

摘要

随着互联网的迅猛发展和相关技术的成熟，在互联网上开展各种服务已经成为许多企业的急切需要，电子商务也逐渐渗透到了人们生活的各个角落。我国某著名酒厂集团地处山区、交通不便、信息闭塞，因此网络营销对其尤为重要。该酒厂要通过网络营销建立会员制俱乐部，给予厂商和消费者沟通的最直接的机会，吸引更多潜在客户，把一般客户发展为重点客户，提高企业竞争力，因此推行网上商城项目的开发。

作者的主要研究工作如下：

(1) 对 ECO (企业核心对象) 技术做了深入的研究，包括建立企业逻辑模型、用状态机建立动态模型、数据库的自动生成和自动演化、EcoSpace、如何使用 OCL 和 EAL 对对象进行处理、ECO 组件组的应用等等。ECO 是基于 UML 模型驱动的面向对象的数据库开发框架，可以用来轻松地实现信息模型设计及基于信息模型的应用程序。ECO 提供了快速模型驱动开发的方法，让设计者将更多的精力投入到考虑如何建立企业模型，而不是编代码。ECO 能实现数据库的自动生成和自动演化，数据库平台无关性，模型的高可复用性。

(2) 根据某酒厂网上商城系统的特点，对系统进行了详细地分析设计。网上商城系统主要包括电子商城子系统和内部业务管理子系统。电子商城子系统主要功能模块有商品展台、购物车、会员中心等，会员中心里主要包括会员活动、我的订单、消费记录、兑奖申请、我的消息、网上调查。电子商城子系统里，根据分类对各种商品进行展示、说明，普通用户可以注册成为会员，选中商品放入购物车，并填写订单内容，生成订单提交。在会员中心里，会员可以查看自己的订单，消费记录，参与会员活动，兑换奖品等。内部业务管理子系统主要功能模块有会员管理、积分管理、订单管理、信息管理、商品管理、货款结算、专卖店监管等。在内部业务管理子系统里，能对商品和信息进行发布和修改，对会员信息进行管理，对订单进行审核等处理，对货款进行审核和结算等处理，对各专卖店进行监管等。

(3) 把 ECO 技术应用于网上商城系统的开发，最终实现了该系统。对 ECO 技术中的订阅机制和状态机的实现进行了阐述。项目中，主要把订阅机制用在

派生属性上，派生属性类似于计算字段，但派生属性是非持久化，不占用数据库资源，并且它是定义在模型里，可复用。在系统中，以销售单的状态机为例，定义好销售单的有效、作废状态，设置触发器，用 EAL 语句描述事件：产生日志。当销售单实例的状态发生转移时，就会产生日志。由于在状态机里用的是 OCL 语句和 EAL 语句，因此可实现数据库平台无关性。

ECO 技术提高了系统的开发效率，能以较小的代价轻松地适应需求变化，是一个优秀的模型驱动的面向对象的数据库开发框架。

关键词：电子商务，ECO，MDA，状态机

Abstract

With the rapid development of Internet and related technology's maturing, many enterprises need to develop various services on the Internet, and the electronic commerce gradually penetrates our lives. A famous distillery of our nation lies in a mountainous area, the traffic is not convenient, so Internet marketing is very important to it. The distillery wants to found member club by the Internet marketing. It can let distillery communicate with customers more directly, and attract more potential customer, and enhance the advantage in competition. So the distillery wants to develop the Web-Shop system.

The main research in this paper is:

(1) Have the research on the ECO technology. ECO is an object-oriented database-development frame of model driven architecture with UML; it can easily achieve the design of information model and the application base on the information model. ECO provides the method of fast MDA; it can let the designer spend more time on the model. ECO can achieve the auto-create and auto-evolve of the database; it also can achieve being independent of database; the model with ECO can be reused.

(2) According to the characteristic of the Web-Shop system, analyze the main user-case and main flow of the system. The Web-Shop system includes the electronic-shop subsystem and the inner operation management subsystem. The main modules in the electronic-shop subsystem are products show, cart and member center. Member center includes member activities, my orders, my short messages, etc. In the electronic-shop subsystem, show various products by sort, the common consumer can register to become the member. The member selects the products, put them into the "cart", and then submit the order. In member center, member can see their own orders and consume-records. They can participate in activities. When they have enough scores, they can apply to exchange award. The main modules in the inner operation management subsystem are member management, score management, order management, information management, products management, etc. In the inner

operation management subsystem, products and information can be released; member information can be managed; checking the orders; checking the payment for goods and balance it; monitor the monopolistic-store, etc.

(3) Achieve the Web-Shop system, ECO technology is applied in it. Expound that how to achieve the Subscription Mechanism and the State Machine. In the project, the Subscription Mechanism is applied in the deriving attribute. The deriving attribute is similar to the calculated field. But the deriving attribute is transient. It does not cost the database's resource. And it is defined in the model, so it can be reused. In the system, take the sale-bill for example, define the states of the sale-bill in the State Machine, set the trigger, and describe the event by the EAL. When the state of the instance transfers, it will spring the event. Because use the OCL and EAL in the State Machine, it is independent on the database platform.

ECO improves the efficiency of the system development. It can adapt the requirement's change with less cost. ECO is an excellent object-oriented database-development frame of model driven architecture.

Keywords: electronic commerce, enterprise core objects, model driven architecture, state machine

独 创 性 声 明

本人声明，所呈交的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。尽我所知，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，也不包含为获得武汉理工大学或其它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了谢意。

签 名: 王秉 日期: 2007.5.18

关于论文使用授权的说明

本人完全了解武汉理工大学有关保留、使用学位论文的规定，即学校有权保留、送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

(保密的论文在解密后应遵守此规定)

签 名: 王秉 导师签名: 王秉 日期: 2007.5.18

第1章 绪 论

1.1 课题研究背景及发展现状

随着因特网的在全球的迅猛发展和广泛应用，电子商务受到人们越来越多的关注，并渗透到人们生活的各个角落，改变着社会经济的各个方面。电子商务与传统的商务方式相比具有明显的特点，如高效性、方便性、安全性、集成性和可扩展性等。^[1]目前，电子商务热潮正席卷全球，成为 IT(Information Technology, 信息技术)产业最为热门的话题和竞争的焦点。美国于 1997 年 7 月 1 日发布了《全球电子商务纲要》(A Framework for Global Electronic Commerce)，1998 年又通过法令，对电子商务实行免税。欧盟于 1997 年 4 月 15 日发布了《欧盟电子商务行动方案》。^[2]美国和欧盟还在 1997 年底共同发表了有关电子商务的联合声明。日本、新加坡、韩国、马来西亚、菲律宾等亚洲国家也先后颁布了有关电子商务的基本法规。日本政府还成立了电子商务振兴委员会，并投入 300 亿美元发展电子商务。^[3]我国政府也积极制定了中国的电子商务发展战略，2000 年 3 月的人大和政协会议期间，30 多位人大代表联合提出制定我国的电子商务法，全国政协把关于电子商务立法的提案列为 1 号议案。大量事实表明，电子商务正在全球范围内蓬勃发展。

1.1.1 电子商务的概念和基本商务模式

(1) 电子商务的概念

简单地讲，电子商务就是利用现代先进的电子技术从事各种商务活动的方式。虽然各国政府和企业、用户对电子商务都十分重视，电子商务发展也很快，但对于电子商务的确切定义，至今没有一个统一的说法，其中比较系统、全面且有代表性的观点有以下几种：

美国的 Emmelhainz 博士在其专著《EDT 全面管理指南》中把电子商务定义为“通过电子方式，并在网络基础上实现物资、人员过程的协调，以便商业交换活动”。

加拿大专家 Jenkins 和 Lancashire 在《电子商务手册》中从应用角度定义 EC

为数据（资料）电子装配线的横向集成。

美国政府在“全球电子商务纲要”中给出的电子商务定义是：电子商务是通过互联网进行的各项商务活动，包括广告、交易、支付、服务等活动。^[4]

最早明确提出电子商务概念的 IBM 公司认为电子商务是当用户（商家）通过互联网融入传统信息技术系统的丰富资源库时所发生的，它是动态的和交互式的，它通过 WEB 技术把客户、生产商、供应商和雇员连接在一起。^[5]

欧洲经济委员会于 1997 年 10 月在全球信息标准大会上的定义：“电子商务是各参与方之间以电子方式而不是以物理交换或直接物理接触方式完成任何形式的业务交易”。这里的电子方式包括电子数据交换（EDT）、电子支付手段、电子订货系统、电子邮件、传真、网络、电子公告系统条码、图象处理、智能卡等。

1997 年在法国首都巴黎，由国际商会组织的世界电子商务会议，给出目前关于电子商务概念的最权威论述：电子商务是指对整个贸易活动实现电子化。从概念的外延看：交易各方以电子交易方式而不是通过当面交换或直接面谈方式进行的任何形式的商业交易；从技术方面看：电子商务是一种多技术的集合体，包括交换数据、获得数据以及自动捕获数据等；从涵盖的业务看：电子商务包括信息交换、售前售后服务、销售、电子支付、组建虚拟企业等。^[6]

虽然人们都对电子商务的认识和理解存在着差异，但大体上可以分为广义电子商务和狭义电子商务两大流派。广义的电子商务是指包括电子交易在内的利用 IT 技术进行的全部商业活动。所谓狭义的电子商务是指借助计算机网络进行网上的交易活动，是商家和客户之间的联系纽带。

（2）电子商务的基本商务模式

目前，主要的电子商务的商业模式^[7]主要有四种：

BtoB（企业对企业）：电子商务主要是进行企业间的商品批发业务。电子商务其实远不止是网络零售业，更核心的是市场潜力比零售业大一个数量级的企业级电子商务。BtoB 电子商务模式是一个将买方、卖方以及服务于他们的中间商之间的信息交换和交易行为集成到一起的电子运作方式。而这种技术的使用从根本上改变企业的计划、生产、销售和运行模式，甚至改变整个产业的基本生产方式。以此，这种企业之间的电子商务经营模式越来越受重视，是电子商务的一个重要的发展方向。^[8]

BtoC（企业对个人）：亚马逊网上书店就是采用这种商业模式的网站。它利

用网络对广大的消费者介绍和推销书，通过网上银行进行交易，利用快递公司运送产品。这种方式也称为电子零售。^[9]

CtoC（个人对个人）：这种模式中，商品直接由消费者出售给消费者，例如个人以分类广告形式出售住房、汽车等。因特网上的个人服务广告和知识技能的出售也是 CtoC 的例子。

CtoB（个人对企业）：这个类别包括个人使用因特网将产品和服务出售给企业的情况，也包括个人寻求买家，与其进行交涉并达成交易的情况。^[10]

1.1.2 模型驱动架构

对象管理组织（OMG）于 2001 年 7 月发布了全新的软件开发框架——模型驱动架构（Model Driven Architecture, MDA）。^[11]MDA 的关键之处就是将模型作为软件开发的核心制品，严格区分系统的功能规约与实现细节。^[12]在 MDA 中，软件开发过程是由对软件系统的建模行为驱动的。它把建模语言当作编程语言而不仅仅是设计语言。它能够创建出机器可读和高度抽象的业务模型，这些模型以独立于实现的技术开发，以标准化的方式存储，然后根据不同的平台，对业务模型进行模型编译，生成相应平台的实现模型。^[13]这样就分离了传统的业务模型和平台相关的实现模型，从而摆脱了对技术平台的依赖。

MDA 将软件系统的模型分离为平台无关模型（PIM）和平台相关模型（PSM）。^[14]平台无关模型（PIM）是对系统高层次的抽象，其中不包括任何与实现技术相关的信息；平台相关模型（PSM）是跟特定平台相关的模型。在 MDA 技术实施时首先搭建平台无关模型（PIM），然后根据已经实现的模型转换规则，将 PIM 转换成 PSM，最终生成应用程序代码和测试框架。^[15]整个 MDA 技术的实现步骤如下：

（1）首先我们使用平台无关模型（PIM）从如何以最好的方式支持商业逻辑的角度来对系统进行建模。PIM 是对系统的一种高层次的抽象，与具体的实现技术无关。在此过程中，我们会不断根据客户需求和其它因素对 PIM 进行精化，最终使得它能够十分精确地描述我们的系统。

（2）然后 PIM 可以被转换到一个或者多个平台相关模型（PSM），对于每种特定的技术平台都会生成独立的 PSM。PSM 是针对你选择的实现技术、平台，对你的系统度身定做的模型。由于现今的很多系统都跨越多种技术，所以对于一个 PIM 拥有多个 PSM 是很正常的。这是 MDA 中最复杂的，也是很重要的一

步。

(3) 有时候从 PIM 自动生成的 PSM 并不能使软件开发人员完全满意，这时根据平台的特性对 PSM 加以修改，但是对 PSM 的改变也要求及时反映到 PIM 中去。

(4) 软件开发人员对 PSM 进行不断的精化，准备最终以此模型指导生成工具生成高质量的应用程序代码和测试框架。

(5) 最后一步就是将每个 PSM 都转换到代码，由于 PSM 与具体的系统实现技术已经很接近，因此这种转换来得比较直接。^[16]

MDA 开发途径看起来和传统的开发方法很类似，但是这里有一个至关重要的不同，就是设计模型在 MDA 中是可执行的，这里的可执行是指通过设计模型能够产生实际可执行的代码，即该模型是能够产生出的。^[17]传统意义上，从模型到模型，从模型到代码的转换主要是手工完成的。虽然有些工具可以从模型中产生部分代码，但是它们一般都是通过指定的模板来生成，有很大的局限性，大多数的工作仍然需要手工来完成。而与之不同的是 MDA 中的转换通常都是通过工具来完成的，这促进了软件开发人员的高效率。^[18]

在基于 MDA 的软件开发过程中，MDA 将抽象层次提高到模型的高度，在 MDA 环境中开发者把重心转移到业务逻辑的建模上，而不管最终目标平台是 Web Services, EJB, COM+, 还是其它基于构件或事务处理的中间件平台，他们投入更多的时间和精力来关注功能性需求的分析和设计，而不会过早的投入到具体的实现中。^[19]当然，MDA 的出现确实给传统软件开发带来了巨大的冲击，但它所带来的促进软件生产率、增强软件可移植性和互操作性、提高文档和维护便利性等方面的好处使人们迫切希望软件产业的又一个黄金时代的到来。^[20]

1.1.3 国内外研究现状

在国外发达国家，电子商务的发展非常迅速，通过 Internet 进行交易已成为一种潮流。世界各大 IT 厂商积极推出面向电子商务的软件产品和解决方案。基于电子商务推出的商品交易系统方案、金融电子化方案和信息安全方案等，已逐步成为国际信息技术市场竞争的焦点。^[21]随着电子商务的日益发展，产生了虚拟银行、网络营销、网上购物、网络广告等新兴信息产业。^[22]

在我国，电子商务的发展前景无疑也是非常远大的。^[23]受到电子商务发展的有力拉动，中国个人网上支付的市场规模发展迅速。2001 年中国网上支付的

市场规模为 9 亿元，2004 年该规模增长为 75 亿元，年均复合增长率（CAGR）为 102.7%，预测，未来几年我国网上支付的市场规模继续扩大，2007 年我国网上支付市场规模将达到 605 亿元。随着传统支付观念的改变，人们越来越多地使用网上支付。2001 年中国网上支付的用户规模为 374 万，2004 年该规模增长为 1830 万，年均复合增长率（CAGR）为 69.8%。预测，未来几年我国网上支付用户规模将继续扩大，2007 年我国网上支付用户规模将达到 5325 万。^[24]

企业电子商务系统的现状及发展趋势如下：

（1）从企业内部到企业之间

过去，我们注意企业内部的资源，传统的 ERP 主要解决的就是企业内部资源的优化问题、企业与企业之间的竞争演变成企业供应链与另一个企业供应链的竞争，企业管理的范围不但包括自身的资源，还要延伸到供应商、客户与合作伙伴，这就出现了 CRM，除了 CRM 之外，还出现了一些新的概念；PRM：合作伙伴关系管理系统；ERM：员工关系管理系统（HR），包括 eLearning 系统；SRM：供应商管理系统（不同 SCM）；这些软件系统采用的技术都是 Web，可以解决企业间的很多实际商业问题，它们代表了企业电子商务的新趋势。^[25]

（2）从后台管理到前台支持

由于企业认识到互联网和电子商务对企业未来的重要性，越来越多的企业不再仅满足传统的后台应用，比如传统的财务、库存及人力资源管理等，一些新的应用，如 CRM，SEM，EIP 等，更接近企业的客户、供应商、员工和合作伙伴，即所谓的“前台”应用，这些应用逐渐在企业级应用中流行起来。

（3）从 ERP 到协同商务

ERP 是企业级应用的一个的经典案例。早期的企业级应用基本上是一些信息“孤岛”，由一些传统的部门级应用如财务、库存管理等组成。20 世纪 90 年代初期，ERP 成功地把企业后端处理和信息流集成在一个单一地系统中。最近几年，一些企业开始实施 CRM、SCM 及数据仓库等有前端处理地应用系统，但是这些功能强大的应用往往不能很好地集成，企业信息“孤岛”再次出现。未来的电子商务应用，不仅要求企业内部的各种应用（ERP、CRM、SCM 等）能够集成，而且要求不同企业之间的应用能够集成，即所谓的“协同商务”。

（4）从分散管理到集中管理

今天，企业局部的、分散的系统到处存在，维护成本居高不下，业务流程难以协调一致，管理数据支离破碎。通过使用多层体系结构，简化桌面配置（从

“Desktop”到“Webtop”), 实现低成本的系统推广和维护(集中维护和发布),可以使企业内部数据和业务流程集中管理。数据集中,集中管理是企业级电子商务应用的大势所趋。^[26]

目前,电子商务系统主要是在.NET 平台或 J2EE 平台上进行开发的。Borland 公司发布的 ECO (Enterprise Core Objects, 企业核心对象), 目前只支持.NET 平台而不支持 J2EE 平台,为基于.NET 的开发提供了快速的 MDA 解决方案。现在,除了 Borland 公司, 国内外其他一些软件公司也已经推出了较为成熟的工具和平台来支持 MDA 方法。如法国 Softeam 公司的 Objecteering 工具, 它声称是第一个完全支持 MDA 方法的工具。诸如此类的产品还有 Interactive Objects 公司的 ArcStyler, TogetherSoft 公司的 Together ControlCenter, Compuware 公司的 OptimalJ, 国内楚凡公司的 Trufun Kant 等等, 其中 OptimalJ 和 ArcStyler 是目前开发 J2EE 平台的首选。IBM(Rational)更加关注自己提出的 RAS(Reusable Asset Specification) 概念并且在 Eclipse 下了很多工夫, 最终的目标是所有的 IBM 软件都基于 Eclipse 框架。Eclipse 中体现 MDA 思想的就是 EMF (Eclipse Model Framework)。国内 ERP 软件领袖企业之一的金蝶软件也声称在其 BOS 系统中成功地应用了 MDA 的思想。因此毫无疑问 MDA 将成为未来软件开发的必然趋势。^[27]

1.2 课题研究的目的和意义

本课题的研究是基于我国某著名酒厂集团信息化的软件开发项目之一: 网上商城系统。该项目主要包括电子商城子系统和内部业务管理子系统。

该酒厂在全国建立以特许经营专卖店为销售终端的分销网络, 建立营销体系: 以总部为中心, 片区为分支, 专卖店为终端的三级营销网络。现在该酒厂要建立以终端消费者为中心的会员资料库以及考虑到营销策略上的需求, 因此推行网上商城项目的开发。

该酒厂销售业务, 现已实现内部日常业务、片区沟通、客户网上业务等基础管理体系的信息化, 在此基础上, 2005 年设计了专卖店分销网络系统, 完成了系统投运的技术测试工作, 现正在进行了一些推广前的配套准备工作, 专卖店分销系统是会员系统的业务支撑平台, 在其实施的前提下, 可保证后续以终端消费者为中心的会员服务网络体系的正常运作。

本课题研究的主要目的是研究 ECO 相关技术, 学会用 MDA 的思想来开发系统, 并且把这些技术应用到该酒厂网上商城系统的设计与开发中, 最终实现

该系统。

本课题研究的意义主要是：学习 ECO 技术，用这种快速 MDA 的思想来开发应用系统大大提高了开发效率，能以较小的代价轻松适应需求的变化，系统模型可复用，提高了软件质量和预见性。由于该酒厂集团地处山区交通不便、信息闭塞，因此网络营销尤为重要。网上商城系统的实现，使该酒厂通过网络营销建立会员制俱乐部，给予厂商和消费者沟通的最直接的机会，吸引更多潜在客户，培养客户忠诚度，通过俱乐部营销把一般客户发展为重点客户或关键客户。网络营销也为企业降低了成本。在网上发布信息，代价较低，将产品直接向消费者推销，可缩短分销环节，发布的信息谁都可以自由地索取，可拓宽销售范围，这样可以节省促销费用，还可根据订货情况来调整库存量，降低库存费用，从而降低成本，使产品具有价格竞争力。

1.3 本文所做工作

1.3.1 主要研究内容

(1) 研究了 ECO 相关技术：建立企业逻辑静态模型（类图）和动态模型（状态机），EcoSpace（执行时期框架），OCL（Object Constraint Language），EAL（ECO Action Language），ECO 组件组。ECO 是基于 UML 模型驱动的面向对象的数据库开发框架，可以用来轻松地实现信息模型设计及基于信息模型的应用程序。ECO 能实现数据库的自动生成和自动演化，数据库平台无关性，模型的高可复用性。

(2) 根据该酒厂网上商城系统的特，对系统的主要用例和主要流程进行了分析与设计。网上商城系统主要包括电子商城子系统和内部业务管理子系统。电子商城子系统里，根据分类对各种商品进行展示、说明，普通用户可以注册成为会员，选中商品放入购物车，并填写订单内容，生成订单提交。在会员中心里，可以查看自己的订单，消费记录。会员有足够的积分后还可以申请兑换奖品。在内部业务管理子系统里，能对商品和信息进行发布和修改等，对会员信息进行管理，对订单进行审核等处理，对货款进行审核和结算等处理，对各专卖店进行监管等。

(3) 将 ECO 技术应用于该酒厂网上商城系统的开发，并且实现了该系统。对 ECO 技术中的订阅机制和状态机的实现进行了阐述。订阅机制是 ECO 架构

的基础核心之一。在对象之间的关系上，ECO 需要建立一种互通互信的“订阅”机制，比普通的虚拟环境更为高级。在状态机里定义好类的若干状态和触发器，当类实例的状态发生转移时可触发一些事件。

1.3.2 本文组织结构

本文共分为五章，主要的内容概要如下：

第 1 章概述了本课题的研究背景，国内外对电子商务系统以及 MDA 的研究现状。阐述了本课题研究的目的和意义。简要的说明了论文所研究的主要内容和论文的组织结构。

第 2 章对 ECO 相关技术进行了综述。包括建立企业逻辑静态模型（类图）和动态模型（状态机），把对象模型映射为关系数据库模型，EcoSpace（执行时期框架），OCL（Object Constraint Language），EAL（ECO Action Language），ECO 组件组。

第 3 章阐述了网上商城系统的需求分析，并对该系统进行了详细地设计。用 ECO 技术建立企业逻辑静态模型（类图）和动态模型（状态机），把对象模型映射到关系数据库，实现 OR-Mapping。

第 4 章对网上商城系统的实现进行了详细地阐述。网上商城系统主要包括电子商城子系统和内部业务管理子系统。电子商城子系统主要包括商品展台模块，购物车模块，会员中心模块等。内部业务管理子系统主要包括会员管理模块，订单管理模块，商品管理模块，货款结算模块等。

第 5 章对所做的工作进行了简要的总结，并展望了进一步的工作。

第2章 ECO 的相关技术和开发架构

近年来，越来越多的人认识到使用面向对象的数据库开发框架来进行大型数据库应用的开发有着很多的好处。在项目设计阶段，使用 UML 建模语言设计业务域对象模型，从模型出发，定义业务域对象，然后使用标准的美观的组件对业务域对象进行操作，设计某些方法将业务域对象保存到数据库，或者从数据库加载，这就是通常所说的 OR Mapping，对象——关系映射问题。

同 RAD (Rapid Application Development) 开发方式相比，面向对象的开发方式比较适合信息模型比较复杂的大型系统，同时面向对象的数据库开发框架可以很容易的实现 Lazy load 的延迟加载技术，因此会产生更少的网络信息 round trip，提供较好的性能。另外，框架一般都提供数据库平台无关性的支持，适合于对移植性有较高要求的开发。此外，可以将 GUI 同数据库解偶合，更容易扩展和维护。不过缺点是对程序员的素质要求的更高。

一般的面向对象数据库开发框架的要求：

对象持续性：框架必须能够以对象的方式存取数据，能存取复杂对象，如复合对象，支持对象之间的关联，比如继承，聚合，关联。

支持对象查询：框架应该提供一种方式可以根据条件查询复合对象以及对象集合。支持标准的对象查询语言，如 Object Constraint Language (OCL) 或者 Object Query Language(OQL)。

支持对象主键：大家都知道在关系型数据库中，主键可以用来唯一标识一条数据记录。面向对象的框架中也应该有一个对象主键来唯一标识一个对象。

支持事务：创建还必须支持事务，一个交易必须是原子级别的，或者完全成功，或者完全失败，支持类似于数据库的提交和回滚操作。交易结果要求是一致的、独立于其它交易。

支持 XML：随着 XML 这种元数据语言的越来越广泛的应用，框架必须能从 XML 文件中存取对象。

性能优化：如果面向对象框架默认产生的操作性能不高的话，应该允许用户优化性能。

支持多种应用：应该即支持桌面型应用开发，也能支持 C/S 和多层数据库

开发以及 Web 开发。

数据库及数据存取 API 无关的：可以随时切换开发及数据库发布平台。

兼容现有的数据感知组件的：这样在使用面向对象开发框架改动已有系统时，可以保证平滑的移植。^[28]

从 Borland 公司收购模型驱动开发的领先厂商 Bold，以及双向建模工具 Together 之后，Borland 公司开始整合两者的资源，终于推出了可视化的基于 UML^[29]模型驱动的 ECO (Enterprise Core Objects) 这样一套优秀的基于 UML 模型驱动的面向对象的数据库开发框架，可以用来轻松地实现信息模型设计及基于信息模型的应用程序。^[30]ECO 是 Borland 公司根据 MDA (Model Driven Architecture, 模型驱动架构) 发展出来的技术，从 ECO 的名称中知道 ECO 的目的是使用来开发企业级的软件。ECO 是一个完整的框架，它是由许多高端的技术所结合而成，可视化设计接口、EcoSpace (执行时期框架)、OCL (Object Constraint Language)、EAL (ECO Action Language)、ECO 组件组等等。

2.1 ECO 的相关技术

2.1.1 可视化设计接口

在产品的需求分析阶段，我们首先要建立数据库程序的信息模型，一般来说信息模型主要是指基于 ER 图的实体关系模型，这是因为我们使用的数据库大部分都是关系型数据库，虽然有些数据库，比如 Oracle 有面向对象的特性，但不是很完善，一般很少使用。而关系型数据库有一个很大的问题就是无法直观的体现面向对象的思想，关系型的 ER 模型能够清晰地描述业务域的静态的数据视图，但你无法从模型获取实体的操作及其相互之间的交互。同时，也很难在关系型数据库中简单地实现继承、重载、多态等等面向对象的技术，因此现代数据库开发方法所提倡的面向对象的编程思想无法简单、清晰、平滑地映射为关系型数据库中的表结构。

统一建模语言 (UML) 是一种以可视化的方式建立软件系统框架，并进行文档化的语言。UML 语言是对当今软件工程领域成熟设计实践的一个总结，并且已经被实践证明是可以成功地描述大型的复杂系统的。^[31]目前国内很多的大型公司已经开始在软件开发过程中使用 UML 作为一种标准的信息模型设计语言了。ECO 就是基于 UML 的，它内置了一套 Together 简化版的 UML 建模工具，

当然我们也可以使用其它的建模工具如 Rose 或者 ModelMaker 来进行 UML 设计，然后将生成的模型通过 XMI 模型文件导入系统中。

面向对象的 UML 类图则可以说是对 ER 模型的一个扩展，它对实体之间的关系以及相互之间的作用也进行了描述，ER 模型只是对要进行保存的数据进行的模型化，而类图则包括了全部的类实体的属性以及它们的操作和相互作用，它可以使我们对业务域问题有一个更精确的视图，通过使用各种类图技术可以更容易地、也更快速地建立正确的软件系统。^[32]

基于 ECO 的数据库开发革命性的一点就是允许我们直接把基于 UML 的类图映射为关系型数据库的存储，而无须手工的通过代码进行转换。要注意一点的是，ECO 同其它建模工具如 Rose，ModelMaker 不同，它生成框架代码时只使用了 UML 中的类图。^[33]而 Rose 等可以利用 UML 图中的类图、协作图等其它 UML 元素来生成代码框架，但是 Rose 不负责生成对象模型对应的关系数据库模型^[34]。

下面举例说明如何使用 ECO 建立企业逻辑静态模型和动态模型，如何将对象模型映射为关系数据库模型。

(1) 建立企业逻辑静态模型

在集成开发环境 Borland Developer Studio 2006 (ECO III) 里先用 UML 类图设计企业逻辑静态模型。打开 BDS，建立一个工程 state，切换页面到 ModelView，可以看到默认有一个 Package_1 节点双击该节点，将调出内置的模型编辑器可以看到 Tool Palette 上提供了 Class(类图)，Association(关联)，Generalization(继承关系)等 UML 模型元素。^[35]

设计 4 个类：销售单 (t_pos_xsd)，销售单明细 (t_pos_xsdmx)，入库单 (t_pos_rkd)，入库单明细 (t_pos_rkdmx)。设置各个类的属性和操作，以及类与类之间的关联，如图 2-1 所示。

武汉理工大学硕士学位论文

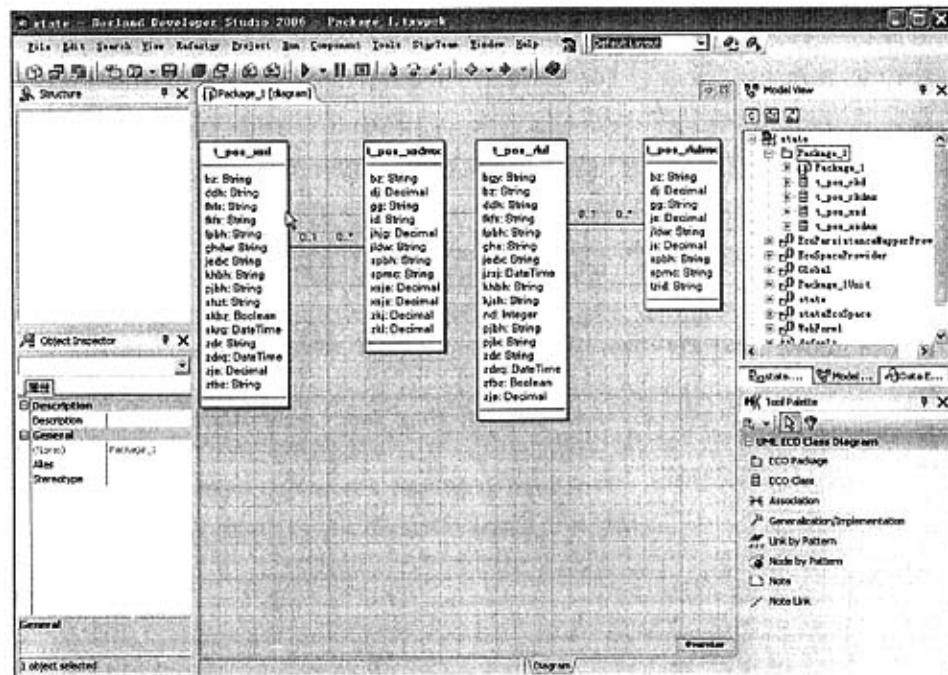


图 2-1 用类图设计企业逻辑静态模型

选中 `t_pos_xsd` 类，执行右键的 Go to Definition 命令，可以看到 Together 自动为我们生成的 ECO 类的代码了，如下：

```
t_pos_xsd = class(System.Object, ILoopBack)
    {$REGION 'ECO generated code'}
public
type
Eco_LoopbackIndices = class
public
const
    Eco_FirstMember = 0;
const
    Eco_MemberCount = (Eco_FirstMember + 18);
const
    skbz = Eco_FirstMember;
const
    skrq = (Eco_LoopbackIndices(skbz + 1));
const
    State_1 = (Eco_LoopbackIndices(skrq + 1);
```

```

const
  ztbz = (Eco_LoopbackIndices.State_1 + 1);
const
.....
t_pos_xsdListAdapter = class(ObjectListAdapter, It_pos_xsdList)
public
  constructor Create(source: IList);
  function get_Item(index: Integer): t_pos_xsd;
  procedure set_Item(index: Integer; Value: t_pos_xsd);
  property Item[index: Integer]: t_pos_xsd read get_Item write set_Item;
default;
  function Add(value: t_pos_xsd): Integer;
  function Contains(value: t_pos_xsd): Boolean;
  function IndexOf(value: t_pos_xsd): Integer;
  procedure Insert(index: Integer; value: t_pos_xsd);
  procedure Remove(value: t_pos_xsd);
end;
.....

```

上面的类图如果使用关系型数据库来实现的话，需要建立主从表，并将人员和账目之间的关联约束通过应用程序代码强制一些运行逻辑来完成，这时通常要通过补充详细的文档来描述需要强制的业务逻辑，如果没有详细的设计文档，实现代码时就很容易遗漏某些重要的商业规则，同时这些文档在整个的数据库开发的生命周期里面都需要人来手工地维护，难免会出现文档和模型不匹配的错误。而且文档的工作量比较大，而程序员数量又相对不足的话，程序员会觉得既要写代码又要写文档，无形中增加了很多工作量，难免会有抵触情绪，这些都会影响工作的效率。

对于这样的问题，ECO 则通过精确描述信息模型，无须详细规则描述文档可以将模型自动的转变为实现代码，商业规则在整个数据库开发生命周期内由 ECO 的类来维护，减少了文档的工作量和出错的可能。

(2) 建立企业逻辑动态模型（状态机）

在类图里定义好类后，可以设置类的状态机。^[36]以上面静态模型中的销售单（t_pos_xsd）类实体为例，设置销售单的状态机。在销售单（t_pos_xsd）类上点反键，Add 里添加它的 StateMachine 和触发器 Trigger（TrXJ 和 TrZF）。在状态机编辑器可以看到 Tool Palette 上提供了 State（状态），Initial（起始），Final（结束），Transition（转变）等元素。^[37]

在销售单的状态机里设置新建(create)、有效(effective)、作废(cancellation)三种状态以及触发器 TrXJ 和 TrZF。TrXJ 是用来触发销售单从新建状态到有效状态的转移，TrZF 是用来触发销售单从有效状态到作废状态的转移。并用 EAL (ECO Action Language) 来描述销售单状态转移时所执行的事件。

销售单新建时建立日志，增加个人积分，EAL 语句如下：

```
Let olog=t_sys_logs.Create in (olog.username:=UserName; olog.atime:=Time;
olog.aObject:='(销售单'+self.pjhbh); olog.aperate:='新建'; olog.notes:='增加积分+Score.asstring)
```

销售单作废时建立日志，EAL 语句如下：

```
let olog=t_sys_logs.Create in (olog.username:=UserName; olog.atime:=Time;
olog.aObject:='(销售单'+self.pjhbh); olog.aperate:='作废';olog.notes:='测试')
```

如图 2-2 所示。

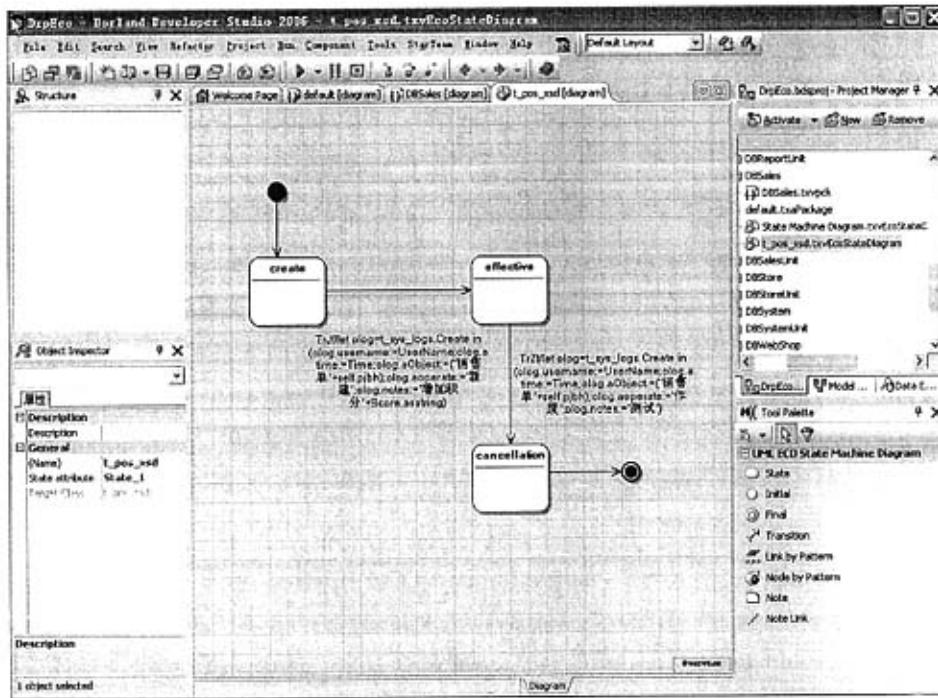


图 2-2 设置销售单 (t_pos_xsd) 类实体的状态机

当系统运行时，新建销售单会调用触发器 TrXJ，使销售单状态从新建转移到有效，同时产生日志并对会员增加积分。把销售单作废会调用触发器 TrZF，

使销售单状态从有效转移到作废，同时产生日志。

点中状态图里的 Transition，在 Object Inspector 里有 Trigger, Guard 和 Effect 属性。如图 2-3 所示。

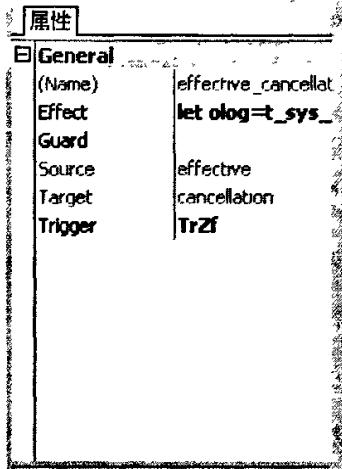


图 2-3 Transition 的属性

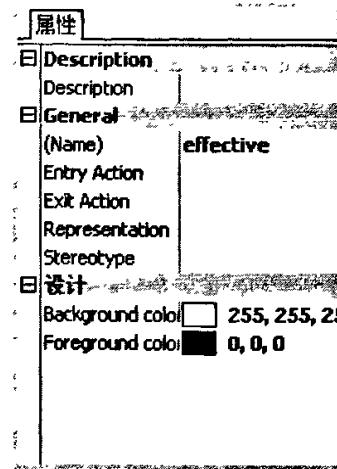


图 2-4 State 的属性

Trigger 里可选择已定义的触发器（TrZF）。

Guard 里定义所需的 OCL 语句，用来检验触发器（TrZF）能否触发，满足条件则发生触发，否则不能。

Effect 里定义所需的 EAL 语句，它是在触发器（TrZF）发生触发的同时所执行的。

点中状态图里的 State (effective)，在 Object Inspector 里有 Entry Action 和 Exit Action 属性。如图 2-4。

Entry Action 里定义所需的 ECO Action Language 语句，它是在状态转移到本状态时所执行的。

Exit Action 里定义所需的 ECO Action Language 语句，它是在状态转移出本状态时所执行的。

以下是触发器触发时，所执行的顺序：

- ①检查 Transition 的 Guard 语句，如果不满足，则整个 Action 被取消 (exception)；若满足，则继续。
- ②前一个状态的 Exit Action 被执行。
- ③触发器（TrZF）的 Effect 里的语句被执行。

④类实例的状态被设为下一个状态。

⑤下一个状态的 Entry Action 被执行。

如果下一个状态是 final state，则这个类实例被删除（在子状态图里除外）。

在状态机里，当类实体的状态改变时，也可以同时修改模型里其它任何类的属性。要实现这一点，必须在模型运行时得到其它类的实例，这就需要使用 OCLPsService。OCLPsService 提供了各种 functions，Execute 方法经常被用到。

若想实现销售单（t_pos_xsd）作废时入库单（t_pos_rkd）的属性 zje（总金额）增加 100，可以如下做：在销售单类 t_pos_xsd 里增加一个 Operation: Change，并把 t_pos_xsd 的状态 cancellation 的 Entry Action 设为：self.Change。

Change 方法的定义如下：

```
procedure t_pos_xsd.Change;
var
  myPs           :IOclPsService;
  oResultList    :IObjectList;
  oObj           :IObjectInstance;
  orkd           :t_pos_rkd;
begin
  myPs:=self.AsIObject.ServiceProvider.GetEcoService(typeof(IOCLPsService))
as IOclPsService;
  oResultList:=myPs.Execute('t_pos_rkd.allInstances->select(pjhb=20070322001)');
  for oObj in oResultList do
    begin
      orkd:=oObj.AsObject as t_pos_rkd;
      orkd.zje:=orkd.zje+100;
    end;
end;
```

（3）由对象模型生成对应的关系数据库模型

设计好企业逻辑静态模型和动态模型，现在我们可以由对象模型生成对应的关系数据库模型了。

ECO 企业逻辑模型必须由一个 EcoSpace 来执行，而为了提供 OR-Mapping 的功能，因此 ECO 提供了由 BDP（Borland Data Provider）技术连结各种数据来源的 PersistenceMapperBdp 组件，企业逻辑模型执行在 EcoSpace 中，而 EcoSpace 必须由 ECO 的 PersistentMapper 完成 OR-Mapping 的功能，此外 PersistentMapper 还提供了当数个 EcoSpace 由相同的 PersistentMapper 来使用 OR-Mapping 功能

时，PersistentMapper 也提供了在这些 EcoSpace 中对象和数据同步机制（Synchronization）的处理，以避免这些 EcoSpace 破坏数据。

但是 EcoSpace 要使用 PersistentMapper，则必须由 PersistentMapperProvider 来连结，PersistentProvider 提供了 EcoSpace 和 PersistentMapper 之间沟通的桥梁。

在上面所建工程 state 中，有两个文件非常重要，即 EcoPersistenceMapperProvider.pas 和 stateEcoSpace.pas。

EcoPersistenceMapperProvider.pas 是 PersistenceMapperProvider 的封装程序单元，可以在这个单元中使用 PersistentMapper 以及 BDP/ADO.NET 组件提供 OR-Mapping 的功能。

stateEcoSpace.pas 就是此 ECO ASP.NET 的 EcoSpace 程序单元。ECO 也已经在这个程序单元中建立了一个内定的 PersistentMapperSharer 以连结到 EcoPersistenceMapperProvider.pas 中包含的 PersistenceMapperProvider。

在 BDS 中打开 stateEcoSpace.pas，选择所建的模型到 Selected UML Packages。打开 EcoPersistenceMapperProvider.pas，拖放组件 BdpConnection，DefaultORMappingBuilder，PersistenceMapperBdp。设置相关属性，选择要连接的数据库（这里以 MS SQL Server 为例），点击 PersistenceMapperBdp，执行右键中的 SQL Server setup，才能让 PersistentMapperBdp 组件产生 OR-Mapping 对映数据。点选下方工具列中从左方起第 2 个 Generate Schema 图像以根据模型自动在 SQL Server 数据库中产生模型使用的数据库纲要，此时 BDS 会显示如下的对话框：

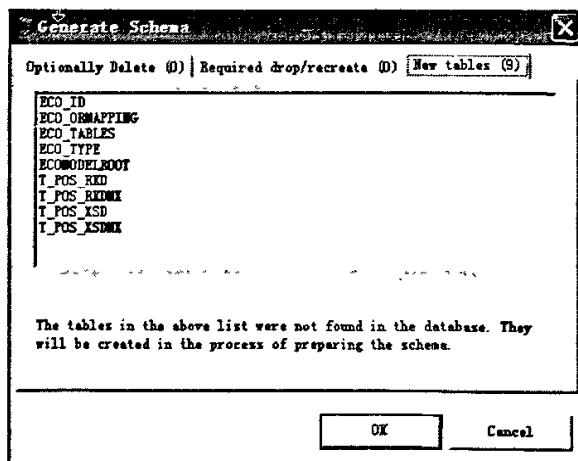


图 2-5 生成的数据库纲要

在 SQL Server 数据库中产生的数据库纲要如下：

名称 /	所有者	类型	创建日期
ECO_ID	dbo	用户	2007-3-15 16:29:49
ECO_ORMAPPING	dbo	用户	2007-3-15 16:29:49
ECO_TABLES	dbo	用户	2007-3-15 16:29:49
ECO_TYPE	dbo	用户	2007-3-15 16:29:49
ECOModelRoot	dbo	用户	2007-3-15 16:29:49
syscolumns	dbo	系统	2000-9-6 1:29:12
syscomments	dbo	系统	2000-9-6 1:29:12
sydepends	dbo	系统	2000-9-6 1:29:12
sysfilegroups	dbo	系统	2000-9-6 1:29:12
sysfiles	dbo	系统	2000-9-6 1:29:12
sysforeignkeys	dbo	系统	2000-9-6 1:29:12
sysfulltextcatalogs	dbo	系统	2000-9-6 1:29:12
sysfulltextnotify	dbo	系统	2000-9-6 1:29:12
sysindexes	dbo	系统	2000-9-6 1:29:12
sysindexkeys	dbo	系统	2000-9-6 1:29:12
sysmembers	dbo	系统	2000-9-6 1:29:12
sysobjects	dbo	系统	2000-9-6 1:29:12
syspermissions	dbo	系统	2000-9-6 1:29:12
sysproperties	dbo	系统	2000-9-6 1:29:12
sysprotects	dbo	系统	2000-9-6 1:29:12
sysreferences	dbo	系统	2000-9-6 1:29:12
sysatypes	dbo	系统	2000-9-6 1:29:12
sysusers	dbo	系统	2000-9-6 1:29:12
t_pos_xsd	dbo	用户	2007-3-15 16:29:49
t_pos_xsdmx	dbo	用户	2007-3-15 16:29:49
t_pos_rkd	dbo	用户	2007-3-15 16:29:49
t_pos_rkdmx	dbo	用户	2007-3-15 16:29:49
t_pos_xsdm	dbo	用户	2007-3-15 16:29:49

图 2-6 在 SQL Server 数据库中产生的数据库纲要

在数据库中，ECO 共生成了 9 个表，其中 ECO_ID，ECO_ORMAPPING，ECO_TABLES，ECO_TYPE，ECOModelRoot 这 5 个是 ECO 相关信息的表，而 t_pos_xsd，t_pos_xsdmx，t_pos_rkd，t_pos_rkdmx 这 4 个表是我们设计的 4 个类实体所对应的，表的字段就是我们设计的类实体的属性。在 ECO 框架里，对业务域对象的操作都是通过对象的属性或者方法来进行的。

工具列中从左方起第 1 个 Wrap Existing Database With Eco 图像的作用是，把已经存在的数据库纲要以逆向工程产生类模型。

在原型设计期间，我们可以使用 PersistenceMapperXml 组件替代 PersistenceMapperBdp 组件。PersistenceMapperXml 组件将使我们的程序可以使用 XML 文件来保存和读取对象，这是一个使用很方便的控件，特别是在快速原型设计期间。因为在原型设计期间，模型经常会被改动，而重新生成数据库表是很费时间的，而 XML 文件可以使我们非常快的变更我们的模型设计。当模型基本稳定后，可以去掉这个控件，转而切换为其他使用关系型数据库进行存储的可持续性控件，这样的开发方式可以使我们不需要改动整个程序就能很容易地改变数据持续层的存储策略。这就是“数据库平台无关设计”。

2.1.2 EcoSpace

ECO 应用程序在执行时期所有的 ECO 对象都是维持在 EcoSpace 中，应用程序也由 EcoSpace 存取 ECO 架框提供的服务，此外 EcoSpace 更负责执行企业逻辑模型。设计时创建的信息模型可以被看做元数据，就象数据库的表和字段结构一样的信息，而 EcoSpace 类是一个对象容器保存模型定义的类的运行时的实例。这些对象实例是对 UML 模型的一种运行时的表达，EcoSpace 是其他 ECO 组件的信息源，用来表达整个系统的业务域元素，可以理解为对象空间。通过对对象空间我们可以在运行时获得设计时元数据表达的对象的运行实例。

下面详细讨论一下 EcoSpace 的生命周期和对象池机制。

由于 EcoSpace 在执行时期控制了企业逻辑模型以及企业逻辑对象，因此 EcoSpace 的生命周期便深度的影响了 Eco ASP.NET 应用程序的执行行为。

在 ECO III 中提供了三种 EcoSpace 的生命周期模式，它们是 Dirty，Never 和 Always。

(1) EcoSpace 的 Dirty 模式

Dirty 模式是默认模式，如果此 EcoSpace 对象中包含任何异动过的对象，就把此 EcoSpace 对象快储在 session 状态中。这个模式允许应用程序在多个客户端请求中维护应用程序的状态（EcoSpace 的状态）。

在 Dirty 模式下，一般 Eco ASP.NET Web 窗体如果使用 EcoSpace 和企业逻辑对象来进行查询数据的动作而没有异动任何的企业逻辑对象的话，那么当客户端移转到其它的 Eco ASP.NET Web 窗体后，那么上一个 Eco ASP.NET Web 窗体使用的 EcoSpace 便会释放回 EcoSpace 对象池，第二个 Eco ASP.NET Web 窗体会从 EcoSpace 对象池取得另外一个 EcoSpace 对象池中管理的 EcoSpace 对象。

然而如果客户端异动了任何企业逻辑对象，那么包含这个企业逻辑对象的 EcoSpace 对象便会自动储存在 ASP.NET/Web Service 应用程序的 Session 对象中，从此以后所有 ASP.NET Web 窗体使用的 EcoSpace 对象便是这个储存在 Session 对象中的 EcoSpace 对象，一直到 ASP.NET/Web Service 应用程序结束后这个 EcoSpace 对象才会释放回 EcoSpace 对象池。

从上面的讨论中我们可以了解下面重要的观念：

在 Eco ASP.NET/Eco Web Service 应用程序中，每一个 ASP.NET Web 窗体使用的 EcoSpace 对象可能都是不一样的，因此一个 ASP.NET Web 窗体建立的 Eco 企业逻辑对象最好传递到其它的 ASP.NET Web 窗体，否则 Eco ASP.NET/Eco Web

Service 应用程序会显示“对象不是由这个 EcoSpace 建立”的错误。这是因为当我们把 Eco 企业逻辑对象传递到另外的 ASP.NET Web 窗体时，另外的 ASP.NET Web 窗体可能使用了不同的 EcoSpace 对象。

那么既然在 Dirty 模式下 Eco ASP.NET/Eco Web Service 应用程序中我们不能直接传递企业逻辑对象到其它的 ASP.NET Web 窗体，那么我们要如何传递企业逻辑对象呢？很简单，那就是传递代表企业逻辑对象的唯一 ID 值，如此一来其它的 ASP.NET Web 窗体便可以使用这个唯一对象 ID 值来取得它代表的企业逻辑对象，之后就可以继续使用这个企业逻辑对象了。

(2) EcoSpace 的 Never 模式

Never 模式特别适合使用来开发以查询为主的 ASP.NET 应用系统，在这种模式下由于每一个 ASP.NET Web 窗体一定都是使用从对象池中重新取出的 EcoSpace 对象，因此 ASP.NET Web 窗体绝不能在不同的 ASP.NET Web 窗体中传递 ECO 企业逻辑对象，开发人员一定要使用 ECO 对象 ID 来取得实际的 ECO 企业逻辑对象，再由企业逻辑模型中定义的关系来存取其它相关的 ECO 企业逻辑对象。

(3) EcoSpace 的 Always 模式

Always 模式则是相对 Never 模式的另一个极端，在这种模式下当 ECO ASP.NET 应用程序第 1 次存取了 EcoSpace 之后，这个 EcoSpace 对象会自动储存在 ASP.NET 应用程序的 Session 之中，随后的所有此 ASP.NET 应用程序包含的 ASP.NET Web 窗体都重复使用这个相同的 EcoSpace 对象。因此在这种模式下，开发人员是可以直接把企业逻辑对象传递到不同的 ASP.NET Web 窗体而没有任何的问题。

Always 模式虽然在使用上和程序设计上都是最简单的，但是由于 Always 模式把 EcoSpace 快储在 Session 之中一直到 ASP.NET 应用程序结束为止，因此 Always 模式会一直占据一些资源，这些资源包含了 EcoSpace 本身，EcoSpace 维护的企业逻辑对象，以及 EcoSpace 开启的数据库连结。

由于 Always 模式需要占用伺服端的资源，因此它的延展性就不如另外两种模式。

和上述 EcoSpace 生命周期相关的就是 EcoSpace 对象池机制了。只有开发人员先启动 EcoSpace 对象池机制，上述 3 种模式才能够运作。要启动 EcoSpace 对象池机制，开发人员可以打开 ECO ASP.NET 工程的 Web.config 文件，在其中

和 EcoSpace 对象池机制相关的就是 Borland.Eco.Web.MaxPool 和 Borland.Eco.Web.MaxAge 这两个设定值。设定 MaxPool 大于零的话就会自动启动 EcoSpace 对象池的机制，并且最多会在 EcoSpace 对象池中建立此设定值的 EcoSpace 对象，MaxAge 指在 EcoSpace 对象池中的 EcoSpace 对象最长会保留此设定值设定的秒数时间。

当开发人员启动了 EcoSpace 对象池机制之后应该会明显的感觉 ECO ASP.NET 应用程序的执行效率增加了许多。当然这两个设定值到底应该是多少需要根据开发人员使用的环境来决定，开发人员在完成 ECO ASP.NET 应用系统的开发之后，在部署之前应该试着调整这两个数值以便提供最好的执行效率以及最佳的服务器端资源使用率。

2.1.3 OCL

OCL (Object Constraint Language) 最早是由 IBM 公司提出的 UML 规范的一个子规范，用来描述对象的约束和相互之间的关联关系，该规范同 UML 标准一样由 OMG (Object Modeling Group) 组织负责制定和维护。^[38]之所以需要 OCL 查询语言是因为标准的 UML 模型对模型特性的描述主要是通过自然语言进行描述的，自然语言的特点就是存在着二义性，这在系统设计的时候还不是太大的问题，但要将 UML 模型转化为真实运行的程序，则要求消除这种模型表述的不确定性。而 OCL 语言则可以精确的描述模型的约束条件以及相互之间的关系，非常适合于模型驱动的开发。OCL 是一种对象查询语言，类似于 SQL 语言，区别就是 OCL 语句返回的是对象实例，而 SQL 返回的是数据库记录。

OCL 查询语言同标准的 SQL 的语言之间的对比如表 2-1：

表 2-1 OCL 查询语言同 SQL 语言对比

OCL 查询语言	SQL 查询语言
面向对象的查询语言，“面向对象的 SQL”，强类型的查询语言	面向静态数据集的查询语言，非面向对象的，弱类型的查询
由 OMG 制定的统一标准	有统一的 ANSI 标准，但大多数厂商对标准进行了扩展，导致事实上的标准不统一
数据库平台无关	数据库平台相关，不同平台的 SQL 语句用法不同
只能对数据进行查询	除了查询，可以通过 Insert, Update, Delete 对数据进行修改

OCL 由许多操作和函数组成，可分类如表 2-2。

表 2-2 OCL 的操作和函数组成

语言结构	说明和举例
对象访问	allInstances, allLoadedObjects, emptyList...
属性访问	navigating via .Attribute name e.g. self.sFirstName
算数操作符和表达式	Addition, Subtraction, Multiplication, Division... e.g. self.iID+5, Round(self.iPrice)...
逻辑操作符	>, <, =, >=, <=, <>, or, and not ... e.g. self.iID>=24
字符串函数	concat, length, pad, subString, toLower... e.g.. self.sFirstName+" "+self.sLastName e.g.. self.sFirstName.Length
日期/时间函数	inDateRange, strToDate, formatDate... allInstances->forAll(..), ->orderBy(...), ->first, ->last, ->count, ->select(..), ->union(..)... e.g.: allInstances->select(sFirstName>'AAA') allInstances->collect(employees->size))
类型分析和类型定制	oclType, oclIsTypeOf, oclAsType, allSubClasses...

2.1.4 EAL

EAL (ECO Action Language) 语言是从 OCL 加以延伸出来的语言，为什么有了 OCL 还需要 EAL 语言？这是因为 OCL 是一个没有任何副作用(side effect)的正规语言 (Formal Language)，这个意思简单的说就是 OCL 不会改变对象的状态，OCL 主要是使用在类图中定义对象的限制条件或是在类图中和程序代码中执行查询的工作。^[39]但是在 ECO 的状态机中却需要能够改变状态，因此需要修改或是改变对象的状态。所以 ECO 扩充了 OCL 成为 ECO Action Language 以便让 EAL 语言具有修改或是改变对象状态的能力。EAL 包含了 OCL 的所有特性和操作，此外它还增加了 assign-operation, list-manipulation operations, object creation/deletion and sequencing of statements。

2.1.5 ECO 组件组

ECO 也提供了一组.NET 组件，这些组件可以让开发人员使用来运用 ECO 提供的各种功能，例如使用 ECO 组件连结到 EcoSpace 以处理对象模型，使用 ECO 组件执行 OCL 查询，使用 ECO 组件把对象模型和后端的数据来源连结在一起以利用 ECO 的 OR Mapping 等功能，以及最重要的是这些 ECO 组件也提供和.NET 可视化连结在一起的能力，以便让 EcoSpace 中的对象能够自动的显示在.NET 的图形使用者接口组件之中。如图 2-7 所示。



图 2-7 ECO 组件组

2.2 ECO 的开发架构

2.2.1 使用 Winform 之单机/主从架构的 ECO 应用系统

ECO 最基本的架构就是开发单机/主从架构的应用程序，在这种架构中客户端的应用程序就是 Winform。ECO 的 Winform 应用程序在执行时期会建立一个称为 EcoSpace 的执行架框环境，EcoSpace 在执行时期提供了各种 ECO 的服务，更重要的是 EcoSpace 执行框架环境会包含开发人员设计的企业逻辑模型，并且根据这个企业逻辑模式来建立对象实例，根据企业逻辑模型来执行企业规则。

简单的说，EcoSpace 就在“执行”设计人员于设计时期设计的模型架构。因此，我们可以了解在这种架构中，客户端的 Winform 由 EcoSpace 来执行我们设计的企业逻辑模型，例如建立模型中的对象，执行模型中定义的企业规则等，并且由 EcoSpace 来存取 ECO 架框提供的服务，例如查询对象，存取对象，储存异动的对象，最后由 ECO 框架把异动过的对象自动由 ECO 提供的 OR-Mapping（Object Relational Mapping）的机制储存到后端的数据来源中。而客户端的 Winform 应用程序从头到尾只需要使用对象来完成工作，而无需面对复杂的数据存取以及对象和数据之间转换的工作。

对于单机和主从架构的 ECO 应用系统而言，这个架构的开发是最简单的，也非常适合初学 ECO 的开发人员学习的开发架构，但是如果以目前热门的程序来说，开发 ASP.NET 的 Web 应用程序似乎更受欢迎，下面来说明 ECO 在 Web 的开发架构。

2.2.2 使用 ASP.NET 之 Web 架构的 ECO 应用系统

ECO 提供了强大的 ASP.NET 开发能力，除了能够使用所有 ASP.NET 本来就提供的各种组件和服务之外，也能够使用所有 ECO 的功能和服务。在这种 ECO ASP.NET 架构中，ASP.NET 应用程序同样也由 EcoSpace 执行我们在设计时期设计的企业逻辑模型，然而由于 ASP.NET 应用程序会有许多客户端同时存取使用，因此 ASP.NET 应用程序需要和后端的数据来源连结时，我们使用 ECO 永续储存映像同步器（PMapper Synchronizer）来同步每一个客户端对于对象的异动，以保证后端数据来源之中数据的一致性。

此外在这种架构中，由于每一个 Web 客户端呼叫 ECO ASP.NET 应用程序时便需要建立 EcoSpace，因此 ECOIII 也提供了 EcoSpace 池的机制，以节省 ECO ASP.NET 应用程序使用的资源以及执行效率。

由于 ECO ASP.NET 应用架构提供了许多先进的功能，因此非常适合使用来开发大型，复杂的 Web 应用系统。

2.2.3 使用 ECO 套件架构的 ECO 应用系统

使用 ECO 套件架构的 ECO 应用系统和前两种架构不同的地方是，前两种架构中企业逻辑模型都是直接设计在 Winform 或是 ASP.NET 的项目之中，这样架构的问题是如此一来企业逻辑模型就只限定使用在这个项目中，如果其它的

项目也需要使用这个企业逻辑模型，那么就比较麻烦（仍然可以做到，只是需要先把企业逻辑模型输出，再汇入到其它的项目中）。

因此 ECO 框架提供了另外一种架构，那就是让开发人员把企业逻辑模型单独设计在一个 ECO 套件中，然后再把这个 ECO 套件加入到其它的 ECO Winform 或是 ECO ASP.NET 中使用，如此一来这个企业逻辑模型就可以在不同的 ECO 项目中重复使用。

ECO Winform 和 ECO ASP.NET 的 EcoSpace 执行了 ECO 套件中的企业逻辑模型，而且如果开发人员又回头修改 ECO 套件，那么仍然可以在 ECO Winform 和 ECO ASP.NET 项目中使用经过修改的企业逻辑模型，非常方便。

2.2.4 使用 ECO 套件之多层架构的 ECO 应用系统

最后一种 ECO 架构比较复杂，但是非常强大。ECO 允许开发人员开发.NET 环境下的多层应用系统，分为两部份。首先开发人员先开发一个 ECO 服务器，这个 ECO 服务器可以是 Web 应用程序或是 Winform 应用程序，也可以使用 ECO 套件。接着再开发客户端的 ECO Winform 应用程序或是 ECO ASP.NET 应用程序，而 ECO 服务器和客户端的 ECO 应用程序则是使用.NET Remoting 来沟通的。

由于这是一个多层的应用系统，因此当然也需要使用 ECO 永续储存映像同步器来同步协调客户端对于对象的异动。ECO 自动封装了一些.NET Remoting 复杂的程序代码。

下面解释一下数种不同的 ECO 应用程序型态。

第 1 种是 ECO Winform 应用程序，在这种应用程序中通常只会建立一个 EcoSpace 来执行企业逻辑模型，因此没有所谓同步的问题，因此在这种应用程序中 EcoSpace 是直接连结 PersistentMapper 来使用 OR-Mapping 的机制。

第 2 种架构是建立多个 EcoSpace 的 ECO ASP.NET 或是 ECO Winform 应用程序，在此应用程序中由于有多个 EcoSpace 需要共同使用一个相同的 PersistentMapper，因此首先我们需要使用 PersistentMapperSharer 来连结 PersistentMapperProvider，再由 PersistentMapperProvider 连结 PersistentMapper。

第 3 种是最复杂的 ECO 执行架构，由于 ECO 应用程序是在不同的执行程序或是机器中执行，因此客户端的 ECO 应用程序（ECO ASP.NET 或 ECO Winform）需要使用 PersistentMapperClient 组件由.NET Remoting 连结远程 ECO 服务器中的 PersistentMapperProvider 来共享 PersistentMapper。

2.3 本章小结

本章主要介绍了 ECO 的相关技术和 ECO 的开发架构。ECO 是一套基于 UML 模型驱动的面向对象的数据库开发框架，可以用来轻松地实现信息模型设计及基于信息模型的应用程序。ECO 是一个完整的框架，它是由许多高端的技术所结合而成，包括可视化设计接口、EcoSpace、OCL、EAL、ECO 组件组等等。

第3章 基于ECO的电子商务系统的设计

3.1 网上商城系统的需求分析

我国某著名酒厂集团地处山区交通不便、信息闭塞，因此网络营销尤为重要。网络营销的核心是以顾客为中心，围绕着新的4C理论，即Customer（顾客），Cost（成本），Convenience（便利），Communication（沟通），体现了现代营销的发展趋势，是利用科学技术促进营销工作的强有力的手段和方法。该酒厂要通过网络营销建立会员制俱乐部。会员制俱乐部有很多优点，可以给予厂商和消费者沟通的最直接的机会；可以吸引更多潜在客户，培养客户忠诚度，通过俱乐部营销把一般客户发展为重点客户或关键客户，根据“百分之二十的客户产生百分之八十的利润”这一著名的商界“二八”理论，因此俱乐部会员是企业的血脉。俱乐部营销也可以通过这些会员的良好口碑，通过个人传播，服务于品牌营造和企业形象塑造。^[40]

网络营销弥补了传统企业宣传方式的不足。首先，是受众群体庞大，针对性强。企业一旦在互联网上建立了宣传基地，其对象就不仅限于某一特定区域，而是面向全球的客户，而且还不存在时间上的限制。此外，企业信息的受众都是对信息本身感兴趣的群体，因此是有的放矢，针对性很强。其次，从传播方式来说，传统媒体形式下信息的受众没有选择的权利，不管于己有用没用都必须被动接受，换句话说就是信息的浪费率惊人，因此造成单位成本的上升。而在互联网背景下，由于网络宣传费用本身就很低，加上信息的命中率极高，所以其“性价比”极佳。^[41]

网络营销为企业降低了成本。在网上发布信息，代价较低，将产品直接向消费者推销，可缩短分销环节，发布的信息谁都可以自由地索取，可拓宽销售范围，这样可以节省促销费用，从而降低成本，使产品具有价格竞争力。前来看望的大多是对此类产品感兴趣的顾客，受众准确，避免了许多无用的信息传递，也可节省费用。还可根据订货情况来调整库存量，降低库存费用。

该酒厂网上商城系统，不是一个简单的网上商城软件，它是涉及多个系统，多个部门，多个营销环节，集网上商品订购、网上支付、财务结算、物流调度、

会员互动、积分管理，能与专卖店前台业务平台互联、与总部业务平台接口的多系统、多平台、多渠道的信息化商务体系。

在软件组成上，系统必须建立如下系列平台软件，才能保证网上商城系统的功能得以实现：

网上商城：终端消费者在网上订购该酒系列产品，客户服务中心进行订单审核后，由专卖店执行，为消费者提供一个快捷的网上商务平台。

商城管理平台：建立客户服务中心，客户服务中心可进行商品信息维护，发布促销方案，审核跟踪会员订单，执行售后服务，管理会员积分，进行会员评价等业务工作。

会员中心平台：为会员提供的操作平台，具有会员注册、会员资料管理、意见反馈（会员论坛）、会员消费记录查询、企业公告、会员促销广告、会员活动、会员积分查询、网上调查等系列功能。

专卖店业务前台接口：与专卖店前台业务系统接口，实现网上订单划转、会员消费记录、积分，专卖店可下载会员基本资料，才能实现会员异地购酒。

销售总部业务系统接口：会员到总部购酒，实现会员积分。

会员服务终端平台：为机场、宾馆酒店提供会员查询、服务管理。

货款结算平台：完成销售总部与专卖店之间的货款结算。

查询统计平台：快速统计商品订购情况、销售行情、调查结果、会员排名、积分排名、地区消费行情等各种市场行情信息，领导可通过因特网直接查询。

平台之间的关系如图 3-1 所示：

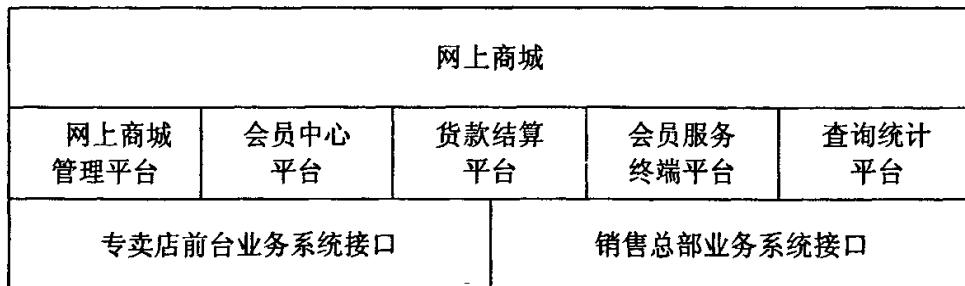


图 3-1 网上商城系统各平台之间的关系

3.2 主要功能分析和用例设计

网上商城系统包括电子商城子系统和内部业务管理子系统。由于会员在网上商城系统里提交的订单主要是全国各专卖店网点去执行，因此网上商城系统要和该酒厂的专卖店前台业务系统^[42]互联。电子商城子系统主要包括商品展台模块，购物车模块，会员中心模块等。内部业务管理子系统主要包括会员管理模块，订单管理模块，商品管理模块，货款结算模块等。网上商城系统主要有以下功能。

电子商城：

- (1) 商品展示：根据分类对各种商品进行查询、展示、说明。
- (2) 商品选购：选择中意的商品放入购物车，并填写订单内容，生成订单提交。
- (3) 购物指南：对商品选购、送货方式、付款方式等细节进行说明。
- (4) 服务中心：具有防伪测试、在线打假、专卖店查询、经销商查询等功能。
- (5) 走进神秘茅台：信息介绍，包括企业业绩、企业领导介绍、集团介绍、企业文化等。
- (6) 公众公告：向所有消费者发布最新通知。

会员管理：

- (1) 会员公告：向会员发布最新通知。
- (2) 会员活动：向会员展开各种活动，可包括有奖问答、抽奖、旅游等。
- (3) 短消息：向会员发布友情提示。
- (4) 订单管理：包括建立订单，订单跟踪。
- (5) 消费记录：记录会员网上购买详情。
- (6) 积分管理：包括积分查询、积分派送、积分兑奖等。
- (7) 网上调查：包括商品满意度调查、服务问卷调查等。
- (8) 客户留言：与会员进行一对一的信息交流。

网上商城管理：

- (1) 商品管理：商品的添加、编辑、删除。
- (2) 订单管理：订单的查询、审核、明细浏览、打印。
- (3) 会员管理：会员的添加、编辑、查询、分类统计、纪念日管理（纪念

日添加、纪念日编辑、纪念日提醒)。

- (4) 信息管理：活动的添加、编辑、查询；客户留言的查询、回复；通知的添加、编辑、查询；问卷的添加、编辑、查询。
- (5) 会员积分管理：积分规则的新建、编辑、列表，积分政策编辑，积分榜查询，积分兑换处理，会员升级。
- (6) 广告管理：广告编辑、列表。
- (7) 礼品管理：礼品的添加、编辑、查询。

货款结算：

- (1) 网上支付货款查询。
- (2) 专卖店货款结算制表。
- (3) 专卖店货款结算。
- (4) 结算明细浏览。
- (5) 专卖店货款发放处理。
- (6) 货款分类账和明细账浏览、查询。
- (7) 专卖店货款总账和明细账浏览、查询。

下面说明网上商城系统主要用例。

3.2.1 电子商城用例

该模块主要相关对象是普通消费者和会员。主要功能如下：

- (1) 商品展示：进行商品展示。
- (2) 会员注册登录：会员注册分为个人注册和企业注册。
- (3) 公众公告：展示公告信息。
- (4) 购物指南：各种与购物相关的帮助信息。
- (5) 购物车管理：购物车列表，选择订购方式，配送方式，提交订单。
- (6) 营销广告：各种广告展示，分布在每个页面。

此业务模块的用例图可表示为图 3-2。

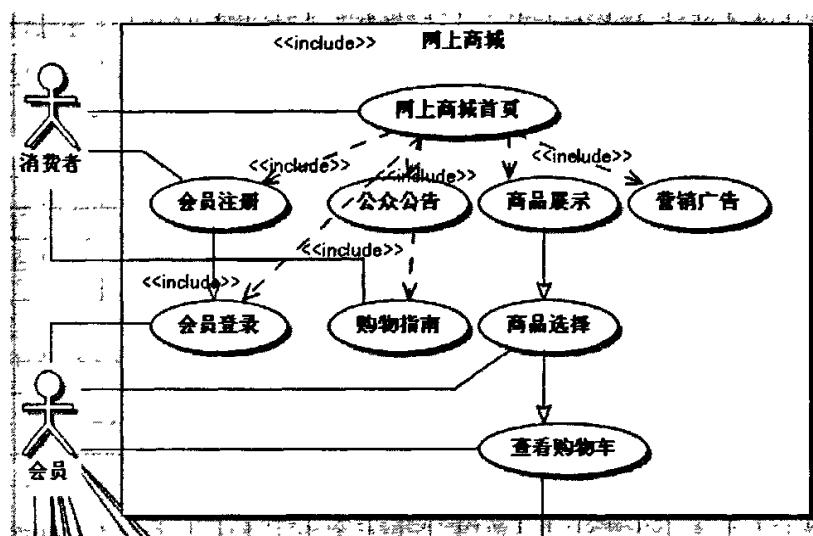


图 3-2 网上商城用例图

3.2.2 会员中心用例

该模块主要相关对象是会员。主要功能如下：

- (1) 积分政策：显示会员积分规则。
- (2) 会员权益：显示会员的权益。
- (3) 会员活动：发布会员促销活动。
- (4) 订单管理：可查看会员自己的订单等。
- (5) 消费记录：可查看会员自己的消费记录。
- (6) 积分榜：所有会员的积分排行榜。
- (7) 积分查询：可查看会员自己的积分。
- (8) 兑奖申请：可申请会员自己积分以内的奖品。
- (9) 会员公告：显示最近关于会员的一些信息。
- (10) 会员论坛：会员的论坛板块。
- (11) 网上调查：问卷调查和统计结果。

此业务模块的用例图可表示为图 3-3。

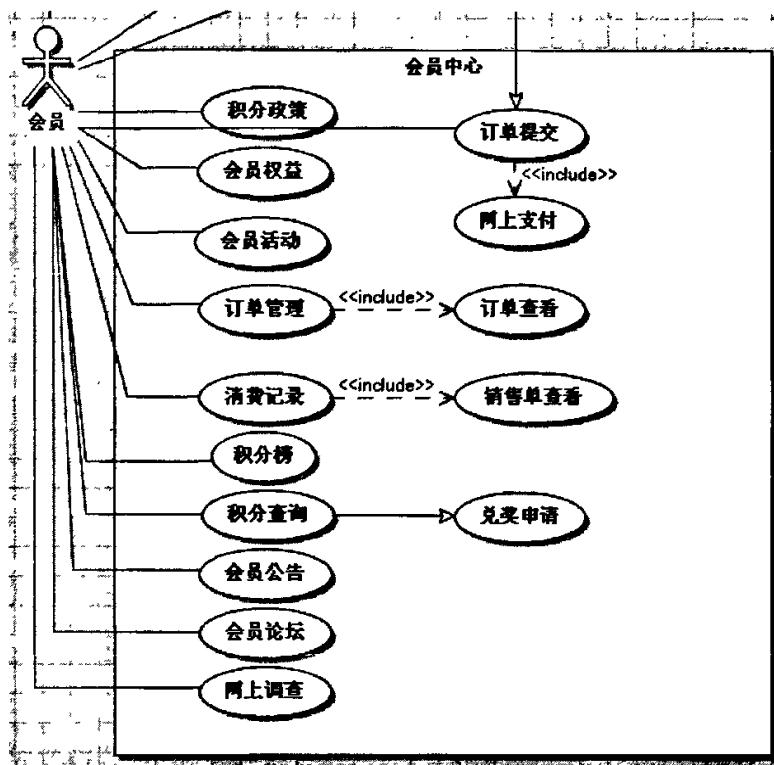


图 3-3 会员中心用例图

3.2.3 网上商城管理用例

该模块主要相关对象是商城管理员。主要功能如下：

- (1) 会员管理：会员信息查询，会员添加，会员信息编辑，纪念日添加、编辑，纪念日提醒。
- (2) 订单管理：订单列表，订单审核，提交打印。
- (3) 商品管理：商品信息查询，商品添加，商品发布，价格设定。
- (4) 积分标准：积分标准查询、新建与修改，设定会员升级标准，导出 xml，打印。
- (5) 广告发布：广告编辑，发布。
- (6) 活动管理：活动查询、新建与修改，导出 xml，打印。
- (7) 问卷管理：问卷查询、添加和查看，题目的增加、修改与删除。
- (8) 兑奖管理：兑现查询，兑现处理。
- (9) 赠品管理：赠品目录，赠品的添加、修改。

(10) 意见回复：留言查询，留言回复。

此业务模块的用例图可表示为图 3-4。

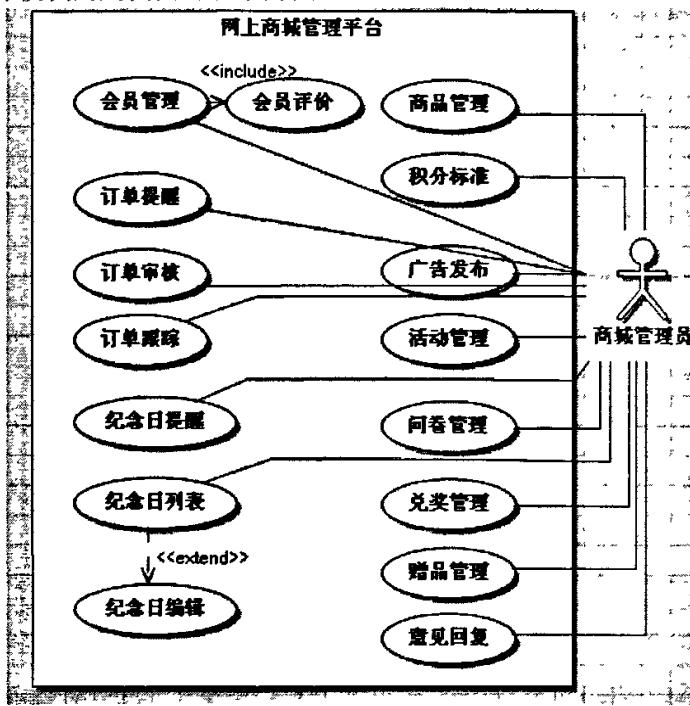


图 3-4 网上商城管理平台用例图

3.2.4 贷款管理用例

该模块主要相关对象是会计。主要功能如下：

- 1) 订单货款审核：审核订单的货款是否到账，是否足够。
- 2) 专卖店货款总账：专卖店执行的订单的总账。
- 3) 专卖店货款明细账：专卖店执行的订单的明细账。
- 4) 货款结算管理：货款结算。
- 5) 服务费用结算管理：服务费用结算。
- 6) 服务费用分类账：专卖店执行订单的相关服务费用分类账。
- 7) 服务费用明细查询：专卖店执行订单的相关服务费用明细查询。

此业务模块的用例图可表示为图 3-5。

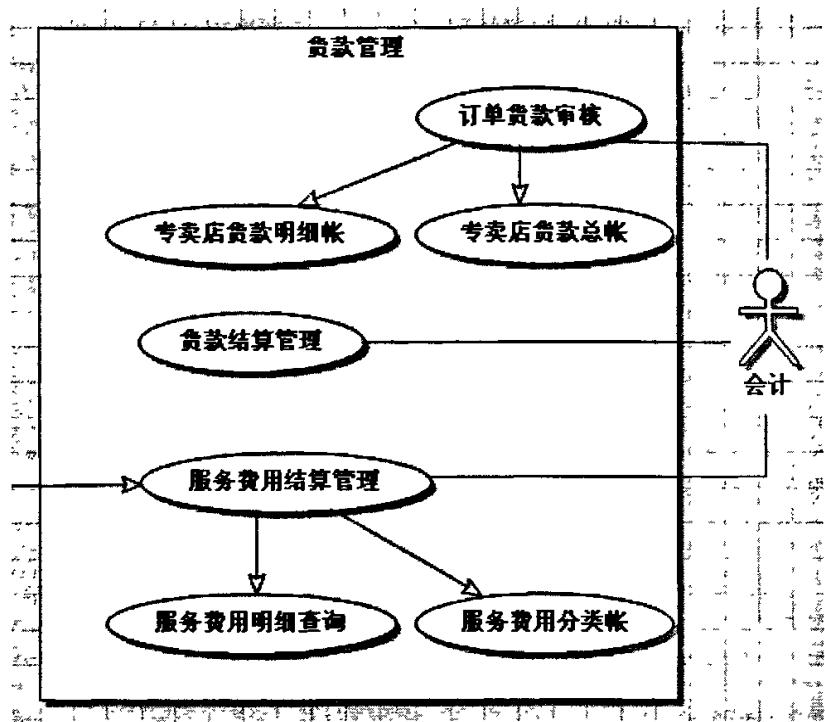


图 3-5 贷款管理用例图

3.3 主要业务流程

3.3.1 网上商城业务流程

该酒厂集团目前在全国建立了近 700 个专卖店，全国所有省区首府，绝大多数地级市已建立网络，专卖店网络已成形，配送覆盖面广。因此，会员在网上商城系统里选购商品，提交的订单主要是由该酒厂全国各专卖店网点去执行的。

消费者在网上商城选购商品放入购物车，提交时，若不是会员则要注册成为会员，若是会员则登录，生成订单。若是小额订单则系统自动批准，若为大额订单则需等待总部审批。小额订单自动批准后会员选择各大银行的网上银行进行支付。网上商城业务活动图如图 3-6 所示。

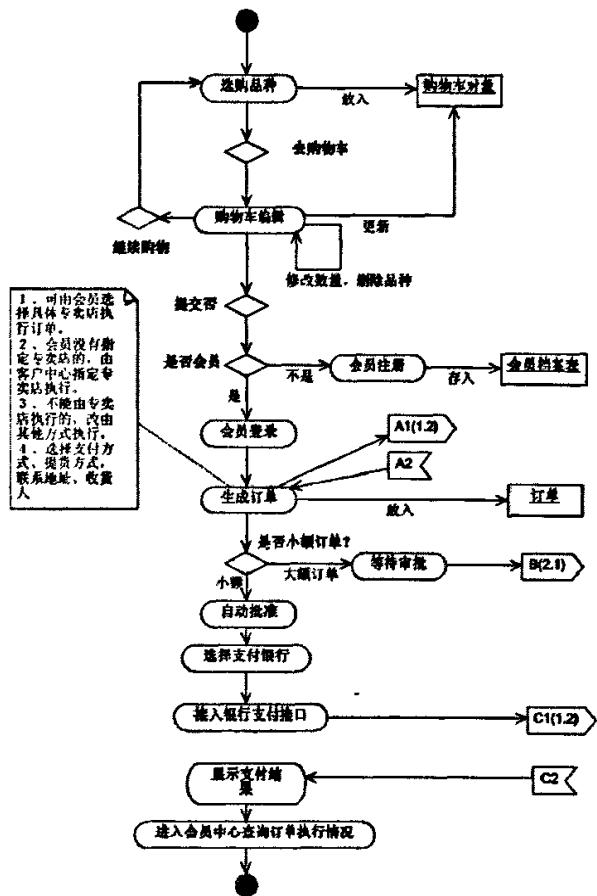


图 3-6 网上商城业务活动图

大额订单等待茅台总部业务科审批，业务科会查询会员购货历史作为判断订单可信度的部分依据，再查询该订单所选专卖店网点的库存是否足够，再向领导汇报，不管是否批准都发邮件通知会员。

会员可以查询自己的大额订单，若批准了，则选择多种方式支付，总部财务科进行货款审核后，客户中心就选择执行网点去执行该订单。

大额订单的审批流程和会员中心大额订单查询处理流程如图 3-7 所示。

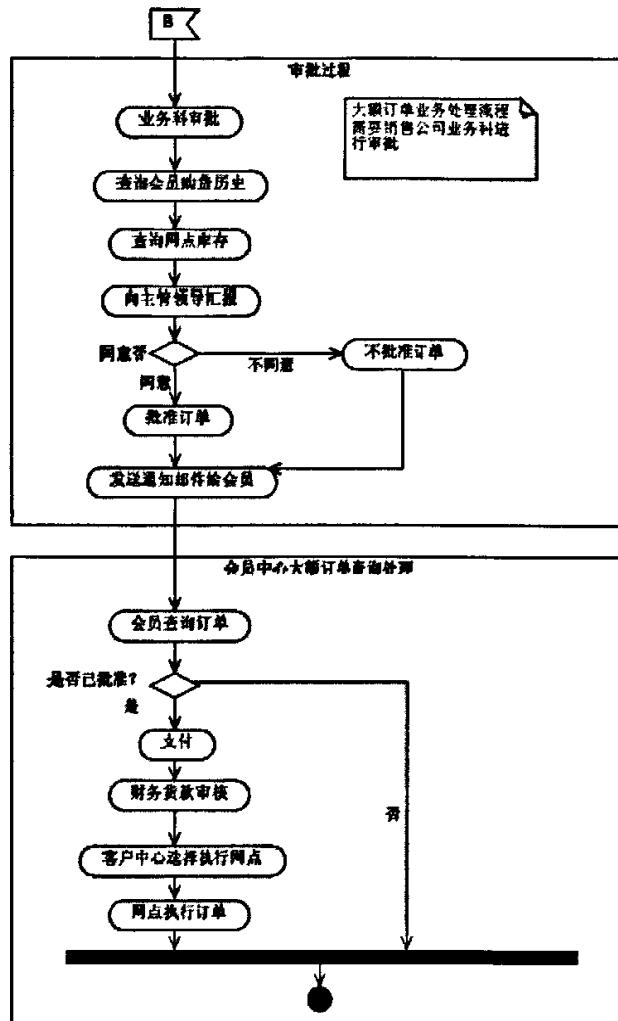


图 3-7 审批流程和会员中心大额订单查询处理活动图

3.3.2 专卖店订单处理业务流程

各专卖店管理员用本专卖店账户登录到专卖店前台业务系统，订单列表会自动刷新，调出新订单并查证它的真实性，若为虚假订单则不执行，若为真订单，则查证本专卖店能否执行。若无力执行，则退回总部，能执行则备货执行。

专卖店订单处理业务活动图如图 3-8 所示。

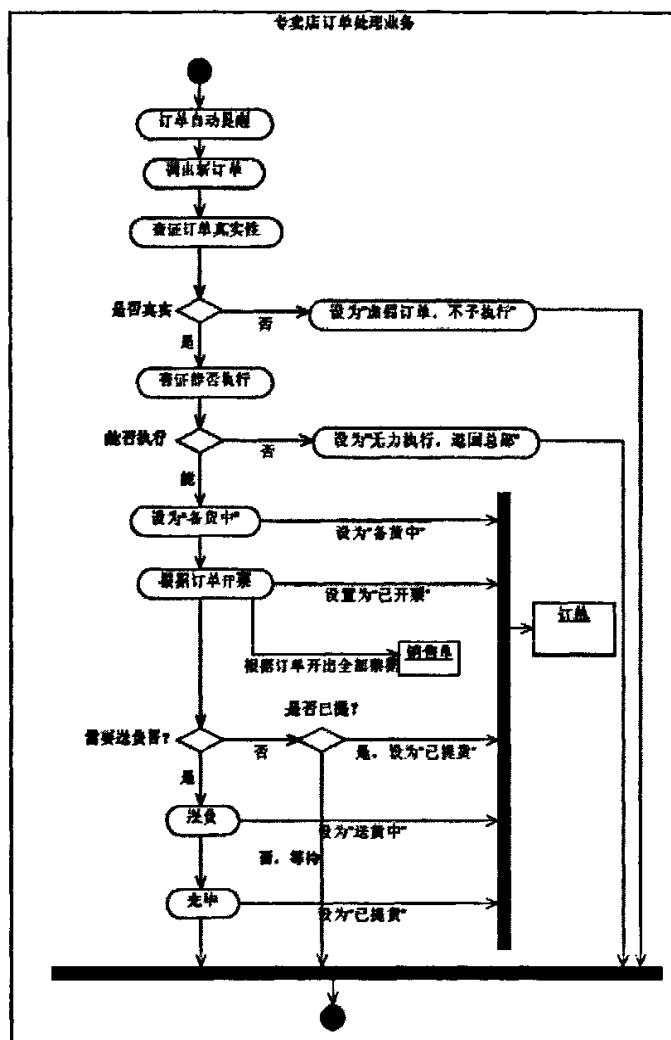


图 3-8 专卖店订单处理业务活动图

3.3.3 订单货款审核及结算业务流程

审核订单，银行会给茅台回单，进行回单查证。到了结算期，出纳根据未结算订单生成结算表，财务核查订单明细账是否与结算表相符。若有错误，查找原因，更正货款结算表；若无错误，则进行结算处理。

订单货款审核及结算业务活动图如图 3-9 所示。

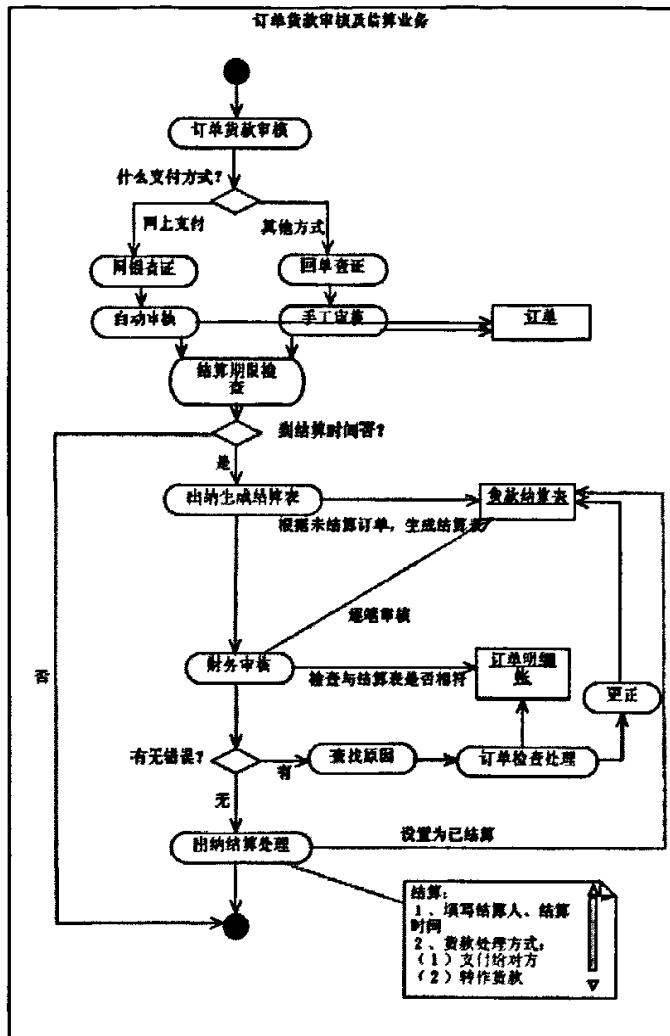


图 3-9 订单货款审核及结算业务活动图

3.3.4 总部订单跟踪业务流程

总部调出新订单，已支付订单不用查证，未支付订单需打电话给客户查证。设置订单状态为“查证真实”或“查证虚假”。再审核货款是否支付，订单是否开票，给客户打电话证实是否提货或送货。

总部订单跟踪业务活动图如图 3-10 所示。

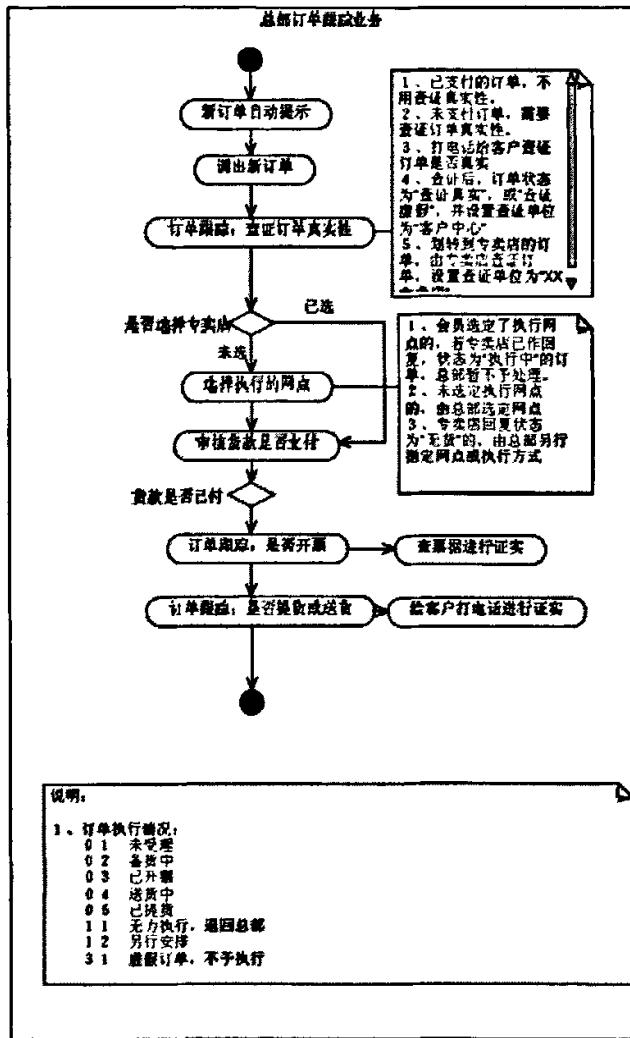


图 3-10 总部订单跟踪业务活动图

3.4 数据库模型

根据对系统的分析，利用 ECO 进行 UML 建模，类图包括七个包，分别为 DBAnalysis（分析包），DBMember（会员中心包），DBReport（统计报告包），DBSales（销售包），DBStore（库存包），DBSystem（系统管理包），DBWebShop（网上商城包）。在类图里设计相关类，并设置类与类之间的关联。关联一般有三种类型：一对一，一对多，多对多。^[43]

网上商城类图如图 3-11 和图 3-12 所示。



图 3-11 网上商城类图一

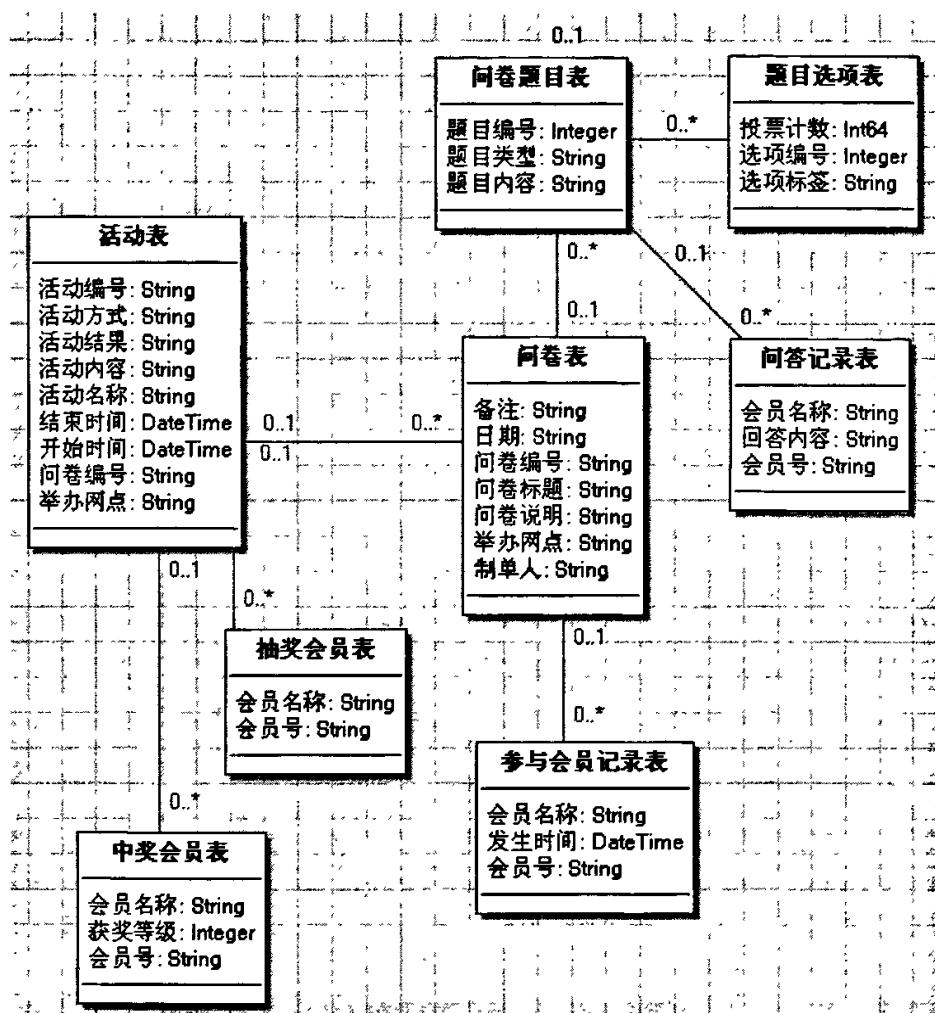


图 3-12 网上商城类图二

会员中心类图如图 3-13 和图 3-14 所示。



图 3-13 会员中心类图一



图 3-14 会员中心类图二

建立了企业逻辑模型，我们就可以用 ECO 直接由企业逻辑模型生成数据库，实现 OR Mapping。

3.5 本章小结

本章根据我国某著名酒厂电子商务系统的特点，对该系统进行了详细的设计。对其主要用例和主要业务流程进行了分析与设计。该酒厂的电子商务系统

与其它企业的电子商务系统不同的是：由于该酒厂建立以总部为中心，片区为分支，专卖店为终端的三级营销网络，在全国已建立 700 多家专卖店，覆盖面广，所以会员在网上商城系统里订购商品提交的订单主要是由全国各专卖店去执行的。到了结算期，总部与各专卖店进行货款结算。

分析了该系统后，又对系统进行了建模，用 ECO 把所建的类图自动生成了数据库，形成 OR-Mapping。基于 ECO 的数据库开发革命性的一点就是允许我们直接把基于 UML 的类图映射为关系型数据库的存储，而无须手工的通过代码进行转换。

第4章 基于ECO的电子商务系统的实现

4.1 系统的开发环境和部署环境

网上商城系统是基于.NET 平台开发的，集成开发环境是 Borland Development Studio 2006，Web 服务器是 IIS (Internet Information Services)，数据库服务器是 MS SQL Server，操作系统是 Windows Server 2003 企业中文版。

网上商城系统部署图如图 4-1 所示。

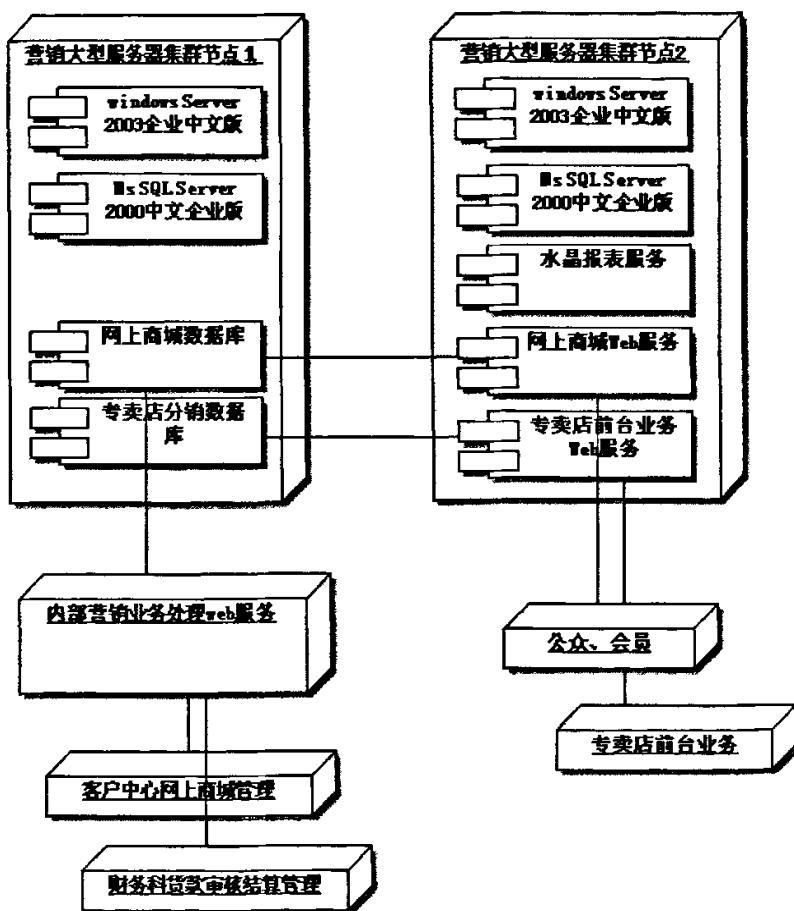


图 4-1 网上商城系统部署图

服务器采用 IBM X460 大型机架构的服务器，双机集群，保证系统可靠性。
网上商城系统硬件结构图如图 4-2 所示。

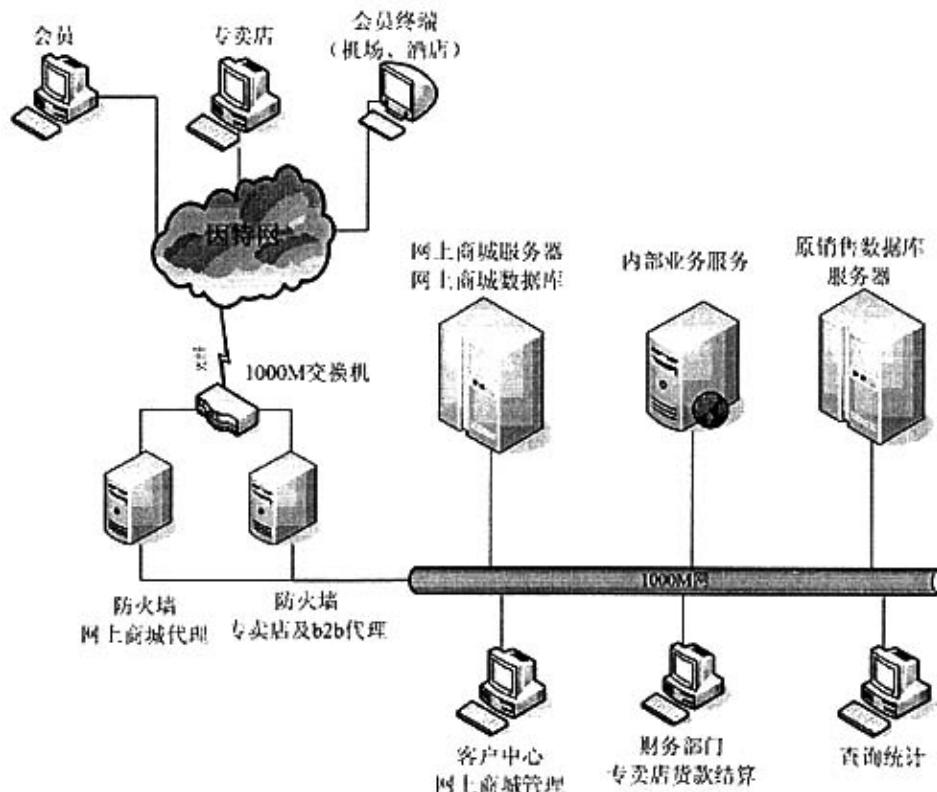


图 4-2 网上商城系统硬件结构图

4.2 ECO 的实现

4.1.1 订阅机制的实现

订阅机制（Subscription Mechanism）是 ECO 架构的基础核心之一。在对象之间的关系上，ECO 需要建立一种互通互信的“订阅”机制，比普通的虚拟环境更为高级，处理诸如级联删除、对象的连锁通知等功能。

例如要实现后台管理人员所属的多种角色在表格里对应显示出来，我们在模型里实现操作人员的派生属性 `jsmcStr`，代码如下：

```

function t_xtgl_czry.jsmcStrDeriveAndSubscribe(reevaluateSubscriber:
Borland.Eco.Subscription.ISubscriber; resubscribeSubscriber:
Borland.Eco.Subscription.ISubscriber): System.object;
var
  s, s1                      : string;
  i                           : Integer;
  o                           : t_xtgl_czryjs;
  oczjs                       : t_xtgl_czjsb;
begin
  s := "";
  if Assigned(self.t_xtgl_czryjss) then
    begin
      if self.t_xtgl_czryjss.Count > 0 then
        for i := 0 to self.t_xtgl_czryjss.Count - 1 do
          begin
            o := t_xtgl_czryjs(self.t_xtgl_czryjss[i]);
            oczjs := t_xtgl_czjsb(o.t_xtgl_czjsb);
            oczjs.AsIObject.Properties['jsmc'].SubscribeToValue(reevaluateSubscriber);
            s1 := oczjs.jsmc.Trim;
            if i = 0 then
              s := s1
            else
              s := s + ',' + s1;
          end;
    end;
  Result := s;
end;

```

这里实现的派生属性类似于计算字段，但它是非持久化，不占用数据库资源，并且它是定义在模型里，可复用。

4.1.2 状态机的实现

建立相关类的状态机。在销售单、入库单、货款核算表、货款结算表等类上建立状态机，系统运行时，当对这些类实例进行操作并且它们的状态发生转移时，可实现日志的生成等功能。以销售单为例，销售单的状态机如图 4-3 所示。

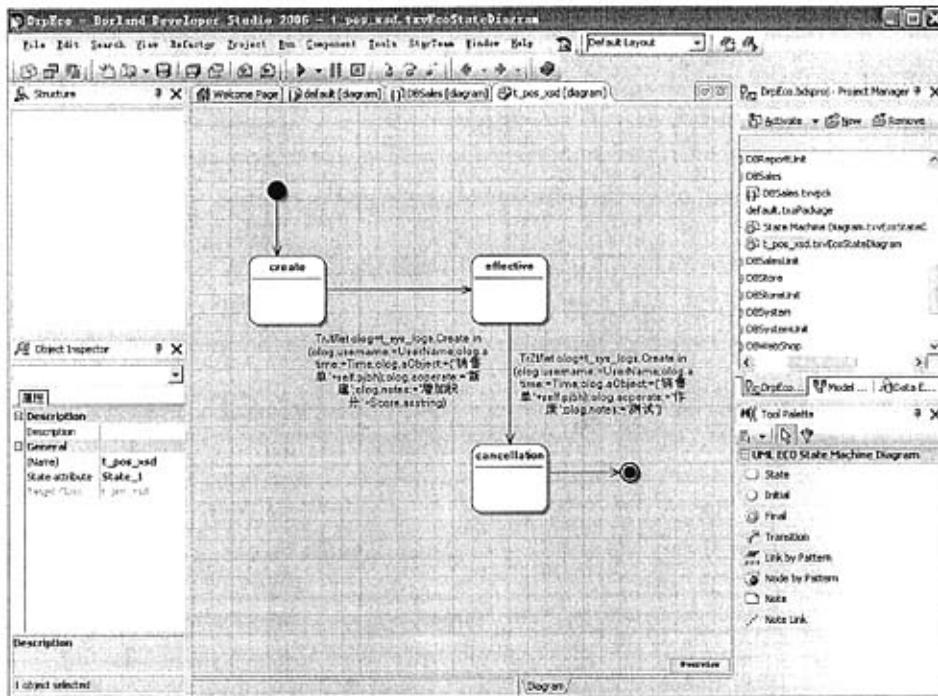


图 4-3 销售单的状态机

在销售单的状态机里设置新建(create)、有效(effective)、作废(cancellation)三种状态以及触发器 TrXJ 和 TrZF。TrXJ 是用来触发销售单从新建状态到有效状态的转移，TrZF 是用来触发销售单从有效状态到作废状态的转移。并用 EAL (ECO Action Language) 来描述销售单状态转移时所执行的事件。

销售单新建时建立日志，增加个人积分，EAL 语句如下：

```
Let olog=t_sys_logs.Create in (olog.username:=UserName; olog.atime:=Time;
olog.aObject:='销售单'+self.pjhb); olog.aoperate:='新建'; olog.notes:='增加积分'+Score.asstring)
```

销售单作废时建立日志，EAL 语句如下：

```
let olog=t_sys_logs.Create in (olog.username:=UserName; olog.atime:=Time;
olog.aObject:='销售单'+self.pjhb); olog.aoperate:='作废';olog.notes:='测试')
```

在模型中实现设置销售单状态标志的方法，代码如下：

```
procedure t_pos_xs.set_ztbz(Value: string);
var
    name :string;
```

```

time          :DateTime;
score         :Integer;
name2         :string;
time2         :DateTime;

begin
  Self.eco_Content.set_MemberByIndex(Eco_LoopbackIndices.ztbz,
(System.Object(Value)));
  if Self.ztbz='0' then
    begin
      name:=mczry.name;
      time:=DateTime.Now;
      score:=10;
      Self.TrXJ(name,time,score);
    end;

  if Self.ztbz='1' then
    begin
      name2:=oczry.name;
      time2:=DateTime.Now;
      self.TrZF(name2,time2);
    end;
end;

```

当销售单的状态标志为 0 时，调用触发器 TrXJ，使销售单状态从新建转移到有效，同时产生日志并对会员增加积分；当销售单的状态标志为 1 时，调用触发器 TrZF，使销售单状态从有效转移到作废，同时产生日志。

状态机是定义在模型里，因此可复用。由于在状态机里用的是 OCL 语句和 EAL 语句，因此可实现数据库平台无关性。

4.3 系统的实现

网上商城系统主要包括电子商城子系统和内部业务管理子系统，下面分别进行说明。

4.3.1 电子商城子系统的实现

电子商城子系统里，根据分类对各种商品进行展示、说明，普通用户可以

注册成为会员，选中商品放入购物车，并填写订单内容，生成订单提交。进入会员中心，可以查看自己的订单，消费记录。会员有足够的积分后还可以申请兑换奖品。

电子商城子系统主要功能模块如图 4-4 所示：

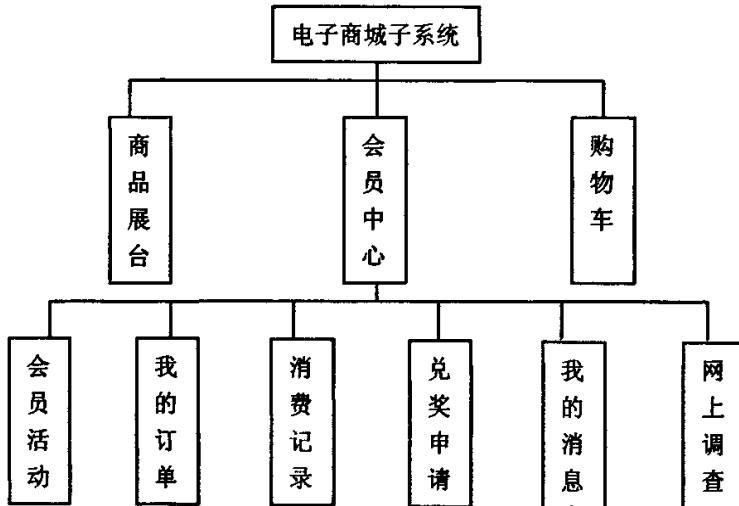


图 4-4 电子商城子系统主要功能模块

以购物车、我的订单和兑奖申请为例进行说明。

(1) 购物车

购物车要能实现添加物品到购物车、从购物车中删除物品、修改购物车中物品的数量和计算总金额等功能。购物车的部分代码如下：

```

procedure TWu_Cart.DeleteItems; {功能：删除品种}
var
  i:integer;
  mCB:CheckBox;
  mCodes:TStringList;
begin
  FCart := TCart(Session['Cart']);
  mCodes:=TStringList.Create;
  for i := 0 to fcart.Count - 1 do
  begin
    mCB := CheckBox(FindControl('CartCheckBox' + IntToStr(i)));
    if mCB.Checked then
      mCodes.Add(FCart.Code(i));
  end;
  FCart.Items := mCodes;
end;
  
```

```

end;

if mcodes.Count > 0 then
for i := 0 to mCodes.Count - 1 do
begin
  self.DeleteCart(mCodes[i]);
end
else
begin
  MsgLabel.Visible := true;
  MsgLabel.Text := '请选择要删除的商品！';
end;
FCart.CalcMoney;
mCodes.Free;
end;

```

(2) 我的订单

会员选择好商品放入购物车后，就可以提交生成购货订单了。大额订单需要等待总部审批，小额订单则自动通过。生成订单的部分代码如下：

```

fUndoService:=EcoSpace.GetEcoService(Typeof(IUndoService))asUndoService
;
fUndoService.StartTransaction;                                //开始事务
try
begin
  FCart:=Tcart(Session['Cart']);
  lecoghdd:= t_pos_ghdd.Create(ECOSpace);
  with lecoghdd do
  begin
    t_khgl_khda:= lecoKhda;
    khbh := fhybh;
    GetNewCode;
    ddh := fddh;
    dhdw := lecoKhda.khmc;
    ztbz:='1';
    .....
  end;
  for i := 0 to FCart.CartItems.Count-1 do
  begin
    lecoghddmx:= t_pos_ghddmx.Create(ECOSpace);
    lecoghddmx.t_pos_ghdd := lecoghdd;
  end;
end;

```

```

lecoghhdmx.spbh:= FCart.Code(i);
FSQL := 'select * from t_shop_spml where spbh =' +
QStr(lecoghhdmx.spbh);
FDT := DBGetDataSet(FSQL);
fpybm := DBFieldValue(FDT, 'pybm').Trim;
.....
lecoghhdmx.jf := FCart.Point(i);
if session['yxwdbh'].ToString<>" then
    lecoghhdmx.yxwdbh := session['yxwdbh'].ToString
else
    lecoghhdmx.yxwdbh:='001000000001';
end;
end;

fUndoService.CommitTransaction;//提交事务
UpdateDataBase;
session['Cart']:= nil;
except
    fUndoService.RollbackTransaction;//回滚事务
    raise;//显示出错信息
end;

```

会员进入会员中心，可以在我的订单里查看自己所有订单记录，以及订单的状态和执行情况等。

(3) 兑奖申请

会员通过购买商品可以获得积分，积累到一定积分后会员可以根据积分的多少申请兑换不同等级的奖品。兑奖申请的部分代码如下：

```

fUndoService :=EcoSpace.GetService(Typeof(IUndoService)) as IUndoService;
//事务控制
fUndoService.StartTransaction;
try
    //在 t_shop_djsq 里增加一条记录
    oDjsq:=t_shop_djsq.Create(EcoSpace);
    oDjsq.t_khgl_khda:=ohy;
//设置关联
    oDjsq.zxqk:='2';
    oDjsq.rq:=strtodatetime(FormatDateTime('yyyy-mm-dd',date));
    oDjsq.yxwdbh:='001000000001';

```

```

//生成编号
ftemp:=session['hybh'].ToString.Trim+FormatDateTime('yyyy-mm-dd',date).Substring(2,2)+'0001';
    ohDjsq.Expression:='t_shop_djsq.allInstances->select(bh='+qstr(ftemp)+')';
    ohDjsq.Execute;
    if ohDjsq.Element.GetAsCollection.Count=0 then
        oDjsq.bh:=ftemp
    else
        begin

ohDjsq.Expression:='t_khgl_khda.allInstances->select(khbh='+qstr(session['hybh'].ToString.Trim)+').t_shop_djsqs->orderDescending(bh)';
    ohDjsq.Execute;
    oDjsqTem:=t_shop_djsq(ohDjsq.Element.GetAsCollection[0].AsObject);

ftemp:=session['hybh'].ToString.Trim+FormatDateTime('yyyy-mm-dd',date).Substring(2,2)+RightStr('0000'+inttostr(strtoint(oDjsqTem.bh.Substring(23,4))+1)),4);
    oDjsq.bh:=ftemp;
end;

//在 t_shop_djsqmx 里增加相应的记录
for i := 0 to fGiftCart.Count - 1 do
begin
    flpbh:=fGiftCart.Code(i);
    flpsl:=strtoint(floattosrt(fGiftCart.Number(i)));
    ohGift.Expression:='t_shop_gift.allInstances->select(bh='+qstr(flpbh)+')';
    ohGift.Execute;
    oGift:=t_shop_gift(ohGift.Element.GetAsCollection[0].AsObject);
    oDjsqmx:=t_shop_djsqmx.Create(EcoSpace);
    oDjsqmx.t_shop_djsq:=oDjsq;
//设置关联
    oDjsqmx.t_shop_gift:=oGift;
//设置关联
    oDjsqmx.sl:=flpsl;
end;
//减少该会员的当前积分
ohy.Score:=ohy.Score-fGiftCart.TotMoney;
fUndoService.CommitTransaction;
UpdateDatabase;
.....

```

```

except
    fUndoService.RollbackTransaction;
    raise;
end;
end

```

4.3.2 内部业务管理子系统的实现

在内部业务管理子系统里，能对商品和信息进行发布和修改等，对会员信息进行管理，对订单进行审核等处理，对货款进行审核和结算等处理，对各专卖店进行监管等。

内部业务管理子系统主要功能模块如图 4-5 所示：

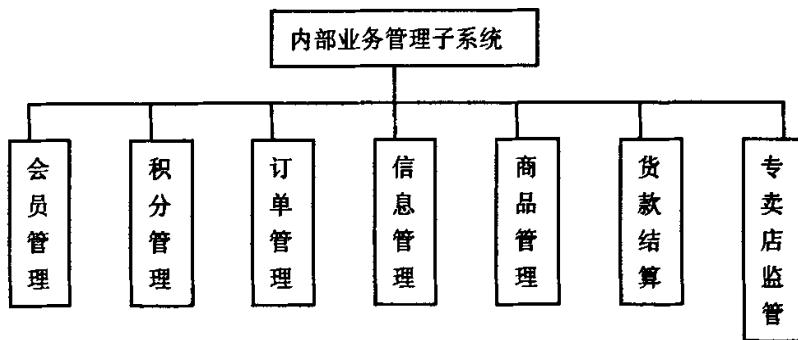


图 4-5 内部业务管理子系统主要功能模块

以订单管理和货款结算为例进行说明。

(1) 订单管理

对订单的审核主要是审核订单是否有效，是否真实，并且查看执行网点的库存量是否足够等。若订单审核通过，就设置订单状态为批准。这时会员就可以去网上银行进行支付，支付完后银行会给总部回单，财务部门对该订单进行货款审核，若货款已付清，则设置状态为已付款。订单审核的部分代码如下：

```

fUndoService := EcoSpace.GetEcoService(Typeof(IUndoService)) as IUndoService;
//事务控制对象
fUndoService.StartTransaction;                                //开始事务
try
    fUndoService.CommitTransaction;

```

```

//提交事务
order.pzbz:=session[fPageName+'pz'].ToString;
order.yxwdbh:=Label_yxwdbh.Text;
order.sprybm:=fusername;
order.spsj:=formatdatetime('yyyy-mm-dd',datetime.Now);
order.ztbz:=session[fPageName+'ddzt'].ToString;
order.bz:=Label_bz.Text;
UpdateDataBase;
btn_Commit.Enabled := False;
except
  fUndoService.RollbackTransaction;
  //回滚事务
  raise;
  //显示出错信息
end;
end
else
begin
  lbl_message.Visible:=True;
  lbl_message.Text:='货款已结算，不能修改！';
end;

```

订单管理页面如图 4-6 所示。

订单号	会员名称	执行网点	金额	日期	支付信息	订单状态	审批情况	货款情况	执行	订单处理
					银行名称	订单编号	支付日期			
0609130805070005	黄姐	专卖店系统测试户	¥16,040.00	2007-01-03	无			待审核	执行	未收
060913080507010002	黄姐	专卖店系统测试户	¥424.00	2007-01-03	无			待审核	执行	未收
0609130805070004	黄姐	专卖店系统测试户	¥8,880.00	2007-01-03	无			待审核	执行	未收
061209080107010001	武汉理工大学	专卖店系统测试户	¥398,750.00	2007-01-03	无			待审核	执行	未收
060913080507010001	黄姐	专卖店系统测试户	¥1,055.00	2007-01-03	无			待审核	执行	未收

图 4-6 订单管理页面

(2) 货款结算

到了结算期，根据未结算的订单生成结算表，财务部门根据订单明细账审核结算表是否相符。审核无误后，总部就与各专卖店进行货款结算。货款结算方式有两种，第一种是直接把货款汇入各专卖店账户，第二种是以货物抵货款，把该货款作为专卖店下次从总部进货的货款。货款结算的部分代码如下：

```
FDT:=DataSet(Session['ReportDataSet']);
if (DBFieldValue(FDT,'shzt')='1') and (DBFieldValue(FDT,'zmdshqk')='1') and
(DBFieldValue(FDT,'zdrq')<>'') then
begin
  fSQL:=' update t_shop_hkjs '+
    ' set      jsr='+Qstr(fUserName)+'
    '      jsrq='+Qstr(FormatDateTime('yyyy-MM-dd',datetime.now))+'
    '      jszt="1" '+
    '      jsfs='+Qstr(ddl_jsfs.SelectedValue)+'
    '      ,hdbh='+Qstr(txt_hdbh.Text.Trim)+'
  where  jsbh='+Qstr(Fjsbh);
if DBExecSQL(fSQL) then
  Return
else
begin
  lbl_message.Visible:=True;
  lbl_message.Text:='提交失败！';
end;
end
else
begin
  lbl_message.Visible:=True;
  lbl_message.Text:='该结算单未审核或审核尚未通过，不能结算！';
end;
```

货款结算页面如图 4-7 所示。



图 4-7 贷款结算页面

4.4 本章小结

本章对该电子商务系统的实现做了详细地阐述。该酒厂的网上商城系统是一个涉及多个系统，多个部门，多个营销环节，集网上商品订购、网上支付、财务结算、物流调度、会员互动、积分管理，能与专卖店前台业务平台互联、与总部业务平台接口的多系统、多平台、多渠道的信息化商务体系。该系统主要包括电子商务子系统和内部业务管理子系统。电子商务子系统主要功能模块有商品展台、购物车、会员中心等，会员中心里主要包括会员活动、我的订单、消费记录、兑奖申请、我的消息、网上调查。内部业务管理子系统主要功能模块有会员管理、积分管理、订单管理、信息管理、商品管理、贷款结算、专卖店监管等。

本章还对 ECO 中状态机和派生属性在该系统中的应用进行了详细地阐述。状态机和派生属性都是定义在模型里，模型可复用。对相似的系统进行开发只需在已建的模型的基础上做一些修改，就能开发新系统。这样使投资可以得到持续回报，提高了软件的开发效率。

第 5 章 总结与展望

5.1 工作总结

本文主要做了以下几个方面的工作：

深入研究了 ECO 技术，包括建立企业逻辑模型、用状态机建立动态模型、数据库的自动生成和自动演化、EcoSpace、如何使用 OCL 和 EAL 对对象进行处理、ECO 组件组的应用等等。ECO 应用程序在执行时期所有的 ECO 对象都是维持在 EcoSpace 中，应用程序也由 EcoSpace 存取 ECO 架框提供的服务，此外 EcoSpace 更负责执行企业逻辑模型。设计时创建的信息模型可以被看做元数据，就象数据库的库表和字段结构一样的信息，而 EcoSpace 类是一个对象容器保存模型定义的类的运行时的实例。ECO 是基于 UML 模型驱动的面向对象的数据库开发框架，可以用来轻松地实现信息模型设计及基于信息模型的应用程序。ECO 提供了快速模型驱动开发的方法，让设计者将更多的精力投入到考虑如何建立企业模型，而不是编代码。ECO 能实现数据库的自动生成和自动演化，数据库平台无关性，模型的高可复用性。

对该酒厂网上商城系统进行了详细地分析设计。网上商城系统主要包括电子商城子系统和内部业务管理子系统。电子商城子系统主要功能模块有商品展台、购物车、会员中心等，会员中心里主要包括会员活动、我的订单、消费记录、兑奖申请、我的消息、网上调查。电子商城子系统里，根据分类对各种商品进行展示、说明，普通用户可以注册成为会员，选中商品放入购物车，并填写订单内容，生成订单提交。进入会员中心，可以查看自己的订单，消费记录。会员有足够的积分后还可以申请兑换奖品。内部业务管理子系统主要功能模块有会员管理、积分管理、订单管理、信息管理、商品管理、货款结算、专卖店监管等。在内部业务管理子系统里，能对商品和信息进行发布和修改等，对会员信息进行管理，对订单进行审核等处理，对货款进行审核和结算等处理，对各专卖店进行监管等。

把 ECO 技术应用于网上商城系统的开发中，并且实现了该系统。对 ECO 技术中的订阅机制和状态机在系统中的实现进行了阐述。订阅机制是 ECO 架构

的基础核心之一。在对象之间的关系上，ECO 需要建立一种互通互信的“订阅”机制，比普通的虚拟环境更为高级。项目中，主要把订阅机制用在派生属性上，派生属性类似于计算字段，但派生属性是非持久化，不占用数据库资源，并且它是定义在模型里，可复用。在状态机里定义好类的若干状态和触发器，当类实例的状态发生转移时可触发一些事件。在系统中，以销售单的状态机为例，定义好销售单的有效、作废状态，设置触发器，用 EAL 语句描述事件：产生日志。当销售单新建或作废时，就会产生日志。由于在状态机里用的是 OCL 语句和 EAL 语句，因此可实现数据库平台无关性。状态机是定义在模型里，因此也可复用。

该酒厂集团地处山区交通不便、信息闭塞，网上商城系统的推出使该酒厂可以实行网络营销，通过网络营销建立会员制俱乐部。会员制俱乐部可以给予酒厂和消费者沟通的最直接的机会，可以吸引更多潜在客户，培养客户忠诚度，通过俱乐部营销把一般客户发展为重点客户或关键客户。网络营销受众群体庞大，针对性强，网络宣传费用低，弥补了传统企业宣传方式的不足。网络营销将产品直接向消费者推销，可缩短分销环节，拓宽销售范围，节省促销费用，从而降低成本，使产品具有价格竞争力。还可根据订货情况来调整库存量，降低库存费用。

5.2 工作展望

由于 ECO 框架里包含的技术和知识点太多以及时间原因，还有很多 ECO 相关知识点我没有学习和应用，例如：the object factory service, the dirty list service, the extent service 等等。还需要对 ECO 框架里的技术进行深入研究。

该酒厂将来还要与一些飞机场和酒店合作，使持有一定级别该酒厂会员卡的会员享受飞机票打折和住酒店优惠的政策。因此，当此网上商城系统运用以后，将来还需要扩展网上商城系统，做飞机场和酒店的终端接口。

参考文献

- [1] Fountain. R., Braithwaite. R., Joyce. P. Teaching electronic commerce: a new focus for business computing[J]. Software Engineering: Education and Practice, 1998: 353-358.
- [2] 王全胜. 电子商务原理[M]. 北京, 北京大学出版社, 2002.
- [3] 甄阜铭. 电子商务基础教程[M]. 大连, 东北财经大学出版社, 2001.
- [4] Chen Xiao-fang, Wang Ying. Model of Multi-Agent Based on Personalized Transactions in Electronic Commerce[J]. Management Science and Engineering, 2006: 208-212.
- [5] 方美琪. 电子商务概论[M]. 北京: 清华大学出版社, 2002.
- [6] 黄敏学. 电子商务[M]. 北京: 高等教育出版社, 2001.
- [7] 阎冰洁. 基于 B-B-C 电子商务模式的研究与应用[D]. 武汉: 武汉理工大学, 2006.4.
- [8] Rui Cao, YueTing Chai. The study of the electronic commerce business models[J]. Systems, Man and Cybernetics, 2004: 4089-4094.
- [9] 韦欣. B2C 电子商务模式的发展[J]. 科技经济市场, 2007 年 02 期: 10-11.
- [10] 崔琦. J2EE 技术在电子商务系统中的应用和实现[D]. 西安: 西北工业大学, 2005.2.
- [11] Edwards. C. Model development[J]. IEE Review, 2003: 42-45.
- [12] Siegel. J. Why use the model driven architecture to design and build distributed applications[J]? Software Engineering, 2005: 37-37.
- [13] Kuhne. T. Understanding metamodeling[J]. Software Engineering, 2005: 716-717.
- [14] Gervais. M.P. Towards an MDA-oriented methodology[J]. Computer Software and Applications Conference, 2002: 265-270.
- [15] 姜泉, 赵建华等. 针对应用系统 ER 模型的 MDA 模型转换方法[J]. 计算机科学, 2006 年 07 期: 236-241.
- [16] 张栋伟. 基于 MDA 软件开发的过程模型及其应用[D]. 上海: 上海师范大学, 2006.
- [17] Hong Wang, Dong Zhang, Jun Zhou. MDA-based development of e-learning system[J]. Computer Software and Applications Conference, 2003: 684-689.
- [18] 徐晓钟. 模型驱动架构(MDA)相关技术研究与实现. 计算机科学, 2006 年 12 期: 278-279.
- [19] Wegmann. A., Preiss. O. MDA in enterprise architecture? The living system theory to the rescue[J]. Enterprise Distributed Object Computing Conference, 2003: 2-13.
- [20] 崔晓斐. 下一代面向对象的软件开发方法——MDA[J]. 山西科技, 2007 年 01 期: 59-60.
- [21] Ahuja. V. Building trust in electronic commerce[J]. IT Professional, 2000: 61-63.
- [22] Il-Woo Lee, Ho-Jin Park, Dae-Ung Kim, etc. A study on B2C based Electronic Commerce Payment System using the telephone number[J]. Computers and Communication, 2003: 620-624.

- [23] 汪荣. 浅谈我国电子商务物流发展对策[J]. 吉林省经济管理干部学院学报, 2007 年 01 期: 75-76.
- [24] 肖林. 电子商务网上支付系统的研究[D]. 贵阳: 贵州大学, 2006.7.
- [25] 王建宏. 电子商务的发展状况之我见[J]. 商场现代化, 2007 年 09 期: 154.
- [26] 金双武. 基于 J2EE 的企业电子商务平台的研究及优化设计[D]. 武汉: 武汉理工大学, 2005.
- [27] 陈亚丽. 基于 MDA 的软件开发方法研究与应用[D]. 西南石油大学, 2006.5.
- [28] 陈省. 基于 ECO 的 UML 模型驱动的数据库应用开发[J]. 程序员, 2003.12: 99-102.
- [29] Ober, I. More meaningful UML models[J]. Technology of Object-Oriented Languages and Systems, 2000: 146-157.
- [30] 何智峰. 基于 Borland ECO II 的数据库开发[J]. 电脑知识与技术, 2006 年 02 期: 30-31.
- [31] Briand, L.C., Labiche, Y., O'Sullivan, L. Impact analysis and change management of UML models[J]. Software Maintenance, 2003: 256-265.
- [32] Haohai Ma, Zhe Ji, Weizhong Shao, etc. Towards the UML evaluation using taxonomic patterns on meta-classes[J]. Quality Software, 2005: 37-44.
- [33] 李维. 开发第一个 ECO 应用程序. <http://liwei.csdn.net/down/Chap01SC.PDF>
- [34] Nguyen, K.D., Zhenxin Sun, Thiagarajan, P.S., etc. Model-driven SoC design via executable UML to SystemC[J]. Real-Time System Symposium, 2004: 459-468.
- [35] 杨东. 基于 UML OCL、支持团队开发的企业组织元模型[J]. 高技术通讯, 2004 年 06 期: 60-64.
- [36] Shen, W., Compton, K., Huggins, J. A method of implementing UML virtual machines with some constraints based on abstract state machines[J]. Software Engineering Conference, 2003: 224-233.
- [37] Wei Dong, Ji Wang, Xuan Qi, etc. Model checking UML statecharts[J]. Software Engineering Conference, 2001: 363-370.
- [38] 陈省. OCL 查询语言[J]. 程序员, 2004 年 05 期: 104-106.
- [39] 陈省. ECO 高级应用[J]. 程序员, 2004 年 03 期: 103-105.
- [40] 鲍旭峰. 电子商务的会员制度[J]. 科技经济市场, 2007 年 01 期: 209-210.
- [41] 王森林. 电子商务与企业成本竞争力[J]. 商场现代化, 2007 年 09 期: 147-148.
- [42] 唐宏超. 基于 Multi-Agent 的专卖店网络分销管理系统[D]. 武汉: 武汉理工大学, 2006.
- [43] 陈艳春. UML 和 WebML 在电子商务信息流系统建模中的应用[J]. 石家庄铁道学院学报, 2007 年 01 期: 77-83.

致 谢

首先，我衷心感谢我的导师吕锋教授三年来给予我的悉心指导和教诲，感谢陈德军老师的悉心指导和教诲。他们严谨求实的治学态度，丰富的知识与经验，前瞻开阔的科学视野和诲人不倦的风范，给我的影响至深，培养了我独立开展工作的科研能力和创新精神，使我受益匪浅！

感谢信息工程学院的全体老师在我学习期间给予我的帮助和关心。

感谢武汉视聆信息技术有限公司总经理高文立老师对我的帮助。

感谢我的父母和女友，及我所有的亲人，学业的完成离不开他们的支持、鼓励和无私的奉献，在此，表示衷心感谢！

感谢马英哲和实验室其他同学、朋友在平时对我的帮助和支持，在此，表示衷心地感谢！

最后让我再一次向所有关心、支持和帮助过我的人表示真诚的谢意！

王 秉

2007 年 4 月

攻读硕士学位期间发表的论文及参加的项目

发表的论文：

- [1] 吕锋, 王秉. 基于 Multi-Agent 的虚拟企业系统模型研究. 武汉理工大学学报, 2007, 29 (1): 95-98.
- [2] 吕锋, 王秉. 基于 ECO 的模型驱动的数据库开发. (已被武汉理工大学信息工程学院模拟国际会议录用)

参与的项目：

- [1] 2006.5 至 2006.7, 参与中国贵州茅台酒厂集团国酒茅台专卖店前台业务系统的编程, 系统集成、测试, 系统用户培训与维护。
- [2] 2006.7 至 2006.10, 参与中国贵州茅台酒厂集团国酒茅台网上商城系统的需求分析, UML 建模, 建数据库, 编程与测试。