

## 摘 要

随着网络技术的飞速发展,电子商务以其方便、快捷、高效的特点被越来越多的企业和顾客所认识和使用。电子商务使用了刊登广告并出售货物的新方法来进行交易,并为动态开放式电子商务环境中的大组客户提供服务和信息。它不仅是中国企业内部改革可以利用的先进技术,还可以促进国家经济结构调整。

本文对基于 CORBA 组件技术的电子商务系统进行了初步和相对实际的研究。论文共分为五部分。第一部分对电子商务系统做了总体地概括和介绍,主要包括对电子商务系统应具有的特性及相关技术的分析。此外,对当前市场上主流的客户关系管理进行对比和分析。论文第二部分分析了三种基于组件的分布式对象技术( DCOM、EJB 和 COBRA ) 的优缺点和适用场合,突出介绍了 CORBA 的优势,并通过工作流图形象地介绍了 CORBA 的工作原理。论文第三、四、五部分详细地介绍了于 CORBA 组件技术的电子商务系统的需求分析、设计和实现。对电子商务系统设计时,结合了 JAVA 和 CORBA 两种用于分布式应用程序开发的 WWW 开发技术,充分体现了该系统简单易行、可扩展、可移植等性能方面的优势。设计时也充分考虑了数据库的设计问题。论文第四部分以一个网上购物系统为例,结合该系统的具体设计过程,对系统功能进行了介绍。

第六部分总结了本文的不足之处,并对今后的工作做了进一步的展望。

关键词: CRM、CORBA、ORB、组件、电子商务

## Abstract

With the development of Internet, e-commerce (electronic commerce) is realized and utilized by more enterprises and customers for its advantages such as convenience, promptness and efficiency. E-commerce performs trades by means of advertising and selling goods, and provides services and information for large groups of customers in the dynamic open electronic commerce environments. It is not only advanced technologies used in enterprise's internal reform, but also can promote the regulation of national economic structure.

This thesis mainly studies on the CORBA-based e-commerce system, which includes five chapters. The e-commerce system, which includes the features and the technology applied, is introduced in chapter one. In addition, chapter one analyses and compares the popular CRM (Customer Relationship Management). Chapter two analyses the three main distributed object component technology (DCOM, EJB and CORBA), emphasizes on the kernel technology- component technology CORBA. CORBA's work principles are interpreted by using workflow diagrams. In chapter three, this thesis analyse the system's requirement. Chapter four introduces how to design the CORBA-based e-commerce system. JAVA and CORBA are combined in the system design, which has many advantages of being simple, transportable, reusable and secure. The design of database is important. Chapter five gives an example of shopping e-commerce system, and interprets the system functions of this system by combining the design process.

Lastly, this thesis summarizes the shortages of this thesis, and gives some advice on the future research.

**Keywords :** CRM、CORBA、ORB、Component、EC(Electronic Commerce)

## 学位论文版权使用授权书

本人完全了解同济大学关于收集、保存、使用学位论文的规定，同意如下各项内容：按照学校要求提交学位论文的印刷本和电子版本；学校有权保存学位论文的印刷本和电子版，并采用影印、缩印、扫描、数字化或其它手段保存论文；学校有权提供目录检索以及提供本学位论文全文或者部分的阅览服务；学校有权按有关规定向国家有关部门或者机构递交论文的复印件和电子版；在不以赢利为目的的前提下，学校可以适当复制论文的部分或全部内容用于学术活动。

学位论文作者签名：  
2006年5月22日

经指导教师同意，本学位论文属于保密，在 年解密后  
适用本授权书。

指导教师签名：

学位论文作者签名：

年 月 日

年 月 日

## 同济大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

签名：

2006年3月22日

# 第一章 引言

## 1.1 电子商务概念

科学技术的突飞猛进直接把我们带进了信息化和数字化的社会，计算机的应用已普及到经济和社会生活的各个领域，网络时代的出现改变了人们的生活方式的方方面面，让人可以足不出门就可以与世界同步，以最快的速度了解到新的消息和咨询，让人置身于一个全新的世界里，其中网上购物与交流对于智能生活来说是一个普遍而重要的部分，由此促成了电子商务的出现，但以前的服务效率不高。而长久以来，人们就希望有一个高效率的电子商务网站，让人们在购物、交流等更加方便。

电子商务（ELECTRONIC COMMERCE，简写为 EC）作为一种全新的交易技术方式，是建立在全新的网络基础之上，指的是利用简单、快捷、低成本的电子通讯方式，买卖双方不谋面地进行各种商贸活动。电子商务可通过多种电子通讯方式来完成。但现在人们所探讨的电子商务主要是以 EDI（电子数据交换）和 INTERNET 来完成的。作为一种新型的商务模式，电子商务具有普遍性、方便性、整体性、安全性、协调性等特征。

### 1.1.1 电子商务的模式

电子商务可以分为企业(Business)对终端客户(Customer)的电子商务(即 B2C) 和企业对企业的电子商务 (B2B) 两种主要模式。

#### (1) B2C

提起 B2C，大家可能更为熟悉一些，它是从企业到终端客户（包括个人消费者和组织消费者）的业务模式。今天所谈的电子商务时代的 B2C 是通过电子化、信息化的手段，尤其是互联网技术把本企业或其它企业提供的产品和服务不经任何渠道，直接传递给消费者的新型商务模式。因为它与大众的日常生活密切相关，所以被人们首先认识和接受。

电子商务 B2C 模式的一种最为大家所熟悉的实现形式就是新兴的专门做电

电子商务的网站，中有网上商店、网上书屋、网上售票等等。但无论怎样，这些新型模式企业的出现，使人们足不出户，通过因特网，就可以购买商品或享受资讯服务。这无疑是时代的一大进步。在这些新涌现出来的互联网公司中，亚马逊公司可以说是最具代表性的一例。

亚马逊最初是一家通过互联网售卖图书的网上书店，就在几乎谁都没有搞清它的店面在哪里的时候，它在短短的两年间一举超过无数成名已久的百年老店而成为世界上最大的书店，其市值更是远远超过了售书业务的本身。通过亚马逊的 WEB 网站，用户在购书时可以享受到很大的便利，比如要在 100 万种书中查找一本书，传统的方法可能要跑上几个书店，花费很多的时间，但在亚马逊，用户可以通过检索功能，只需点击几下鼠标，不久就会有人把想要的书送到家里了。亚马逊另一个吸引人的方面是提供了很多的增值服务，包括提供了众多的书籍评论和介绍。而在传统销售方式下，这些增值服务会变得非常昂贵。在“成功”地将自己发展成超越传统书店的世界最大规模书店之后，今天亚马逊的业务已扩展到音像制品、软件、各类日用消费品等多个领域，成为美国、也是全世界最大的电子商务网站公司。但它的这个“成功”现在还是划引号的，人们质疑它在建立起自己的规模和客户群的同时，给投资人所留下的巨额亏损。

人们在反思亚马逊的亏损原因时意识到，也许不应该将建立起电子商务时代 B2C 的任务全都寄托于这些白手起家的网站上，传统行业自觉的互联网和电子商务革命也许会更经济，更实惠，也更必要，不至于给投资人、给股民带去那么多的压力和担忧。也许，只有当这两股力量都齐齐奔向同一个山顶时，这样的电子商务世界才更精彩，真正的电子商务时代也才会更快一些到来。

B2C 的这二种实现方式还有一点很大的不同，由网站起家的 B2C 较难发展起自有品牌的产品、实业，因此它们更象是一个百货商店，当然与百货商店最不同的就是百货店是用户上门的，而 B2C 网站是送货上门的，而由传统企业改造而来的 B2C 更可能象是一个专卖店，专营自己品牌的产品，与传统专卖店不同的是：这里用户和厂商互动性更强，可以量身定做，同时由于省去了建物理店的开销，成本可能会降低。

## (2) B2B

企业与企业之间的业务模式被称作 B2B，电子商务 B2B 的内涵是企业通过内部信息系统平台和外部网站将面向上游的供应商的采购业务和下游代理商的销售业务都有机地联系在一起，从而降低彼此之间的交易成本，提高满意度。

实际上面向企业间交易的 B2B，无论在交易额和交易领域的覆盖上，其规模比起 B2C 来都更为可观，其对于电子商务发展的意义也更加深远。

与 B2C 相似，B2B 在企业间的应用也有两种主要实现形式。

B2B 的一种实现是其在传统企业中的应用。一些传统企业的实质性业务，正在逐步向 B2B 转变，更多地以 WEB 方式来传递信息和实现网上订单，但物流方式就和以前没什么变化，依然是供应商到本企业，本企业再到代理商或最终客户。以通用汽车为例，通用汽车建立了一个 B2B 电子商务网站——TradeXchange，计划在今年年底之间，将其每年高达 870 亿美元的采购业务完全通过该网站进行。并且这个网站不仅满足通用自身的采购业务，其 30000 多家供应商也将在这一系统上进行交易，它将对通过 TradeXchange 进行的电子商务交易收取 1% 的手续费，专家们估计这将为通用汽车带来每年 50 亿美元的收入。

不要把互联网和电子商务仅仅看作是一个工具，它同样可能对营销模式和管理模式带来变革，B2B 的第二种实现方式就有这样的意味。这一类的 B2B 公司并不是为自身企业的采购或销售服务的，它自身可能不生产任何产品，但它通过建立统一的基于 WEB 的信息平台，为某一类或某几类的企业采购或销售牵线搭桥，此时物流的方式就和上一类有很大不同了，它是由供应商直接到代理商。

### 1.1.2 电子商务的作用

电子商务极大提高了传统商务活动的效益和效率，与传统商务活动相比它具有下列竞争优势：降低交易成本、减少库存、缩短生产周期、增加商机、减轻物资的依赖。可以说，电子商务在生活的各个方面具有举足轻重的作用。电子商务的作用突出表现在以下四个方面：

1、大幅度降低信息成本，提高信息使用效率。电子商务降低了信息来源成本；突破行业和产品物理特性的限制，使交易范围急剧放大；弥补信息的不对称性，实现交易信息互换和交易行为的虚拟市场化。信息成本的低廉形成了对电子商务生存的最有力的支撑。

2、大量减少中间环节，降低销售成本和购买成本。电子商务为买卖双方在网上直接交易提供了现实可能性，缩短了供求之间的距离。绕过传统的经销商

而直接与客户沟通，企业可以将需求直接转化为企业的生产指令，减少了许多中间环节，使得零库存生产成为可能。

3、有利于形成高效流通、交换体制。电子商务构成了虚拟社会中的整个商品交易庞大网络，实体社会中商品的盲目实物移动转变为有目标的实物移动。借助于电子商务的信息沟通和需求预测，企业可以组织有效生产，形成高效流通、交换体制。

4、有利于推动产业结构的优化和升级。电子商务以新供给创造了新需求，又以新需求推动了新供给，促进了社会总供求的良性互动，推动了经济的持续稳定增长，从而延长了经济扩张期，减小了经济波动的幅度。

### 1.1.3 我国电子商务的现状

20世纪90年代以来，电子商务在催生新经济和推动经济全球化中所表现出来的巨大能量，成为评价一国经济发展水平和可持续发展能力的重要指标。

2005年1月25日国务院办公厅《关于加快电子商务发展的若干意见》正式出台，4月1日电子签名法正式实施，身份认证、社会诚信和银行支付体系也不断完善，我国电子商务的发展环境总体趋好。2004年度全国企业信息化500强入选企业总体规模有所提高，入选企业最低年主营业务收入由2003年的2亿元提高到5亿元。从行业分布来看，81.2%的企业来自制造业，比2003年度的77.4%又有所增加；从企业规模分布来看，年主营业务收入超过百亿的企业75家，其中千亿级企业近10家；从企业CIO制度建设方面来看，36.4%的500强入选企业设置了CIO职位；从企业信息化建设的导向来看，业务部门平均拥有对项目规划、实施和预算决策的参与和支配权达50%以上；从电子商务的普及应用来看，42.2%的500强入选企业不同程度的开展了电子商务活动，比2003年度的39.2%有所增长。但在总体上还没有达到应有的水平。

目前我国电子商务活动总体水平较低。据赛迪顾问(CCID Consulting)对10个城市13个行业600多家企业进行调查，只有22.3%的企业参与了电子商务，9.3%的企业开展供应链管理，15.6%的企业开展分销渠道管理，51.4%的企业进行网上销售，72.9%的企业进行信息查询，72.4%的企业进行网上信息发布。2004年，中国约26.1%的企业参与电子商务。大型企业参与电子商务最积极，比例达32.1%，中、小型企业比例分别为27.8%和21.4%。目前中国

有超过 1100 万家中小企业，其中已实现信息交互的占 1/4，而真正实现 B2B 网上交易的不到 18%<sup>[1]</sup>。

我国不少企业对电子商务认识不到位。在受访的企业中，80. 4%的企业认为开展电子商务有助于产品宣传，77. 7%的企业认为可以提高企业知名度，只有 43. 6%企业选择了降低生产、销售、运营成本。转型过程中的经济体制、企业制度以及经济运行方式也不尽适应发展电子商务的要求。

#### 1.1.4 电子商务涉及到的技术及存在问题

电子商务系统中，涉及到许多新的技术，如：

- (1) 网络通讯技术。包括网络设备、移动通讯系统。
- (2) 安全技术。保护 Internet 上的交易；保证数据的安全传送；提高站点及企业网络抵抗攻击的能力。包括防火墙技术，网络安全监控技术，信息加密技术等。
- (3) 认证技术。Internet 是一个开放的系统，信息发送者，接收者的身份仅仅依靠 Internet 是无法确定的，需要有一个权威的认证机构，以执行 Internet 上用户及商家的身份认证；另外银行还有信用认证问题。
- (4) 电子支付技术。电子资金转帐技术（电子支票、数字现金、网络银行），数据自动俘获技术（磁卡、IC 卡、信用卡读写设备），银行清算系统等。

目前的电子商务系统，或多或少的存在一定问题。

现在的电子商务只是在对通用方针和平台意见一致的参与者间的封闭组织内进行。例如，电子数据交换（EDI）被用来在一个机构的多个分支之间，或者在建立了契约联系的机构之间安全地传输数据。而在这些早期阶段，电子商务系统只处理某几个方面的完全商务事务。

当客户可以通过他们的 Web 浏览器来使用的第一批基于 web 的商店出现时，建立了电子商务的一个更全面的概念作为 Internet 上传递货物和价格的方式。现在，大多数电子商务系统是基于 web 的，并且允许客户通过他们的 web 浏览器购买货物并用信用卡结帐。然而，基于 web 的应用程序的局限功能使得很难向客户提供全范围的服务。

未来，电子商务解决方案的需求将超过当前级别。将来的系统将必须通过多个自治的服务提供商来满足动态开放式环境中的需求，电子商务将成为一个

包含多个交易实体间复杂的交互作用的分布式过程。在一个开放式市场中，有许多独立的货物和服务的供应商，并且可能有通过合并第三方提供的服务来提供服务的调解者。客户本身也可能合并随选（on-demand）产品或者服务来实现合成包。因此，现代的电子商务系统必须能集成不同种类参与系统和不同政策领域中互相不信任的用户。

Internet 的爆发增长，使得通过为一大群顾客和供应商提供一个通用通讯环境的方法可以发挥电子商务的独一无二的潜力。今天，网上有数以千计的面向消费者和面向交易的商务站点，并且这个数目正在快速增长。然而，电子商务成为世界新热点的同时，其安全性和可操作性也随着信息化的深入也随之要求愈高了。快速和不受控制的增长产生了组织和技术天性方面的不同问题。市场依旧是封闭的，并且常常没有完全符合顾客和提供者的需求。今天的电子商务系统在私人拥有的平台上运行，因此应用程序并不能互操作，也不能建立在对方的基础上。安全性和支付系统仍然不成熟，并且常常是不相称的。只有用标准的电子商务框架才能解决这些问题。

未来的电子商务系统的友善性和安全性方面的问题是它们必须通过复杂的组件技术和信托关系在一个动态并开放的，从而也是不受控制的环境中操作。多数电子商务使用的电子支付系统必须很容易使用的，也必须透明地提供鉴定、完整性保护、机密性保护和认可。另外，客户和提供者之间的通讯连接必须保持数据的机密性和完整性，首先保护客户的隐私，其次是确保客户购买的服务不能被篡改<sup>[2]</sup>。

## 1.2 客户关系管理背景

电子商务的三大主要功能即提高效率、降低交易成本和密切客户关系，在跨国公司中的应用已充分显示出了其竞争优势。在这三大功能中随着商业竞争全球化的加剧，客户关系已成为企业制胜之道。

客户关系管理，是企业为提高核心竞争力，达到竞争致胜，快速成长的目的，选择、争取、发展和保持客户需要的全部商业过程。客户关系管理本身已经是企业的一种战略选择，因为在今天，客户已成为企业最宝贵的资源之一。管理好客户关系，不是保护资源最有效的战略，就是实现利润最大化的有效方

法。

客户关系管理目的是通过企业不断改进与客户关系的流程，提高客户满意度和忠诚度，提高企业效率和利润水平，因为所有企业都离不开客户，没有客户一切都是空谈！

客户关系管理不仅帮企业从竞争对手那里争夺客户、留住客户，也帮助企业提高客户满意度，最大化客户价值，从而提高企业利润。而客户关系管理的研究核心在于对电子商务网站登陆系统的研究，毕竟与客户直接打交道的是你的电子商务登陆平台，因此，要想抓住这一商机，吸引更多的客户，就必须要让你的客户管理系统与众不同，要更具有友善的界面、强大的功能和完善的安全机制，毕竟客户直接面对的是你的客户管理系统。很不幸，今天的（基于 web 的）客户管理系统不太能迎合这些关于功能性和安全性的需求。

### 1.3 主流软件对比分析

目前市场上主流的客户管理系统有用友的 CRM、创能 CRM 等。

创能 CRM 作为专门管理企业前台的软件，提供了一个收集、分析和利用各种客户信息的系统，帮助企业充分利用其客户关系资源，也为企业在电子商务时代从容自如地面对客户提供了科学手段和方法。使企业充分共享内部的资源，通过一个统一的视角，借助多渠道的方式与其客户进行交流<sup>[3]</sup>。

用友CRM (Customer Relationship Management) ——是一整套用于改变企业的传统思维模式的企业哲学；是一整套用于帮助企业持续盈利和发展的商业模式；是一整套用于帮助企业规范业务流程管理的管理工具；用友客户关系管理软件，是实施于企业市场营销、服务与技术支持等与客户相关的领域，通过多渠道（面对面直接销售、渠道销售、呼叫中心、Internet、现场服务等）的客户交互管理，集中式数据管理，和数据分析以实现客户价值最大化的信息系统和管理系统。是用友经过充分深入调研、精心细致开发的又一套优秀系统软件，它通过提供快速和周到的优质服务吸引和保持更多的客户；通过对业务流程科学有效的管理降低企业的成本；帮助企业搜集、追踪和分析每一个客户，充分了解并满足用户的个性化需求<sup>[4]</sup>。

无论在影响还是在技术上，用友的CRM都要超过创能CRM。用友的CRM跨

平台能力——便于部署和移植、可伸缩性——便于系统配置和扩展、高效运行、安全可靠。这些都是创能无法拟的。

## 1.4 实现技术原理简介

CORBA (Common Object Request Broker Architecture, 即公共对象请求代理体系结构) 是由 OMG (Object Management Group, 对象管理组织) 提出的应用软件体系结构和基于 OMA 对象的技术规范。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(OMG 组织是一个国际性的非盈利组织，其职责是为应用开发提供一个公共框架，制订工业指南和对象管理规范)。与过去的面向过程的 RPC(Remote Procedure Call)不同，CORBA 是基于面向对象技术的，它能解决远程对象之间的互操作问题。CORBA 是真正跨平台的，平台独立性正是 CORBA 的初衷之一。另一种做到平台无关性的技术是 Java RMI(Remote Method Invocation)，但它只能用 JAVA 实现。CORBA 与此不同，它通过一种叫 IDL (Interface Definition Language) 的接口定义语言，能做到语言无关，也就是说，任何语言都能制作 CORBA 组件，而 CORBA 组件能在任何语言下使用<sup>[5]</sup>。

因此，可以这样理解 CORBA：CORBA 一种异构平台下的语言无关的对象互操作模型。

## 1.5 本文所做的工作

本文工作主要集中于以下几点：

分析讨论了现有电子商务系统的现状和开发技术的优势与限制，论证了建设基于 CORBA 组件技术的电子商务的现实性和必要性。总结了电子商务系统开发技术的现状和发展前景。

在以上分析的基础上，本文经过分析比较，选择了以 CORBA 组件技术为基础进行了网上电子商务系统的模型的设计，并进行了实际系统的开发。

论文各章节安排如下：

第 2 章对三种主要的组件技术做了分析比较，并详细介绍了 CORBA 组件

技术的原理。包括 ORB(对象请求代理)、IDL（接口定义语言）和 GIOP/IOP 等。

第 3 章介绍了电子商务系统的需求分析。包括网站模式的确定、网站的功能分析、工作流程分析和系统的开发运行环境等。

第 4 章是对基于 CORBA 组件技术的电子商务系统的设计，分别介绍了系统的解决方案、系统的总体设计和数据库的设计。

第 5 章是实现部分。介绍了 CORBA 的 ORB 代码的实现代码、模块代码实现、网站的主要功能界面。

第 6 章对所做的工作做了个总结，并提出了系统的不足之处，最后对组件技术的发展进行了展望。

## 1.6 本章小结

在这一章中首先讲述了电子商务的概念、作用、模式以及目前的电子商务存在的问题；其次分析了客户管理关系的背景，由客户管理关系的背景来引出在电子商务系统开发中使用 CORBA 组件技术的必要性；然后对目前市场上主流的客户管理系统用友的 CRM、创能 CRM 等进行了对比分析；接着简单介绍了 CORBA 组件技术，最后对本文所做的工作和章节安排进行了简单介绍。

## 第二章 技术原理综述

### 2.1 组件技术原理综述<sup>[6]</sup>

分布对象技术是伴随网络而发展起来的一种面向对象的技术。以前的计算机系统多是单机系统，多个用户是通过联机终端来访问的，没有网络的概念。网络出现后，产生了 Client/Server 的计算服务模式，多个客户端可以共享数据库服务器和打印服务器等等。随着网络的更进一步发展，许多软件需要在不同厂家的网络产品、硬件平台、网络协议异构环境下运行，应用的规模也从局域网发展到广域网。在这种情况下，Client/Server 模式的局限性也就暴露出来了，于是中间件应运而生。中间件是位于操作系统和应用软件之间的通用服务，它的主要作用是用来屏蔽网络硬件平台的差异性和操作系统与网络协议的异构性，使应用软件能够比较平滑地运行于不同平台上。同时中间件在负载平衡、连接管理和调度方面起了很大的作用，使企业级应用的性能得到大幅提升，满足了关键业务的需求。但是在这个阶段，客户端是请求服务的，服务器端是提供服务的，它们的关系是不对称的。

随着面向对象技术的进一步发展，出现了分布式对象技术。可以这么说，分布式对象技术是随着网络和面向对象技术的发展而不断地完善起来的。分布对象计算中，通常参与计算的计算体（分布对象）是对称的。分布对象往往又被称为组件（Component），组件是一些独立的代码的封装体，在分布计算的环境下可以是一个简单的对象，但大多数情况下是一组相关的对象复合体，提供一定的服务。分布环境下，组件是一些灵敏的软件模块，它们可以位置透明、语言独立和平台独立地互相发送消息，实现请求服务。组件是包含可以更改的数据形式的代码的对象，是可用代码的封装，这些代码可以用来执行应用程序的一些功能，如从数据库中检索某些信息等功能。它提供客户端在运行时候请求的对象，将这些文件注册到服务器上，然后就可以在脚本文件中创建它的实例，并使用它的属性和方法（即接口）来完成一定功能。组件像 VB、VC 中的控件一样，把功能和实现细节封装隐藏，并且可被不同应用程序重复使用。组件的功能可以做得十分强大，甚至可以调用 Win32API 或编程语言的任何其它

底层功能。组件有许多好处，例如将你的程序做成组件后，你的程序代码就被保护起来，别人也看不见了。而且在多线程调用和需要早期绑定的情况下，使用 DLL（动态链接库）会大大增强性能。

使用现成的组件来开发应用程序时，组件工作在两种模式下：设计时态和运行时态。

设计时态下组件的方法不能被调用，组件不能与最终用户直接进行交互操作，也不需要实现组件的全部功能。

在运行状态下，组件工作在一个确实已经运行的应用程序中。组件必须正确地将自身表示出来，它需要对方法的调用进行处理并实现与其他组件之间有效的协同工作。

设计时态下所有的组件在窗体中都是可见的，但在运行时态下不一定可见。如 TTable、TQuery 和 TDataSet 在运行时态下就不可见，但他们均完成了重要的功能。

创建组件就是自行设计制作出新的组件。

设计组件是一项繁重的工作。自行开发组件与使用组件进行可视化程序开发存在着极大的不同，要求程序员熟知原有的 VCL 类库结构，精通面向对象程序设计。设计组件是一项艰苦的工作。对于组件的开发者，组件是纯粹的代码。组件的开发不是一个可视化的开发过程，而是用 C++ 或 Object Pascal 严格编制代码的工作。实际上，创建新组件使我们回到传统开发工具的时代。虽然这是一个复杂的过程，但也是一个一劳永逸的过程。创建组件的最大意义在于封装重复的工作，其次是可以扩充现有组件的功能。组件创建过程包括设计、开发、调试（就是所谓的 3D's）工作，然后是组件的使用。组件开发者应该掌握的三项主要内容是：属性、事件和方法。

引入中间件（MiddleWare）作为事务代理，完成客户机（Client）向服务对象方（Server）提出的业务请求。

引入中间件概念后分布计算模式如图 2.1 所示。

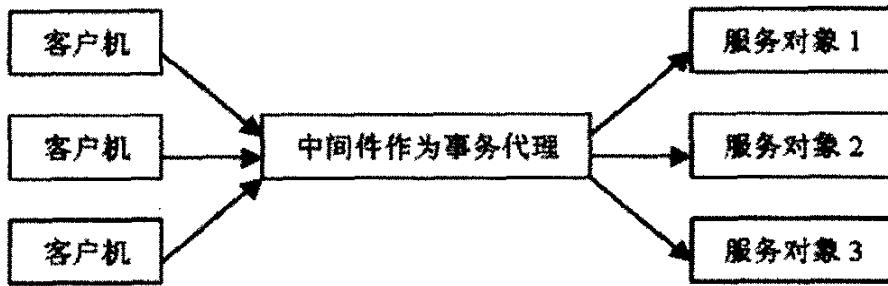


图 2.1 引入中间件技术后的分布模式

## 2.2 三大组件技术对比分析

电子商务系统从技术的角度来看也是一个计算机系统，同样包括硬件和软件系统两大部分。另外电子商务本身是必须建立在网络之上的，由于网络计算的分布和异构性，所以在结构上，电子商务系统也必须采用分布式计算的系统结构。目前，构架电子商务服务应用框架的分布式对象技术主要有三种：Microsoft 的 COM/DCOM 技术、Sun 公司的 RMFEJB 技术和 OMG 的 CORBA 技术。其中 CORBA 技术是三者之中出现最早的，它已经成为一种标准，它的分布计算能力、互操作性和开放性都非常好；微软的 COM/DCOM 技术一般只局限于 Windows 平台；Java 技术的跨平台性很好，其分布对象技术 RMI 和 EJB 也不错，现在 JAVA 技术和 CORBA 技术融合是一种趋势。

### 2.2.1 Sun 的 EJB

#### 1、J2EE/EJB 技术简介

Sun 在 1999 年底推出了 Java2 技术及相关的 J2EE 规范，J2EE 的目标是：提供平台无关的、可移植的、支持并发访问和安全的，完全基于 Java 的开发服务器端中间件的标准。

在 J2EE 中，Sun 给出了完整的基于 Java 语言开发面向企业分布应用规范，其中，在分布式互操作协议上，J2EE 同时支持 RMI 和 IIOP，而在服务器端分布式应用的构造形式，则包括了 Java Servlet, JSP(Java Server Page), EJB 等多种

形式，以支持不同的业务需求，而且 Java 应用程序具有’ ‘Write once, run anywhere” 的特性，使得 J2EE 技术在分布计算领域得到了快速发展。

J2EE 简化了构件可伸缩的、基于构件服务器端应用的复杂度，J2EE 是一个规范，不同的厂家可以实现自己的符合 J2EE 规范的产品，J2EE 规范，是众多厂家参与制定的，它不为 Sun 所独有，而且其支持跨平台的开发，目前许多大的分布计算平台厂商都公开支持与 J2EE 兼容技术。

EJB 是 Sun 推出的基于 Java 的服务器端构件规范 J2EE 的一部分，自从 J2EE 推出之后，得到了广泛的发展，已经成为应用服务器端的标准技术。SunEJB 技术是在 Java Bean 本地构件基础上，发展的面向服务器端分布应用构件技术。它基于 Java 语言，提供了基于 Java 二进制字节代码的重用方式。EJB 给出了系统的服务器端分布构件规范，这包括了构件、构件容器的接口规范以及构件打包、构件配置等的标准规范内容。EJB 技术的推出，使得用 Java 基于构件方法开发服务器端分布式应用成为可能。从企业应用多层结构的角度，EJB 是业务逻辑层的中间件技术，与 JavaBeans 不同，它提供了事务处理的能力，自从三层结构提出以后，中间层，也就是业务逻辑层，是处理事务的核心，从数据存储层分离，取代了存储层的大部分地位。从分布式计算的角度，EJB 像 CORBA 一样，提供了分布式技术的基础。提供了对象之间的通讯手段。从 Internet 技术应用的角度，EJB 和 Servlet}JSP 一起成为新一代应用服务器的技术标准，EJB 中的 Bean 可以分为会话 Bean 和实体 Bean，前者维护会话，后者处理事务，现在 Servlet 负责与客户端通信，访问 EJB，并把结果通过 JSP 产生页面传回客户端。

## 2、在电子商务应用平台中的评价

J2EE 的优点是，服务器市场的主流还是大型机和 UNIX 平台，这意味着以 Java 开发构件，能够做到” Write once run anywhere”，开发的应用可以配置到包括 Windows 平台在内的任何服务器端环境中去。

但 EJB 纯语言的优势同时也是它的一个劣势。实际的应用系统一般是基于多种不同高级语言开发的，只依赖一种语言构建的大型网络应用是很少见的。正是这一点限制了它在解决复杂的分布式问题上有更大作为。目前，Java 技术和 CORBA 技术有融合的趋势。

## 2.2.2 Microsoft 的 DCOM

### 1、DCOM/COM 技术简介

DOOM 是 Microsoft 的 COM(组件对象模型, Component Object Model)的分布式扩展, 是 Microsoft 与其它业界厂商合作提出的一种分布构件对象模型。COM 服务器能创建多对象类的对象实例。一个 COM 对象可以支持多个接口, 每个接口代表对象的一种不同的视图或行为。一个接口由一套功能相关的方法组成。COM 的客户程序通过获取指向一个对象接口的一个指针, 并通过该指针来调用方法以实现与 COM 对象之间的互相作用, 就好像对象驻留在客户程序的地址空间中一样。

在设计和实现分布式应用系统时, 一个普遍的问题就是为开发一个特定的组件而选择语言以及工具的问题。语言选择是一个典型的在开发费用、可得到的技术支持以及执行性能之间的折衷。作为 COM 的扩展, DCOM 规范是二进制级别的, 具有语言独立性。任何语言都可以用来创建 COM 组件, 并且这些组件可以使用更多的语言和工具。Java, Microsoft Visual C++, Microsoft Visual Basic, Delphi, PowerBuilder 和 Micro Focus COBOL 都能够和 DCOM 很好地相互作用。

### 2、DCOM/COM 特点

遵循 COM 的规范标准, 使组件与应用、组件与组件之间可以相互操作, 极其方便地建立可伸缩的应用系统。从工程的角度看, 组件在应用开发方面具有以下特点:

(1) 组件与开发的工具语言无关。开发人员可以根据需要和爱好选择特定语言工具实现组件的开发。语言独立性还使得一些原型组件开始时可以用诸如 Visual Basic 这样的高级语言来开发, 而在以后用一种不同的语言, 例如 Visual C++ 和 Java 来重新实现, 而这种语言能够更好地支持诸如 DCOM 的自由线程/多线程以及线程共用这些先进特性。

(2) 通过接口有效保证了组件的复用性。一个组件具有若干个接口, 每个接口代表组件的某个属性或方法。其它组件或应用程序可以设置或调用这些属性和方法来进行特定的逻辑处理。组件和应用程序的连接是通过其接口实现的。负责集成的开发人员无须了解组件功能的如何实现, 而只须创建组件对象与其接口建立连接。在保证接口一致性的前提下, 可以调换组件、更新版本, 也可

以将组件应用在不同的系统中。

(3) 组件运行效率高、便于使用和管理。因为组件是二进制代码，比 ASP 脚本运行效率高，所以核心的商务逻辑计算任务必须由组件来负责，ASP 脚本只起组装组件的角色。而且组件在网络上的位置可被透明地分配，组件和使用的程序既能在同一进程中运行，也可在不同进程或不同的机器上运行。组件之间是相互分离和独立的。利用 MTS 或 COM+对组件管理更加方便。

### 3. 在电子商务平台中应用的评价

它致命的弱点就是跨平台性较差，开发一个 DCOM 应用程序，分布式应用中所有参与的节点都必须以 Windows 风格运行。如何实现与第三方厂商的互操作性始终是它的一大问题。这样极大的局限了它在解决分布式网络资源整合方面的应用。

#### 2.2.3 OMG 的 CORBA

CORBA 是为了实现分布式计算而引入的。首先，在企业级应用中，硬件系统集成商基于性能、价格、服务等方面的考虑，通常在同一系统中集成来自不同厂商的硬件设备、操作系统、数据库平台和网络协议等，由此带来的异构性给应用软件的互操作性、兼容性以及平滑升级能力带来了严重问题。另外，随着基于网络的业务不断增多，传统的客户/服务器(C/S)模式的分布式应用方式越来越显示出在运行效率、系统网络安全性和系统升级能力等方面局限性。

为了解决分布式计算环境中不同硬件设备和软件系统的互联，增强网络间软件的互操作性，解决传统分布式计算模式中的不足等问题，对象管理组织(OMG)提出了公共对象请求代理体系结构，以增强软件系统间的互操作能力，使构造灵活的分布式应用系统成为可能。

CORBA (Common Object Request Broker Architecture) 表示公共对象请求代理体系结构，是由对象管理工作组 (Object Management Group , OMG) 基于众多开放系统平台厂商提交的分布对象互操作内容的基础上制定的公共对象请求代理体系规范<sup>[9]</sup>。属于网络分布式体系结构，核心为一套标准的语言、接口和协议，以支持异构分布应用程序间的互操作性及独立于平台和编程语言的对象重用。软件实现上一般采用总线结构，建立动态的客户程序和服务器程序之间的调用关系，即对象中间件。它被列为是主流远程体系结构。CORBA 分

布计算技术，是由绝大多数分布计算平台厂商所支持和遵循的系统规范技术，具有模型完整、先进，独立于系统平台和开发语言，被支持程度广泛的特点，已逐渐成为分布计算技术的标准。

应该说，这三者之中，COBRA 标准是做的最漂亮的。COBRA 标准主要分为 3 个层次：对象请求代理、公共对象服务和公共设施。最底层是对象请求代理 ORB，规定了分布对象的定义（接口）和语言映射，实现对象间的通讯和互操作，是分布对象系统中的“软总线”；在 ORB 之上定义了很多公共服务，可以提供诸如并发服务、名字服务、事务(交易)服务、安全服务等各种各样的服务；最上层的公共设施则定义了组件框架，提供可直接为业务对象使用的服务，规定业务对象有效协作所需的协定规则。总之，CORBA 的特点是大而全，互操作性和开放性非常好。

对于 CORBA 应用的保密性，完整性，身份鉴别我们可以采用加密的方法解决。SSL 包括：各种加密算法（DES, RSA, IDEA, RC2, RC4, Blowfish），各种检验和机制（MD2, MD5, SHA），证书函数（X.509）等。我们可以用 SSL 协议对 IIOP 通信进行加密，实现 Internet 上的 CORBA 应用的保密性，完整性和身份鉴别。

另一种常见的安全技术是防火墙技术。一般的基于 Tcp/ip 层的防火墙对低层的网络层、传输层的攻击能很好的防护。而应用层防火墙能提供很好的授权和访问控制功能，同时它还能对内容进行检查。因此，可以将防火墙技术嵌入 CORBA 应用的模型里，为 Internet 上的 CORBA 应用提供安全保护。

最后，CORBA 安全服务(CORBASec)是 CORBA 中一项重要的公共对象服务，它在 CORBA 客户对象和服务对象之间建立安全语言环境，为 CORBA 应用提供很好的安全服务。CORBA 在接口实现定义和接口实现的分离，大大提高了安全性和分布性，是当今要达到信息安全化的一个首要的选择。支持多种现存语言、可在分布式应用中混用多种语言、支持分布对象、提供高度的互通性。而 EJB 虽然非常简洁，但是安全性不如 CORBA，这主要是由于 EJB 层只包含很少的约束，必定造成系统的不安全隐患。

## 2.3 基于 CORBA 电子商务的优点

CORBA 作为电子商务系统的底层结构有许多优点，本节概述其中几个优点。

开放式电子商务系统的两个主要需求是互操作性和完整性。所有的客户和供应商应用程序都应该可以在一个灵活的、动态的、开放的框架中，越过不同平台，不同编程语言和商业布局来互操作。CORBA 可以从开放的电子商务环境的复杂性中抽象出来。CORBA 方便了电子商务系统和其它系统之间的交互作用，比如股票管理系统、会计系统、行销系统等，并使得和以前的应用程序之间集成变得容易，例如，一个旧的股票数据库系统。

从软件开发者的观点来看，CORBA 使得一切都变得比较简单，尤其是如果打算进行不同的商店配置时。CORBA 抽象了网络和动态的远程商店调用，允许应用程序开发者集中精力在实际的程序上，而不是集中在底层结构的内部工作方式上。应用程序开发者可以再利用已存在系统中的部分(例如安全性系统)来开发新程序。CORBA 的灵活结构也使得开发者可以实现整个商业街的一部分来迎合特殊的贸易需求，并为进一步增强系统和容易地升级这部分商业街软件提供坚实的基础。将来，个别基于 CORBA 的可定制的商业街组件就可用，可以购买它，并可以很容易的将之即插即用进已存在的商业街中，来增强或升级商店系统，。

为了使得商店组件的动态互用性运转起来，一套定义良好的标准服务需要在电子商务环境下可用。例如，用来描述对象（例如货物、服务、合同、发票或帐单等）的语义需要广泛定义。因此，OMG 和商业网（CommerceNet）共同定义了一系列电子商务服务的需求，也就是说，语义数据工具，选择/商议工具和支付服务。语义数据工具提供了对电子市场参与者之间语义信息交换的支持，商议服务提供了一组从事商业事务的参与者，在服务或工具的选择和配置上相互协定的支持，而电子支付工具关注支付协议的调用。

实际上，CORBA 还是经常被认为是一项不成熟的技术，尤其是 CORBA 没有实现许多服务，例如安全性服务。除了与不成熟的 ORB 实现相关的问题外，软件开发者也往往没有完全训练到可以熟练编写基于 CORBA 的组件的地步。就本地程序来说，基于 CORBA 的应用程序开发几乎与普通的应用程序开发相

同，因此并没有真正造成问题，但是例如实现一个透明的提供了 ORB 层安全性的安全性服务就需要专业知识<sup>[10]</sup>。

目前，对基于 CORBA 的电子商务系统的评价使得这些开发足以以为一些公司盈利了。例如，银行配置基于 CORBA 的个人银行业，或者股票交易系统可能因为它是顾客的最主要地边缘技术的服务提供者而获益。

## 2.4 CORBA 实现技术原理

CORBA 是一个用于创建分布式应用程序的开发的标准解决方案。使用 CORBA，系统可以直接在网络上进行通信，而不管使用了什么程序设计语言来创建系统，也不管对象运行在哪些操作系统和平台上。每个常用的环境都可以使用 CORBA 解决方案，并且可以将它用于集成 Java、C++、Visucal Basic 和 COBOL 编写的应用程序，它可以运行在嵌入式系统、PC、UNIX 主机和大型机上。如果 CORBA 在这些环境中运行，那么应用程序可以无缝地交互。除了它广泛的可用性以外，CORBA 还具有广泛的基层结构，该结构支持分布式业务对象需要的所有功能。该结构包括诸如交易、安全性和消息发送等重要的分布式服务。

CORBA 的体系结构如图 2.2 所示。

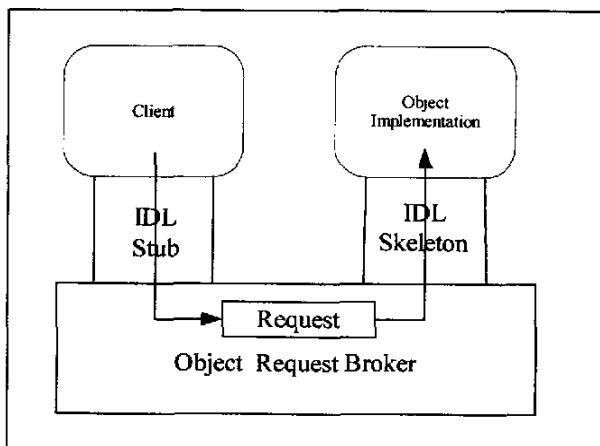


图 2.2 CORBA 体系结构

CORBA 上的服务用 IDL 描述，IDL 将被映射为某种程序设计语言如 C++

或 Java，并且分成两分，在客户方叫 IDL Stub(桩)，在服务器方叫 IDL Skeleton(骨架)。两者可以采用不同的语言。服务器方在 Skeleton 的基础上编写对象实现(Object Implementation)，而客户方要访问服务器对象上的方法，则要通过客户桩。而双方又要通过 ORB (Object Request Broker，对象请求代理) 总线通信。

与传统的 Client/Server 模式（我们称为 Two-tier client/server）不同，CORBA 是一种 multi-tier client/server architecture，更确切的说，是一种 three-tier client/server 模式。双重客户/服务器模式存在的问题是两者耦合太紧，它们之间采用一种私有协议通信，服务器的改变将影响到客户方。多重客户/服务器与此不同，两者之间的通信不能直接进行，而需要通过中间的一种叫代理的方式进行。在 CORBA 中这种代理就是 ORB。通过它，客户和服务器不再关心通信问题，它们只需关心功能上的实现。从这个意义上讲，CORBA 是一种中间件(Middleware)技术<sup>[11]</sup>。

### 2.4.1 ORB (Object Request Broker) 对象请求代理

ORB (Object Request Broker 对象请求代理) 是 CORBA 体系中最重要的组成部分，它为体系中的所有对象提供通信中枢。

在传统的基于客户机/服务器模式的应用程序开发过程中，开发人员必须遵循公开的标准或自由设计模块间的协议，这样的协议依赖于网络类型、实现语言、应用方式等。引入了 ORB 后，客户只要遵循服务对象的对外接口标准向服务对象提出业务请求，由 ORB 在分布式对象间建立客户/服务对象关系。

#### 2.4.1.1 ORB 在 CORBA 中的作用

- (1)、接受客户发出的请求，完成请求在服务对象端的映射；
- (2)、自动设定路由寻找服务对象；
- (3)、提交客户参数；
- (4)、携带服务对象计算结果返回客户端。

CORBAR 的工作过程中，ORB 代理取得不可缺少的作用，也就是说 ORB 工作流程就是 CORBAR 工作流程。

由图 2.3 可以看出，客户(Client)向服务对象的实现(Object Implementation)

发出事务请求，其中客户是欲对服务对象发出方法请求的实体，服务对象应包括该方法的数据资源以及实现代码。对象请求代理的作用就在于定位服务对象，接收客户发出的服务请求并将服务对象的执行结果返回给客户。请求发出后，客户对象采用轮询等方式以获取服务对象计算的结果。上面的图形说明的是客户端发送一个请求到对象的实现。客户端是希望对某对象执行操作的实体。对象的实现是一片代码和数据来实际实现对象。ORB 负责下面的必要的机制：对该请求找到对象的实现，让对象的实现准备好接受请求，和请求交换数据。客户端的接口完全独立于对象的位置，其实现的语言和其他不影响对象接口的东西<sup>[12]</sup>。

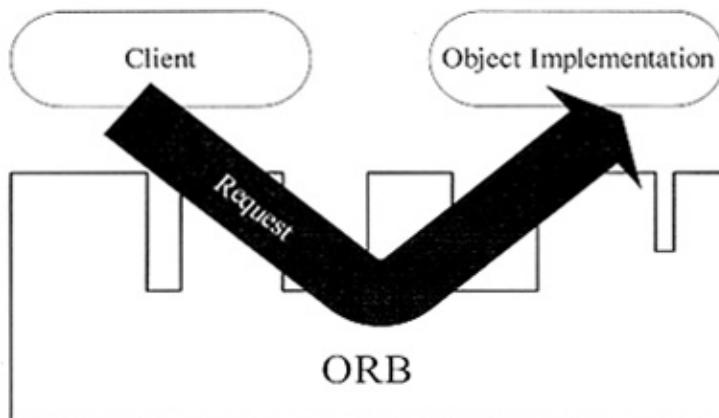


图 2.3 ORB 在 CORBA 中的作用

#### 2.4.1.2 ORB 的结构及类型

ORB 通过一系列接口和接口定义中说明的要实现操作的类型，确定提供的服务和实现客户与服务对象通信的方式。通过 IDL 接口定义、接口库或适配器（Adapter）的协调，ORB 可以向客户机和具备服务功能的对象实现（Object Implementation）提供服务。作为 CORBA 体系结构的核心，ORB 可以实现如下三种类型的接口<sup>[13]</sup>：

- (1)对于所有的 ORB 实现具有相同的操作；
- (2)针对特定类型对象的操作；
- (3)与对象实现类型有关的操作。

图 2.4 显示的是一个独立的对象请求代理 (ORB) 的结构。ORB 的接口是灰色的矩形。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 Interface) (和目标对象的接口独立) 或者一个 OMG 的 IDL 占位程序 (具体的占位程序依赖于目标对象的接口)。客户端也可以直接和 ORB 在某些地方交互。

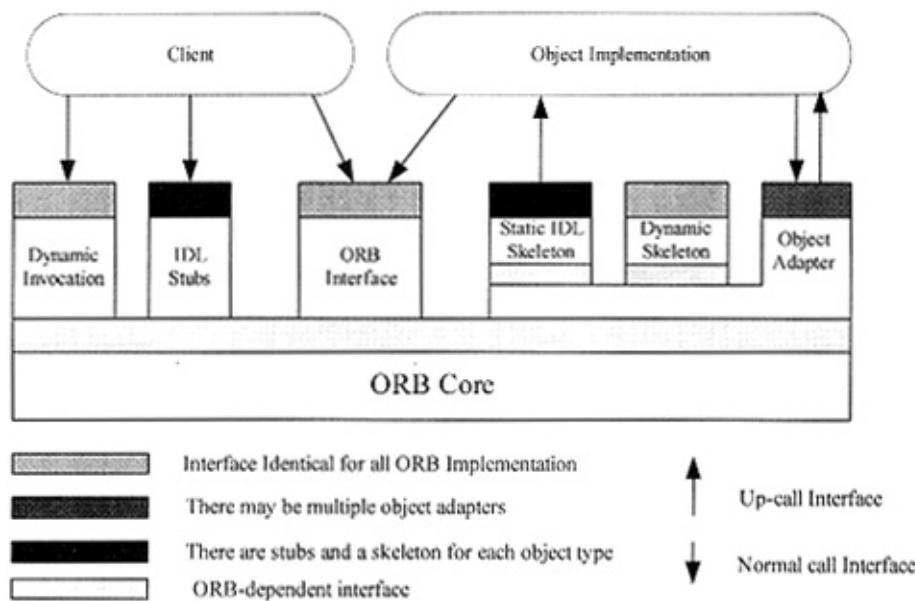


图 2.4 ORB 接口结构

对象的实现通过 OMG 的 IDL 产生的骨架或者是一个动态骨架的调用来接受请求。对象的实现可能在处理请求或其他的时候调用 ORB。

对象接口定义的定义可以有下面两种方式。接口可以通过接口定义语言静态的定义，这叫做 OMG 的 IDL，在下一节有详细介绍。该语言按照可以进行的操作和该操作的参数定义对象类型。或者（也可以作为补充），接口可以加入到 Interface Repository service。该服务描述了该接口作为一个对象的组件，并允许运行时访问这些组件。在任何 ORB 实现中，IDL 和 Interface Repository 有相同的表达能力。

客户端使用占位程序或者动态调用接口如图 2.5 所示。

客户端通过访问对象的对象引用和了解对象的类型及要求执行的操作来发布一个请求。客户调用占位程序例程来请求或者动态构造请求。

无论动态还是占位程序的接口都可以相同实现。接收方不可能知道请求是如何发布的。

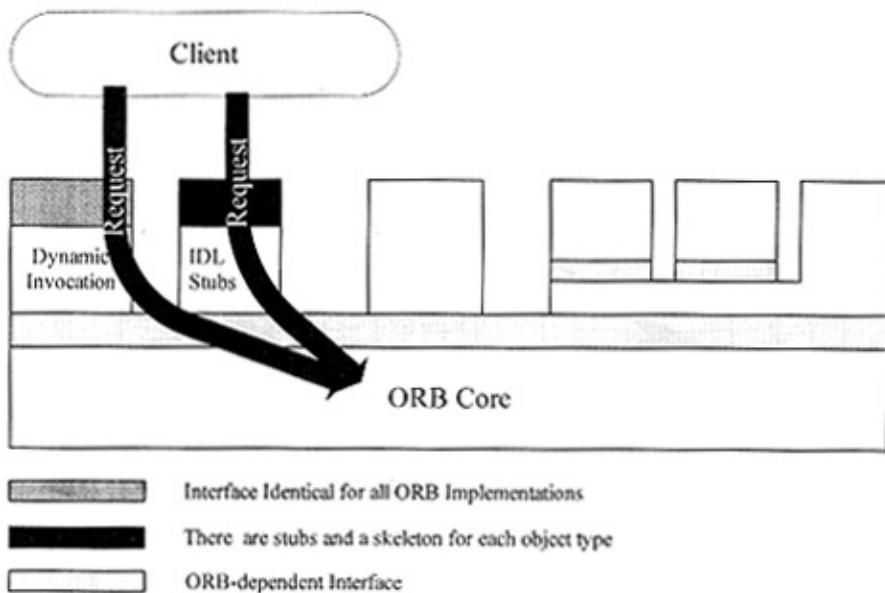


图 2.5 ORB 动态调用接口

对象的实现接受请求过程如图 2.6 所示。

ORB 向对象实现定位适当的代码，传递参数，传输控制。这一切都通过 IDL 骨架或者动态骨架。骨架对于不同的接口和对象适配器是不同的。在执行该请求的时候，对象的实现可能由 ORB 通过对象适配器来获得一定的服务。当请求完成，控制和输出值返回给客户。

对象的实现可能会选择使用的对象适配器。该决定基于对象的实现要求的服务。

基于 ORB 实现的不同类型接口，一个客户端请求可以同时访问多个由不同 ORB 实现通信管理的对象引用。在实际应用中，只要遵循公共的 ORB 体系结构，程序设计可以选择 ORB 的多种实现方式，其中包括<sup>[14]</sup>：

(1) 客户和实现驻留 (Client-Implementation Resident) ORB：采用驻留在客户和服务对象实现程序的方式实现 ORB。在这种实现方式下，客户端可以通过

桩（Stub）程序，以位置透明的方式向具体的实现对象提出服务请求，实现客户与服务对象的通信。

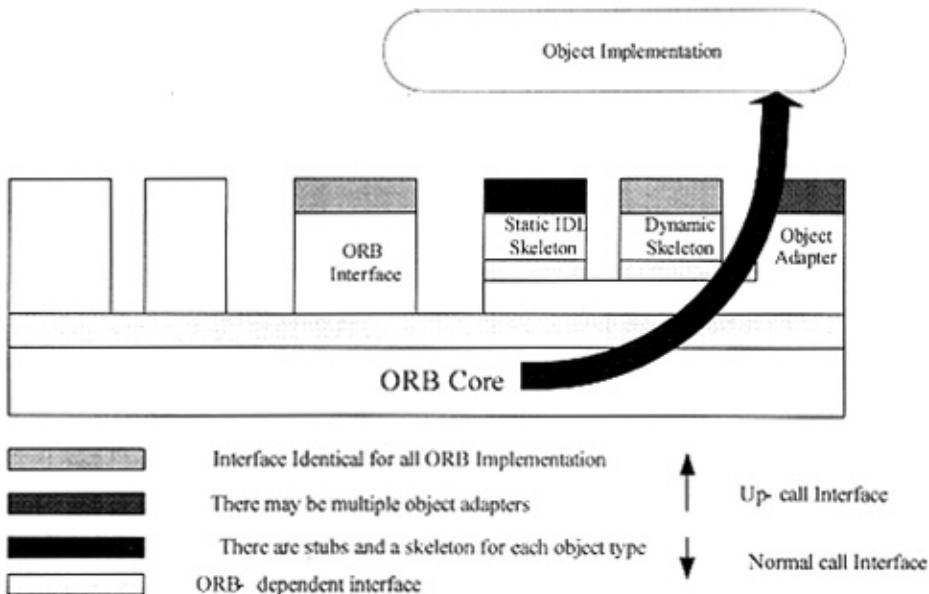


图 2.6 对象的现实接受请求

(2) 基于服务 (Server-based) ORB：客户对象和实现对象均可以与一个或多个服务对象进行通信，服务对象的功能是将请求从客户端发送到对象实现。在这种方式中，ORB 的作用是完成客户对象与实现对象的通信，为对象之间的交互提供服务。

(3) 基于系统 (System-based) ORB：在这种实现方式中，ORB 被操作系统认为是系统所提供的一项基本服务。由于操作系统了解调用方与服务对象的位置，因而可以充分地实现 ORB 功能的优化。

(4) 基于库 (Library-based) ORB：如果认为对象实现可以共享，则可以将实现功能放入实现库 (Implementation Repository) 中，从而创建基于库的 ORB。

#### 2.4.1.3 ORB 中的主要方法

客户端和服务对象端均可以自由选择使用 ORB 对象中定义的方法来实现操作。下面举例说明 ORB 对象中的重要方法<sup>[15]</sup>：

(1)ORB ORB\_init(inout arg\_list argv, in ORBId orb\_identifier)

作用：ORB 初始化方法。

(2)String object\_to\_string(in Object obj)

作用：将对对象的引用转换成客户可以用字符串方式存储的对象信息。

(3)Object string\_to\_object(in String str)

作用：上述方法的逆过程。

(4)Object resolve\_initial\_references(in ObjectID identifier)

作用：获取初始对象引用。

(5)Policy get\_policy(in PolicyType policy\_type)

作用：返回指定类型的服务策略对象。作为公共对象请求代理体系结构中的核心内容，ORB 提供了相当丰富的方法。读者可以查阅相关资料详细了解 ORB 中的方法。

#### 2.4.2 IDL(Interface Definition Language) 接口定义语言

IDL，接口定义语言，是 CORBA 体系中的另一个重要概念。如果说 ORB 使 CORBA 做到平台无关，那么 IDL，则使 CORBA 做到语言无关。

正像其名字中显示的那样，IDL 仅仅定义接口，而不定义实现，类似于 C 中的头文件。实际上它不是真正的编程语言。要用它编写应用，需要将它映射它相应的程序设计语言上去，如映射到 C++ 或 JAVA 上去。映射后的代码叫 Client Stub Code 和 Server Skeleton Code。Stub code 和 Skeleton Code 是由 IDL Complier 自动生成的，前者放在客户方，后者放在服务器方。不同厂商的 IDL complier 生成的 Stub 和 Skeleton 会略有区别，但影响不大。

IDL 的好处是使高层设计人员不必考虑实现细节而只需关心功能描述。IDL 可以说是描述性语言。设计 IDL 的过程也是设计对象模型的过程。它是编写 CORBA 应用的第一步，在整个软件设计过程中至关重要。

IDL 的语法很像 C++，当然也像 Java。如今的程序设计人员对于 C 或 Java 都或多或少会有一定程度的了解，所以，几乎所有的程序设计人员都能迅速理解 IDL。

下面是一个 IDL 定义的简单例子<sup>[16]</sup>：

```
// grid.idl
```

```

// IDL definition of a 2-D grid:
module simpleDemo{
interface grid {

    readonly attribute short height; // height of the grid
    readonly attribute short width; // width of the grid

    // IDL operations
    // set the element [row, col] of the grid, to value:
    void set(in short row, in short col, in long value);

    // return element [row, col] of the grid:
    long get(in short row, in short col);
};

};

}

```

这个接口定义语言定义了一个 CORBA 对象二维数组 GRID 的接口 T，客户通过他可以进行访问和远程编辑。

Module 类似于 Java 中包（Package）的概念，实际上 module simpleDemo 映射到 JAVA 正是 package simpleDemo。而 Interface 类似于 C++中的类(classss) 声明，或是 Java 中的 Interface 定义服务请求的实现方式。

CORBA 规范中定义客户程序可以用动态调用接口（DII，Dynamic Invocation Interface）方式或通过 OMG IDL 文件经编译后在客户端生成的桩（Stub）方式提出服务请求。这两种实现方式的区别在于通过 OMG IDL 桩（Stub）文件方式实现的调用请求中，客户能够访问的服务对象方法取决于服务对象所支持的接口；而动态调用接口调用方式则与服务对象的接口无关。尽管实现调用请求的方式有所区别，但客户发出的请求服务调用的语义是相同的，服务对象不去分析服务请求提出的方式。

ORB 通过 IDL 客户桩（Stub）方式或动态调用接口（DII）方式定位服务对象的实现代码、传递服务对象应用参数以及完成对请求传送方式的控制。服务对象的实现（Object Implementation）通过对对象适配器（Object Adapter）提供对客户请求的服务。客户对服务器端方法的调用方式见图 2.7。

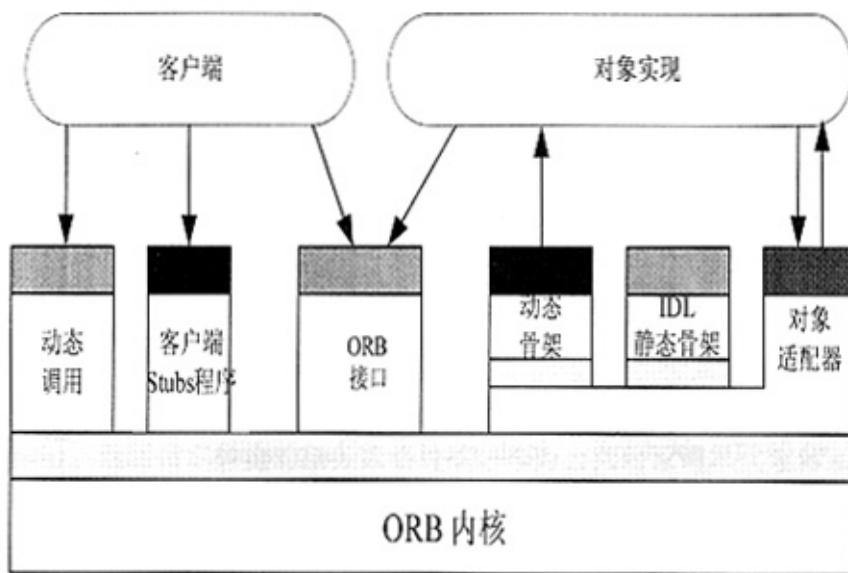


图 2.7 客户对服务器的调用方式

从以上对服务请求调用方式的分析可以看出：服务对象接口的定义在整个 CORBA 系统服务功能的实现中起着相当重要的作用。那么如何实现服务对象的功能接口呢？CORBA 规范中定义了两种接口实现方式：第一种形式是利用 OMG IDL 接口定义语言对接口进行描述。IDL 语言有详细的语法规则。这样，经过对所实现功能的系统分析，利用 IDL，依据服务对象可能实现的服务以及服务中需要的参数，对接口进行定义。另外一种方式是将接口放入接口仓库（Interface Repository）中，在运行期间对接口库中的接口进行访问。

### 2.4.3 GIOP (General Inter-ORB Protocol) 和 IIOP (Internet Inter-ORB Protocol)

我们知道，客户和服务器是通过 ORB 交互的，那么，客户方的 ORB 和服务器方的 ORB 又是通过什么方式通信呢？通过 GIOP(General Inter-ORB Protocol)。也就是说，GIOP 是一种通信协议，它规定了两个实体：客户和服务器 ORBs 间的通信机制<sup>[17]</sup>。

ORBs 通信机制的原理如图 2.8 所示。

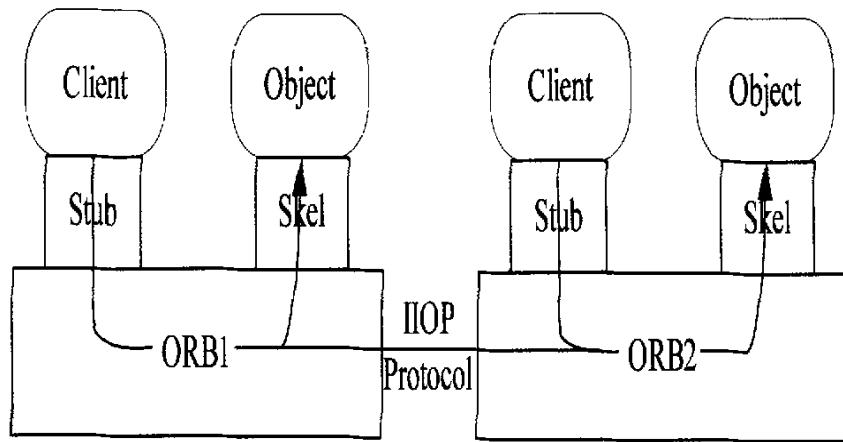


图 2.8 ORBs 通信机制

GIOP 在设计时遵循以下目标:

Widest possible availability

Simplicity

Scalability

Low cost

Generality

Architectural neutrality

也是说, GIOP 设计的尽可能简单, 开销最小, 同时又具有最广泛的适应性和可扩展性, 以适应不同的网络。

GIOP 定义了以下几个方面:

1. The Common Data Representation (CDR) definition.

通用数据表示定义。它实际上是 IDL 数据类型在网上传输时的编码方案。

它对所有 IDL 数据类型的映射都作了规定。

2. GIOP Message Formats.

它规定了 Client 和 Server 两个角色之间要传输的消息格式。主要包括 Request 和 Reply 两种消息。

一个 Request 消息有以下几部分组成:

A GIOP message header

A Request Header

### The Request Body

相应的，一个 Reply 消息则包括

A GIOP message header

A Reply Header

The Reply Body

GIOP1.1 规定 GIOP message header 格式如下：

```
// GIOP 1.1
struct MessageHeader_1_1 {
    char magic;
    Version GIOP_version;
    octet flags; // GIOP 1.1 change
    octet message_type;
    unsigned long message_size;
};
```

Request Header 格式如下：

```
// GIOP 1.1
struct RequestHeader_1_1 {
    IOP::ServiceContextList service_context;
    unsigned long request_id;
    boolean response_expected;
    octet reserved; // Added in GIOP 1.1
    sequence<octet> object_key;
    string operation;
    Principal requesting_principal;
};
```

Request Body 则按 CDR 规定的方式编码，它主要对方法调用的参数进行编码，如方法：

double example (in short m, inout Principal p);

可表示成：

```
struct example_body {
```

```
short m; // leftmost in or inout parameter  
Principal p; // ... to the rightmost  
};
```

### 3. GIOP Transport Assumptions:

主要规定在任何面向连接的网络传输层上的一些操作规则。如：  
Asymmetrical connection usage, Request multiplexing, Overlapping requests,  
Connection management 等。

另外，因为 CORBA 是基于对象的，GIOP 还需定义一套 Object Location 的机制。

GIOP 因为是一种通用协议，所以不能直接使用。在不同的网络上需要有不同的实现。目前使用最广的便是 Internet 上的 GIOP，称为 IIOP (Internet Inter-ORB Protocol)。IIOP 基于 TCP/IP 协议。IIOP 消息格式定义如下：

```
module IIOP { // IDL extended for version 1.1  
struct Version {  
    octet major;  
    octet minor;  
};  
struct ProfileBody_1_0 { // renamed from ProfileBody  
    Version iiop_version;  
    string host;  
    unsigned short port;  
    sequence <octet> object_key;  
};  
struct ProfileBody_1_1 {  
    Version iiop_version;  
    string host;  
    unsigned short port;  
    sequence <octet> object_key;  
    sequence <IOP::TaggedComponent> components;  
};  
};
```

## 2.5 本章小结

这一章是对技术原理的综述，本章首先简单介绍了中间件技术的原理，分布对象技术（中间技术）是伴随网络而发展起来的一种面向对象的技术；然后对比分析了几种中间技术，分别阐述了 EJB、DCOM、CORBA 等组件技术的优缺点，得出 CORBA 组件技术开发电子商务系统具有优越性；接着分析了基于 CORBA 组件技术的电子商务的优点；最后花大量的篇幅详细讲述的 CORBA 组件技术的实现原理。

## 第三章 电子商务系统的需求分析

创建电子商务网站系统需要做很多工作。首先要确定网站的目标。商业性的目标如开拓市场、营销商品等；如意见征询、信息发布等。其次分析该网站各个参与角色及其作用。即分析挖掘该网站来访者（客户、合作伙伴、供应商或内部员工等）来访的目的、所求、作用。接着，围绕网站的目的和各个参与角色及其作用架构网站的商业模式，即如何提供商品、服务及信息流。具体来说就是顾客群的确定，市场的定位，网站页面风格的制定，软件件的选择，信息的收集，数据库的选择等。针对以上任务，应根据实际情况确定需要，纳入实施计划。

### 3.1 前台网站模式确定

由于我们所开发的商务系统面对的是终端用户，顾客范围一般是个人消费者和组织消费者，因此我们的商务网站的模式应选择 B2C 模式。

仔细分析现在流行的 B2C 模式，变会发现它和现实中的大型超市有些相似的地方：通过种类繁多的商品来吸引购买者；提供商品的价格一般比较低廉；辐射范围一般较大；将盈利寄托在薄利多销上。然而，它和现实中的大型超市还有很多并不相似的地方：超市里是现金当场交易，B2C 网站却是通过汇款或网上银行支付的方式交易；B2C 网站必须送货上门，而大型超市不需要这么做（除了大件商品在市区范围内送货上门外）。

### 3.2 系统功能分析

系统的功能划分要从软件工程的角度出发。软件设计过程中应当遵循一些基本的原则，如模块化，抽象，信息隐蔽和局部化模块独立等。展开来讲，模块化就是要将程序划分成若干模块，每个模块完成一个子功能，把这些模块集

中起来组成一个整体，可以完成指定的功能。模块化使一个复杂的能易于被人的智力所管理。采用模块化原理可以使得软件结果清晰，不仅容易设计也容易阅读和理解。抽象是指把相似的方面集中起来和概括起来，暂时忽略它们之间的差异。由于人类思维能力的限制，对付复杂系统的有效方法是用层次的方式构造和分析它，也就是用一些高级的抽象概念构造和理解。局部化就是把一些关系密切的软件元素物理上放得彼此靠近。具体来说一个电子商务系统应具备前台管理功能模块、后台管理功能模块。

前台管理功能模块主要完成用户信息的管理、用户身份的认证、商品选购和搜索引擎。它的功能主要通过前台购物网站的功能模块来实现，其中一般要具有会员管理、商品展示、购物车管理、订单管理、邮件服务、网站论坛和客户服务等功能。

后台管理包括客户管理、商品管理和销售管理等。

### 3.3 工作流程分析

该系统的主要工作流程如下：

任何人通过浏览器均可进入前台网站浏览商品。会员客户在登入时可以以会员的身份，而非会员客户则系统应该给出是否要在线注册为会员来进行登陆。当顾客看到了合适的商品时点击选购后该商品的信息就会进入到客户的购物车内。当客户不在进行选购时就可对已选购的商品进行确认，形成定单，并确认送货时间、地点和付款方式等事项，这样就完成了一次购物过程。

### 3.4 系统体系结构分析

一般的分布式 WEB 应用系统并不包括应用服务器的程序，程序员只需要开发 WEB 应用程序即可，这样客户浏览器提出的各种请求和回复全部有 WEB 应用程序完成，造成了 WEB 服务器的沉重负担，而高效的电子商务系统在同一时刻具有众多的访问者。在此我们采取了多层架构的模式，引入了应用服务器，当 WEB 服务器在接受这些请求后直接将这些请求交给应用服务器来处理，应用服务器在完成了处理以后将反馈的结果经 WEB 服务器传递给客户浏览器，

WEB 服务器只负责通信方面的事项。在这种多层架构的电子商务系统中与客户浏览打交道的 WEB 服务器和 WEB 服务程序，而控制客户密码的数据操纵权限的是应用服务器的应用服务程序和数据库，这样系统的安全性就更强了。

### 3.5 系统的软硬件环境分析

作为一个功能完善，高效的电子商务购物系统，对其软硬件环境要求进行必要的分析是非常重要的。

由于系统是基于 CORBA 组件技术的，采用的是三层架构的模式，WEB 服务器和组件服务器的配置要求不是特别的高，数据库服务器由于要安装 oracle9i，所以配置要求较高，作为工作站用的 PC 机要求不会很高。除了这些之外还需要一些网络方面的硬件设施。

软件方面采用比较稳定的 win2000server 作为操作系统平台，数据库系统采用 oracle9i。

### 3.6 本章小结

这一章是对基于 CORBA 电子商务系统的需求进行的分析。分别对前台网站的模式、系统的功能模块、工作流程、系统的体系架构和软硬件环境进行了分析，这些是进行下一步工作的基础。

## 第四章 基于 CORBA 电子商务系统的设计

### 4.1 系统解决方案的设计

#### 4.1.1 系统结构模式

本系统设计使用 B/S 模式，主要是基于以下优点考虑：

- (1) B/S 模式具有分布性特点，可以随时随地进行查询、浏览等业务处理；
- (2) B/S 模式业务扩展简单方便，通过增加网页即可增加服务器功能；
- (3) B/S 模式维护简单方便，只需要改变网页，即可实现所有用户的同步更新；
- (4) B/S 模式开发简单，共享性强。

采用 B/S 模式，便于管理员的维护，因为维护可能不在本地执行，这时候通过传统的模式很难得以处理，而 B/S 模式就可以解决这种窘态。利用快捷的网络可以随时维护后台的数据，进行网页的更新处理。同时，使用 B/S 模式，在客户端的安装不需要如同其它模式般复杂，只要客户端的计算机可以上网，具备浏览网络的功能，不需要安装任何软件就可以实现对系统的访问<sup>[18]</sup>。

#### 4.1.2 系统的网络拓朴图

我们开发的这个电子商务系统的架构引入了一种新的 WEB 应用开发方法，也就是应用服务器的方式，系统的分布式架构如图 4.1 所示。

应用服务器应当具备的首要特性是至少三个层次的服务器端体系结构。所有的应用请求，都将通过请求接收层，一般就是 Web 服务器，转给应用处理层中的应用服务器处理。应用服务器是独立的进程，对业务进行处理，并进行事务管理，将其中的所有数据操作转给第三层，也就是数据处理层的数据库服务器。在一定的情况下，也可以转给后面的其他系统。应用服务器体系结构的核心在一般的 Web 服务器和数据库服务器之间，增加专门的应用服务器来完成业务处理，而不是直接从 Web 服务器访问数据库服务器。

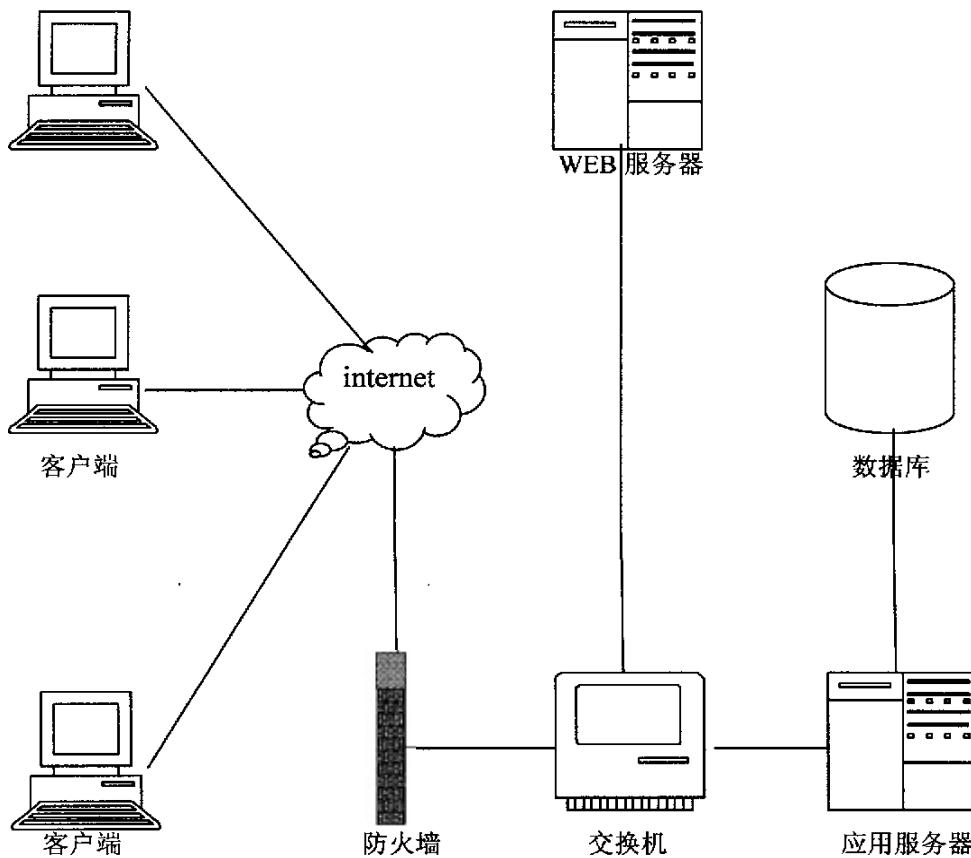


图 4.1 系统的网络拓朴图

应用服务器应当具备的首要特性是至少三个层次的服务器端体系结构。所有的应用请求，都将通过请求接收层，一般就是 Web 服务器，转给应用处理层中的应用服务器处理。应用服务器是独立的进程，对业务进行处理，并进行事务管理，将其中的所有数据操作转给第三层，也就是数据处理层的数据库服务器。在一定的情况下，也可以转给后面的其他系统。应用服务器体系结构的核心在一般的 Web 服务器和数据库服务器之间，增加专门的应用服务器来完成业务处理，而不是直接从 Web 服务器访问数据库服务器。

当然，无论电子商务是怎样一个网络应用体系，其底层仍然离不开作为核心的企业关键任务应用系统和数据资源库。这里就电子商务应用服务器的作用简单介绍如下：

电子商务应用服务器的作用是让网络应用的开发、部署、管理变得更加容

易。其功能有：

- 1、提供在服务器端的分布式应用的部署，包括对象生命周期管理、线程管理、状态管理、安全管理等；
- 2、数据源连接访问管理、交易管理等；
- 3、大规模并发网络用户管理、均衡负载、容错等；
- 4、与现有系统的无缝连接。

其中，对象生命周期管理、线程管理、状态管理、安全管理、策略管理等，都是由预置在电子商务应用服务器中的各类服务构件来支持的，在应用的运行效率上有很好的保障，同时大大简化了应用开发的周期与成本。通过基于图形的集中化控制，用户能够随时掌握分布在网络上众多对象的状态，快速建立对象与对象之间的后援关系，设定对象异常终止时后备对象接替运行的策略。

采用电子商务应用服务器的逻辑结构如图 4.2 所示。

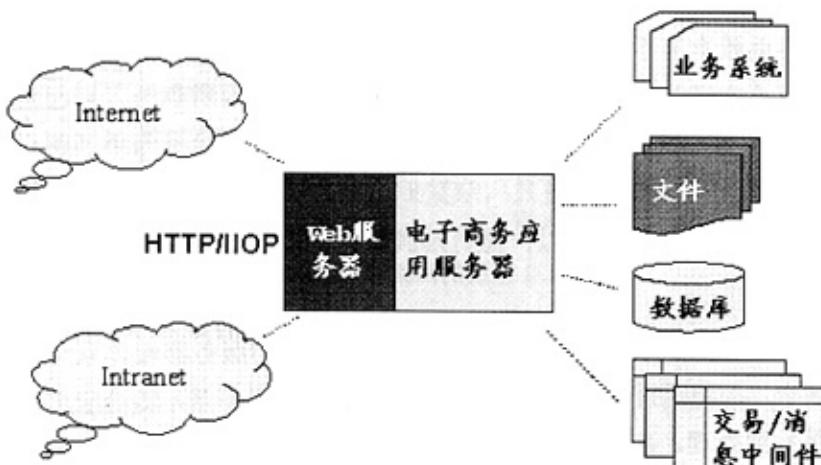


图 4.2 电子商务应用服务器逻辑结构图

电子商务应用服务器可以用一种灵活的方式来代表一个商业进程，把商业过程转化到一个包含若干个阶段的框架结构，每一个阶段代表对一个商业对象（如定货单）的分离的操作。在每一个阶段，一个或多个专门的组件对对象进行操作。电子商务应用服务器的另一个重要功能是可以与标准的交易中间件实现集成，这意味着整个平台可以作为一个单一的操作，这在一些必须维护进程的完整性的场合是非常重要的。

## 4.2 软硬件平台设计

### 4.2.1 硬件平台

#### 1、WEB 服务器、组件服务器

P2 或者赛扬 850 或者 AMD850 以上 CPU、250M 内存、40G 硬盘、3D 以上显卡、

#### 2、数据库服务器、

此服务器由于要安装 oracle9i，要求配置比较高，尤其是内存和 CPU，建议使用 250M 以上内存，主频为 130 以上的主流 CPU。

#### 3、PC 机(工作站用)

要求不高，只要可以运行 IE5 或者 Navigator4 以上就可以了。

#### 4、HUB 或者交换机、路由器、网线

此设备不做太大的要求，可以连接网络即可。

### 4.2.2 软件平台

#### 1、WEB 服务器

win2000server、Tomcat4.0、Visual Broker for java 4.5、IIS5

#### 2、数据库服务器

win2000server、oracle9i

#### 3、组件服务器

win2000server、word2000、Jbuilder7、浏览器等

#### 4、PC 机

windows9X/2K/XP、浏览器(IE5X 或 NV5X)

### 4.2.3 软件安装和配置

以上的软件，主要讲是 Jbuilder7 的配置至关重要，其它的不需要什么配置，方法如下：

安装包含 ORB 的应用程序服务器，这里安装了 visual broker4.5 for java。

选择 Tools|Enterprise Setup 命令，打开 Enterprise Setup 对话框，并选择

CORBA 选项卡。

在 Configuration 下拉列表选择一种配置，这时选择 VisiBroker  
单击 Edit 按钮，打开 Edit Configuration 对话框。

在 Edit Configuration 对话框中，单击 Path For ORB Tools 域旁边的省略号  
按钮选择合适的路径以使 Jbuilder 可以访问 ORB 工具。我们选包含 osagent.exe  
文件的路径。

在 Edit Configuration 对话框中，单击 Library For Projects 域旁边的省略号  
按钮来打开 Select A Different Library 对话框，在其中选择框的位置。

如果当前工程要使用默认工程的 ORB 设置，则在 Enterprise Setup 对话框  
中选择 Apply This Configuration To The Current Project 选项。

配置完成。

### 4.3 系统设计目标与原则

本系统的前台向客户提供方便购物、方便付款的网上电子虚拟平台，客户  
一旦注册登录到本系统即可浏览到本网站所提供的所有商品，并进行方便快捷  
的购买与支付活动；一旦客户与本网站发生了商业活动，即会产生相应等级状  
态并与企业建立起不同的友好联系与服务。同时，通过后台管理系统，按不同  
权限登录的用户则可以很方便地进行客户信息管理、商品信息及商品促销信息  
管理，对商品的销售情况、客户的各种情况进行分析，并产生相应的报表，这  
样就有利于网站管理人员及时随市场前景对商品及与客户的联系情况作出调  
整。

通过对组件技术的实践，开发小组的成员必须加深对该体系结构的了解，  
熟练该体系结构的开发细节，提高自身的软件开发能力；通过对电子商务系统  
的建设，熟悉该系统的业务流程和所包含的功能模块，扩大自己的知识面，完  
成毕业设计的任务。

## 4.4 系统总体设计

整个电子商务功能的实现是由网上在线订购子系统和信息管理子系统两部分组成。订购子系统主要实现用户注册、登录，产品选购，货款支付，用户个人信息管理等；管理子系统则实现网站设置，用户订单处理，产品信息管理等。通过数据库中数据表之间的关联，实现子系统之间相互联系，实现数据传输、访问和共享。

系统需要一些客户信息（譬如姓名、住址等），才能提供完成的商务活动，所以访问者首先要成为注册用户，才可以在网站上订购商品。用户登录后，在服务器端存储了与之相关的信息，包括用户 ID、登录时间、购买的商品品种及数量。系统允许用户修改自己的购物车中的商品，例如放弃该商品，修改购买该商品的数量等。在用户完成本次购物时，系统提醒用户去收银台，选择不同的支付方式进行付款，付款后生成订单号，用户可以凭借该订单号查询购买物品的处理进程。当用户完成此次购物后，应该把本次的购买信息存储到该用户历史购买信息数据表中，以便统计该用户的购买力、兴趣爱好等，也可以为商户提供商品信息参考。

在设计流程中我们需要考虑很多的因素，设计中最重要的原则主要包括：灵活性，可监测性，可靠性，安全性及可管理性。

### 1) 灵活性。

灵活性的核心就是整合。整合之后可以创造出新的战略优势。因此，当设计电子商务解决方案时，非常重要的一个问题是整合所有独立的应用，数据库和业务流程以保持其竞争力。所以设计人员需要考虑所有已经存在的或者计划实现的应用软件，数据库以及业务流程等。当业务需求发生了变化的时候，整合得很好的系统基础可以灵活快速地，经济节约地满足新的业务需求。完成整合的主要策略是使用公开的，标准化的基础，兼容的中间件或者中间层。

### 2) 可监测性。

系统运行中总会有很多计划中难以预料的变化。例如流量的剧增，业务的复杂化等。为了防止这些变化降低系统的性能或者带来高额的维护费用。系统从一开始就应该考虑其可监测性的实现。

### 3) 可靠性。

可靠性对于安全，无间断的运行系统至关重要。当用户数量剧增，可靠性可以保证提供不间断的快速响应速度，24 小时全天候的优质服务。自动的流量调节工具和系统测试工具可以帮助我们对系统的可靠性进行检验。

#### 4) 安全性。

为了优化电子商务系统，系统需要考虑其安全和隐私的要求。首先，需要明确现有系统的安全隐患。其次，新系统需要能有效地预防和处理意外，攻击和一些灾难性事故并能从事故中及时地恢复。另外，检测到潜在的威胁也至关重要。最好，所有的安全功能和过程要能做到可管理。

#### 5) 可管理性。

解决方案中需要包括如何在一个异构的环境中去管理系统的详细方案。这个方案包括修改系统的能力，删除不需要的功能，适应新的运行环境，按需重建系统等能力。另外还有监控功能也十分必要。例如，自动的收集系统信息，比较终端用户的功能、性能、实施情况，对系统的综合评价等。

除了上述几大原则之外，还有一些重要因素需要考虑。他们分别是：

1、技术支持问题。包括对终端用户，消费者的技术支持以及信息反馈的处理机制。

2、电子商务 365X7X24 全天候服务的要求和系统性能要求。

3、合法性。涉及到电子签名，文件存档，隐私，税收和关税，知识产权，银行条例，会计实施等诸多问题。

4、系统结构，运行的管理以及使用的简易性问题。这个问题涉及到软件的发布，版本控制，版权控制，提醒服务，系统管理日志等问题。

5、数据库管理问题。涉及数据结构设计，数据存储，数据备份和恢复，数据存档和查询数据的可管理性，异构系统数据转换问题，数据保护问题等。

WWW 技术迅速发展，它已不再仅仅是超媒体信息的浏览工具，以逐步成为人们进行事务处理的前端；由于分布对象的计算技术，特别是 CORBA 技术对于提高 WEB 的网络计算能力有着巨大作用，CORBA 与 WWW 技术融合，可以为电子商务系统架构提供良好的解决方案。

CORBA 与 WWW 结合，构架出真正的三层体系结构。这种三层的体系结构，以分布对象技术为基础构架，增加了应用层，将客户层与资源层隔开，降低了 Web 服务器的负载，避免了 Web 服务器的性能缺陷对整个性能的影响。并且具有连接缓冲，负载均衡，安全管理等功能，从而提高了 Web 应用整体的灵活性，

可伸缩性，可扩展性。

该结构中，CORBA 客户方程序从 Web 服务器上下载执行，与应用服务器上的 CORBA 应用对象通过 IIOP(Internet-Inter-ORB Protocol) 协议进行通讯，调用其指定的操作。CORBA 应用对象首先对客户的请求进行认证和解释，根据客户请求的内容，或是直接访问资源层的数据库，或是与网络上的其它 CORBA 对象交互，共同完成客户请求。CORBA-WEB 体系与 DCOM, JAVA RMI (Remote Method Invocation) 比较起来，有明显优势。

电子商务系统处在典型的异构和分布式环境中，CORBA 所具备的分布式计算处理能力为完成大型复杂的电子商务系统提供了技术上的保证，在当今各种开发工具积极提供与 CORBA 的接口情况下，使更好地把 CORBA 应用到电子商务系统的开发中去成为可能。

主要的组件功能描述如下：

#### 4.4.1 前台管理功能

用户信息管理：必须完成注册信息的修改、密码修改、购物车（在此只进行购物车模块的跳转（调用）、订单查询、常用收货人管理、统计信息、投诉及商品评价、注销。

用户身份认证：包括注册新用户，用户登录，确认用户信息，显示用户信息。

商品选购：管理购物车，显示商品信息。

搜索引擎：必须完成通过商品的名称、价格、折扣等属性对商品的查询，以及通过对订单的编号、日期等属性对订单进行查询。

#### 4.4.2 后台管理管理

客户管理：确定客户等级、

商品管理：商品基本信息、库存管理（入库），商品销售计划

销售管理：根据定单发货

#### 4.4.3 前台购物网站

作为一个电子商务网站，应能完成为企业收集客户信息，了解客户需求，宣传企业文化，树立企业形象，展示企业产品，以及在线销售的功能。所以在我们开发的商务网站中主要设计了以下几个功能模块：

- 1、会员管理模块
- 2、商品展示模块
- 3、购物车模块
- 4、订单模块
- 5、邮件服务模块
- 6、网站论坛模块
- 7、客户服务模块

网站面对的客户是企业消费者。顾客可以方便的查询并订购商品；产品订购手续直观、简单、快捷；订购信息准确；

前台购物网站的总体结构如图 4.3 所示。

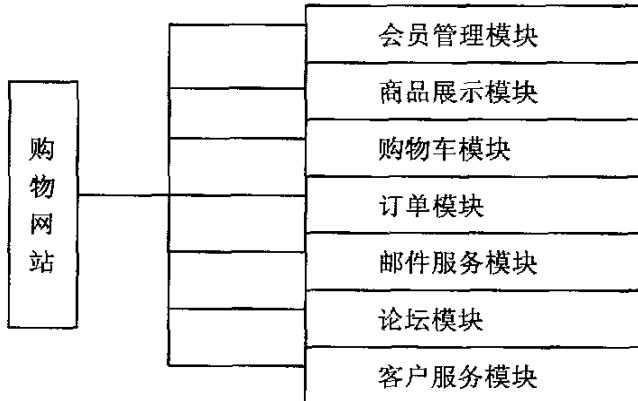


图 4.3 网站总体结构

#### 4.4.4 会员管理模块

主要用来收集用户信息，实现权限管理，方便用户浏览网站选购商品和身份验证识别，为其它功能提供方便是产品展示在线销售的基础，也是实现会员制的基础。

会员管理模块结构如图 4.4 所示。

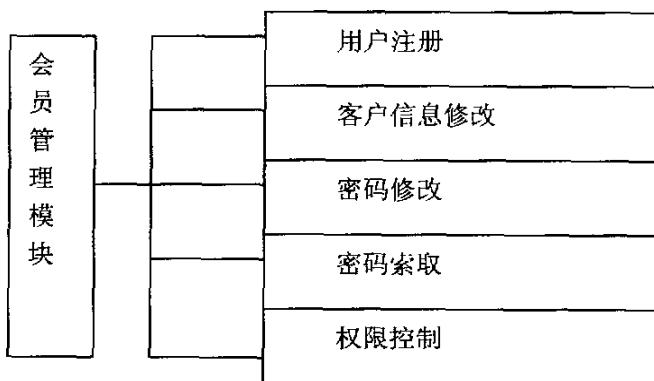


图 4.4 会员管理模块结构图

#### 业务描述：

由用户自己操作，按提示填写相关信息进行注册，成功后系统将给该用户一个唯一的编号，使之成为该系统的一个用户。任何时候客户都可以来买东西，用户注册的好处在于：

- 1、了解客户群体的结构，以便在商品的准备上有相对性；
- 2、控制和管理客户信息；
- 3、便于统计客户。

#### 4.4.5 商品展示模块

作为电子商务网站，展示宣传企业的产品尤为重要。这一部分同样要放在首页的醒目位置，将企业的产品分类，及商品的规格、信息、图片，详细的展示出来。

使用方便快捷的操作接口，准确明了的显示商品信息。当用户点击选择的商品时展现商品的全貌、规格及价格信息。是这一部分首要的功能。

这一部分我设计了以下部分功能：

- 1、商品的分类信息
- 2、商品的列表显示
- 3、商品的详细数据显示
- 4、商品图片的查看显示
- 5、商品的搜索

- 6、添加商品
- 7、修改商品
- 8、删除商品
- 9、商品的类别管理设定

#### 4.4.6 购物车模块

这里为方便用户管理自己的订购而设定，购物车中将存储显示此用户的所  
有未进入付款程序的订购信息，包括商品名称、商品单价、订购数量、总计金  
额等。在这里，提供给用户进行修改和删除订购商品信息的功能。

当用户选择并订购商品后，将会在购物车中显示出来，当用户确认无误后  
可点击付款，进入最后的付款程序。

#### 4.4.7 订单模块

作为电子商务网站，实现在线销售是其最终的目的。这一部分与商品分类  
展示并存，操作一定要简便快捷，处处为用户考虑，但又不能失去必要的安全  
检验步骤。要在操作前识别验证用户的身份信息，检验用户是否已经登陆成功，  
是否为本网站的合法注册用户。当检测到用户已成功登陆后才能允许其进行交  
易操作否则不能操作。这样强制用户进行注册，有效的保证了系统数据库中数  
据的安全防止恶意购买产生大量的无用订单。

这一部分应为企业实现商品的在线销售，它依附于商品展示管理子系统。  
当用户浏览选择完商品后即可进入订购程序。它与商品展示管理子系统并存在  
同一浏览接口中。

这里我设计了以下部分功能：

- 1、订购商品
- 2、查看购物车
- 3、删除定购
- 4、清空购物车
- 5、收银台付款
- 6、提交订单
- 7、交易支付

- 8、订单查询
- 9、查看新订单
- 10、删除订单
- 11、查看已确认订单

## 4.5 购物流程图

在确立了网站的总体架构和各个功能模块后我们对客户的购物流程进行了设计，客户购物流程如图 4.5 所示。

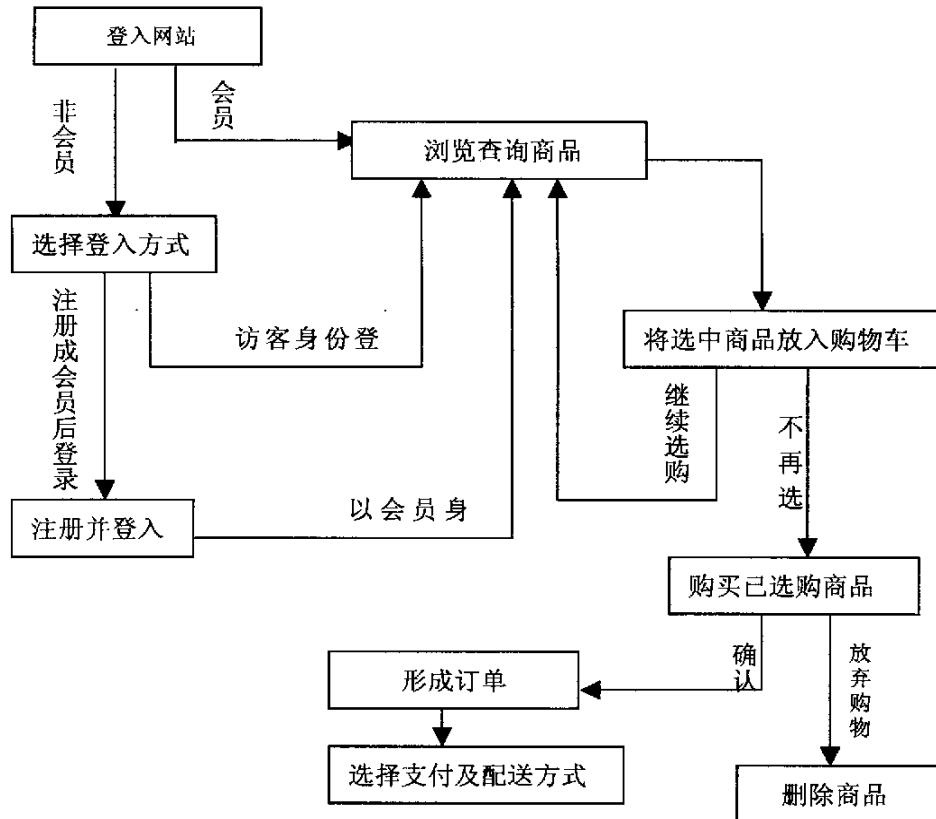


图 4.5 客户购物流程图

## 4.6 系统数据库设计

### 4.6.1 数据库设计原则

数据库设计主要是指数据库的物理设计。数据库物理设计的工作任务主要是在数据库逻辑设计的基础上，运用关系数据库规范化理论，设计出适合应用环境的数据库物理结构。数据库在物理设备上的存储结构与存取方法成为数据库的物理结构，它依赖于给定的计算机系统。为一个给定的逻辑数据模型选取一个最合适应用要求的物理结构的过程，就是数据库的物理设计。数据库设计的内容是：对于一个特定的环境，进行符合应用语义的逻辑设计，以及提供一个确定存贮结构和物理设计，建立实现系统目标，并能有效存取数据和数据模型。

数据模型是指数据库中记录与记录之间联系的数据结构形成的。不同的数据管理系统有不同的数据模型，数据库设计的核心问题是设计好的数据模型。在目前的数据库管理系统中有层次模型，网状模型、关系模型三种数据模型。其中，关系模型具有较高的数据独立性，使用也较为方便。

网站信息管理数据繁杂，重复性很大，数据使用频繁。这样，就需要一种能正确反映用户的现实环境，易于维护、效率较高的数据管理方法。考虑到以上特点，我们的这一电子商务购物网站的数据库系统采用的是 Oracle 数据库，Oracke 9i 是一种功能十分强大的关系型数据库管理系统。Oracke 9i 可以用于构建大型的、分布式的网络数据库系统，分布式数据库系统是分布式网络技术与数据库技术相结合的产物，Oracle 9i 具有较高具有图形化管理、丰富的编程接口、多线程系统、良好的并发控制、与操作系统良好的接口等特点。

在确定了网站的主要商务的业务对象和业务流程后，可以确定了网站的数据流，也就可以进行数据库设计。在进行数据库设计时，同样和一般的应用系统开发一样，应该注意信息的完整性和数据的独立性。数据库表数据的独立性和数据冗余直接影响数据的存取效率，影响网站的运行速度，所以我们在数据库设计时尽量避免数据的冗余性，同时还避免长数据库表的设计。

#### 4.6.2 数据库架构图：

我们在该电子商务网站的数据库架构如图 4.5 所示。

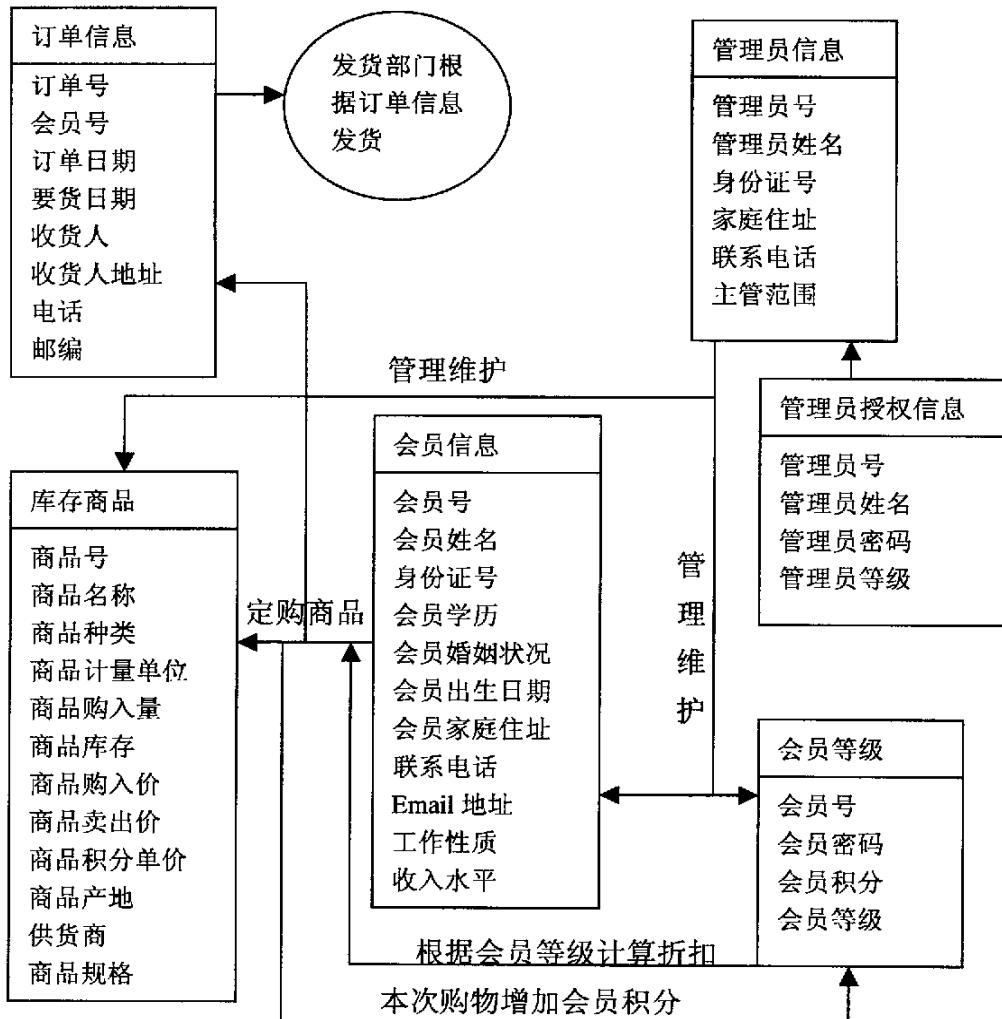


图 4.6 电子商务网站数据库架构图

#### 4.6.3 数据表

我们在设计数据表的过程中本着高效、低冗余的原则，以下就是该购物网站设计中的 6 个主要的数据表。

会员基本情况表：用来存放会员用户的基本信息；

会员等级表： 用来记录会员用户的购物积分和等级；

库存商品表： 用来存放当前的库存商品信息；

后台管理员基本信息表： 用来存放后台管理员的基本信息；

后台管理员授权表： 用来存放后台管理员的授权信息， 授权的级别决定管理员操作功能模块的权限；

订单信息表： 用来存放顾客购物的订单信息。

这 6 个表的表结构如下：

会员基本情况表如表 4.1 所示。

表 4.1 MEMBER\_BASIC 会员基本情况表

列名	数据类型及长度	可否为空	说明
MEM_NO	CHAR(6)	NOT NULL	会员号（主键）
MEM_NAME	VARCHAR(10)	NULL	会员姓名
MEM_ID	CHAR(2)	NULL	会员身份证号
MEM_SCHOOL	VARCHAR(20)	NULL	会员学历
MEM_MARRIAGE	VARCHAR(2)	NULL	会员婚姻状况
MEM_BIRTHDAY	DATE	NULL	会员出生日期
MEM_HOMEADDRESS	VARCHAR(20)	NULL	会员家庭住址
MEM_TELENO	CHAR(10)	NULL	会员联系电话
MEM_EMAIL	CHAR(10)	NULL	会员 EMAIL 地址
MEM_WORK	VARCHAR(10)	NULL	会员工作性质
MEM_INCOME	CHAR(6)	NULL	会员收入水平

会员等级表如表 4.2 所示。

表 4.2 MEMBER\_GRADE 会员等级表

列名	数据类型及长度	可否为空	说明
MEM_NO	CHAR(6)	NOT NULL	会员号
MEM_PASSWORD	CHAR(6)	NOT NULL	会员密码
MEM_INTEGRAL	CHAR(6)	NULL	会员积分
MEM_SCALE	CHAR(2)	NULL	会员等级

库存商品表如表 4.3 所示。

表 4.3 MERCHANDISE\_STOCK 库存商品表

列名	数据类型及长度	可否为空	说明
MER_NO	CHAR(6)	NOT NULL	商品号 (主键)
MER_NAME	VARCHAR(10)	NULL	商品名称
MER_KIND	VARCHAR(6)	NULL	商品种类
MER_UNIT	CHAR(2)	NULL	商品计量单位
MER_BUYCOUNT	CHAR(6)	NULL	商品购入量
MER_STOCK	CHAR(6)	NULL	商品库存
MER_BUYPICE	CHAR(3)	NULL	商品购入价
MER_SALEPRICE	CHAR(3)	NULL	商品卖出价
MER_INPRICE	CHAR(3)	NULL	商品积分单价
MER_CITY	VARCHAR(10)	NULL	商品产地
MER_SPEC	VARCHAR(4)	NULL	商品规格

后台管理员基本信息表如表 4.4 所示。

表 4.4 MANAGE\_BASIC 后台管理员基本信息表

列名	数据类型及长度	可否为空	说明
MAN_NO	CHAR(6)	NOT NULL	管理员号 (主键)
MAN_NAME	VARCHAR(10)	NULL	管理员姓名
MAN_ID	CHAR(2)	NULL	管理员身份证号
MAN_HOMEADDRESS	VARCHAR(20)	NULL	管理员家庭住址
MAN_TELENO	CHAR(10)	NULL	管理员联系电话
MAN_FIELD	VARCHAR(10)	NULL	管理范围

后台管理员授权表如表 4.5 所示。

表 4.5 MANAGE\_BACKGROUND 后台管理员授权表

列名	数据类型及长度	可否为空	说明
MAN_NO	CHAR(6)	NOT NULL	管理员号 (主键)
MAN_NAME	VARCHAR(10)	NOT NULL	管理员姓名
MAN_PASSWORD	CHAR(6)	NOT NULL	管理员密码
MAN_GRADE	CHAR(2)	NULL	管理员等级

订单信息表如表 4.6 所示。

表 4.6 ORDER\_INFORMATION 表

列名	数据类型及长度	可否为空	说明
ORD_NO	CHAR(6)	NOT NULL	订单号(主键)
MEM_NO	CHAR(6)	NOT NULL	会员号
ORD_DATE	DATE	NULL	订单日期
GET_DATE	DATE	NULL	要货日期
ORD_NAME	VARCHAR(2)	NULL	收货人姓名
ORD_HOMEADDRESS	VARCHAR(20)	NULL	收货人家庭住址
ORD_TELENO	CHAR(10)	NULL	收货人联系电话
ORD_MAIL	CHAR(2)	NULL	收货人邮编
ORD_SEND	VARCHAR(2)	NULL	是否已发货

## 4.7 本章小结

该章是对基于 CORBA 组件技术的电子商务系统设计。本章首先阐述了该系统解决方案的设计，系统的结构模式采用 B/S；然后简单介绍了系统的软、硬件平台的设计；然后描述了系统的模块设计方案和该电子商务系统的购物流程，最后，简单介绍了该系统的数据库设计及几个主要的数据表。

## 第五章 系统的实现

### 5.1 关键技术与算法

#### 5.1.1 CORBA 的 ORB 代理的实现代码

目的：把客户端的桩程序和服务器的框架程序连接起来，以便调用 IDL 文件定义的方法。

实现原理：客户端的桩程序发出请示，组件服务器的端 ORB 程序接受并解析服务，由服务器端的框架程序进行响应，再按原路返回结果

```

public void init() {
    try {
        String [] args = null;
        String name = "backstageuser";
        if (System.getProperties().get("ORBagentPort") == null) {
            System.getProperties().put("vbroker.agent.port", "14000");
        }
        if (System.getProperties().get("org.omg.CORBA.ORBClass") == null) {
            System.getProperties().put("org.omg.CORBA.ORBClass" ,
"com.inprise.vbroker.orb.ORB");
        }
        if (System.getProperties().get("org.omg.CORBA.ORBSingletonClass")
== null) {
            System.getProperties().put("org.omg.CORBA.ORBSingletonClass",
"com.inprise.vbroker.orb.ORB");
        }
        if (_orb == null) {
            _orb = org.omg.CORBA.ORB.init(args, System.getProperties());
        }
    }
}

```

```

        if(ifacebackstageuser == null) {
            ifacebackstageuser =
backstageadmin.modBackstage.backstageuserHelper.bind(_orb ,      "/" + name +
"_poa",   name.getBytes());
        }
    }
    catch (org.omg.CORBA.OBJECT_NOT_EXIST cone) {
        System.out.println(cone.getMessage());
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}

```

### 5.1.2 模块代码实现

目的：把纪录以表格的形式返回到客户端，并且自动增加和更新的字段，点击可以进入相应的页面，进行相应的操作。

实现原理：在组件的实现里面，根据要求查询出纪录，并把他们连接成字符串，返回到客户端程序。

```

public String displayUsualUser(short level,   short sort)  {
    //xq add 2004-5-29 11:00
    //ver 1.0.0.0
    //入口参数  level:会员或后台管理员的级别，前者有 1-3，后者有 1-5，0 代表显示所有级别
    //          sort:如果是 1，代表 level 是指会员的级别，2 为管理员的级别
    //功能:按等级显示会员或者管理员列表
    //.....定义部分省略.....
    if(sort==1)  //如果是会员，则针对会员的表进行操作
    {
        bFlag=false;

```

```

strSql="select Customer.Customer_Name , Customer.Customer_Enterprise ,
Customer.EnterpriseSize , Customer.Address , Customer.Email , memlevel.level_name ,
Customer.CreateDatetime";

strSql=strSql+" from memlevel, CustomerLevel, Customer ";

strSql=strSql+ " where memlevel.Level_ID=CustomerLevel.MemLevel and
Customer.Customer_Name=CustomerLevel.Customer";

if(level!=0) //如果用户级别不为 0，则根据相应的级数进行显示
    strSql=strSql+" and memlevel.Level_ID="+String.valueOf(level);
else          //如果用户的级数为 0，则显示所有的会员
    strSql=strSql+" order by CustomerLevel.MemLevel";

}

else //如果是管理员，则针对会员的表进行操作
{
    strSql=strSql+"select admin.id, admin.admin_Department, admin.PwdQuestion ";
    strSql=strSql+" from power, admin, PowerAssign";
    strSql=strSql+"      where PowerAssign.admin_ID=admin.admin_ID      and
PowerAssign.Power_ID=power.Power_ID";

    if(level!=0) //如果管理员级别不为 0，则根据相应的级数进行显示
        strSql=strSql+" and power.Power_ID="+String.valueOf(level);
    else          //如果管理员的级数为 0，则显示所有的管理员
        strSql=strSql+" order by power.Power_ID";

}

Connection con=conn();
if(con==null)
    return "cann't connection webserver";

try
{
    strResult=<table width=535 border=1 cellspacing=0 bordercolor=#A7ADFC>;
    Statement stmt= con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE ,
ResultSet.CONCUR_UPDATABLE);

    ResultSet rs=stmt.executeQuery(strSql);
}

```

```

while(rs.next())
{
    if(bTrFlag)
    {
        strResult=strResult+"<tr align=center bgcolor=#CCCCCC>";
        bTrFlag=false;
    }
    else
    {
        strResult=strResult+"<tr align=center bgcolor=#E8E8E8>";
        bTrFlag=true;
    }
    strFirstField= rs.getString(1);
    strResult=strResult+"<td width=65>" +strFirstField+"</td>";
    for(i=2;i<=3;i++)
    {
        strResult=strResult+"<td width=65>" +rs.getString(i)+"</td>";
    }
    if(bFlag==false)
    {
        for(i=4;i<=5;i++)
        {
            strResult=strResult+"<td width=65>" +rs.getString(i)+"</td>";
        }
        strResult=strResult+"<td width=80>" +rs.getDate(7)+"</td>";}
        strResult=strResult+"<td width=65><a href=del.jsp?name="+strFirstField+"&sort="+sort+"onclick='"+""+"'"+if(window.confirm('Are you sure to delete it?')) return true;else return false;+"+">del</a>"+"</td><td><a href=update.jsp?name="+strFirstField+"&sort="+sort+">update</a></td></tr>";}
    }
strResult=strResult+"</table >";

```

### 5.1.3 网站的主要功能模块界面

在这一节中我将通过几个主要的操作界面图以展示该电子购物系统的购物

过程。

### 1、购物网站登陆界面图

应随时随地的方便用户登录网站。无论用户处于站点的那一层，哪怕在深层网页也应有登录窗口。登录窗口的密码输入部分一定是暗码输入，以免被他人窃取。

当客户在登陆窗口输入其用户名和密码后应打开数据库取出用户详细信息并在合适的位置显示用户的信息以便用户及时查验及时更正。

购物网站的登陆界面如图 5.1 所示。



图 5.1 登陆网站界面图

该功能模块是购物网站的首页，提供会员用户登陆的界面，非会员用户也可从此进入在线注册页面，由于我们的开发经验有限，而且缺乏必要的素材，该界面还不够美观，并且还应进一步插入一些动态的热销商品的展示，这些有待我们在以后进行进一步的补充和完善。

### 2、在线注册会员用户界面图

在网站首页的醒目位置，我们放置了用户在线注册的进入按键，注册操作尽可能的方便用户使用，有效的收集关于用户的必要信息，同时注意保护用户的隐私，步骤要简便快捷不可繁琐。

在向数据库添加新数据即把用户所填注册信息存储到数据库前先检查用户名是否已经在数据表中存在，若已有用户使用返回提示信息，这样也能减少和避免信息的存取错误。

在线注册会员用户的操作界面如图 5.2 所示。

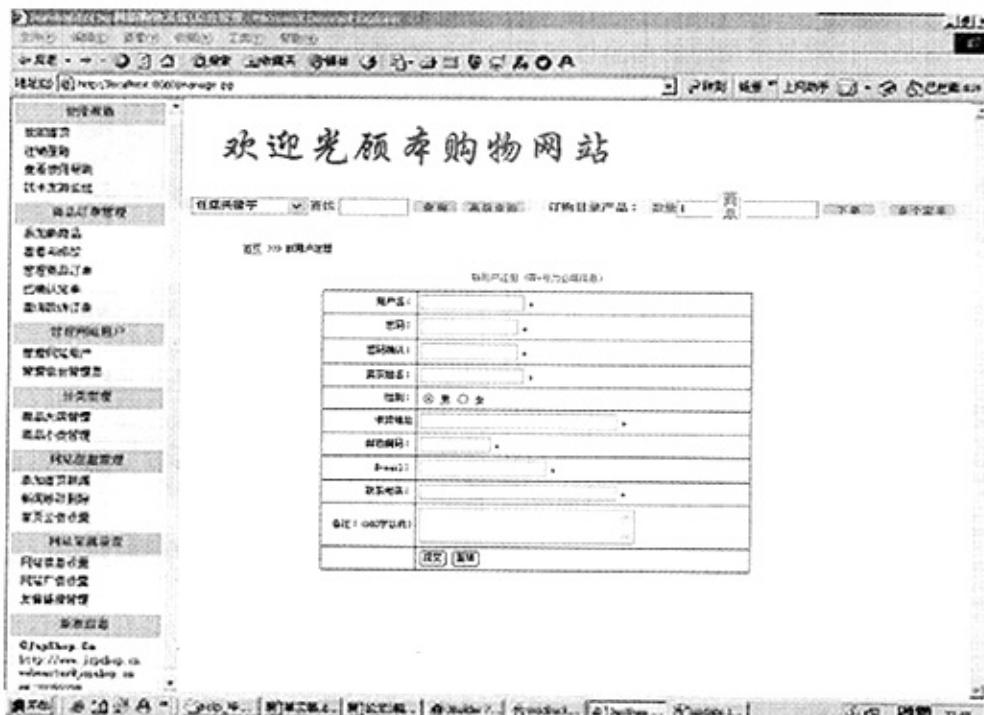


图 5.2 会员注册界面图

这一功能模块实现的是非会员用户通过在线注册成为会员用户的操作界面图，其功能较简单。

### 3、会员用户选购商品界面图

为方便客户对商品进行选购，应将商品的展示信息放在首页的醒目位置，操作应尽可能的方便用户使用，从数据库中的相应数据表中取出商品信息，并以时间的顺序进行显示列表。把最近发布的最新商品信息置于最顶端。但显示的信息不可过多，由于一条商品的信息包含内容很多，所占页面空间也不小，所以最好显示为 4-6 条左右。如果显示的信息过多，将破坏页面的美观，影响网页的整体效果。同时也要设计一个“more...”按钮，以便用户选择浏览跟多

的商品信息。

会员用户在选购商品的界面如图 5.3 所示。



图 5.3 会员选购商品界面图

该功能模块实现的是顾客选购商品的界面图，提供了商品检索的功能和列视了顾客购物车的商品，该模块的界面布局不太理想，还有就是商品展示功能还应增强。

#### 4、购物车信息展示界面

这里为方便用户管理自己的订购而设定，购物车中将存储显示此用户的所有未进入付款程序的订购信息，包括商品名称、商品单价、订购数量、总计金额等。在这里，提供给用户进行修改和删除订购商品信息的功能。

当用户选择并订购商品后，将会在购物车中显示出来，当用户确认无误后可点击付款，进入最后的付款程序。

查询购物车信息的界面如图 5.4 所示。



图 5.4 会员用户购物车信息展示界面图

该功能模块在展示用户在购物的过程中购物车信息的同时还列示了会员用户本次购物的信息和积分信息等。

## 5.2 本人所实现的模块

本人所写的是后台的管理方面功能，主要实现的是会员的管理方面的一些功能，通过近一年的工作努力，我胜利完成了后台管理模块的编写工作，由于缺少系统开发经验，虽然在大家的帮助下简单实现了一些管理功能，但这一模块还存在许多的不足之处，以下分别展示了该模块的一些操作界面图。

### 5.2.1 删除功能：删除指定的会员或者后台管理员

为管理员提供定期清除查询无效的注册用户及恶意注册的功能，以减少数据库负担和数据冗余。

这里主要是为管理而设计，由网络管理员进行操作，程序应首先识别用户身份，当具有相应的管理权限时才能允许其进行操作否则不能操作。这样就有效的保证了系统数据库中数据的安全防止恶意攻击及黑客破坏。

当管理员选择要删除的用户后点击删除按钮，程序将此用户及其相关信息从数据库中清除掉。

会员管理的删除界面如图 5.5 所示。



图 5.5 会员管理的删除界面图

输入项：会员或者后台管理员的用户名

输出项：显示删除是不是成功的提示

算法：主要是通过调用客户端的 delUser() 函数，来调用 CORBA 的方法

delUsualUser(String , short)完成删除功能

限制条件：无

没有解决的问题：删除界面不美观

### 5.2.2 更新功能管理：更新会员或者后台管理员的等级

当用户的基本信息发生变更时，应能提供给用户修改的功能，但不能修改用户名。其操作要求同样应是方便简捷不可繁琐。

选择点击修改按钮，然后页面打开并显示出该用户的信息内容。当用户修改完毕，点击确定保存后，程序将更新后的信息内容覆盖更新到数据库中相应数据表的记录。并返回成功操作的提示。

会员管理的更改界面如图 5.6 所示。



图 5.6 会员管理的更改界面

输入项：会员或者后台管理员的用户名

输出项：显示更新是不是成功的提示

算法：主要是通过调用客户端的 `execute_updateUsualUse()` 函数，来调用 CORBA 的方法 `updateUsualUse()` 完成更新功能

限制条件：无

没有解决的问题：只用来更改级别

### 5.2.3 查询管理：查询会员或者后台管理员

要对会员及后台管理员的资料进行管理操作时，管理人员应该能够方便的查询到会员及后台管理员的基本信息的内容才能进行相应的操作。

会员管理的查询界面如图 5.7 所示。



图 5.7 会员管理的查询界面图

**输入项：**会员或者后台管理员的用户名

**输出项：**显示出查询结果

**算法：**主要是通过调用客户端的 `execute_selectUsualUse()` 函数，来调用 CORBA 的方法 `selectUsualUse()` 完成显示功能

**限制条件：**无

**没有解决的问题：**只可以根据等级和用户名来显示，其它的附加的条件没有

### 5.2.3 增加后台管理员：增加不同权限的后台管理员

由于后台管理人员随时可能会发生人事上的变动，在此为管理员提供了增加后台管理员的功能。

这项功能主要是为管理而设计，同删除、更新功能模块一样由网络管理员进行操作，程序应首先识别用户身份，当具有相应的管理权限时才能允许其进行操作否则不能操作。这样就有效的保证了系统数据库中数据的安全防止恶意

攻击及黑客破坏。

增加后台管理员的操作界面如图 5.8 所示。



图 5.8 增加后台管理员的操作界面

输入项：后台管理员的用户名、昵称、密码、提问、回答、等级

输出项：显示出增加是不是成功

算法：主要是通过调用客户端的 `execute_addUsualUse()` 函数，来调用 CORBA 的方法 `addUsualUse()` 完成显示功能

限制条件：无

没有解决的问题：无

### 5.3 本章小结

这一章中主要内容是电子商务系统的实现，该章首先介绍了系统实现的一些关键技术与算法，实现的关键代码，购物网站的几个主要操作界面图；接着介绍的我所实现的后台管理方面的一些功能。

## 第六章 总结与展望

### 6.1 总结

由于当前得电子商务平台往往涉及到大量不同的网络、不同的操作系统、不同的数据库、异构的网络环境以及数个应用、不同的技术软硬件构架等等，如果不使用中间件技术就必须对每种情况都进行处理，其复杂程度是显而易见的。而使用了中间件技术以后，就可以将如此复杂的各种平台当成统一的平台来开发，使整个开发过程变得更加容易、高效。分布式应用的开发，尤其是面向对象的分布式应用的开发，对于系统开发者而言是极具挑战性的。**CORBA**、**DCOM** 和 **EJB** 是目前针对此类应用的主要分布对象计算模型和支持系统。一般来说，如果所开发的应用只面向 Windows 平台（Windows 3.x/95/NT），那么使用 **DCOM** 是最佳的选择；如果所开发的应用完全是使用 Java 语言编制的，那么就应该采用 **EJB** 的方案。但是，如果所开发的应用需要集成异构的平台，那么无疑应该选择 **CORBA**。通过 **CORBA** 系统的支持，分布应用的开发者可以采用不同的语言、操作系统和硬件平台来开发面向对象的分布式应用。

本文对基于 CORBA 组件技术的电子商务的开发项目进行了一些相关研究工作，现总结如下：

- 1、概述电子商务发展的历史，运作模式，主要作用及趋势。
- 2、介绍了客户关系管理的背景，分析了主流的两个客户关系管理软件。
- 3、探讨中间件技术的原理和工作机制，分析了几种中间件技术的优缺点，并详细介绍了 CORBA 组件技术的核心技术及工作原理等。
- 4、利用 CORBA 组件和 Java 技术实现了电子商务系统的设计开发，给出了具体实现的解决方案、对网上购物系统的数据库进行的设计。

### 6.2 系统的不足之处

经过前期一段时间的努力，毕业设计总算有了一定的成果。虽然仍存在着许多的不足之处，但看着经过自己的努力和付出而终于获得了一些的成果，还

是感到欣慰的。因为我们只是模拟现实中的电子商务系统而做的，因此在需求分析时没有完全到位，而且当今市场上基于 CORBA 技术的产品不是很多，因此给我们毕业设计带来了比较大的困难，所有的技术都是我们对照现有资料，一点一滴的摸索出来的，再加上时间和人手的限制，本系统不是做的很完善，有些功能比如统计分析、网上银行支付的设计等方面没有完全实现，系统的安全性方面也不是做的很好，这些都是我们以后应该进一步完善的。

### 6.3 技术展望

随着 Internet、Intranet 及 Extranet 在全球范围内的普及和发展，下一代的 Web 必须能够处理企业—企业（business-to-business）交易和客户—企业（consumer-to-business）多步骤交易过程中的复杂需求。为了实现这一目的，Web 必须发展成为一种能够执行在线交易软件的一种成熟的客户机/服务器结构。现在的 HTTP/CGI 结构存在一些不足，它不能满足上述这些新的要求。而各种不同的 CGI 扩展规范，例如 cookie、Microsoft Internet Services API (ISAPI)、Netscape Server API (NSAPI) 以及 Active Server page 等，也都不能从根本上解决问题。为了能够满足不断发展的电子商务的要求，Web 需要有分布式对象的功能。这就导致了人们所称的下一波的 Internet 革命——对象化的 Web (Object Web) 有产生。

建立 Object Web 的一种可行的方法，是使用 COBRA (通用对象请求代理结构，Common Object Request Broker Architecture) 和 Java。如果没有 Object Web 的话，对大多数对企业级的客户机/服务器市场有兴趣的厂商，以及喜欢面向对象编程的人而言，CORBA 及 Java 将只不过是一种容易使用的技术而已。因此业界都希望能将 COBRA 和 Java 各自的功能结合起来形成一种新的 Object Web 环境。软件界中除了 Microsoft 以外几乎所有其它的厂商所组成的联盟都力促 COBRA 和 Java 的结合以建立一个通用的 Object Web 环境。而 Microsoft 目前也正要据它的 ActiveX/DCOM 技术建立自己的 Object Web 环境。本文首先将分别对 CORBA 和 Java 进行介绍，接下来将会详述 CORBA/Java 以及 Object Web 的各种细节。

首先，那些 Java 的拥护者必须清楚，CORBA 不止是一种对象请求代理

(ORB)，它同时也是一个非常完整的分布式对象操作平台。CORBA 能跨越网路、程序语言、组件界限以及操作系统，因此能够大大扩展 Java 应用软件的使用范围。

对于那些 CORBA 的支持者，我们必须要指出的 Java 也不光只是一套能和 CORBA 结合的程序语言。Java 不仅是一种面向对象的语言，同时也是具备对象功能的可移植性操作系统。Java 允许 CORBA 对象在各种不同的机器上执行，无论是大型机、网路电脑(NC)，甚至是蜂窝式电话等。Java 简化了大型 CORBA 系统的程序分发——Java 利用字节码使得用户开发的对象能到处运行，这同时也为 CORBA 便携式代理提供了一个切实可行的方案。现在人们都知道 Java 几乎是开发客户机/服务器 CORBA 对象的最理想的语言。它内建的多线程(multithreading)、垃圾收集(garbage collection)及错误管理(error management)功能，使人们可以很容易开发出可靠的面向对象的应用。

这个方案成功的关键在于这两种技术的基础结构能够彼此互补。Java 弥补了 CORBA 的一些不足，而 CORBA 则可以为 Java 带来可移植的面向对象的软件环境<sup>[19]</sup>。

## 6.4 本章小结

本章是对该项目的总结与展望，首先对所做的工作做了一个总结；然后分析了系统的一些不足；最后是对组件技术的发展展望。

## 第七章 结束语

当前，全球信息化正势不可的改变世界，电子商务购物网站为了使人们的购物更加方便，快捷，成了电子商务的主角，而为了使网站的运营更加顺畅，针对网站的访问者（主要是购物者）的分析是必不可少的，因此网站的后台管理系统是必不可少的，这里主要是对客户的关系管理。有分析表明客户关系管理：“争取一个新客户的成本，相当于维护五个老客户”。“20%的大客户，为公司创造了 80%的利润来源”<sup>[20]</sup>。客户关系的重要性，已经不再是简单的买主和主顾的关系，让客户真正的体验服务，并且保持忠诚度，才能保持公司的竞争力，更好的促进销售。开放式电子商务系统的两个主要需求是互操作性和完整性。所有的客户和供应商应用程序都应该可以在一个灵活的、动态的、开放的框架中，越过不同平台，不同编程语言和商业布局来互操作。CORBA 可以从开放的电子商务环境的复杂性中抽象出来。CORBA 方便了电子商务系统和其它系统之间的交互作用，并使得和以前的应用程序的之间集成变得容易。

电子商务发展速度之快，让人难以想象。要跟上时代的步伐，就要掌握电子商务技术，运用电子商务技术。其次，网上安全和资源共享也越来越受关注，组件技术就很好解决了一要求。本系统结合组件技术和电子商务技术为一体，详细地设计了电子商务的流程，体验了电子商务带来的便利。

许多 CORBA 的核心概念对电子商务系统是有用的，例如，互操作性和综合性，平台、编程语言和安全机制等的灵活性，底层组件布局和网络的抽象，安全性功能的透明性，安全性的自动增强。

CORBA 是一个用于创建分布式应用程序的开发的标准解决方案。使用 CORBA，系统可以直接在网络上进行通信，而不管使用了什么程序设计语言来创建系统，也不管对象运行在哪些操作系统和平台上。每个常用的环境都可以使用 CORBA 解决方案，并且可以将它用于集成 Java、C++、Visual Basic 和 COBOL 编写的应用程序，它可以运行在嵌入式系统、PC、UNIX 主机和大型机上。如果 CORBA 在这些环境中运行，那么应用程序可以无缝地交互。除了它广泛的可用性以外，CORBA 还具有广泛的基层结构，该结构支持分布式业务对象需要的所有功能。该结构包括诸如交易、安全性和消息发送等重要的分布式服务<sup>[21]</sup>。

因此我们选择当今 3 种具有代表性的主流技术之一的 CORBA 组件技术作为系统开发的主要技术有着其无可比拟的先进性与适应性。

本系统比较全面地完成了网上购物和顾客管理等方面的功能，运用当前最新的技术来开发，适应时代的趋势。开发过程中遇到了种种问题，然后再去解决，带来了非常大的成就感，尤其是 CORBA 组件开发中遇到的问题，比如移植性、兼容性等。

总之，这次为了完成硕士毕业论文参加了这个模拟电子商务平台的开发，虽然我们的这一电子商务交易平台还不能形成一个完整的产品，但是通过这次开发，使我对电子商务概念和组件技术都有了一个全新的认识，不管从经验上还是从技术上，都给我一个很大的提高。在今后我将利用工作之余积极关注有关电子商务系统开发方面的文章，进一步加强我对 CORBA 组件技术和 JAVA 语言的了解，通过这些方面的知识的积累以进一步对该系统的一些功能进行完善。由于本人对论文相关理论及技术的研究还欠深入透彻，对基于 CORBA 组件技术的电子商务系统实际开发的认识还不够全面，论文叙述中必然存在着一些错误和不足之处，希望读者能够谅解。

## 致谢

历时近一年的论文研究与撰写终于完成了。这是用无数个节假日的闭门攻读，无数个夜晚的精心耕耘，收获了这3万多字的文稿。看着凝聚着我的汗水与心血的成果，心里涌动着这样的感叹：采得百花酿成蜜，不畏辛苦不畏难。

然而我付出的这一切，与给予过我帮助的老师、同学和朋友的无私奉献，是多么的微不足道。

在论文的撰写过程中，我得到了导师范存海、王力生两位教授的悉心指导和谆谆教诲，他们的渊博的知识和严谨的治学态度将牢记于我心。尤其是王力生教授克服距离的障碍，在百忙之中不断地通过电话、Email 和我联系，从我确定开题意向以后一直对我的论文完成情况进行关注，督促我的写作进度，指导我的写作方法。在整个论文撰写过程中，王力生教授一直是我领跑人，在此我难以用文字来表述我的感激之情。一日为师，终身难忘。王力生教授将在我今后的人生道路上成为我永远的良师益友。

南京迪玛斯公司老总范存海老师也给予了我许多关心和帮助，对我的设计和论文提出了宝贵的意见。在此，我表示衷心的感谢和深深的敬意！

同时，向参加论文答辩会的专家、老师，以及其他给予我帮助的老师、同学、朋友表示诚挚的谢意！

2006年3月

## 参考文献

- [1] 王衛生.电子商务发展新论.民邮电出版社,2003
- [2] 吴小明.Internet 上一种新型的 CORBA 应用安全防护模型.华中科技大学出版社,2001
- [3] 创能公司.创能客户关系管理软件概述. <http://www.chuangneng.com/fangan/crm.htm>
- [4] 用友集团.用友 CRM 技术原理 <http://www.ufsoft.com.cn/product/crm/beijing.pdf>
- [5] CORBA 组件技术 Services Specification <http://www.omg.org>
- [6] 烂柯.什么是组件的概述.人民邮电出版社,2002
- [7] 李文军, 同晓聪.分布式对象技术.机械工业出版社,2004
- [8] Rogers Cadenhead.循序渐进 Java2 教程.人民邮电出版社,2003
- [9] 蔡小文.中间件在电子商务中的应用.  
<http://www.b2bbeijing.net/info/wenzhai/infowz21060108.htm>
- [10] 秦学文.基于 CORBA 的电子商务系统的安全性  
<http://www.mydown.com/tests/207/207398.html>
- [11] 苏洋. CORBA 第一讲. 北京东方通信科技发展有限公司  
[http://www.ccw.com.cn/htm/app/course/01\\_11\\_7\\_3.asp](http://www.ccw.com.cn/htm/app/course/01_11_7_3.asp)
- [12] 陈文新.方兴未艾的 CORBA.清华大学出版社,2003
- [13] ORB 结构.<http://www.huihoo.com/corba/orb.html>
- [14] 魏东平, 郭文豪.CORBA 体系中 ORB 互操作性实现机理的研究.  
<http://engine.cqvip.com/content>
- [15] 对象请求代理 (ORB).  
<http://dev-club.esnai.com/club/bbs>
- [16] 吴明.IDL 定义.  
<http://www.w3.org/TR/2000/REC-DOM-Level-2-Style-20001113>
- [17] 龙湘明.CORBA 程序设计指南 (入门).华东科技大学出版社,2002
- [18] 高新.关于 C/S 结构和 B/S 结构  
<http://www.chinaui.com/article/detail/20051011113425.html>
- [19] 江明.Java 与 CORBA 技术结合的前景展望  
<http://www.javaresearch.org/article/showarticle.jsp>
- [20] 电子商务网站的客户管理关系.  
[http://www.haodong99.com/home/Article\\_Print.asp](http://www.haodong99.com/home/Article_Print.asp)
- [21] 蒋建新.对象标准 COM 和 CORBA.  
<http://edu.100down.com/it/program/VC/102217323.htm>