

文章编号:1001-9081(2006)11-2769-04

集成 J2EE 框架构建可复用的电子商务系统

刘婷婷, 蒋玉明, 傅静涛

(四川大学 计算机学院, 四川 成都 610065)

(jiangyuming@cs.scu.edu.cn)

摘要:针对 J2EE Web 应用传统开发方法的不足,在实际系统分析的基础上,集成 Struts、Spring 和 Hibernate 三种 J2EE 框架技术的体系结构,并快速构建和开发了一个多层 B/S 的旅游电子商务系统,减少了开发复杂性,缩短了系统的开发时间,同时该系统具有良好的复用性、扩展性和维护性。

关键词:J2EE; 电子商务; Struts; Hibernate; Spring; 快速构建; 复用性

中图分类号: TP393.09 **文献标识码:** A

Integration with J2EE framework to build reusable E-commerce system

LIU Ting-ting, JIANG Yu-ming, FU Jing-tao

(College of Computer Science, Sichuan University, Chengdu Sichuan 610065, China)

Abstract: Considering the disadvantages of traditional development methods for J2EE Web applications, based on the system analysis, an architecture integrating Struts, Spring and Hibernate frameworks was proposed. A multi-tier E-commerce system for tourism was rapidly built and developed using the architecture. The system can greatly reduce development complexity and time while providing the system with favorable reusability, expansibility and maintainability.

Key words: J2EE; E-commerce; Struts; Hibernate; Spring; rapid build; reusability

由于电子商务的内部逻辑复杂、安全性要求苛刻、商务形式发展变化快^[1],基于 J2EE 的 Web 应用以其层次性、平台无关性、安全性和可扩展性已成为目前电子商务系统主要的解决方案。但是 J2EE 传统的 JSP/Servlet 开发模式却常常带来显示、业务和数据的高耦合,造成软件难以复用,不能满足目前大型的 Web 应用开发中普遍要求降低开发团队的技术成本和分工协作难度,缩短开发时间的要求。成熟的 J2EE 框架技术是解决上述问题的重要技术^[2],本文在实际系统分析的基础上,集成 Struts、Spring 和 Hibernate 三种 J2EE 框架技术快速构建一个多层的可复用的旅游电子商务系统。

1 J2EE 框架

框架是可重用的、半成品的应用程序,可以用来产生专门的定制程序^[3],一般具有即插即用的可重用性、成熟的稳定性以及良好的团队协作性^[4]。

1.1 Struts 框架

在 J2EE 应用程序架构的发展路程中,主要经历了两大阶段:1) Model 1。在 JSP 页面中结合业务逻辑、服务器端处理程序和 HTML,同时实现显示、业务逻辑和流程控制。2) Model 2。Servlet 处理数据存取和导航流,JSP 处理表现。它和 Model 1 最大的区别在于引入了 MVC 模式的概念。MVC 设计模式由控制器(Controller)、模型(Model)、视图(View)三部分组成,通过控制器将表现逻辑和业务逻辑解耦,弥补了 Model1 的不足,提高了系统的可维护、可扩展和可重用性^[5]。

Struts 框架就是基于 Model 2 的架构^[3],主要用于 J2EE 表示层。Struts 有三个主要部件:一个 Controller 及负责具体业务处理的 Action 类;JSP 页面(View);应用的业务逻辑封装(Model)。Struts 的中心控制器(ActionServlet)接受来自客户

端的请求,并根据系统的配置(Struts-config.xml)路由 HTTP 请求到其他 Action 对象。在这些 Action 对象中会进行业务操作,处理完毕再由 ActionServlet 转向到 JSP 页面,将处理结果返回给客户端。同时还通过 ActionForm 对象封装数据在 Model 和 View 之间交互使用。

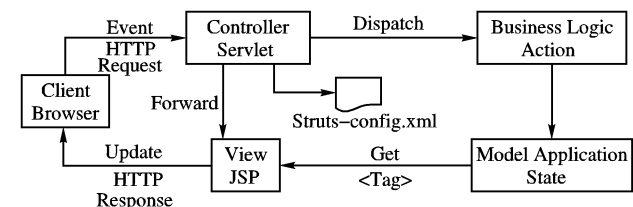


图 1 Struts 体系结构^[6]

1.2 Hibernate 框架

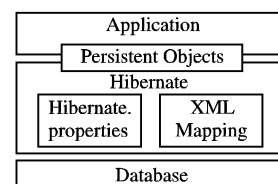


图 2 Hibernate 体系结构^[7,8]

关系数据库的非对象化的管理模式给面向对象开发带来了异构的数据结构,因此数据库对象化具有重要意义。Hibernate 就是一种容易使用的、功能强大的持久层框架,一种轻量级的对象-关系数据映射(Object/Relation Mapping, ORM)^[7]。它通过 Properties 文件建立数据库连接,通过映射文件(*.hbm.xml)将数据库表映射为 Java 类,表中的每条数据作为一个类实例,而数据列值则映射为实例属性;还提供了专门的查询语言 HQL 来查找和操纵对象。这样通过持久性对象将数据库数据以同构的数据对象形式传送到其他层中加

收稿日期:2006-05-15;修订日期:2006-07-02

作者简介:刘婷婷(1980-),女,四川乐山人,硕士研究生,主要研究方向:计算机网络、信息系统;蒋玉明(1964-),男,四川南充人,教授,主要研究方向:数据库与集成信息系统、ERP;傅静涛(1974-),男,四川雅安人,讲师,硕士,主要研究方向:计算机网络、信息系统。

以处理,就能够以面向对象思想处理数据,并且可以不再使用冗长的 JDBC 语句。

1.3 Spring 框架

Spring 是一种轻量级 J2EE 应用程序框架,它使应用程序能够抛开 EJB 的复杂性,同时具有传统 EJB 相关的关键服务。而且 Spring 不是一个强制性框架,开发者可以选择某个模块(或组件)独立使用。图 3 所示为 Spring 的模块图。其中 Spring Core 是 Spring 的基础,提供了一个控制反转(Inversion of Control, IoC)容器管理业务组件;而 Spring DAO 和 Spring ORM 不仅提供数据访问的抽象模块,还集成了对 Hibernate 的支持,并提供缓冲连接池、事务处理等重要的服务功能。IoC/DI(Dependence Injection, 依赖注射)是 Spring 框架的核心^[9],即组件之间的关系由容器来控制,而非传统的由程序代码直接操控。Spring 的另外一个重要性质就是对面向对象编程思想的支持,可以用此机制来拦截事务处理^[10]。

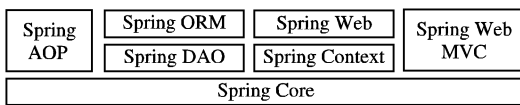


图 3 Spring 模块图

2 旅游电子商务系统的分析与设计

本旅游电子商务系统的目标是通过旅游电子商务平台的建设实现:吸引,使更多客户了解风景区的旅游产品体系,特别是面向广大消费者的产品链;交易,建立面向旅游企业的 B to B 销售模式和面向终端消费者的 B to C 销售模式,实现旅游产品、运营、数据服务的一体化交易;分析,通过定制搜索引擎和数据分析系统,及时掌控行业市场动态,发现潜在客户和关键客户,充分跟踪市场机会,建立完善的科学决策体系。

同时,本系统应与企业信息系统、门户系统无缝集成一体化。由于风景区的电子商务应用业务流程处在不断的变化中,因此兼顾未来发展,本系统的整个硬件系统、软件系统和编码系统还应具有柔性和调整扩充空间。

2.1 系统角色分析

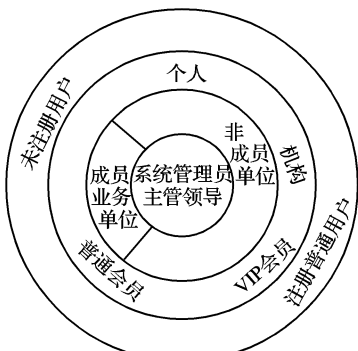


图 4 系统角色

本系统角色分为四类:系统管理员、业务单位、会员和普通游客,如图 4 所示,他们具有不同的权限。

2.2 主要业务流程

1) 预订和在线支付流程,如图 5 所示。

2) 退订流程。用户想要退订,应通过网站申请,如果该订单未付款,则立即取消预订,扣减一定数额的积分,向业务单位发送退订通知,并向用户返回退订成功的消息;如果该预订已经付款,则人工电话联系所有相关业务单位,检查该预订是否已经发生费用,如果发生费用,扣除退订要缴的手续费,返回用户一个实际退款。用户查询后确定是否肯定退订。

3) 换订流程。用户想要修改预订内容,应先做退订再重新预订。如果新订单的总额超过原订单,退换订中间产生的费用由业务单位承担;如果新订单的总额小于原订单,则差额部分要扣除手续费等,再退回用户。

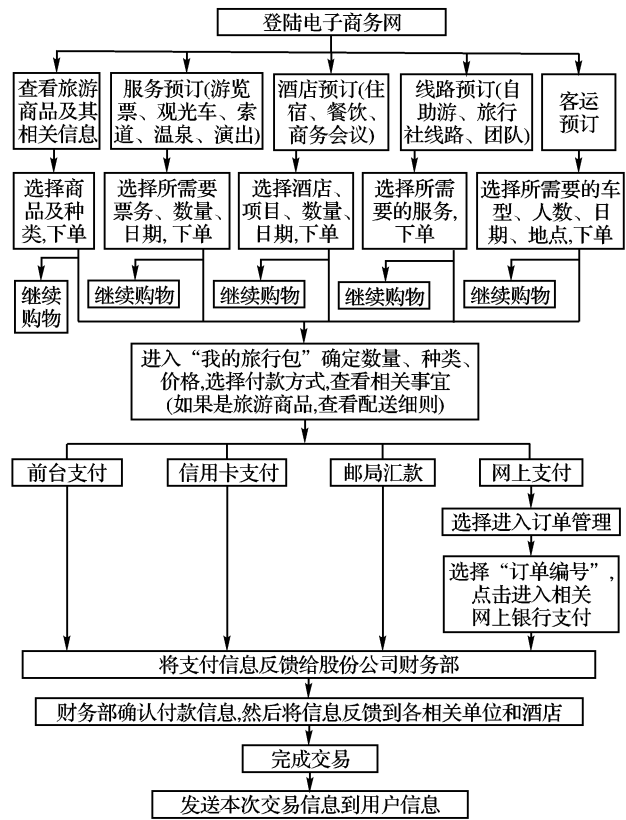


图 5 预订和在线支付流程

2.3 系统功能设计

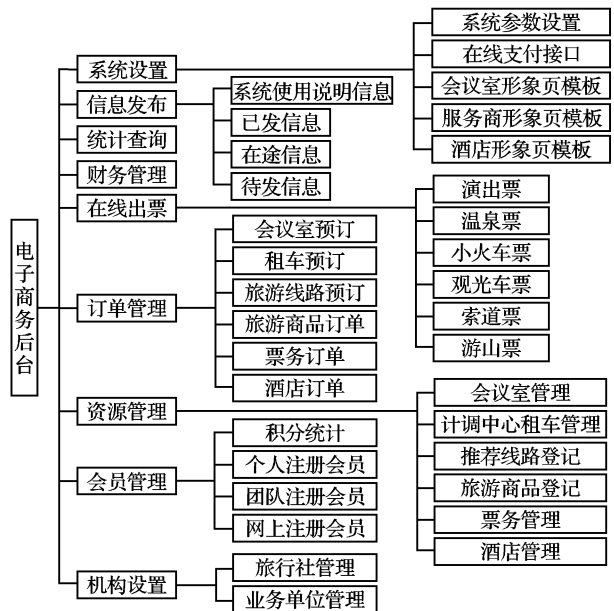


图 6 电子商务后台管理系统功能树

前台功能 1) 预订资源(酒店、票务、商品等信息和价格)列表显示和查询; 2) 推荐酒店详细资料查询; 3) 预订及在线支付,用户选择票务、酒店等放入购物车,系统自动为每个业务单位生成一张订单; 4) 退订或变更; 5) 查询已预订情况。

酒店推荐功能(供管理员使用) 1) 特推酒店的增删改功能; 2) 主要推荐酒店特推的服务、套房及相关设施。

后台管理功能(供管理员使用) 1) 维护业务单位信息

和管理员账号; 2) 添加预订类别, 设置不同的形象显示页和列表; 3) 发布预订信息; 4) 查看和处理、改变、删除或废弃订单; 5) 系统使用说明信息的维护; 6) 语音提醒。

后台管理功能(供业务单位使用) 1) 发布预订信息, 可以同时用多种语言发布预订信息; 2) 查看订单和付费信息、处理退换订单; 3) 由财务部门根据用户支付反馈回的信息修改订单状态。

2.4 系统体系架构

依据 J2EE 的典型构架^[11], 集成前述的 Struts、Spring 和 Hibernate 框架, 本旅游电子商务系统划分为五个层次(图 7):

客户层 客户端用户使用 Web 浏览器(IE)进入本系统访问, 与服务器端软件进行交互;

表现层(Web 层) 使用 Struts, 根据 ActionServlet 接收到的 Request 调度相应的 Action;

业务逻辑层 由 Spring IoC 容器负责向 Action 提供业务模型 Model 组件和该组件的协作对象 DAO (Data Access Object) 组件实现业务逻辑;

数据持久层 由 Hibernate 从数据源中获取数据, 生成 PO (Persistent Object), 再把 PO 传给 DAO 组件; 在应用了 Hibernate 的 J2EE 构架中, PO 完全可以充当 VO (Value Object)^[9];

数据源层 数据库系统负责管理、存储、组织和分析系统的相关业务数据和用户相关数据。

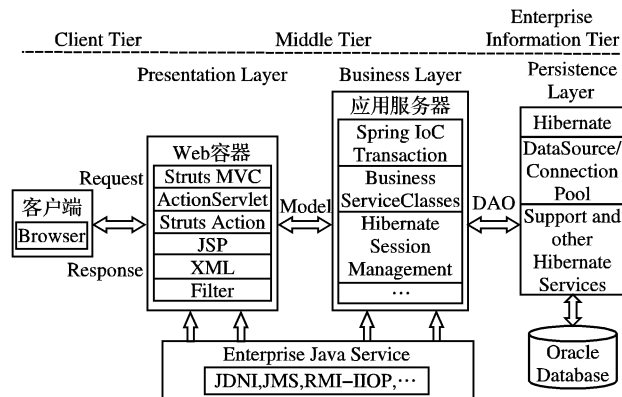


图 7 电子商务系统体系结构

以上架构充分利用了三种框架各自的优势, Struts 引入 MVC 模式实现了表示逻辑和业务逻辑的解耦; Hibernate 实现关系数据库对象化; 而 Spring 则提供了 IoC 容器, 完成持久访问对象的注入和业务逻辑的事务管理, 关联 Web 层和持久层。采用多层体系结构, 进一步降低了各层间的耦合度, 有利于组件复用和系统扩展、维护; 有利于项目小组明确分工, 多人同时并行开发, 缩短开发周期。

3 旅游电子商务系统的实现

3.1 表现层的实现

使用 Struts 框架, Java 代码与页面表示逻辑分离。由 ActionServlet 实现请求(*.do)到业务逻辑(Action)的映射, 再由 Action 调用相应的业务逻辑的服务组件, 并根据处理结果调用 Forward 对象所指定的响应。以酒店客房管理模块为例, 在 struts-config-eps-resource.xml 中有如下一个映射:

```
<-- EbsHotelRoomAction.do 请求调用 EbsHotelRoomAction 对象,
返回"EIDHotelList"跳转至列表显示页面, 返回"AddHotel"跳转至
添加/编辑页面 -->
< action path = "/EbsHotelRoomAction"
```

```
name = "EbsHotelRoomForm" scope = "request"
type = "com.bstar.eps.hotel.action.EbsHotelRoomAction" >
< forward name = "EIDHotelList" path = "/HotelList.jsp" />
< forward name = "AddHotel" path = "/AddEditHotel.jsp" />
...
</action >
```

新增加客房信息时, ActionServlet 收到客户端填写客房信息表的 Request 后, 将用户提交过来的数据填充到 EbsHotelRoomForm 类中, 该 Form 对象将被作为参数传递给 EbsHotelRoomAction 类的 execute 方法。再由该 Action 根据 Request 参数, 从 IoC 容器中获取业务模型完成业务逻辑处理。EbsHotelRoomAction 类的部分实现如下:

```
public ActionForward execute(.....) throws Exception{
    EbsHotelRoomModel eh = (EbsHotelRoomModel) this;
    wac.getModel("ebsHotelRoomModel");
    //从 Spring 中获取业务模型
    String Action = ParamUtil.getString(request, "action", "error");
    String orgid = WebSite.getOrgID(request);
    if (Action.equals("AddHotel1")) {
        if (eh.addHotelRoom(request)) {
            Action = "success"; //新增成功
        } else {
            Action = "error"; //新增失败
        }
    }
    ...
    return mapping.findForward(Action); //返回 Forward 对象 }
```

3.2 业务逻辑层的实现

业务逻辑层由 Spring 框架实现, 主要包括 Model 组件, 使用 DAO 模式封装具体的数据持久层操作以供 Model 组件调用数据的 DAO 组件以及由 Spring 提供的事务处理、缓冲连接池等服务组件。它们统一由 Spring IoC 容器管理, 从而使开发者摆脱了开发一个组件还需要开发其相应的工厂类、代理类、单例模式、多例模式等所产生的繁琐劳动, 并且能够方便地在其他项目中重用这些组件。例如, EbsHotelRoomModel 封装了添加或修改客房信息、取得客房信息列表等客房管理的各种业务。EbsHotelDAO 则封装了对 Hibernate 的数据操作的实现。通过配置 Spring 的 ApplicationContext.xml, EbsHotelRoomModel 调用 EbsHotelDAO 实现各种业务操作:

```
<-- 创建 Hibernate 的 session 工厂 -->
< bean id = "sessionFactory" class =
    "org.springframework.orm.hibernate.LocalSessionFactoryBean" >
< property name = "mappingResources" > < list >
< value > com/bstar/hbmxml/eps/org/EbsHotelRoom.hbm.xml
</value > </list > </property > ..... </bean >
<-- 创建容器提供的事务管理组件, 对 Model 组件进行管理 -->
< bean id = "proxy" class =
    "org.springframework.transaction.interceptor.
    TransactionProxyFactoryBean" >
< property name = "transactionManager" >
< ref local = "transactionManager" /> </property >
< property name = "target" >
< ref local = "ebsHotelRoomModel" /> </property > ...
</bean >
<-- 创建 Model 组件, 并调用 DAO 组件作为协作对象 -->
< bean id = "ebsHotelRoomModel"
    class = "com.bstar.eps.hotel.model.EbsHotelRoomModel" >
< property name = "ebsHotelRoomDAO" >
< ref local = "ebsHotelRoomDAO" />
</property > </bean >
<-- 创建 DAO 组件, 并调用 Hibernate 的 session 工厂作为协作对
```

象 -->

```
<bean id = "EbsHotelRoomDAO"
class = "com. bstar. ebs. hotel. dao. EbsHotelRoomDAO" >
<property name = "sessionFactory" >
<ref local = "sessionFactory" / > </property > </bean >
```

3.3 数据持久层的实现

图 8 和图 9 为本旅游电子商务系统前、后台界面。



图 8 电子商务前台界面



图 9 电子商务后台管理界面

数据持久层基于 Hibernate 框架,主要包括值对象 (VO)、数据访问对象 (DAO) 和映射文件 (Hibernate Mappings)。系统的主业务流程主要用到以下 10 类 VO: 会员信息、系统信息、酒店信息、日志信息、订单信息、业务组织信息、票务信息、商店信息、在线信息和银行信息。每一类中有数个相关 VO。这些 VO 封装了持久层对象的数据,在业务逻辑层和数据持久层之间进行流动,以传递所需数据。每个 VO 对应于数据库的一张表,是一个 POJO (Plain Old Java Object),即普通的 JavaBean,由一些属性及对应的 getter/setter 方法组成。与 VO 相对应的 DAO 用于持久化 VO,如 MemberDAO、EbsSystemDAO、EbsHotelDAO 等,每个 DAO 均包含插入、删除、更新等数据库基本操作;每个 VO 还有相应的映射文件,用来定义 Hibernate 把对象持久化到关系数据库中的具体规则,例如 EbsStoreType. hbm. xml、EbsCarinfo. hbm. xml 等。同

时还需要在 hibernate. cfg. xml 中配置各映射文件。实际开发中,可以在 Eclipse 中使用 Hibernate 插件直接从已经建好的电子商务系统数据库的数据表结构中自动生成映射文件和对象文件。

4 结语

本文在实际的电子商务系统分析基础上,集成 Struts、Spring 和 Hibernate 框架技术快速地实现了一个 B/S 旅游电子商务系统,独立了显示页面、轻量化了业务组件、对象化了数据库,使得通过配置外部文件、显示模块、业务逻辑模块及数据处理模块就可以复用到其他相关应用开发中,极大地提高了系统开发效率和复用性。

参考文献:

- [1] SCHNEIDER GP, PERRY JT. ElectronicCommerce [M]. USA : CourseTechnology-ITP, 2000.
- [2] JOHNSON R. J 2 EE Development Frameworks [J]. Computer, 2005, 38(1): 107 - 110.
- [3] HUSTED T, DUMOULIN C, FRANCISCUS G, *et al.* Struts in Action: Building Web Applications with the Leading Java Framework [M]. Manning Publications Company, 2002.
- [4] WONG W - C, EYADAT M, NG S. Degree of Freedom - Experience of Applying Software Framework[A]. ITNG'06[C]. Las Vegas, Nevada, USA, 2006.
- [5] GAMMA E, HELM R, JOHNSON R, *et al.* Design Patterns: Elements of Reusable Object-Oriented Software[M]. 北京: 机械工业出版社, 2002.
- [6] DAVIS M. Struts, an open-source MVC implementation[EB/OL]. <http://www-106.ibm.com/developerworks/library/j-struts/?dw-zone=java>, 2003.
- [7] KING G. HibernateReference[EB/OL]. http://www.hibernate.org/hib_docs/v3/reference/en/html/architecture.html#architecture2overview, 2005.
- [8] 孙卫琴. 精通 Hibernate: Java 对象持久化技术详解[M]. 北京: 电子工业出版社, 2005.
- [9] JOHNSON R, HOELLER J, ARENSEN A, *et al.* Spring - Java/ J2EE Application Framework Reference Documentation Version 1. 2. 8[EB/OL]. <http://www.springframework.org/docs/reference/index.html>, 2006.
- [10] JOHNSON R, HOELLER J. Expert One-on-One J2EE Development without EJB[M]. Wiley Publishing Inc, 2004.
- [11] SUN Microsystems, Inc. Java 2 Platform Enterprise Edition Specification, v1. 4 PublicDraft[EB/OL]. <http://java.sun.com/javae/reference>, 2003.

(上接第 2768 页)

- [5] DAI YH, YUAN Y. Some properties of a new conjugate gradient method[A]. Advances in Nonlinear Programming[C]. YUAN Y, ed. Kluwer Academic Publishers, 1998. 253 - 266.
- [6] 飞思科技产品研发中心. 神经网络理论与 MATLAB7 实现[M]. 北京: 电子工业出版社, 2005. 287 - 288.
- [7] 李学伟, 关忠良. 经济数据分析预测学[M]. 北京: 中国铁道出版社, 1998: 225 - 241.
- [8] 王文成. 神经网络及其在汽车工程中的应用[M]. 北京: 北京理工大学出版社, 1998. 61 - 89, 232 - 268.
- [9] RAVDIN PM, CLARK GM, HILSENBECK SG, *et al.* A demonstration that breast cancer recurrence can be predicted by neural network analysis[J]. Breast Cancer Research and Treatment, 1991, 21(1): 47 - 53.
- [10] De LAURENTIIS M, RAVDIN PM. Survival analysis of censored data: Neural network analysis detection of complex interactions between variable[J]. Breast Cancer Research and Treatment, 1994, 32(1): 113 - 118.
- [11] WHITE H. Comment on nonparametric regression: Multilayer feedforward networks can learn arbitrary mapping[J]. Neural Networks, 1990, 3(1): 47 - 51.
- [12] 向小东. BP 算法的改进及其在股票价格预测中的应用[J]. 西南交通大学学报, 2001, 36(4): 425 - 427.
- [13] 鲍立威, 何敏. 关于 BP 模型的缺陷的讨论[J]. 模式识别与人工智能, 1995, 8(1): 1 - 5.