基于 MVC 模式的客户关系管理系统设计*

邹　鹏, 尚　维, 李一军
(哈尔滨工业大学 管理学院, 黑龙江 哈尔滨 150001)

摘　要: MVC(Model-View-Controller) 模式是一种适用于交互式系统的软件设计模式。首先对客户关系管理系统的交互性、可扩展性、数据操作复杂性和集成性等特点进行了分析。针对客户关系管理系统的特点提出了基于 MVC 模式的系统设计方案, 重点论述了系统的功能和结构方面的特点。

关键词: 客户关系管理; MVC 设计模式; J2EE
中图法分类号: TP302　　**文献标识码**: A　　**文章编号**: 1001-3695(2005) 02-0021-03

Design of CRM System Based on MVC Pattern

ZOU Peng, SHANG Wei, LI Yi-jun
(School of Management, Harbin Institute of Technology, Harbin Heilongjiang 150001, *China)*

Abstract: Model-View-Controller (MVC) pattern is a software design pattern that is suitable for interactive system. This paper analyzes the CRM characteristics: interactive, expansibility, the complexity of data operation and integration. To these characteristics of CRM, it provides a system design scheme on MVC pattern, and describes the advantages of function and structure in the system.

Key words: Customer Relationship Management (CRM) ; MVC Design Pattern; J2EE

1　引言

客户关系管理(Customer Relationship Management, CRM) 是企业管理中集成信息技术的管理方法和应用解决方案的总和, 它既是帮助企业组织管理客户关系的一系列信息技术、方法和手段, 又是运用信息技术对企业涉及销售、客户服务等业务流程自动化的软件及硬件系统。CRM 系统的设计和开发是企业在实施 CRM 解决方案过程中的重要环节。

2　CRM 系统特性

随着客户关系管理理论和企业实际需求的不断深化和发展, CRM 系统的功能和结构特性也在不断演进, 但是搜集来自企业外部(市场和客户) 的信息, 同时处理企业内部经营管理数据, 经过综合分析, 为企业不同层面的经营管理人员提供信息和知识的反馈, 这一基本功能是不变的。这也就决定了 CRM 系统具有交互性、扩展性、数据多样性和集成性[1] 等主要特点。

2.1　交互性

CRM 系统是一种典型的交互式系统。首先它要在业务的流程中合理地提供与客户更多的接触点, 使企业保持与客户的紧密联系, 跟踪客户关系的全过程。其次, 它还要方便快捷地将客户信息传递给企业有关人员, 为管理人员的决策和业务人员的具体执行提供支持。在业务流程中, 与系统发生交互的对象数量众多, 且多种多样。有企业的决策者、日常管理人员、营销人员、业务员、分销商和客户, 不同的用户对系统的需求不同, 特征各异, 而且数量也是在不断变化。另外系统的交互终端有 PC 机、网上自助服务、ATM、PDA 和各种无线, 移动通信设备等多种形态。所以 CRM 系统应该是一个提供强大交互功能的系统。

2.2　扩展性

一个成熟的 CRM 系统应该是可扩展的, 必须要能很好地适应企业日常业务的变化。因此, 同企业的日常业务一样, CRM 系统也不是绝对稳定的。在 CRM 系统实施运行过程中, 通常每隔一段时间会要求增加新的模块; 如果企业员工或用户的需求发生变化, 会要求系统更改, 企业业务流程的变化也会要求系统的更改。对于功能完善的 CRM 系统来讲, 它应该能按不同功能分为很细的模块, 由最终用户根据自身要求组装适合自己的系统。

2.3　数据操作复杂性

CRM 系统的核心功能是处理有关客户的数据, 因此, 客户数据直接决定了系统的特点。而客户关系的数据具有以下三个特点: ①分散性, 企业所收集的客户信息来自于不同的渠道。②多样性, 客户数据应包括客户的姓名、年龄、工作等个人信息, 还要包括客户曾经购买企业产品的列表, 对产品的偏好等非常个性化的数据, 这些都是对统计分析有用的信息, 是提供个性化服务所必需的, 而不同的数据使用者对数据需求的内容和显示格式也不相同。③动态性, 客户的数据是一种变动的数据, 随时间和客户情况变化而变化, 具有一定的动态性。这些都要求系统具有处理复杂数据的能力。

2.4　集成性

CRM 系统在企业中不是孤立存在的, 它应该与其他内部和外部的 IT 系统结合起来, 充分利用现有的信息资源, 使所有系统的信息实现共享, 满足全局性的需求, 达到网络规模效益

收稿日期: 2004-03-13; 修返日期: 2004-05-11
基金项目: 国家自然科学基金资助项目(70171013)

最大化。这要求系统能够提供系统间的接口,集成基于不同操作系统和数据库的其他系统。

CRM 系统还有如安全性、Internet 访问等普适于一般系统的功能要求。我们要根据 CRM 系统的特点选择相适应的模式,为 CRM 系统提出一种合理的设计方案。

3 MVC 设计模式介绍

MVC(Model-View-Control) 设计模式,即模型 - 视图 - 控制器模式,是为那些需要为同样的数据提供多个视图的系统而设计的[2,3]。它很好地实现了数据层与表示层分离,将系统对象分为三类: ①Model 类,主要实现系统的业务逻辑和数据逻辑; ②View 类主要实现系统的显示逻辑; ③Control 类主要实现系统的控制流程。MVC 体系保持系统的业务逻辑和数据逻辑、显示逻辑、控制逻辑的相互独立性,从而实现面向业务领域构建业务逻辑和数据逻辑,面向具体的应用领域设计控制逻辑和显示逻辑。在系统业务流程调整时,无须修改或尽可能少修改业务逻辑和数据逻辑。在业务领域本身的业务规则、算法变化时,只修改 Model 类组件,从而实现系统组件的独立性。MVC 模式分离了数据访问和数据表现。其结构如图 1 所示。

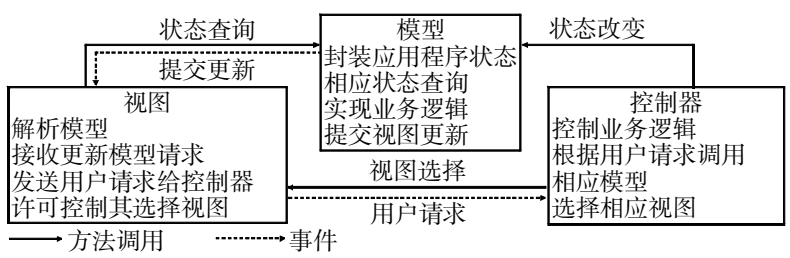

图 1 MVC 组件体系结构图

由于 MVC 模式有效地实现了表达与内容的分离,通过将数据库查询语句这样的数据层代码和像 HTML 这样的表示层代码分离开来,为那些需要多种数据操作的程序的用户提供了最低的代码重复,方便地提供多个视图显示多套数据格式。很好地实现了数据表现与控制的分离,此外,MVC 模式普遍应用于 J2EE 构架,J2EE 平台以其丰富的系统功能,通过 JDBC, JMS, XML, JNDI, CORBA 等 API 可以与几乎所有关系型数据库、事务处理服务器、消息处理服务器、目录服务器和邮件服务器等进行无缝对接,还对 Web Service 的支持都使 J2EE 对原有系统具有很好的集成能力[4]。从而保护原有的投资,并且为将来的发展留出广阔的空间。

4 基于 MVC 模式的 CRM 系统设计方案

4.1 系统功能模块划分

系统包括以下五个功能模块:

(1) 销售自动化

①模块用户: 销售人员、销售管理人员;

②模块功能: 账户管理、订单管理、联系人管理、销售机会管理、活动管理、报价管理、销售预测、佣金管理、报表管理、费用管理;

③信息集成: 产品目录价格、购买记录、服务记录、存货情况、促销记录、信用记录;

④应用集成: 电子邮件、传真、办公软件、促销管理模块、联络中心渠道、产品配置及定价。

(2) 市场营销

①模块用户: 营销人员;

②模块功能: 促销项目管理、营销文本管理、客户管理、价格管理、市场划分管理、脚本管理、竞争管理;

③信息集成: 客户信息、客户商务智能信息、产品信息、营销知识;

④应用集成: 报告软件、商务智能、工作流执行引擎、办公软件、电子邮件、传真。

(3) 客户服务

①模块用户: 客户服务人员;

②模块功能: 现场服务管理、电子邮件管理、客户投诉管理、网络自助服务管理;

③信息集成: 客户信息、产品和配件信息、订单信息、联系人信息等;

④应用集成: 工作流应用、邮件服务器、销售自动化、呼叫中心。

(4) 呼叫中心

将销售模块和服务模块集成,可以进行实时的销售和服务。它通过管理账户、合同等信息,知识库支持,可以满足客户的多方面的需求。呼叫中心提供全面的计算机电话集成技术(CTI) ,可以在用户拨叫的过程中获得客户的资料,灵活地进行业务处理。

(5) 电子商务

电子商务模块是以上所有模块的一个逻辑集成,它提供了一个个性化、高度集成且易于使用的用户界面。在这个用户界面上客户可以进行各种活动,诸如购买、付款、寻求服务支持、查询产品与服务目录、查询订单状态等。

4.2 基于 MVC 的系统实现方式

下面通过分析设计销售自动化模块中的订单管理功能的实现过程来说明系统基于 MVC 的实现方式。系统是在 J2EE 平台上设计开发的,将 MVC 模式映射到 J2EE 应用上,如图 2 所示。

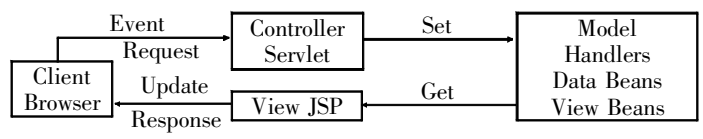

图 2 基于 J2EE 平台的 MVC 体系结构图

例子中的 CRM 系统利用 Java 语言进行服务器端应用程序的开发,使用 Apache Tomcat 作为应用服务器,数据库管理系统选用 SQL Server 2000,数据库接口程序使用 JDBC2 接口。销售自动化模块使用基于 Web 的方式来实现客户管理、职员管理、订单管理、商品管理、出库管理、入库管理等业务处理、流程控制、权限控制、查询统计以及打印等功能。其中详细分析订单管理部分的查询功能。

4.2.1 数据库设计

对于订单部分,需要定义订单信息表 Db_order,表中的字段有订单编号(OrderNo) 、客户编号(CustNo) 、商品编号(PrdtNo) 、商品单价(Price) 、订购数量(Amnt) 、合计总价(TotPrc) 、收到金额(PayAmnt) 、收到日期(PayDate) 。为了能够在应用中使用 Db_order 表,建立相应的数据库连接。

4.2.2 业务功能实现设计

在系统的开发过程中,基于 MVC 设计模式的思想,结合系统的实际情况,进行如下设计:

(1) Model 层设计

我们将 Model 层按照分工进行了进一步划分。Model 层由

许多 Java Beans 组成, 根据这些 Beans 在系统中起的不同作用, 将它们分为 Handlers, Data Beans, View Beans 三种类型。其中 Handlers 用来实现业务逻辑即对对象实例的处理。具体工作是根据请求参数提取一个或多个 Data Bean 并组装成 View Bean 送至 Control 层; Data Beans 是用来描述和定义从现实世界中抽象出来的对象模型; 而 View Beans 其中包括页面显示信息及用户访问期信息的集合。它是盛有数据包的容器, 用来将处理完的对象实例进一步封装并返回到客户端。此外还设计了一个 Data Manager, 它具有存取数据的全部功能, 具体表现形式就是一组静态的 SQL 函数, 在数据读取时, 能够产生 Data Beans。

(2) Control 层设计

Control 层的功能由 Servlet 来完成。当它接收到一个用户请求时, 向 Handlers 发出指令, 要求其根据请求参数提供一个专门处理该类请求的 Handler, 再要求该 Handler 处理该请求。

(3) View 层设计

View 层的功能在 JSP 页面中实现。在我们的设计中每个 JSP 只和一个 View Bean 对应, 这减少了 Servlet 出错的可能。

4.2.3　系统的工作过程

下面以订单查询过程为例, 简述以上设计方案的工作机制。根据功能需求, 设计了以下实例化的类(表 1), 实现了订单查询的部分功能。

表 1　实现订单查询功能的部分实例化类

对象名	功　能
Handlers:	
. QueryAmntHandler	查询订货量
QueryClientHandler.	客户查询
QueryPriceHandler.	订单价格查询
QueryPayAmntHandler.	到款情况查询
Data Beans:	
ClientInfo	客户信息
AmntData	订货数量
PriceInfo	价格信息
PayAmntData	到款信息
View Beans	
OrderHomeViewBean	订单主页面显示内容
OrderQueryViewBean	订单查询页面显示内容
QueryResultViewBean	订单查询结果显示内容

用户输入订单号, 要求查询该订单的客户信息和到款情况, 系统进入以下过程:

(1) Servlet 接收到浏览器发来的请求后根据请求参数在 Handlers 列表中搜索专门处理该类请求的 Handler, 向 Handlers 发出指令, 要求其提供一个专门处理订单客户查询的 QueryClientHandler. 和查询到款情况的 QueryPayAmntHandler. , 并要求其处理请求。

(2) QueryClientHandler 和 QueryPayAmntHandler. 获得 Servlet 传递过来的信息, 根据业务逻辑对信息进行处理。为获得查询请求中的信息, 它们将调用 Data Manager 相应的 SQL 函数对数据库进行操作。数据库操作完成后, 再将返回的记录集封装成 ClientInfo 和 PayAmntData 的对象实例, 进行一定的处理后, 将这些 Data Beans 再封装到事先定义的 QueryResultViewBean 中去。Handler 将组装好的 QueryResultViewBean 返回给 Servlet。

(3) Servlet 将用户请求及回应参数传递给 QueryResultViewBean, 在 Servlet 的控制下, 与这个 QueryResultViewBean 相对应的 JSP 从 QueryResultViewBean 得到所有的信息, 然后转变成 HTML 页面送还给终端用户。

以上过程可以用图 3 来表示。

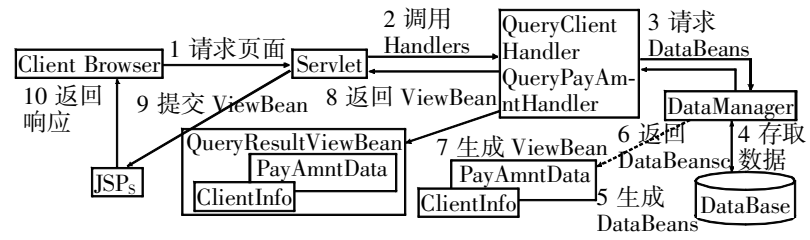

图 3　系统工作流程示例

5　结论

本文提出的设计方案较好地适应 CRM 系统的特点, 主要体现在以下方面:

(1) 设计中使用 JSP 与 Servlets 联合使用来实现动态内容服务的方法, 用 JSP 生成表达层的内容, 让 Servlets 完成深层次的处理任务。这样清晰地分离了表达和内容。

(2) 由 Data Manager 完成对数据库的操作, 产生 Data Beans, 这种设计使数据存取的细节得以被封装, 从而使业务逻辑层与数据层有更好的分离。由于设计方案对显示、业务逻辑及数据的清楚分离使 CRM 系统在进行功能扩展和改动时只需对相应的业务逻辑进行重组, 不用去改动整个系统。此外也可以很方便的重用现有构件。

系统中的每个 Handler 被设计成只完成一个功能, 而所有的 Handler 都由 Servlets 控制、调用。这种设计便于明确定义不同用户的角色权限, 使 CRM 系统可以很方便地支持其他新的客户类型。

设计中每个 View Beans 都携带有用户访问期的信息, Servlets 可以根据这些信息找到相应的 JSP 页面, 这样系统就可以向用户显示个性化的界面, 该设计满足了 CRM 系统用户对数据操作的多样性和复杂性的要求。

由于设计方案是在 J2EE 平台上开发的, 所以可以很好地与其他系统集成。

由于 MVC 模式更适合复杂系统的开发, 对于未接触过该模式的开发者来说, 学习起来有一定难度, 所以在设计实现 CRM 系统时还要根据任务和开发团队的实际情况来选择方案。

参考文献:

[1] 杨骐, 王加阳, 刘连浩, 等. 基于 J2EE 体系结构的 CRM 系统的设计与实现[J]. 计算机应用研究, 2002, 19(11): 153-154.

[2] 袁梅冷, 黄烟波, 黄家林, 等. J2EE 应用模型中 MVC 软件体系结构的研究与应用[J]. 计算机应用研究, 2003, 20(3): 147-149.

[3] 张普朝, 王愚. UML 与 MVC 设计模式在社区信息系统中的应用与实现[J]. 计算机应用, 2003, 23(6): 103-105.

[4] 张建奋, 王申康. 使用 J2EE 构建的网上公安信息系统[J]. 计算机工程, 2002, (9): 195-197.

[5] Erich Gamma, Richard Helm, Ralph Johnson. Design Patterns: Elements of Reusable Object-Oriented Software[M]. New York: Addison Wesley Professional, 1995.

作者简介:

邹鹏(1975-), 男, 辽宁沈阳人, 博士研究生, 主要研究方向为客户关系、数据挖掘; 尚维(1978-), 女, 黑龙江哈尔滨人, 博士研究生, 主要研究方向为电子商务; 李一军(1957-), 男, 黑龙江宾县人, 教授, 博士生导师, 主要研究方向为电子商务、管理信息系统与决策支持系统。