

基于 SSH 的网上超市系统的研究

郑步芹^{1,2}, 姜利群¹, 张瑜慧²

(1. 中国矿业大学 计算机学院, 江苏 徐州 221008; 2. 宿迁学院, 江苏 宿迁 223800)

摘要:该文首先对网上超市系统进行了需求分析, 给出了相应的功能模块; 其次介绍了 SSH 整合框架相关技术; 最后讲解了 SSH 框架在网上超市系统中的具体应用。

关键词: Struts2; Spring; Hibernate; MVC

中图分类号: TP311 文献标识码: A 文章编号: 1009-3044(2009)35-9957-02

Studies of on-line Supermarket System Based on SSH

ZHENG Bu-qin^{1,2}, JIANG Li-qun¹, ZHANG Yu-hui²

(1. College of Computer Science, China University of Mining and Technology, Xuzhou 221008, China; 2. Suqian College, Suqian 223800, China)

Abstract: This essay, firstly, through requirement analysis of the on-line supermarket system, gives the system functional modules. Secondly, introduces the related technologies of the SSH's integrated framework. Finally, explains the detailed application of the Integrated framework in the on-line supermarket system.

Key words: struts2; spring; hibernate; MVC

随着 Internet 的不断发展, 网上购物已成为一种时尚。电子商务的双向信息沟通给消费者提供了更多选择, 灵活的交易手段为用户带来了更好的便利性。网上超市就是一个典型的电子商务系统。网上超市经营者可以以最快的速度将商品分类和上架。用户可以不受时间和空间的限制, 查看检索商品的信息和购买商品。这种快速便利的交易方式, 大大提高了企业的生产效益, 降低了企业成本, 有助于企业更快更好的发展。

1 网上超市的总体分析和设计

1.1 网上超市的需求分析

对于传统超市, 消费者购物、挑选、交换、比较购买商品。超市管理员要对商品进货、分类和统计, 对过期商品要进行相应的处理。网上超市不同于传统超市, 但也应能完成普通超市的一般功能。网上超市用户可以分成两种类型, 一类是一般会员可以注册更改用户信息, 查看和检索商品信息、购买商品。一类是系统管理员, 可以对会员信息进行管理, 可以对商品进行增加、分类和统计。

1.2 网上超市的功能设计

根据需求分析可知, 可以将网上超市系统可以分为前台用户模块和后台管理模块。前台用户模块可以分为: 会员登录注册、商品浏览、商品查找、购物车、订单查询。后台管理员管理模块可以分为: 会员管理、商品管理、订单管理。功能模块图如图 1 所示

商品管理: 可以对商品查看、分类、增加、删除、修改、统计等功能。

购物车管理: 会员可以将选购商品放入购物车, 同时还可以对购物车里的商品进行修改和删除, 确认购物后可以生成订单。

会员管理: 可以查看、修改、删除用户, 及进行身份认证。

订单管理: 用户可查询订单信息, 管理员根据情况修改订单。

2 系统的相关技术

2.1 MVC 模式

MVC 这个概念是由 SmallTalk 开发社区提出来的, 它是第一个分开表示逻辑和业务逻辑的设计模式。MVC 引入 View 视图(表示层)、Model 模型(数据)和协调两者的 Controller 控制器。在 MVC 模式下, 在客户端用户通过表单等手段和视图进行交互, 确认后, 将事件传给控制器。控制器在收到事件后对事件进行初步处理, 选择发给哪一个模型, 通知该模型层进行相应的更新。模型更新后将更新的结果再反馈给视图, 进而最后再向用户反馈。如图 2 所示。

MVC 设计模式是一种用于分离数据维护和数据表现的方法, 在 J2EE 中引入 MVC 设计模式, 有助于把应用划分为合理的组件, 从而可以方便的进行系统的开发、维护和扩充^[1]。

2.2 Struts 框架

Struts 是基于 Sun 公司 J2EE 平台的 MVC 框架。Struts 框架完全采用 Java 语言编写, 在编写时调用了标准的 J2EE API, 因此具有“一次编写, 随处运行”的优点。同时, 它继承了 MVC 设计模式的各种特性。Struts 基本组件包括控制器组件、模型组件和视图组

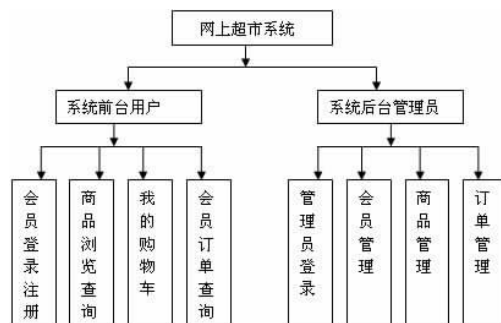


图 1 网上超市系统功能模块图

收稿日期: 2009-09-20

作者简介: 郑步芹 (1981-), 女, 中国矿业大学计算机学院, 硕士, 宿迁学院, 讲师, 主要研究方向为 J2EE; 姜利群 (1956-), 女, 江苏海门人, 副教授, 硕士生导师, 主要研究方向为网络技术; 张瑜慧 (1979-), 女, 硕士, 讲师, 主要研究方向图像处理。

的操作也就变成了对这些 POJO 的操作。

1) 持久化对象(Persistentobject)

持久化对象(Persistentobject)是介于具体应用和数据库之间的一层数据,负责应用和数据库之间的交互。持久化对象用来维护数据库表的记录,它是一个完全符合 JavaBean 规范的纯 Java 对象,属性通过 getter 和 setter 方法访问,对外隐藏了内部实现的细节。

会员管理需要涉及会员信息表(user),该表包含会员 ID、会员名称、密码等信息。下面我们给出 User.java 文件

```
public class User {private Long id;
/* 会员表中其它字段 */
public Long getId() { return id; }
public void setId(Long id) { this.id = id;}
/* 其它 getter 和 setter 方法 */ }
```

2) O/R 映射文件

下面是以会员信息 User 实体为例,讲解映射文件描述数据库表字段与持久化类的属性的映射关系,如代码示例所示。

User.hbm.xml 代码

```
<hibernate-mapping>
<class name="com. supermarket.bean.User" table="User" catalog=" supermarket ">
/* 在持久化类 User 和表 User 之间映射关系。下面进行的是在持久化类 User 的属性和表 User 的字段建立一一对应的映射。 */
<id name="id" type="long" column="id">
<generator class="increment"></generator>
</id>
/* 其它持久化类的属性和表的字段的一一映射 */
</class>
</hibernate-mapping>
```

3.3.2 数据持久层实现

数据持久层使用数据访问对象(Data Access Object,简称是 DAO)来抽象和封装所有对数据源的访问。

DAO 主要是实现对持久化类的插入、查询、更新、删除操作。一般一个持久化对象使用一个独立的 DAO 接口和一个对接口的实现,下面是会员管理中的会员信息的 DAO 对其接口的具体实现,代码如下所示。

UserDAOImpl.java 代码

```
public class UserDAOImpl extends HibernateDaoSupport implements UserDAO
{@SuppressWarnings("unchecked")
public List<User> listUserById(Long UserId)
/* 将用户信息按升序显示 */
/* 其他操作 */}
```

3.3.3 表示层实现

表示层是 Web 应用系统与用户交互的窗口,负责管理用户的请求和响应,并提供一个控制器将调用委托到业务逻辑和其他上游处理,包括视图部分和控制部分,具体是指 JSP 页面,以及对客户请求进行处理的自定义 Action 等部分。

1) 控制器 Action:Action 充当了控制器的角色,用于页面逻辑处理,分发数据,链接转向。会员管理中的显示所有用户页面逻辑处理类功能都放在 listUser.java 文件中。

```
public class ListUser extends ActionSupport{}
```

2) 视图:视图简单的说就是用户所看到的 JSP 页面。本系统中,页面采用了 Struts 标签。会员管理涉及诸多的页面,如会员的添加、修改、删除、等页面,因代码比较长,我们就不在此列出。

3) Struts.xml 配置:显示所有用户信息的 Action 在 Struts.xml 配置文件的配置代码如下所示

```
<action name="list User " class="list User ">
<result name="success">listUser.jsp</result>
</action>
```

3.3.4 业务逻辑层实现

业务逻辑是整个系统中比较复杂的部分,可以说是一个核心部分。业务处理代码主要包括 xxxservice 接口与其实现类 xxxserviceImpl。例如会员管理的业务逻辑代码 UserService 接口和其实现类 UserserviceImpl 构成。UserService 接口定义了针对问题信息的相关业务操作,具体实现则由 UserServiceImpl 完成。

下面给出显示所有会员信息 action 在配置文件 applicationContext.xml 中的配置代码:

```
<bean id="listUser" class="com. supermarket.action.User.ListUser" scope="prototype">
<property name="service" ref="UserService"></property>
</bean>
```

4 总结

利用 J2EE 中的 Struts2+Spring+Hibernate 的轻量级框架实现 MVC 模式是当前比较流行的一种技术。该整合框架能够很好的将网上超市系统分成表示层、业务逻辑层、数据持久化层和模型层。Struts 框架应用在表示层的框架,用于完成表现层的表现逻辑和业务逻辑;Hibernate 框架应用在数据持久化层,实现关系数据库的对象化;Spring 框架应用于各层关联,完成持久访问对象的注入和业务逻辑的事务管理。通过这种分层处理,使得系统系统的结构清晰,各层次之间独立,模块划分明确,降低了 MVC 设计难度,提高了系统的开发效率。

参考文献:

- [1] 李金娟.基于 JSP 的 MVC 开发模式在访问数据库中的应用[J].宁夏师范学院学报,2008(3).
- [2] 杨少波.J2EE 项目实训-Hibernate 框架技术[M].北京:清华大学出版,2008.
- [3] 张恒汝.精通 Eclipse 整合 Web 开发[M].北京:人民邮电出版社,2008.