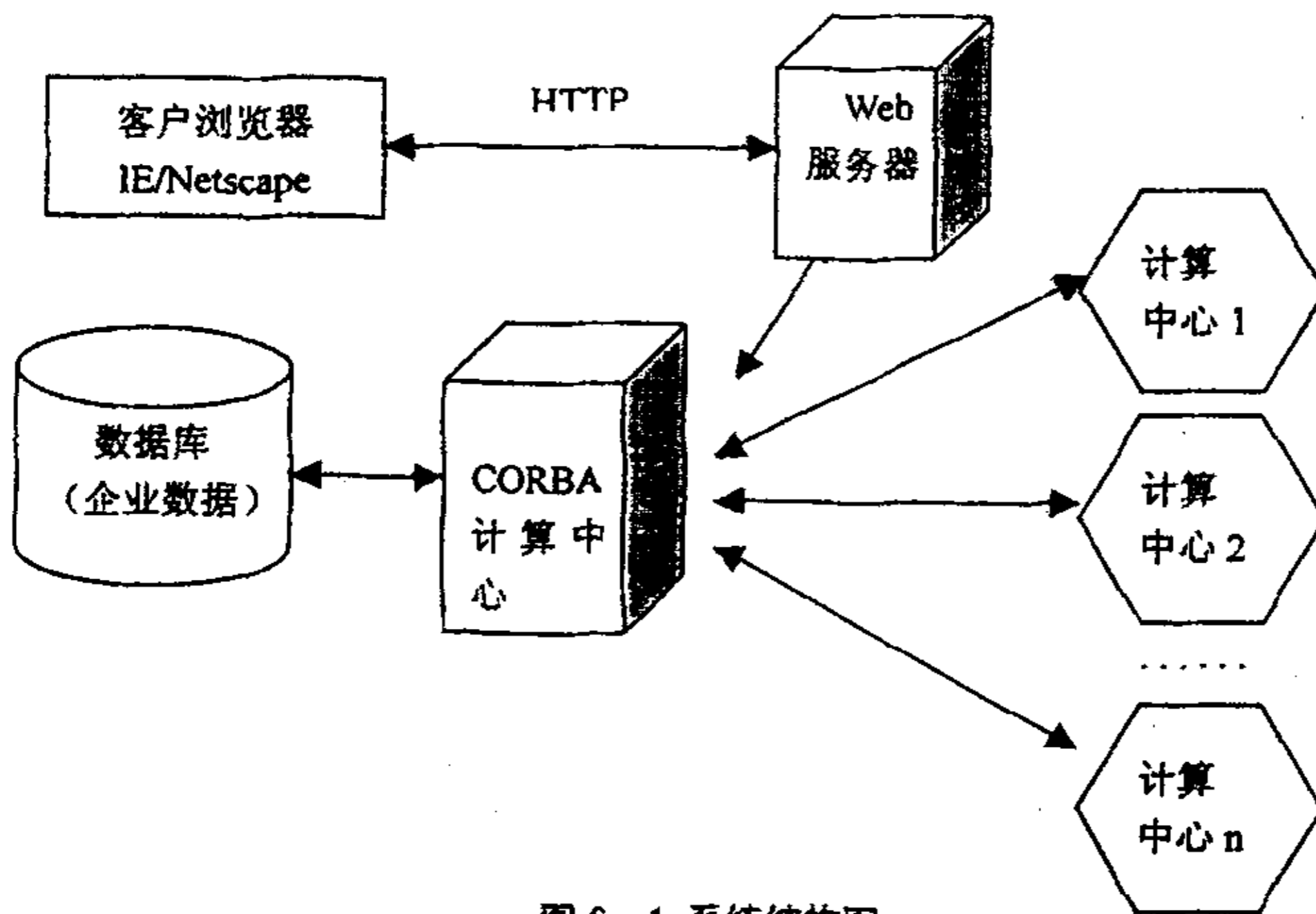


图 6-1 系统结构图

Fig. 6-1 The system infrastructure

最后，此系统要求替代传统的单机版的电力系统专业计算软件，所以对其界面的友好性和容易操作性都提出了比较高的要求。在这方面 JAVA 不仅在跨平台方面表现出了强大的生命力，而且在界面显示和图形操作方面也具有强大的功能。所以 JAVA 是本系统主要的开发语言。

### 6. 2 系统的流程

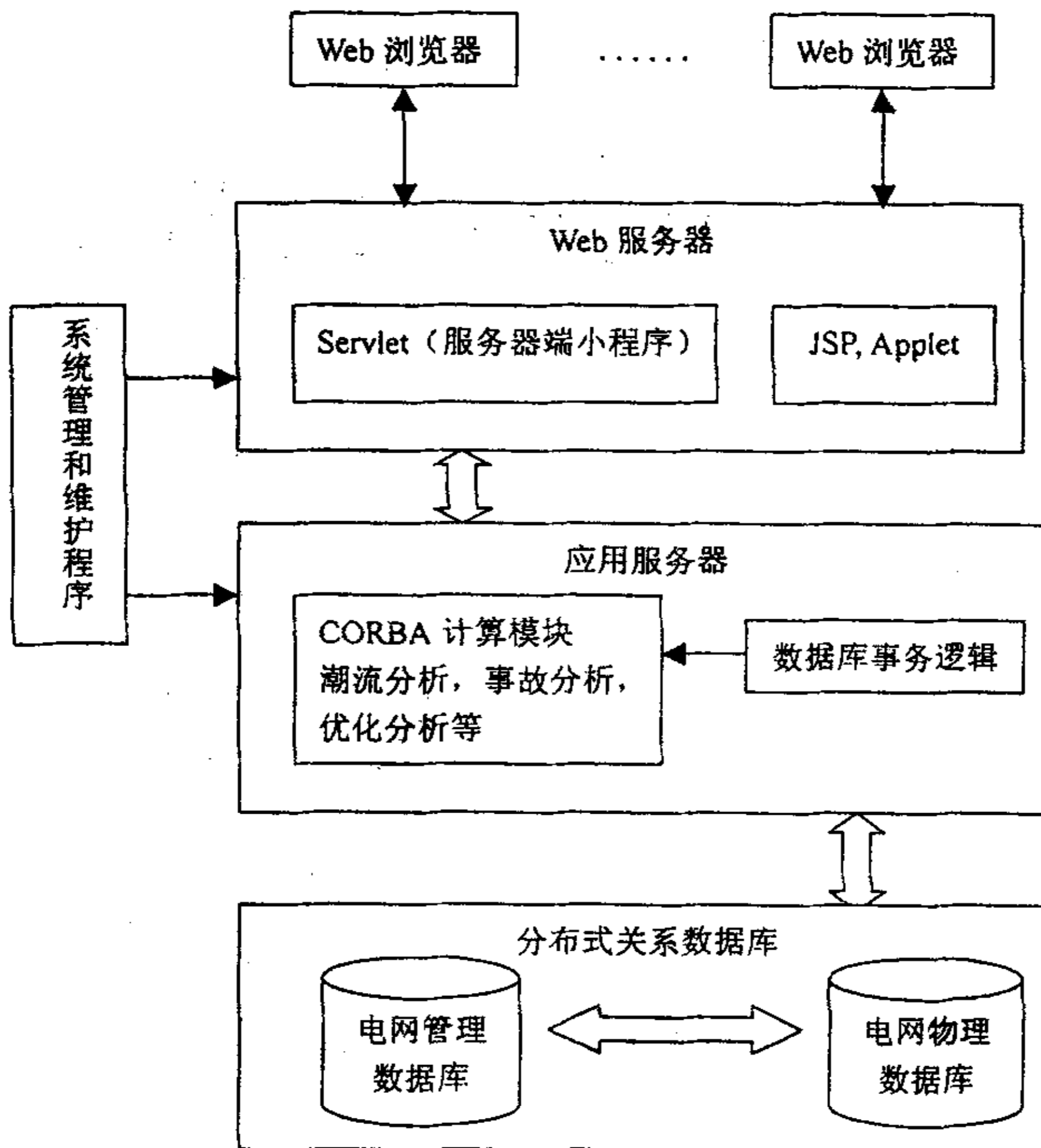


图 6-2 系统结构

Fig.6-2 The multi-tier architecture of the computing system

由图 6-2 可以看出本系统的具体设计采取了多层的分布式结构, 采用了 JAVA 和 CORBA 的结合。客户端和服务端程序用 JAVA 实现, 系统采用的 JAVA 类库均符合 JDK1.0 (Java Development Kit) 版本的要求, 其余所需的类均为自己编写。浏览器 IE5.0 以上版本自带有 JVM1.0 (Java Virtual Machine) 完全可以运行本系统, 而不需要额外的类库, 真正实现了跨平台。客户端和服务端之间的数据交换用 Servlet 进行。服务器端的计算用 CORBA 连接。数据存储采用集中存储在固定的数据服务器上。计算服务器存取数据通过标准的事务层进行。把数据库的绑定、数据操作和数据显示独立开来, 使得系统更容易向其他的大型数据库 (如 ORACLE, SYBASE, INFORMIX, DB2 等) 进行移植。

用户只需要一台接入 Internet 的计算机, 通过 Web 浏览器, 登陆指定的服务器网页, 进行身份验证后, 便可以下载 Java Applet, 在上面绘制电网图和调出已存在的电网图进行查看、修改、计算等。

### 6.3 Web 客户终端设计

客户终端采用了 Java Applet 技术, 在 Web 浏览器上实现。利用 JAVA 的完全面向对象功能, 建立起强大的可以自由扩展的图形输入输出终端。主要包括了电网元器件图形输入模块和数据动画输出模块。

#### 6.3.1 2D 矢量图形输入终端的设计

输入模块封装了国际电子电工委员会标准的绝大多数元器件图形库。所有的元器件都是矢量图形, 此图形的设计模式按照第五章的矢量图形设计模式设计。可以任意角度的放大、缩小、旋转和改变颜色, 满足各种复杂电网图的需求。在绘制图形的同时, 要求用户输入元件数据, 并同时将数据传输到服务器端的数据库中。

所有的图形基类为 BaseGraphProp, BaseGraphProp 类中封装了所有图元对象公有的一些属性和必要的接口, 如图元的位置属性和一些公有的物理属性和通用的函数接口。图元类都是 BaseGraphProp 的子类, 在子类中封装了它所独有的一些特性, 如此图元代表物理元件的物理属性以及这些图元对象所对应的特殊操作。下面以输电线类 LineGraphProp 为例, 列出具体的设计模式和代码实现。输电线类的设计见图-6-3 所示。

```
/**类功能描述: 所有图元对象的基类定义和实现
```

```
* 入口参数:
```

```
* 出口参数:
```

```
* 版本: 2.0
```

```
*/
```

```
public class BaseGraphProp implements Serializable{
```

```
    public byte m_yType;    //标明元件的种类.
```

```
    public int m_nPx;    //左上角横坐标
```

```
    public int m_nPy;    //左上角纵坐标
```

```
    public int m_nWidth;    //宽度
```

```
    public int m_nHeight;    //高度
```

```
public int m_nRed; //边框颜色
public int m_nGreen; //边框颜色
public int m_nBlue; //边框颜色
public byte m_yDrawWidth; //线条的宽度
public Boolean m_bColse; //开关属性
public double m_dAngle; //图形的旋转角度
public String m_str; //对象对应的注释
public int m_nNO; //记录所有元件的序号
public int m_nSelfNo; //对输电线和母线有用, 单独计数从 1 开始
private static int sdRadius = 7; //亮显标志的半径
```

```
public int getRadius(){
    return sdRadius;
}
```

//Serializable 接口中声明的方法, 是对象可以序列化, 有利于在网络中传输

```
private void writeObject(ObjectOutputStream out)
    throws IOException {
    out.defaultWriteObject();
}
```

//Serializable 接口中声明的方法

```
private void readObject(ObjectInputStream in)
    throws IOException, ClassNotFoundException {
    in.defaultReadObject();
}
```

//对对象数据进行初始化

```
public BaseGraphProp()
```

```
{
```

(省略)



```

}
//基类的成员函数,以便子类进行扩展
void Draw() //绘制图形
{
}
//绘制对象的热点标志 HOTSPOT
final void DrawStandOut(Graphics g,Color cl){
    (省略)
}
}
}
    
```

2D图元对象UML设计图

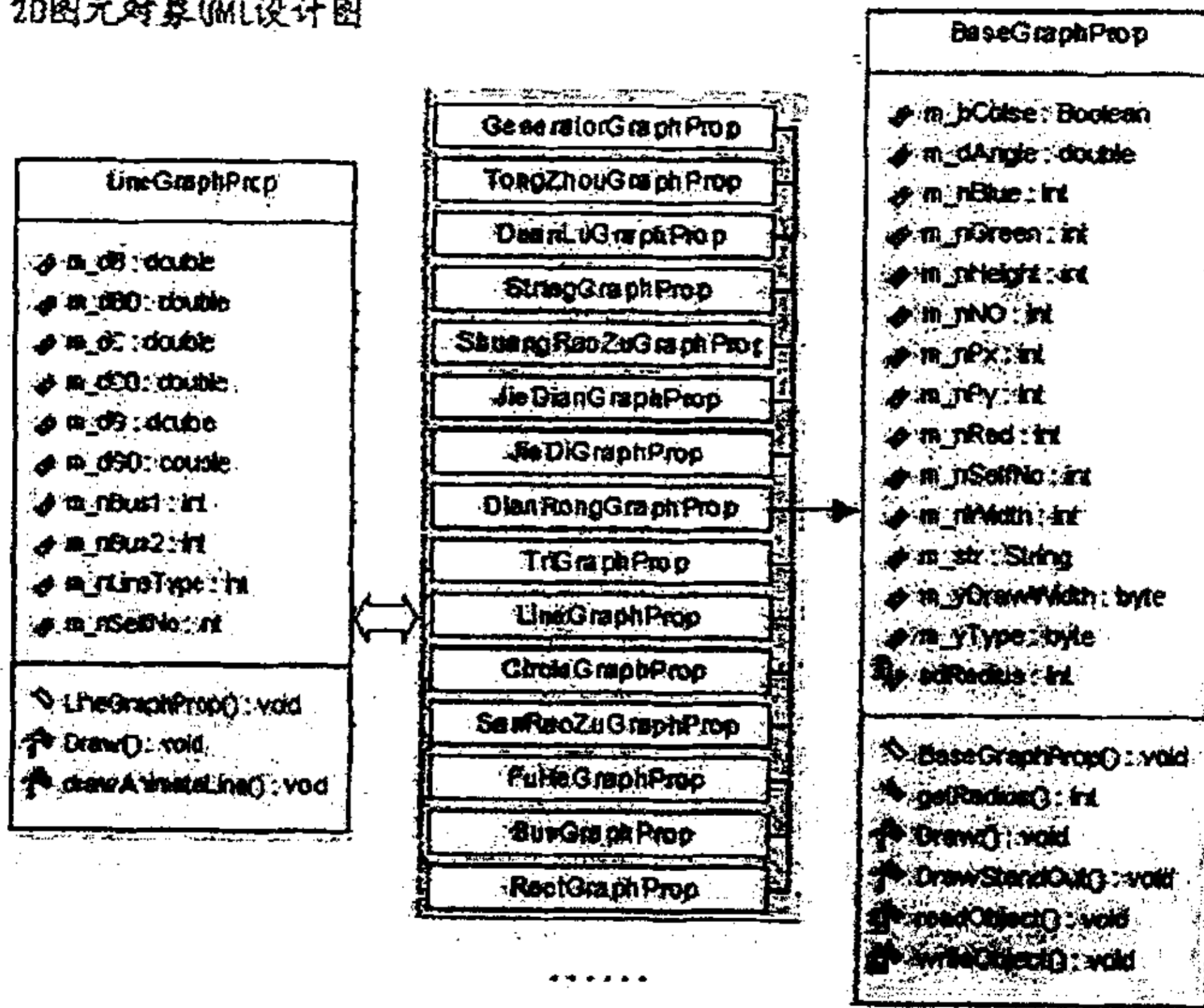


图 6-3 2D 图元对象设计图

Fig. 6-3 The 2D graphic objects design

```
/**类功能描述：输电线图元对象的定义
 * 入口参数：
 * 出口参数：
 * 版本：2.1
 */
//输电线
class LineGraphProp extends BaseGraphProp
{
    //添加电气属性
    public int m_nSelfNo; //对输电线和母线有用，单独计数从 1 开始
    public int m_nLineType; //输电线的类型
    public int m_nBus1; //连接总线 1
    public int m_nBus2; //连接总线 2
    public double m_dG; //参数 G
    public double m_dB; //参数 B
    public double m_dC; //变比 C
    //成员函数
    public LineGraphProp()
    {
        (省略)
    }
    void Draw(Graphics g) //重载 Draw() 实现自己的绘图操作
    {
        (省略)
    }
    //画出动画片断
    void drawAnimateLine(Graphics g, int nDistance)
    { (省略)
    }
}
```

}

### 6. 3. 2 动画模块的设计

在 Java 流行的初期,各种网页上有趣的动画 Applet 和游戏 Applet 对人们认识 Java 起了极大的推动作用。可以说 Java 在设计动画上有着与生就来的优势。

下面说明一下动画的基本原理。就跟手绘的动画卡通原理一样,电脑上的动画其实也是利用快速度播放许多张图片所造成的视觉暂留效果来实现的。而动画的质量,除了取决于图片本身的好坏之外,动画中动作的平顺度,也是一大重点。简单来说,同一时间内放映的图片数量越多,动画中人物的动作就越看起来越平顺。据研究,每秒 24 张的速度,在人的眼中已经感觉不到任何的停顿了。一般的手绘动画通常 1 秒钟放映的图片张数从最粗糙的 8 张,到中等的 1 秒 12 张,甚至最高级的 1 秒 24 张图片都有,而电脑动画的习惯则是每秒 10 张到 20 张不等。

Java 具有多线程和图形的双缓冲技术<sup>[71]</sup>,这两大技术保证了制作动画的高效率。多线程使得动画线程和主线程之间不冲突,更好的使程序无阻碍的运行。双缓冲技术使得动画的显示没有闪烁,更自然。

Web 终端上的动画模块主要包括两部分:

1. 输出潮流的流向在输电线上动态显示。
2. 用色彩的明暗显示节点电压的高低情况。

客户终端图形输入和输出模块属于 Java Applet,由 Borland 公司的 Jbuilder7 开发,经测试能无阻碍的运行 Internet Explore5.0 以上的版本。

## 6. 4 数据库服务器的设计

在分布式程序中,数据库服务器占据着非常重要的位置。数据库设计的好坏涉及到查询速度的快慢,以至影响到整套系统性能的优劣。在本系统中,我们充分考虑到了此图形计算系统的共享、效率和与软件的集成,我们采用了 Borland 公司的 Data Store 数据库。此数据库在中小型和嵌入式数据库的处理上效率很高,并且数据中可以存储多种格式的文件,便于数据的集成。而本系统是多数据库的系统,每个数据库的数据量不会很大,但数据库的数量很多。为

为了避免单个数据库所造成的资源浪费，故没有选用如 Oracle, Sybase, DB2, MS SQL Server 等大型数据库。再加上 Data Store 数据库是纯粹的 Java 数据库，与本系统的 Java 主程序更能有机的结合。对数据库的操作选用了标准的 SQL 语句来进行。与传统的专用数据库编程相比，可提高程序的可移植性、可靠性、可维护性以及缩短了开发的时间。

下面详细介绍了数据库的设计和运行时数据库中各种表的关系。

由于本套系统是多用户系统，每一个用户可以拥有多个电网系统。每一个电网系统对应一个数据库文件，对每一个新用户新建一个目录来保存他的所有电网系统。故系统建立一个用户信息表，来标识每个路径。

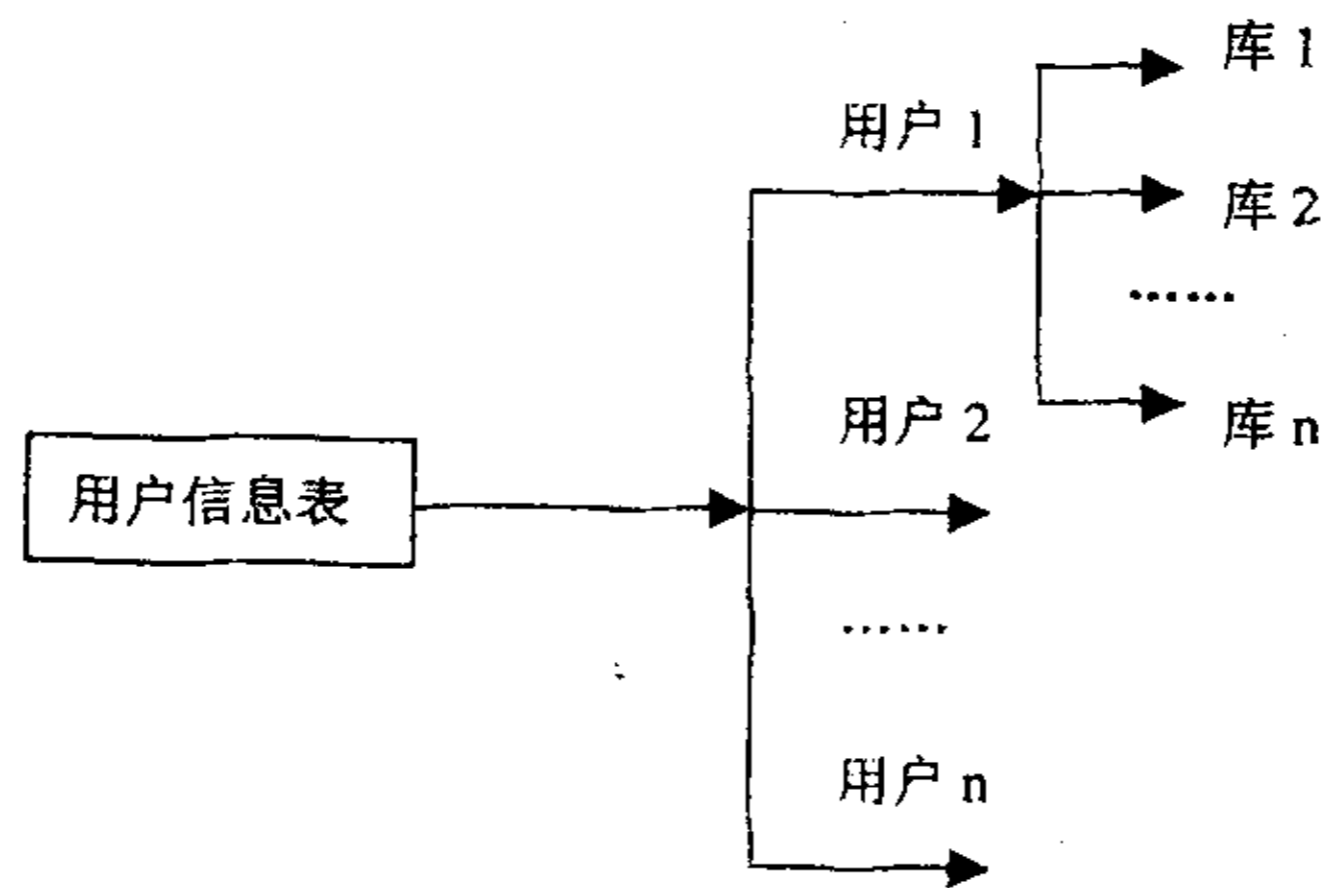


图 6-4 多用户系统设计

Fig. 6-4 The multi-user system design

用户数据表的结构如下。ID 代表用户唯一标识，PASSWORD 表示用户的密码。ID 和 PASSOWRD 是在用户登录的时候验证用户身份的。DBPATH 记录了此用户对应的数据库的存储位置，其存储位置可是在网络的任意一个地方，可以实现数据存储的分布式处理。PRIORITY 表明了此用户的等级和计算优先权，不同的用户可以根据他所拥有的优先权来进行不同等级和权限的操作。

	Column Name	Data Type	Precision	Scale	Required
1	ID	STRING	15	-1	<input checked="" type="checkbox"/>
2	PASSWORD	STRING	20	-1	<input type="checkbox"/>
3	DBPATH	STRING	20	-1	<input type="checkbox"/>
4	PRIORITY	INT			<input type="checkbox"/>

图 6-5 用户数据表的结构

Fig.6-5 The structure of user table

对每一个电网系统库<sup>[62,63,64]</sup>的设计,分为元件物理属性表和元件位置属性表。数据库由元件位置属性表、发电机属性表、负荷属性表、输电线属性表、母线属性表、运行结果表组成。元件位置属性表记录了电网图上每一个元件的矢量信息。数据库的表之间的结构如下图 6-5 所示。

数据库通过标准的 JDBC 桥连接。对数据库的操作通过中间层事务逻辑进行。

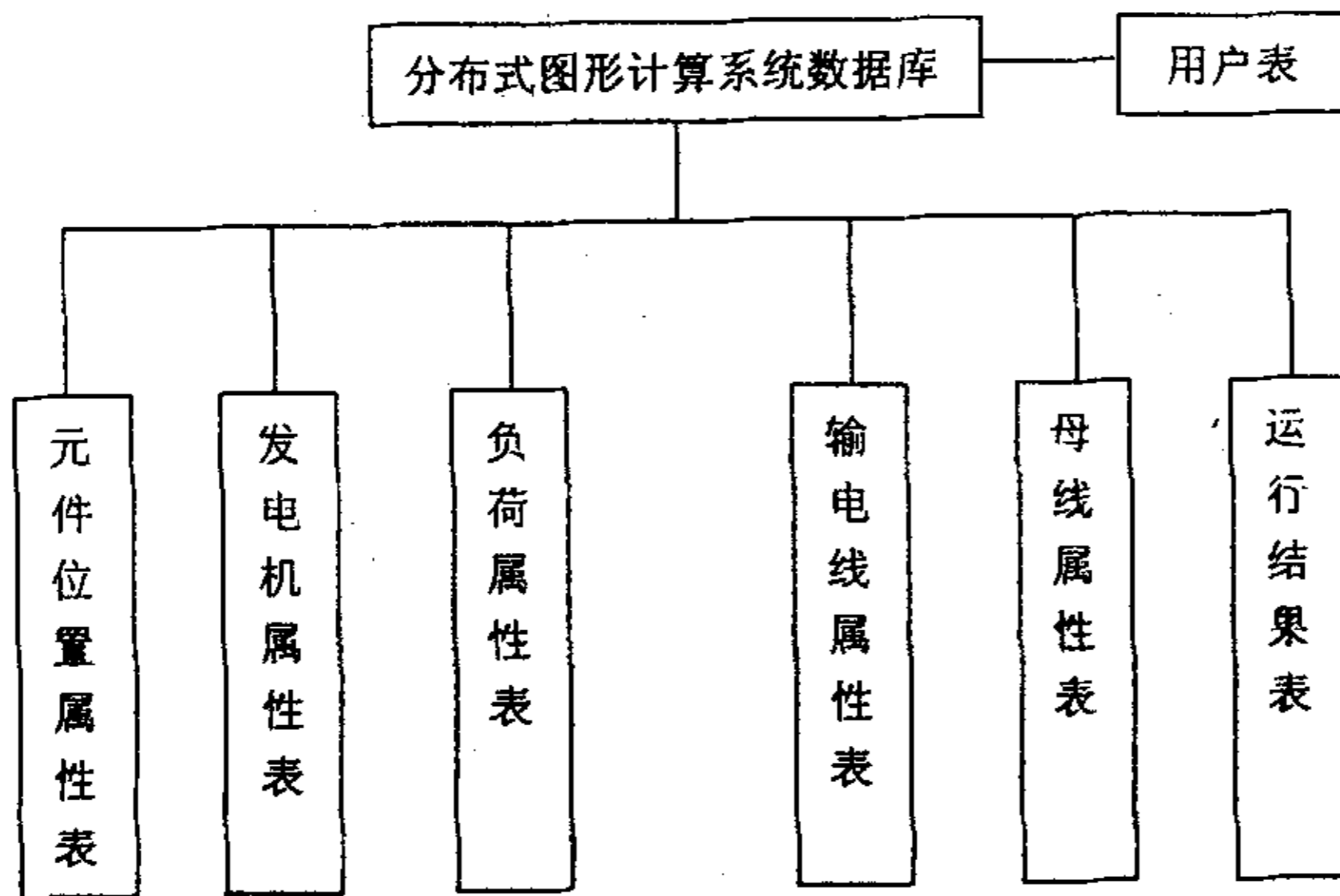


图 6-6 数据库的组成

Fig.6-6 The structure of the database

	Column Name	Data Type	Precision	Scale	Required
1	NO	INT	-1	-1	<input checked="" type="checkbox"/>
2	TYPE	INT	-1	-1	<input type="checkbox"/>
3	PX	INT	-1	-1	<input type="checkbox"/>
4	PY	INT	-1	-1	<input type="checkbox"/>
5	WIDTH	INT	-1	-1	<input type="checkbox"/>
6	HEIGHT	INT	-1	-1	<input type="checkbox"/>
7	RED	INT	-1	-1	<input type="checkbox"/>
8	GREEN	INT	-1	-1	<input type="checkbox"/>
9	BLUE	INT	-1	-1	<input type="checkbox"/>
10	ANGLE	DOUBLE	-1	-1	<input type="checkbox"/>
11	TEXT	STRING	100	-1	<input type="checkbox"/>

图 6-7 元件位置属性表的结构

Fig. 6-7 The structure of position table

#### 6. 4. 1 元件位置属性表

元件位置属性表见图 6-7 所示, 它用来记录电网的元器件在 Web 终端上显示的位置信息。它在数据库中占据着非常重要的地位, 它不仅对电网图的正确显示很重要, 而且负责联系与之对应的元件属性表。NO 是对元件的总编号, 是检索元件链表的关键字段。TYPE 表示元件的对应的类型号, 它标识着不同的物理元件。PX, PY 表示元件最大外接四边形的中心点坐标。WIDTH, HEIGHT 表示元件最大外接四边形的宽和高。RED, GREEN, BLUE 表示元件的 RGB 颜色。ANGLE 表示元件最大外接四边形与水平线的偏离角度。TEXT 表示元件的注释和说明字符串。

#### 6. 4. 2 输电线属性表

输电线属性表记录着电网图中每个输电线所具有的属性。NO 代表在整个元件链表中的编号。SELFNO 表示在输电线中它的编号。TYPE 表示输电线的类型。BUS1、BUS2 表示输电线所连接的两条母线的编号。G、B、C 分别表示

输电线的其他物理参数。

	Column Name	Data Type	Precision	Scale	Required
1	NO	INT	-1	-1	<input checked="" type="checkbox"/>
2	SELFNO	INT	-1	-1	<input type="checkbox"/>
3	TYPE	INT	-1	-1	<input type="checkbox"/>
4	BUS1	INT	-1	-1	<input type="checkbox"/>
5	BUS2	INT	-1	-1	<input type="checkbox"/>
6	G	DOUBLE	-1	-1	<input type="checkbox"/>
7	B	DOUBLE	-1	-1	<input type="checkbox"/>
8	C	DOUBLE	-1	-1	<input type="checkbox"/>

图 6-8 输电线属性表的结构

Fig.6-8 The structure of line table

#### 6. 4. 3 母线属性表

母线属性表记录着电网图中每个母线所具有的属性。NO 代表在整个元件链表中的编号。SELFNO 表示在母线中它的编号。TYPE 表示母线的类型。VOLTAGE 表示母线的电压。

	Column Name	Data Type	Precision	Scale	Required
1	NO	INT	-1	-1	<input checked="" type="checkbox"/>
2	SELFNO	INT	-1	-1	<input type="checkbox"/>
3	TYPE	INT	-1	-1	<input type="checkbox"/>
4	VOLTAGE	DOUBLE	-1	-1	<input type="checkbox"/>

图 6-9 母线属性表的结构

Fig. 6-9 The structure of bus table

#### 6. 4. 4 发电机属性表

发电机属性表记录着电网图中每个发电机所具有的属性。NO 代表在整个元件链表中的编号。TYPE 表示发电机的类型。BUS 表示发电机所接母线的编号。APOWEROUT 表示发电机有功输出。MINRPOWEROUT 表示发电机的最小无功输出。MAXRPOWEROUT 表示发电机的最大无功输出。

	Column Name	Data Type	Precision	Scale	Required
1	NO	INT	-1	-1	<input checked="" type="checkbox"/>
2	TYPE	INT	-1	-1	<input type="checkbox"/>
3	BUS	INT	-1	-1	<input type="checkbox"/>
4	APOWEROUT	DOUBLE	-1	-1	<input type="checkbox"/>
5	MINRPOWEROUT	DOUBLE	-1	-1	<input type="checkbox"/>
6	MAXRPOWEROUT	DOUBLE	-1	-1	<input type="checkbox"/>

图 6-10 发电机属性表的结构

Fig. 6-10 The structure of generator table

#### 6. 4. 5 负荷属性表

负荷属性表记录着电网图中每个负荷所具有的属性。NO 代表在整个元件链表中的编号。BUS 表示负荷所接母线的编号。APOWERLOAD 表示负荷有功。RPOWERLOAD 表示负荷的无功。

	Column Name	Data Type	Precision	Scale	Required
1	NO	INT	-1	-1	<input checked="" type="checkbox"/>
2	BUS	INT	-1	-1	<input type="checkbox"/>
3	APOWERLOAD	DOUBLE	-1	-1	<input type="checkbox"/>
4	RPOWERLOAD	DOUBLE	-1	-1	<input type="checkbox"/>

图 6-11 负荷属性表的结构

Fig. 6-11 The structure of load table



6. 4. 6 表之间的关系



图 6-12 表的关系

Fig. 6-12 The relations between the tables

由图 7-9 可以看出，表与表之间通过字段 NO 进行连接。

6. 4. 7 数据库连接池

一个动态的网站会频繁地从数据库中取得数据来构成 HTML 页面。每一次请求一个页面几乎都会发生数据库操作。但连接数据库却是一个需要消耗大量时间地工作，应为请求连接需要建立通信，分配资源，进行权限认证。这些工作很少能在一两秒钟完成。所以，建立一个连接，然后在后续地查询中都可以使用此连接会大大提高数据库地性能。因为 Servlet 可以在不同的请求间保持

状态, 因此采用数据库连接池是一个直接的解决方案。

Servlet 在服务器的进程中驻留, 可以方便而持久地维护数据库的连接。

## 6.5 专业计算服务器的设计

本系统用来实现电力系统的专业计算, 它要求实现基本潮流运算、事故分析、开关控制和优化求解。

计算服务器的调用过程如图 6-13 所示, 可以分为 4 个过程: 启动、获取数据、计算和返回结果。

首先启动 CORBA 服务器, 通过指定接口指明调用哪一个计算模块; 接着计算服务器从数据库中调出网络拓扑结构数据、元件参数等数据; 然后执行计算模块; 最后将计算结果直接显示在 Web 浏览器的电网图上, 或者通过 Web 服务器形成 JSP 页面将计算结果返回给用户。

计算模块为了提高效率, 可以由 Fortran/C/C++ 等语言开发, 它们是控制台的可执行程序。它们与外界交换的方式为数据文件。所以计算模块可以动态添加。它兼容了以前 DOS 下的算法程序。

CORBA 服务器计算中心程序由 JAVA 开发, 它主要完成读出数据库中的原始数据, 并形成计算模块的输入数据文件, 然后调用计算模块, 最后把计算模块所形成的输出文件传给 Web 服务器程序。

进一步, 可以在各个省电力局建立起自己的计算中心 (如图 6-14 所示), 部署自己的计算模块, 提供给 CORBA 计算中心统一的接口。然后任意的浏览器客户端都可以通过 CORBA 计算中心访问到这些计算中心, 使得信息得以共享。

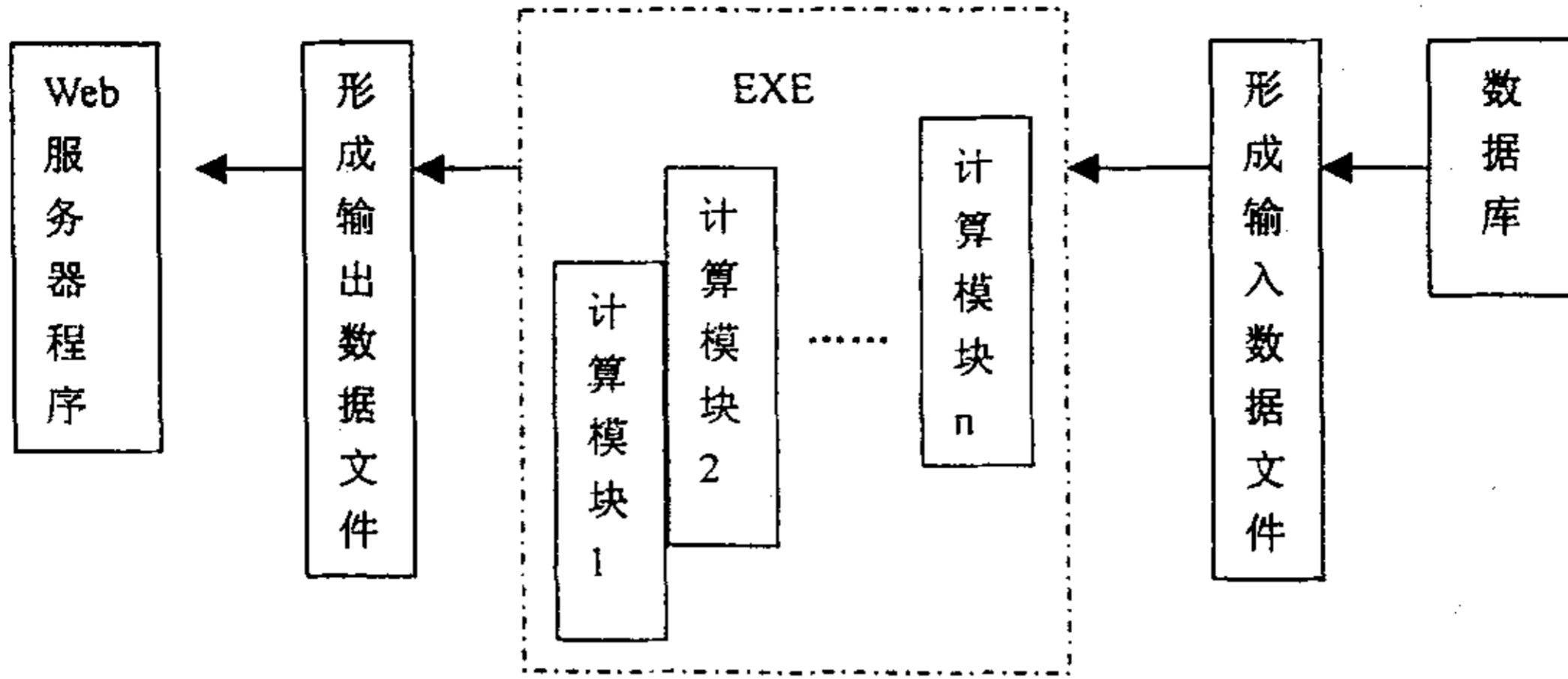


图 6-13 计算模块调用

Fig.6-13 The invoking diagram of computing modules

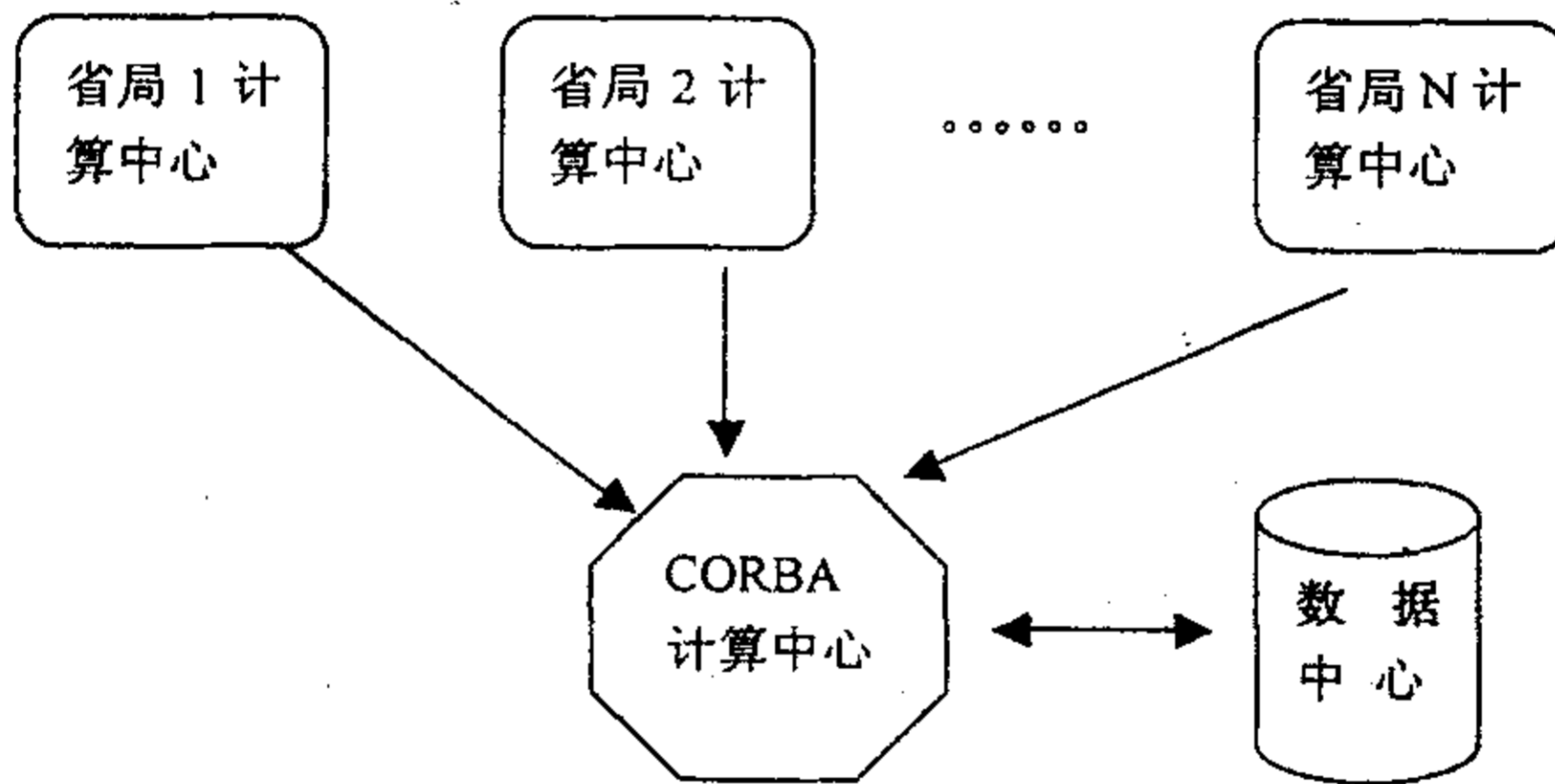


图 6-14 分布式计算中心

Fig 6-14 The distributed computing centers

## 6. 6 安全策略

密码技术是保护信息安全的主要手段之一。使用密码技术可以防止信息被篡改、伪造和假冒。

6. 6. 1 加密和认证

加密是把称为“明文”的可读信息转换成“密文”，即不可读信息的过程；解密则是把已加密的信息恢复成“明文”的过程。加密和解密使用密码算法来完成这些工作。密码算法是用于隐藏和显露信息的可计算过程，算法越复杂，结果密文就越安全。在加密技术中，密钥是不可缺少的。密钥是使密码算法按照一种特定方式运行并产生特定密文的值。密钥越大，结果密文越安全。

加密是为了保护数据的安全，但也可用作几乎相反的用途——认证。认证是信息安全的一个重要方面。它主要包括用户认证和信息认证两个方面。前者用于鉴别用户身份，后者用于保证通信双方的不可抵赖性和信息的完整性。

6. 6. 2 加密算法

加密通信的一般模型如图 6-15 所示。密码技术按其密钥 ( $K_e$ ,  $K_d$ ) 管理的方式分为公钥密码体制 ( $K_e$  不等于  $K_d$ ) 和私钥密码体制 ( $K_e$  等于  $K_d$ ) (也称为非对称密码体制和对称密码体制)。

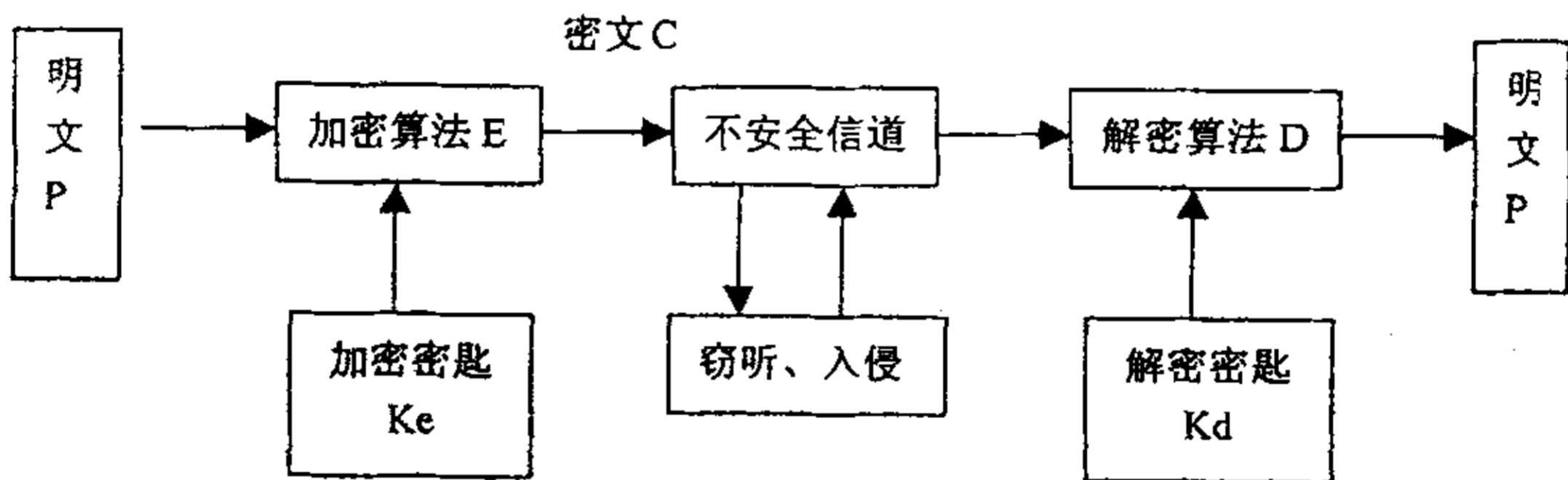


图 6-15 加密通信的一般模型

Fig. 6-15 The general model of encrypted communication

最著名的私钥密码体制是 DES (Data Encryption Standard)。DES 是由 IBM 公司开发的数据加密算法，它的核心是乘积变换。它是指信息的发送方和接收方共同使用同一把密钥进行加解密。通常，使用的加密算法比较简便高效，密钥简短，加解密速度快，破译极其困难。美国于 1997 年 1 月将其定为非机密数据的

正式数据加密标准。它可有效地防止穷尽搜索攻击。还有一些其他常用的私钥加密算法,如 3DES、IDEA 等。

采用私钥密码体制加密的信息,其安全性有赖于密钥的保密性,因而密钥的秘密分配和安全管理成了私钥密码体制最重要的问题。针对这一问题,美国的 M.E.Hallman、W.Diffie 和 R.Merkle 于 1976 年提出了“公开密钥体系”(Public Key System)。此后人们相继提出了一些具体实现方案,其中最著名的是由美国 MIT 的 Rivest、Shamir、Adleman 于 1977 年实现的 RSA 算法。RSA 算法的优点是大大简化了密钥管理,但也存在一些缺点。例如,公开密钥很有可能被攻击者私自改为自己的公钥,从而轻易获得被攻击对象的一切信息。

公钥密码体制和私钥密码体制各有其优缺点,因此在使用中大都采用这两种密码体制相结合的方法。一般是在对密钥的传输中使用公钥加密体制,而在文件的加密过程中使用私钥加密体制。这种结合不仅有效地保护了密钥,同时也保护了密文,效率得到了很大的提高。

### 6. 6. 3 加密和认证的作用

在网络中使用加密算法主要可保护下面几个方面的安全:

#### 1. 私有通信

计算机中加密技术最常用于保护计算机用户之间和通信设备之间的通信。如在网络上传输电子邮件时,在发送邮件之前将电子邮件信息加密而接收到这封邮件时通过软件包或一个外部程序来解密这个信息,从而保证邮件内容不被窃取。

#### 2. 保护存储文件

加密不只是仅仅在通信中使用,它也可以来保护存储文件,如硬盘中的文件、数据等。

#### 3. 用户和计算机的身份认证

计算机使用加密来对用户的身份及信息进行认证,如用户的登录身份认证、数字签名和数字凭证等。

#### 4. 安全口令交换

当你登录到你的网络文件服务器或连接到你的因特网服务供应商时,你要提供你的用户名和口令。这两部分信息控制着你对网络的访问,并代表你在网

络上的身份，必须保证不被偷听。大部分操作系统都通过将用户名和口令进行加密的方法来保证安全。

#### 6. 6. 4 本系统的安全策略

自从网络普及以来，安全便是系统设计必须考虑的问题。由于本系统是个多用户的系统，便需要对用户进行身份识别。

首先在服务器端建立用户数据库，它存储了用户的必要信息，包括用户名，密码和权限等。

用户通过网页输入用户名和密码提交给 Web 服务器，Web 服务器的应用程序 Servlet 与用户数据库进行比较，并返回结果。这里需要注意的是用户必须使用强口令，即由系统规定最短口令长度，限制使用常用单词和要求使用数字和特殊字符。

为了防止黑客侵入用户数据库后，窃取有用的信息。对用户数据库用 DES 进行加密，并定期更换密匙，以达到保护用户资料的目的。

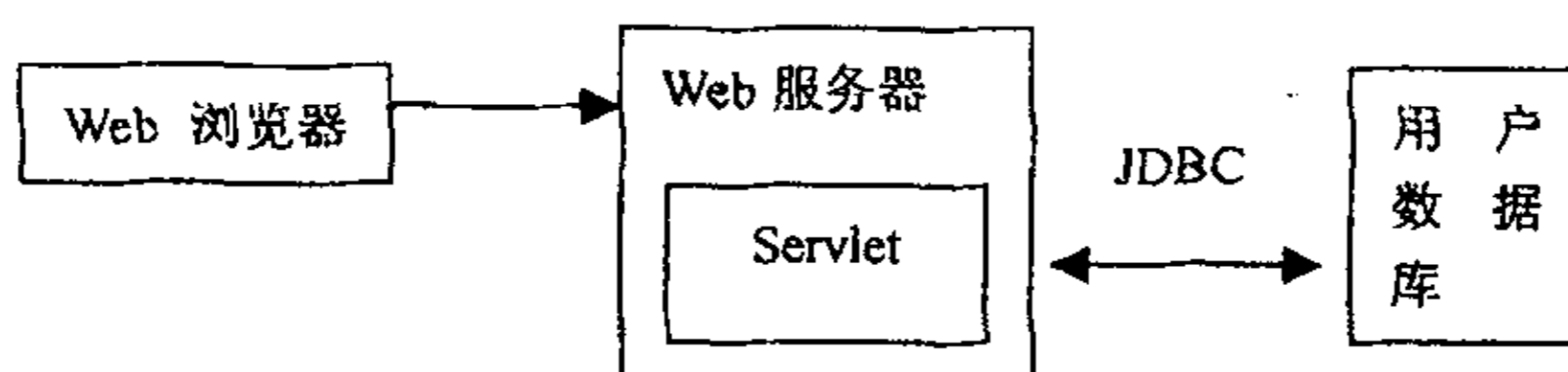


图 6-16 身份识别过程图

Fig.6-16 The user identification flow

## 第七章 系统实例

本部分给出了基于 Web 的电力系统分布式图形化网络计算的运行实例。将本系统分别应用于 IEEE6 节点系统和 IEEE30 节点系统，下面分别介绍了应用情况。

### 7.1 系统登录和注册

图 7-1 显示了系统运行和登录页面：

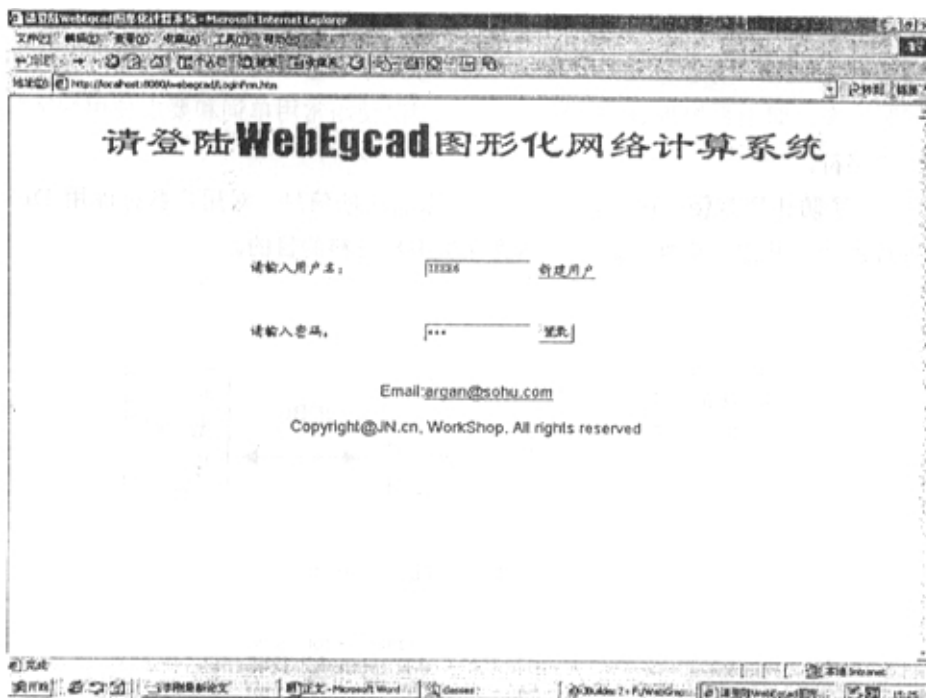


图 7-1 系统登录页面

Fig. 7-1 The home page of the computing system

用户如果已经拥有了用户名和账号, 则可以直接登录, 如果没有账号, 可以注册新用户。注册新用户的界面图 7-2 所示。



图 7-2 新用户注册页面

Fig. 7-2 The page of new user

当用户登录成功后, 就可以进入系统运算的主界面, 进行绘制电网图、输入电气数据、运算、结果显示等, 下面以 IEEE6 节点系统和 IEEE30 节点系统



为例来图示系统的运行情况。

## 7.2 IEEE6 节点系统应用情况

对于 IEEE6 节点系统。进入系统运算的主界面后, 运行情况见图 7-3 所示。

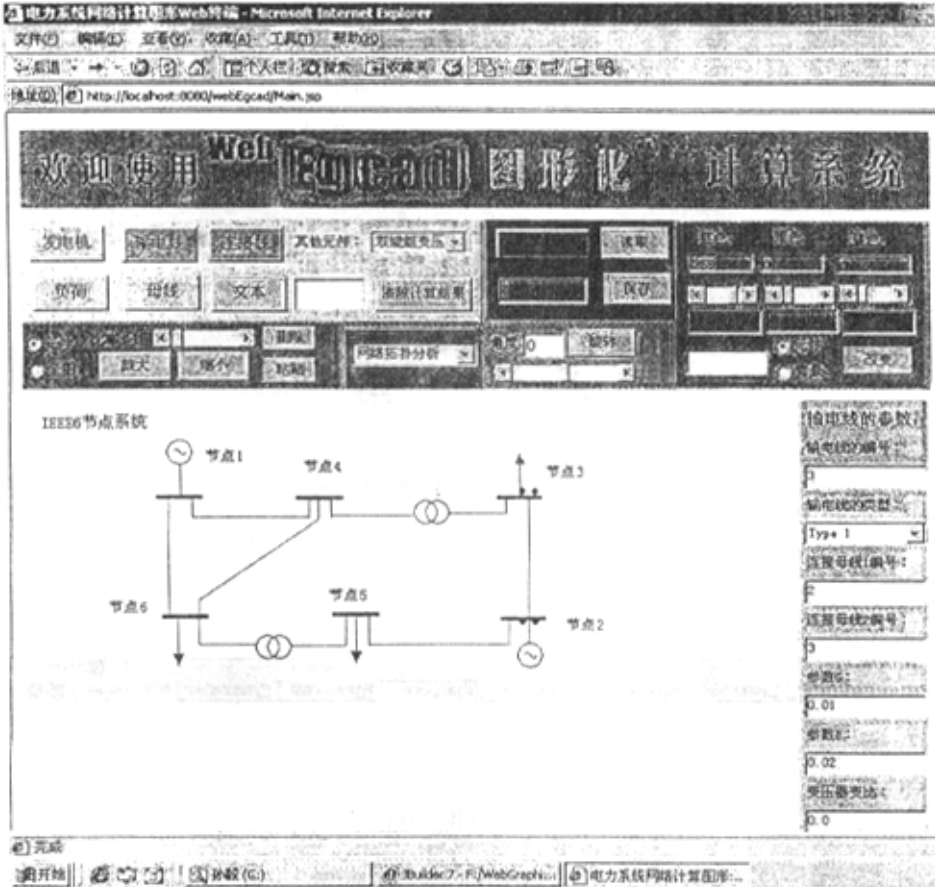


图 7-3 IEEE6 系统计算主界面

Fig. 7-3 The graphic interface of computing system

这里选取的是基本的潮流计算，对 IEEE6 系统进行潮流运算后的结果如图 7-4 所示。各个节点的功率和相位以及每一条输电线的有功和无功都被显示出来。对每一条输电线，其有功和无功的流动方向在输电线上动态的被表示出来。

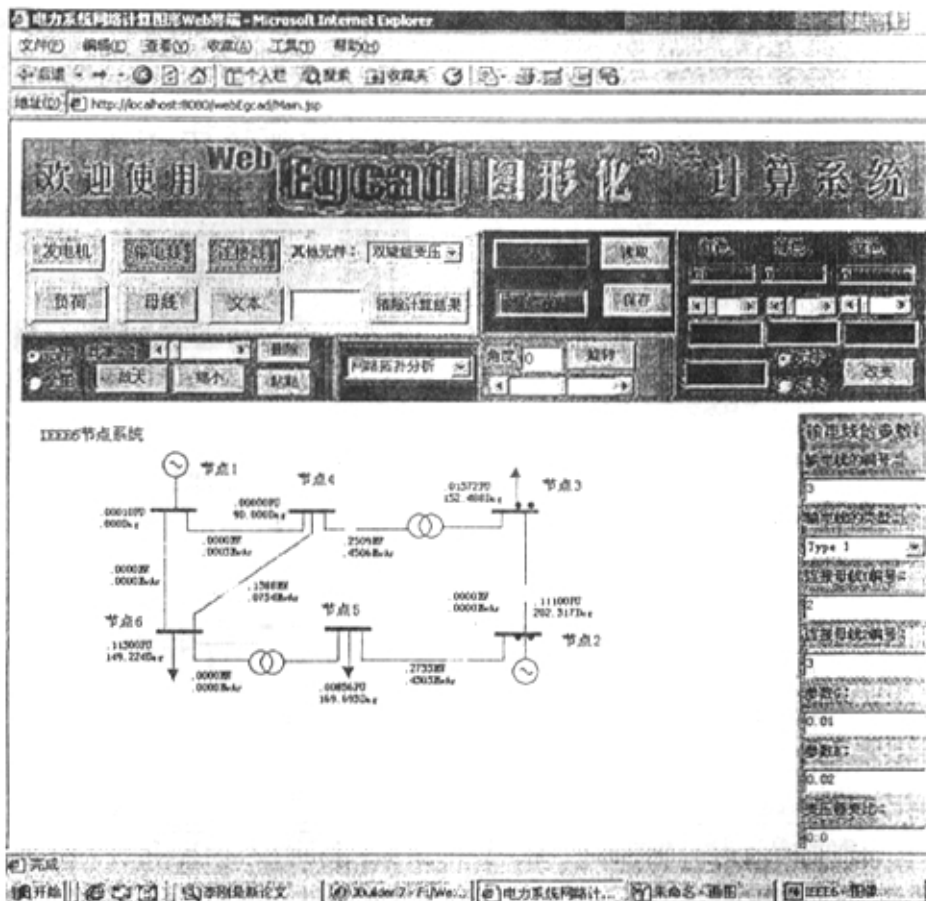


图 7-4 IEEE6 运算结果显示图

Fig. 7-4 The results display of IEEE6

### 7.3 IEEE30 节点系统应用情况

对于 IEEE30 系统，进入系统运算的主界面后，运行情况见图 7-5 示。

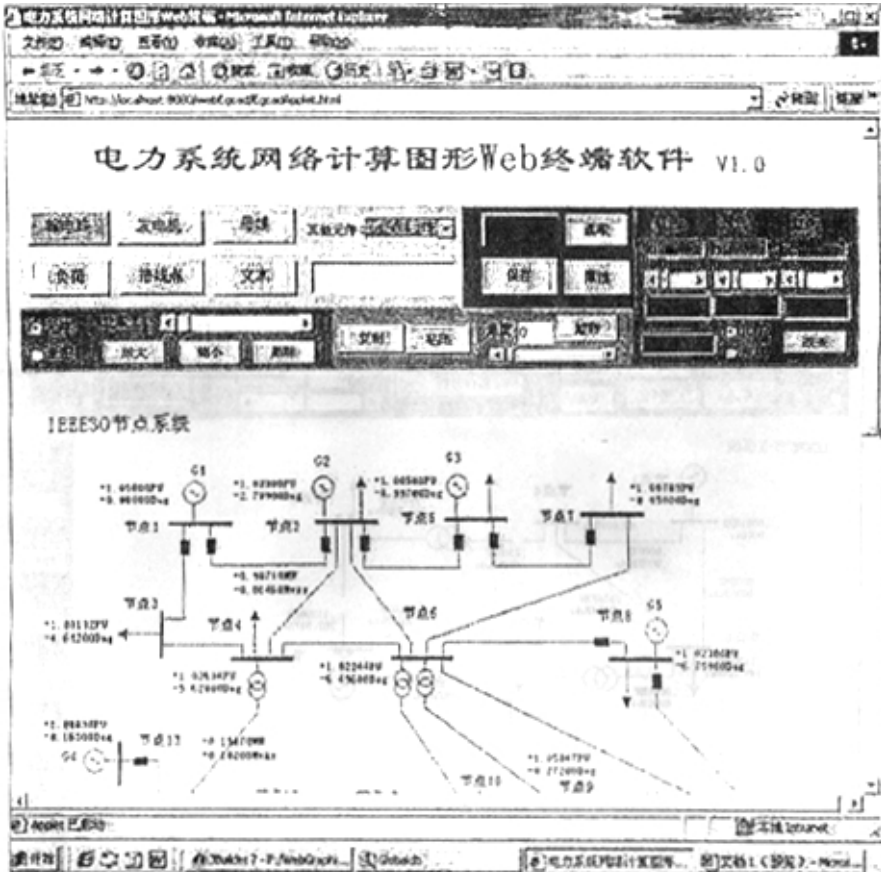


图 7-5 IEEE30 运算结果显示图

Fig. 7-5 The results display of IEEE30

## 第八章 结论

本文详细讲述了基于 Web 的电力系统网络计算系统的设计方案和具体实现, 并采用了 Borland 公司的 JBuilder7 开发出了具有一定使用价值的计算系统。随着计算机技术的进一步发展, Web 技术和分布式网络计算技术在电力系统中开始广泛的使用。实践证明, 分布式的系统架构和浏览器充当瘦客户端的设计方法, 是符合时代发展的潮流。它将使得系统的可维护性、扩展性、重用性都得到大大的提高。

本文首先分析了国内外电力系统领域中的专业计算的发展动态, 比较得出了它们的不足。然后提出了当前在电力系统中专业计算采取基于 Web 的分布式架构的必要性。

随后简要介绍了电力系统中的专业计算。接着分析网络计算发展的历史和未来。着重比较了传统的客户/服务器计算模式、客户/浏览器计算模式和分布式计算模式的区别, 并展望了未来网络的计算模式。

然后就本系统采用的主要开发语言 Java 和一种分布式系统 CORBA 进行了较为详细的介绍。同时还简要介绍了在系统中所使用的计算机图形技术—矢量图形的设计模式。为下一章的具体系统开发做好铺垫。

接下来, 本文给出了基于 Web 的图形化网络计算系统的设计方案。包括了系统的体系结构和具体流程。接着具体介绍了系统的主要部分如 Web 终端、数据库服务器、算法服务器、安全策略等关键技术的实现过程。

最后, 本文把开发出来的系统应用于 IEEE6 节点系统和 IEEE30 节点系统的潮流计算, 并给出了相应的图示。

随着计算机技术的发展, 新的计算模式的出现。也将不断丰富电力系统计算模式。作者就当前技术发展的趋势, 谈一谈对本系统的几点改进和看法:

1. 网络技术和 Web 服务的兴起, 使得服务器为中心的计算机模式有了另外的选择。将系统扩展到兼容 Web 服务的模式, 使系统可以得到更广泛的使用和扩展。
2. 本系统所实现的在 Web 上的可视化技术, 可以推广到其他的专业计算领域。以前的网络计算系统只能实现计算和对数字的操作, 友好性和可操作性

都比较差。基于 Web 的可视化技术在地理信息系统 (GIS) 中发展最为迅速, 现在已经实现了基于 Web 的 3D 的可视化技术。把这些技术应用到传统的领域, 必然拥有良好的发展前景。

总之, 基于 Web 的电力系统网络计算系统是随着计算机技术的发展和 Web 的普及出现的新型的计算系统。它适应了人们对计算机系统方便、快捷、友好、安全并且功能强大的要求。也必将进一步提高电力系统的信息化程度。由于作者的水平有限, 本文难免出现错误和疏忽之处, 请各位专家不吝赐教。

## 独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知,除了文中特别加以标注和致谢的地方外,论文中不包含其他人已经发表或撰写过的研究成果,也不包含为获得四川大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

本学位论文成果是本人在四川大学读书期间在导师指导下取得的,论文成果归四川大学所有,特此声明。

签名: 李刚 导师签名: 刘波 日期: 2003.5.12.

## 致 谢

本文是在导师刘俊勇教授的悉心指导下完成的。刘老师对课题的选择和对课题的顺利进展都倾注了大量心血。导师渊博的学识、治学的严谨和对科学方法的透彻理解让学生由衷的钦佩,必将使学生终身受益。刘老师除了在学术研究上给予了悉心指导,而且也让学生明白做人的道理。在此论文完成之际,向导师刘俊勇教授表示深切的感谢和崇高的敬意。

在作者攻读硕士学位期间,得到了本教研室各位老师的热心指导和帮助,在此,谨向各位老师表示衷心的感谢。

衷心感谢评审论文的各位专家和教授为提出本文提出了宝贵的意见。

在作者学习和生活中,得到了许多同窗好友的支持和帮助,在此向他们表示感谢。

作者深深感谢女友黄乐长期以来给予的关心和鼓励。

最后,深深感谢父母在我漫长的求学过程中所给予的理解和支持,谨以此文献给我的家人和关心作者的人们!

## 参考文献

1. BinQiu. Internet-Based SCADA Display System. IEEE Computer Application in Power, 2002, Vol. 15 Number1, pp 14-19s
2. S.Chen, Web Based Simulations of Power Systems, IEEE Computer Application in Power, 2002, Vol.15 Number1, pp 35-40
3. 诸俊伟, 电力系统分析, 北京: 水利电力出版社, 1995
4. 易文韬, JAVA 2 程序设计实务入门, 北京: 中国铁道出版社, 2001.
5. 袁林, 基于 WWW 技术的电网实时信息系统, 电力系统自动化, Jun.1998, pp 55-57
6. 孙晓慧, 开发基于 Internet/Intranet 电网管理信息系统, 电网技术, Jan.1999, pp 9-13
7. 王永福, 基于 CORBA 多代理机制的新型 EMS 图形 Web 发布系统, 电力系统自动化, Apr.2002, pp 65-67
8. 李捷, Web 技术在 EMS 系统中的应用, 中国电力, Mar.2000, pp 60-63
9. 何君臣, 电力系统 Web 网络信息模型的研究, 电网技术, Jun.2000, pp 62-62
10. 位元文化, JSP 动态网页入门实务, 北京: 科学出版社, 2001
11. Oswald campesato.. Java Graphics Programming Library. Beijing: Publishing House of Electronics Industry, 2002
12. Danny Ayers. Professional Java Server Programming. Beijing: China Machine Press, 2001
13. Steve Graham. Building Web Services with Java. Beijing: China Machine Press, 2003
14. Qusay H. Mahamoud. Distributed Programming with Java. Beijing: China Defense Industry Press, 2001
15. Berners-lee, Tim. World-wide Computer. Communications of the CAN, Feb.1997
16. Comer, Douglas.E. and David L. Stevens. Internet working with TCP/IP. Vol. 2: Design, Implementation, and Internals, 2<sup>nd</sup> edition, Prentice Hall, 1994
17. Couloirs, George, Jean Dollinore, and Tim Kindberg. Distributed System: Concepts and Design. Addison-Wesley, 1994
18. Miranda V, Ranito J V. Genetic algorithms in optimal multi-stage distribution network planning. IEEE Trans Power System.1994, Vol. 9 Number4, pp 1927-1933
19. Terrell T J. Building a geographic information system. IEEE Trans on Computer



- Application in Power. 1991, Number 7, pp 50-54
20. Otte R, Patrick P, Roy M. Understanding CORBA: The Common Object Request Broker Architecture. New York: Prentice-Hall Inc, 1993
  21. Weber J D. Individual welfare maximization in electricity markets including consumer and full transmission system modeling. PHD thesis, University of Illinois, 1999
  22. Mitsui H, Christie R D. Visualizing voltage profiles for large-scale power systems. IEEE Computer Application in Power, July 1997:32-37
  23. Gronquist J, Sethares W, Alvarado F, Animated vectors for visualization of power system phenomena. IEEE Trans on Power Systems, 1996, 11(1): 267-273
  24. Shubha Pandit, S.A.Soman, S.A.Khaparde. Object-Oriented Design for Power System Applications. IEEE Computer Applications in Power. October 2000
  25. Jun Zhu, David L.Lubkeman. Object-Oriented Development of Software Systems for Power System Simulations. IEEE Transactions on Power System, Vol.12, NO.2 May 1997
  26. Nikos Hatziargyriou, Jakub Jaszczaynski, George Atsaves, Dimosthenis Agoris. Building GUIs for Interactive Network Analysis. IEEE Computer Applications in Power. October 1998
  27. Mike Foley, Anjan Bose. Object-Oriented on-Line Network Analysis. IEEE Transactions on Power Systems. Vol.10, No, 1, Feb. 1995
  28. Andreas F. Neyer, Felix F.Wu, Karl Imhof. Object-Oriented Programming for Flexible Software: Example of a Load Flow. IEEE Transactions on Power Systems, Vol.5, NO.3, Aug. 1990
  29. Mike Fole, Anjan Bose, William Mitchell, Antony Fastini. An Object Based Graphical User Interface for Power Systems. IEEE Transactions on Power Systems. Vol.8, NO.1, Feb.1993
  30. Dennis G. Flinn, Roger C. Dugan. A Databse for Diverse Power System Simulation Applications in Power. Jan. 2000
  31. Qing Liu, Shijie Cheng. Object-Oriented Methods Drive Protective Relay System. IEEE Computer Applications in Power. Jan. 2000
  32. B. Hakavik, A.T. Holen. Power System Modeling and Sparse Matrix Operations Using Object-Oriented Programming. IEEE Transactions on Power Systems. Vol.9, No. 2, May

1994

33. Jay Britton. An Open, Object-Based Model as the Basis of an Architecture for Distribution control Centers. IEEE Transactions on Power Systems, Vol.7, No.4, Nov.1992
34. S.Li, S.M. Shahidehpour. An Object-Oriented Power System Graphics Package for Personal Computer Environment. IEEE Transactions on Power Systems, Vol.8, NO.3, Aug. 1993
35. E.Z. Zhou. Object-Oriented Programming, C++ and Power Simulation. IEEE Transactions on Power Systems, Vol.11, NO.1, Feb.1996
36. Gilberto P. Azevedo, Bruno Feijo, Monica Costa. Control Centers Evolve with Agent Technology. IEEE Computer Applications in Power. July 2000
37. Douglas Proudfoot, Dave Taylor. How to Turn a Substitution into a Database Server. IEEE Computer Applications in power April; 1999
38. Andre Bergholz. XML Introduces a Family of Languages to Provide a More Semantic Management of Information than HTML IEEE Internet Computing. July, 2000
39. Chungnan Lee, Chuanwen Chiang, Minfong Horng. Collaborative Web Computing Environment: An Infrastructure for Scientific Computation. IEEE Internet Computing March, 2000.
40. Feng He, Fengquan, Wei Li, Xiaolin Han, Jianhua Liu. Object Request Brokers for Distributed Measurement. IEEE Computer Applications in Power. Jan. 2001
41. E.-C. Yeh(M), Hahn Tram(SM). Information Integration in Computerized Distribution System Planning. IEEE Transactions on Power Systems. Vol.12, NO.2, May 1997
42. 王海猷, 贺仁睦, 基于组件的电力系统实时信息处理软件的结构, 电机工程学报, Vol.19, No.9
43. 曾克娥, 谢庆国, 基于构件技术的能量管理系统设计思想, 电力系统自动化, Vol.25, No.9 2001
44. 刘林, 连迹遐, C/S 体系结构下电力企业 MIS 设计的新考虑, 电机工程学报, Vol.18, No.6, Nov.1998
45. 周佃民, 廖培金, 多层 C/S 环境遐地区电网运行管理系统设计, 继电器, Vol.28, No.1, 2000
46. 贾长朱, 王剑锋, 基于 Web 的电力系统继电保护远程分布式信息管理系统的开发和设

- 计, 继电器, Vol.28, No.10, 2000
47. 汪源生, 基于 Web 的继电保护信息管理系统的开发和应用, 电力系统自动化, Vol.25, No.5, Mar.10, 2001
  48. 吴文传, 张伯明, 基于图形数据库的网络拓扑及其应用, 电网技术, 2002, 26 (2): 14-18
  49. 汪晶, 李晓明, 基于 Web 的电能量远程采集与分析系统的开发和应用, 电网技术, 2002, 26 (4): 66-69
  50. 乔毅, 孙岩, 可视化输配电网状态分析软件包的开发与实现, 电网技术, 25 (2): 1-5
  51. 张伯明, 陈寿孙, 高等电力网络分析, 北京: 清华大学出版社, 1996
  52. 黄远寿, 李新良, 电路基础, 成都: 成都科技大学出版社, 1994
  53. 黄德斌, 唐毅, 基于平台概念的继电保护计算及管理系统, 电网技术, 26 (7): 53-55
  54. 汤红卫, 王华, 一种基于地理信息系统德配电网规划方法, 电网技术, 26 (2): 79-82
  55. 刘理峰, 孙才新, 基于地理信息系统构建电气设备在线状态信息系统德方法, 电网技术, 25 (4): 27-31
  56. 王明俊, 于尔铿, 配电系统自动化及其发展, 北京: 中国电力出版社, 1998
  57. 何江, 吴杏平, 基于组件技术的电力系统实时数据库平台, 电网技术, 26 (3): 64-67
  58. 张洁, 基于 Web 技术的 EMS/DMS 图形和数据访问, 电力系统自动化, 1999, 23 (15)
  59. 杨茂江, 孙星明, 基于 CORBA-Web 的分布式应用系统开发策略, 计算机工程与应用, 36 (2): 21-23
  60. 黄宏涛, 王慧, 基于 Web 的分布式计算, 计算机应用研究, 17 (2): 38-40
  61. 吴复立, 文福拴, Internet 在电力市场中的应用, 电力系统自动化, 27 (5): 1-5
  62. 栗然, 卢锦玲, 基于关系数据库电网结构知识表示, 电力系统自动化, 26 (9): 71-74
  63. 段振国, 高曙, 电网结构知识表示方法研究和管理系统开发, 中国电力, 32 (2): 34-39

64. 孙雅明, 张沛, 电力系统的知识获取和知识库维护管理, 中国电力, 29 (8): 13-16
65. 周建华, 胡敏强, 基于数据库的电气设备故障诊断智能系统, 计算机工程与应用, 35 (2): 105-109
66. 曹阳, 姚建国, XML 技术在电网自动化系统中的应用探讨, 电力系统自动化, 26 (21): 73-76
67. 辛耀中, 新世纪电网调度自动化技术发展趋势, 电网技术, 25 (12): 1-10
68. 张慎明, 黄海峰, 基于 IEC61970 标准的电网调度自动化系统体系结构, 电力系统自动化, 26 (10): 45-47
69. 金敏, 周翔, 变电站自动化系统中嵌入式 Web 服务器的设计和实现, 电力系统自动化, 26 (18): 65-69
70. 朱磊, 基于 Web 的电力通信网监控系统的设计和实现, 电力系统自动化, 25 (2): 56-59
71. 闵勇, 孙希健, 电力系统动态过程的可视化方法研究, 电网技术, 25 (11): 9-11
72. 周明磊, 康重庆, 基于 Internet/Intranet 的电力市场技术支持系统, 电力系统自动化, 26 (23): 18-22
73. 曹荣章, 方艺, PMOS-2000 电力市场即时信息发布子系统, 电力系统自动化, 24 (17): 8-10
74. 胡泳, 张志浩, Java 与 Servlet 通信机制的探讨, 计算机系统应用, 2000 (4): 33-36
75. 贾智平, 李明, 基于 Internet 的电力监控组态软件的关键技术, 电力系统自动化, 26 (15): 62-65
76. 王萍, 王利军, 面向对象电网知识库系统的研究与实践, 电力系统自动化, 26 (11): 62-65
77. 曹文君, 知识库系统原理及其应用, 上海: 复旦大学出版社, 1995
78. 王萍, 岳斌, 电网计算数据库图形管理系统的研究与实践, 19 (5): 114-116
79. 吕飞鹏, 米麟书, 面向对象的电网零序保护整定计算 CAD 系统的研究, 电力系统自动化, 21 (11): 1997
80. 韩肖清, 图形化故障计算软件的开发, 电力系统自动化, 25 (4)
81. 张沛超, 胡炎, 继电保护专家系统中知识的面向对象表示法, 继电器, 29 (2)

82. 张强, 沈南, 基于信息网络平台和 Web 技术的变电站遥视系统, 中国电力, 34 (6)
83. 田华, 李青, 刘智勇, 基于 Web 电网调度检修管理系统的研究, 中国电力, 34 (7)
84. 提兆旭, 李源, 基于客户/服务器计算的电力企业 MIS 构造与优化研究, 中国电机工程学报, 19 (5)
85. 孙敏, 黄勇, 基于 Internet、CGI、Java 技术的电力信息发布系统及在线绘图, 继电器, 29 (7)
86. 张鹏, 郭勇基, 基于 Web 的配电系统远程检测的实现, 继电器, 28 (11)
87. 刘华奎, 韩建民, 基于 WWW 技术对传统信息系统改造的对策和方法, 微型机与应用, 18 (3)
88. 刘茂诚, 建立 3 层 C/S 结构的 MIS 权限设计, 微型机与应用, 2000 年 3 期
89. 陈文博, 夏长虹, 以组件对象为中心的动态 Web 开发方法, 计算机应用, 20 (6)
90. 张均, 姜永昊, 分布式 Web 组件模型分析, 微型机与应用, 18 (2)