

文章编号:1672-6197(2006)02-0049-04

# 基于J2EE的MVC设计模式的研究与实现

孙福振, 李 艳, 李业刚

(山东理工大学 计算机教学部, 山东 淄博 255049)

**摘 要:** 分析讨论了J2EE构架和面向对象的MVC设计模式,并将MVC设计模式引入J2EE体系之中,在北京市电信卡综合管理系统的应用和实现中收到良好的效果,而且给该管理系统的开发、维护和升级带来了便利.

**关键词:** J2EE; 多层体系结构; MVC

**中图分类号:** TP302.1

**文献标识码:** A

## Research and realization of the MVC design pattern based on J2EE platform

SUN Fu-zhen, LI Yan, LI Ye-gang

(Computer Teaching Department, Shandong University of Technology, Zibo 255049, China)

**Abstract:** This article describes the concept and principle of the J2EE platform and the MVC design Pattern. The MVC design pattern based on J2EE platform is studied deeply in this paper, which is applied in the system of Beijing telecom card integrated management and bring us advantage during the development, maintenance and upgrade of the system.

**Key words:** J2EE; multilayer architecture; MVC

随着 WebLogic, Websphere 等 Application-Server 对 J2EE(Java2 Enterprise Edition)的全面支持, J2EE 正逐渐成为 Web 应用软件开发的

标准. 本文所研究的 MVC (Model/ View/ Controller) 模式是软件工程学中一个非常重要的设计模式, 为交互式系统提供了一个非常优秀的开发模型, MVC 开发模式对类对象的强制分离使系统开发变得非常灵活, 同时又提高了系统设计的可重用性. 正是 MVC 模式在交互式系统开发中所体现的强大优势, MVC 模式被开发人员逐步引入 J2EE 体系之中, 成为 J2EE 进行交互式应用开发时, 特别是

Web 应用开发时一个非常重要的设计模式.

## 1 J2EE

J2EE 是一个涉及多个层面的复杂的概念<sup>[1]</sup>, 是 SUN 基于 java 的体系结构. J2EE 为开发跨平台、具有高度可扩展性和健壮性的企业应用提供了极大便利, 显示出了其巨大的发展前景, 几乎所有大型软件公司, 如 SUN, IBM, ORACLE, BEA, HP 等, 都推出了遵循 J2EE 标准的应用服务器产品; 并且, 它们又进一步促进了 J2EE 技术的发展与完善.

## 1.1 J2EE 构架

J2EE 构架是一个 4 层结构<sup>[2]</sup>.

(1) 用户层. 用来与用户交互, 并把来自系统的信息显示给用户. J2EE 平台支持不同类型的用户, 包括 HTML 用户、Java Applets 和 Java 应用等.

(2) Web 层. Web 层产生表示逻辑, 并接受来自客户端的用户反馈. 在所接收的客户端请求的基础上, 表示层对用户的请求产生相应的回应. 在 J2EE 平台中, 由 Web 容器内的 Servlet 和 JSP 来实现这一层.

(3) 业务层. 业务层处理应用的核心业务逻辑. 业务层为低层业务提供必要的接口. 业务组件通常被实现为 EJB 容器内的 EJB 组件. 其中, EJB 容器提供组件生命周期、管理持久性、事务和资源分配等.

(4) EIS 层. 该层为企业的信息系统服务, 包括数据库系统、事物处理系统、企业资源计划系统等. EIS 是 J2EE 应用与非 J2EE 应用的连接点.

层次结构如图 1 所示.

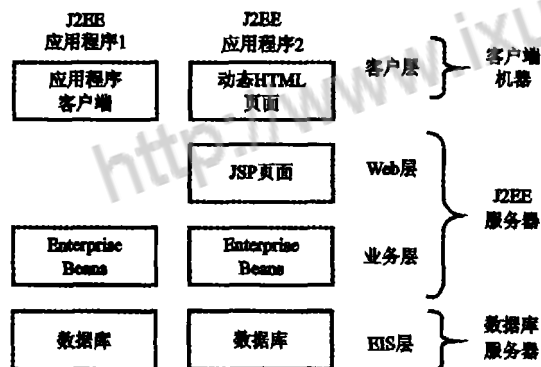


图 1 J2EE 层次体系结构图

## 1.2 J2EE 容器和应用组件

J2EE 应用程序是由组件构成的. J2EE 组件是具有独立功能的软件单元, 它们通过相关的类和文件组装成 J2EE 应用程序, 并与其他组件交互. J2EE 规范中定义了以下的 J2EE 组件:

- (1) 组件: 应用客户端程序和 applets;
- (2) Web 组件: Java Servlet 和 JavaServer Pages;
- (3) 业务组件: JavaBeans 和 Enterprise JavaBeans

在系统的部署过程中, 组件被安装到容器中. 容器就是组件与支持组件的特定平台功能之间的接口. 容器设置定制了 J2EE 服务器所提供的内在支持, 包括安全、事务管理、JNDI 寻址、远程连接等服务.

## 2 MVC 设计模式

### 2.1 MVC 模式的概念

谈到 MVC 的起源, 不能不提 Small Talk —— 20 世纪 70 年代由 Xerox 公司所开发的, 最早的完全面向对象的程序设计语言之一. MVC 就是当时 Small Talk 用来创建用户界面的方法, 体现了将状态与表示分离的思想.

MVC(Model - View - Controller)即模型 - 视图 - 控制设计模式, 是一种面向对象的设计模式, 旨在实现表示和逻辑的分离, 使软件体系结构层次清晰, 便于开发升级和维护, 也为软件的健壮性提供了保障.

### 2.2 MVC 工作模型原理

MVC 模式将所面对的系统分为 3 个部分, 分别是数据模型、视图和控制器, 并定义了这 3 个部分之间实现通信的一种模式, 使每个部分不必卷入到其他部分的状态表示和方法实现的细节中去. 每个部分有自己的数据管理规则, 各个部分对象之间的通信只能使用已定义的一个受限连接集合进行. 保持这种分离性通常是好的面向对象编程和设计要追求的目标, 让各个对象只需专注于自己的事务.

Model 也叫模型, 本质上封装了数据及行为, 其中包含对数据控制及修改的规则, 提供了一套查询、改变 Model 状态的方法.

View 也叫视图, 是 Model 所表示出来的图形界面, 它主要用于提交 Model 的信息展示给用户.

Controller 也叫控制器, 是 Model 和 View 之间的协调者, 它的主要作用有以下几点:

- (1) 定义用户界面对用户输入的响应方式.
- (2) 解释用户的输入, 并命令 Model 进行相应的操作创造相应模型.
- (3) 负责将模型信息传递给 View, 必要时还要负责创建新的 View 和 Controller.

这 3 者的工作方式可用图 2 来表示<sup>[3]</sup>.

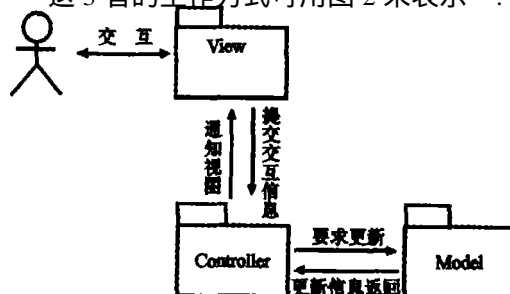


图 2 MVC 工作方式示意图

2.3 MVC 设计模式的优势

使用 MVC 的好处,一方面,分离数据和其表示,使得添加或者删除一个用户视图变得很容易,甚至可以在程序执行过程中动态地进行用户视图的更新.模型和视图能够单独开发,即业务逻辑和用户表现的分离,增加了程序的可维护性,可扩展性,并使测试变得更为容易.

另一方面,将控制逻辑和表现界面分离<sup>[4]</sup>,允许程序能够在运行时根据工作流、用户习惯或者模型状态来动态选择不同的用户界面.现在一个应用程序需要用越来越多的方式访问,MVC 设计模式支持多个视图能共享一个模型,因此在 MVC 设计模式中应用开发人员可以最大化的重用他的代码,大大提高了软件开发的效率和质量.

总之,运用 MVC 的 Web 应用程序的 3 个部件(模型 - 视图 - 控制)是相互独立的,改变其中一个不会影响到其他两个,所以依据这种设计思想开发者能构造良好的松耦合的构件.

3 基于 J2EE 的 MVC 模式的设计与实现

北京通讯公司的电信卡综合管理信息系统涉及到电信系统的各种卡,如 IP、IC、201 等,实现从卡的制作到入库,到出库,到销售,到回库,到支付的整个流程的管理,以及盘库,综合查询,综合报表等的业务管理,并为决策提供一定支持.

针对北京电信卡综合管理系统,我们将面向对象编程中的 MVC 模型应用于基于 J2EE 构架的 Web 应用程序的开发中,下面以北京电信卡综合管理系统为例,论述基于 J2EE 的 MVC 模式的设计与实现,着重讨论其模型、视图和控制器的设计与实现.

3.1 开发环境的选取

经过以上分析,我们已明确将采用 MVC 模式和 J2EE 构架的技术,经过对相关的当前流行开发工具的分析比较,确定开发环境如下:

(1)开发工具.包括 Servlet 和相关类的开发工具(JBuilder 9.0)、JSP 页面开发工具(Dream-Weaver2004)、后台数据库(Oracle9i)及 Oracle9i 数据库开发工具(toad).

(2)软件环境.系统平台采用了 Windows 2000 Server;Web 服务器采用了 Tomcat.

3.2 控制器( Controller)实现

控制器流程如图 3 所示,它是整个 MVC 三个部件的核心调度.

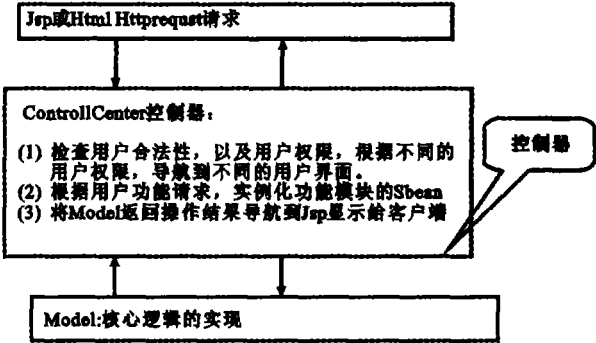


图 3 控制器工作流程图

控制器(ControllCenter)以 J2EE 的 Servlet 实现如下:

(1) ControllCenter 将根据不同的用户,不同的权限,导航到不同的主界面,假设以管理员的身份登陆,ControllCenter 的比对权限并导航主要操作代码片段:

```
// 以系统管理员身份登陆导航到 main_sys.jsp
if (info.DeptLevelId.equals("0"))
    url = "main_sys.jsp";
if ( ! url.equals("") )
    { // 获取相关用户信息,并保存在 session,以便跟踪用户.
        HttpSession session = m_ControlCenter.m
        _Request.getSession(true);
        session.setAttribute("userInfo", info);
    }
else
    // 如果用户信息错误,导航到错误提示
    logonFailure.htm
    url = "logonFailure.htm";
    // 导航到与用户权限匹配的主页
    rd = m_ControlCenter.m_Request.get
    tRequestDispatcher(url);
    m_ControlCenter.m_Response.sendRedirect(url);
    rd.forward(m_ControlCenter.m_Request,
    m_ControlCenter.m_Response);
```

(2) 控制器 ControllCenter 实例化对应的

Sbean,进而调用 Ebean,Dbean 来实现用户操作.其主要操作代码片段如下:

```
public boolean Do_Dispatch()
{
    // 获取 Sbeans 类名
    String ClassName = XMLAnalyze. GetS-
    BeanClassName(m_szClassName,m_XMLFile);
    // 根据类名创建 Sbeans 对象,即实例化
    Sbeans 类
    Class theSBeanClass = Class. forName
    (sbeanClassName);
    HiEnd_SBaseBean theSBean = (HiEnd_
    SBaseBean) theSBeanClass. newInstance();
}
```

(3)由 Do\_Dispatch()方法来实现根据用户请求实例化对应功能模块的 Sbean.当功能执行完毕后的将操作结构以 JSP 反馈给用户的代码片段如下:

```
RequestDispatcher rd;
rd = m_ControlCenter. m_Request. ge-
tRequestDispatcher (" cardtype/ cardTypeFrm.
htm");
rd. include (m_ControlCenter. m_Request,
m_ControlCenter. m_Response);
```

### 3.3 模型(Model)实现

在模型(Model)设计部分我们提出了模型(Model)的3层的设计,即 Sbean 处理用户参数,Ebean 完成逻辑运算,Dbean 完成数据库操作.而在数据层,考虑到数据操作的效率我们封装了数据库连接池.如图4所示.

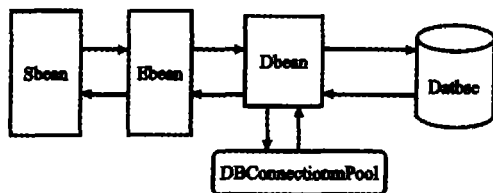


图4 Model 三层体系结构示意图

模型(Model)部分有3层的JavaBeans实现,功能结构上层次明晰、耦合度低,便于维护和升级.在北京市电信卡综合管理系统的所有功能模块即模型(Model)部分,都采用了3层的体系结构,具体实现主要代码如下:

```
(1) public class SB_ApplyCard extends
```

```
HiEnd_SbaseBean
{
    public void DP_SubmitApply()// Sbeans 统
    一标准接口
}
(2) public class EB_ApplyCard extends
HiEnd_EbaseBean
{
    // 各种具体业务操作接口
    public boolean NewApply (ApplyCardData
    A_ApplyCard, ArrayList A_BillDetail)
}
(3) public class DB_ApplyCard extends
HiEnd_DbaseBean
{
    // 后台数据库操作接口
    public boolean NewApply (ApplyCardData
    A_ApplyCard, ArrayList A_BillDetail)
}
```

### 3.4 Jsp 界面( View)实现

采用 MVC 设计模式,其中视图(View)是由 Jsp 来实现的.作为用户界面要求友好美观,因而在界面的外观设计请美工完成.作为程序设计人员实现以下两方面的工作:

(1)将模型(Model)的返回结果,嵌入到页面中,反馈给用户.嵌入实现:

```
< %
// 处理、显示
String realTotal = (String) request. getAt-
tribute("realTotal");
String ok = ((String) request. getAttribute
("ok")). trim();
%>
```

(2)在 Jsp 页面中编写 JavaScript 脚本,完成用户的输入合法性检查,在客户端而不是在服务器端完成用户输入的检验,提高系统的效率. JavaScript 嵌入实现:

```
< script language = "JavaScript" type = "
text/JavaScript" >
function Validate()
{
    if(用户输入合法)
```

### 3 结束语

本文首次给出了用精细积分解决时滞系统动力学问题的方法. 通过 2 个算例题可以看出, 同其他方法相比, 用精细积分解决时滞系统动力学问题有解题过程简单、结果精确度高的优点.

#### 参考文献:

- [1] Hale J K, Lunel S M V. Introduction to functional differential equations[M]. Berlin :Springer-Verlag, 1993.
- [2] Zennaro M. P-stability properties of Runge-Kutta methods for delay differential equations [J]. Numer. Math., 1986, 49:305-318.
- [3] Halle H C. Singly-implicit Runge-Kutta methods for retarded and ordinary differential equations [J]. Computing, 1990, 43:202-222.
- [4] Jackiewicz Z, Lo E. The numerical solutions of neutral functional differential equations by Adams predictor-correctors methods[J].

Appl. Numer. Math., 1991, 8:477-491.

- [5] Ford Neville J., Lunel Sjoerd M. Verduyn. Charactering small solutions in delay differential equations through numerical approximations[J]. Applied Mathematics and Computation, 2002, 131: 253-257.
- [6] Suleiman Mohamed bin, Ismail Fudziah. Solving delay differential equations using componentwise partitioning by Runge-Kutta methods [J]. Applied Mathematics and Computation, 2001, 122:301-323.
- [7] Kainhofer Reinhold. QMC methods for the solution of delay differential equations [J], Journal of Computational and Applied Mathematics, 2003, 155:239-252.
- [8] Agrawal Om Prakash, Gregory John. The complete solution for constrained delay problems in the calculus of variations by unconstrained methods[J]. Journal of Mathematics Analysis and Applications, 1998, 218:127-135.
- [9] 钟万颢. 暂态历程的精细计算方法[J]. 计算力学及其应用, 1995, 12(1):1-6.
- [10] 丁丽娟. 数值计算方法[M]. 北京:北京理工大学出版社, 1997.

(接 52 页)

```
{  
// 提交服务器  
submit()  
}  
else // 否则提示用户,重新输入  
}  
</ script >
```

### 4 结束语

本系统是基于可视化面向对象建模技术研发, 遵循网络化发展需要, 采用先进的 MVC 设计模式, 基于 J2EE 技术进行设计和开发的. MVC 模式就是开发人员在经过长时间经验的积累之后所提出的面向交互式系统开发的解决方案, 它的强制性

带来了系统开发的灵活性、可重用性和易维护性.

本文设计的 MVC 模式在北京电信卡综合管理系统的应用中收到了良好的效果, 它作为一个设计模式可复用到其他企业信息应用中, 并可以在实践中不断地完善和发展.

#### 参考文献:

- [1] 胡松利, 刘 涇. 基于 J2EE 构建多层分布式物资管理信息系统[J]. 现代电子技术, 2005, 28(4):50-52.
- [2] 武 伟, 陆建德. 层模式在 J2EE 中的应用开发研究[J]. 微机发展, 2005, 15(1):126-129.
- [3] 李赤林, 王 琳. Model - View - Controller 设计模式实例研究 [J]. 计算机与现代化, 2003, (3):11-14.
- [4] 彭 雷, 李伟生. 利用设计模式构造高效 Web 应用开发的模型[J]. 计算机应用研究, 2005, 22(1):174-176.



论文写作，论文降重，  
论文格式排版，论文发表，  
专业硕博团队，十年论文服务经验



SCI期刊发表，论文润色，  
英文翻译，提供全流程发表支持  
全程美籍资深编辑顾问贴心服务

免费论文查重：<http://free.paperyy.com>

3亿免费文献下载：<http://www.ixueshu.com>

超值论文自动降重：[http://www.paperyy.com/reduce\\_repetition](http://www.paperyy.com/reduce_repetition)

PPT免费模版下载：<http://ppt.ixueshu.com>

---