

基于 Struts + Hibernate 的 Web 应用的设计与实现

李发英¹, 朱海滨²

(1. 湘南学院 计算机科学系, 湖南 郴州 423000;

2. Nipissing 大学 计算机科学与数学系, 加拿大 诺斯贝 PIC1L2)

摘 要:详细描述了基于 Struts + Hibernate 整合框架开发 Web 应用程序的基本思想。Struts 是 MVC 模式的典型标准体系结构, Hibernate 是数据库访问封装框架。以在线作业提交系统为应用实例, 给出了在 Struts + Hibernate 环境下开发应用程序的关键技术。总结了整合框架开发 Web 应用程序的优点。实践证明, 基于 Struts + Hibernate 的在线作业提交系统具有很好的灵活性和易维护性。

关键词:Struts; Web 应用; 作业提交系统; MVC; Hibernate

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2009)04-0091-04

Design and Realization of Web Application Based on Struts + Hibernate

LI Fa-ying¹, ZHU Hai-bin²

(1. Department of Computer Science of Xiangnan University, Chenzhou 423000, China;

2. Department of Computer Science and Mathematics of Nipissing University, North Bay PIC1L2, Canada)

Abstract: The basic idea of Web application based on Struts and Hibernate is described. Struts is a standard framework of MVC pattern. Hibernate is a framework of database access. Take on-line assignment submission system for example, the main technology is given in the Struts + Hibernate environment, the excellent of Struts + Hibernate is summarized. As a result, assignment submission system based on Struts and Hibernate is more flexible and maintainable.

Key words: Struts; Web application; on-line assignment submission system; MVC; Hibernate

0 引言

Struts 框架是一种基于 Java 的技术。Hibernate 框架使用 O/R 映射技术, 对 JDBC 进行轻量级的对象封装, 让普通的 Java 对象 (Plain Old Java Objects, POJO) 变成持久化类, 管理 Java 类到数据库表的映射, 并提供数据查询和获取数据的方法。Struts 框架与 Hibernate 框架的整合实现 Web 应用的开发, 既便于表示层的结构化开发, 又可以减少数据访问层的代码编写量, 进一步提高应用程序开发效率, 因此与传统的基于 ASP 的作业管理系统相比, 基于 Struts 与 Hibernate 框架设计的 Web 应用系统具有高效性、可伸缩性、稳定性, 同时还具有灵活性、易维护性和松散耦合等优点。

1 Struts 框架整体架构

1.1 Struts 框架介绍

Struts 是在 J2EE 的基础上基于 MVC 模式的 Web

应用程序开发框架^[1,2]。整个框架可以分为三个层次, 即表示层、视图层和控制层, 层与层之间具有相对独立性。

(1) 视图。Struts 视图部分采用 JSP 实现。Struts 提供了丰富的标签库 (taglib), 通过这些标签库可以最大限度地减少脚本的使用。

(2) 控制器。在 Struts 框架下, 控制器由 Java Servlet 承担, Servlet 负责视图和模型之间的交互^[3,4]。Struts 的控制器由两部分组成。一是系统核心控制器, 就是系统中的 ActionServlet; 二是业务逻辑控制器, 就是用户自己实现的 Action 实例。

(3) 模型。Struts 的模型部分主要由底层的业务逻辑组件充当, 这些业务逻辑组件封装了底层数据库访问、业务逻辑方法实现。它可能是一个或多个 EJB 组件^[5], 也可能是一个 Web 服务。

1.2 Struts 业务逻辑处理时序图

为了更好地描述 Struts 框架下三个部分之间的协作关系以及在特定时间引发的特定动作, 图 1 给出了 Struts 框架下的处理时序图。

收稿日期: 2008-07-24

作者简介: 李发英 (1972-), 女, 硕士, 讲师, 研究方向为 Web 服务、分布式计算。

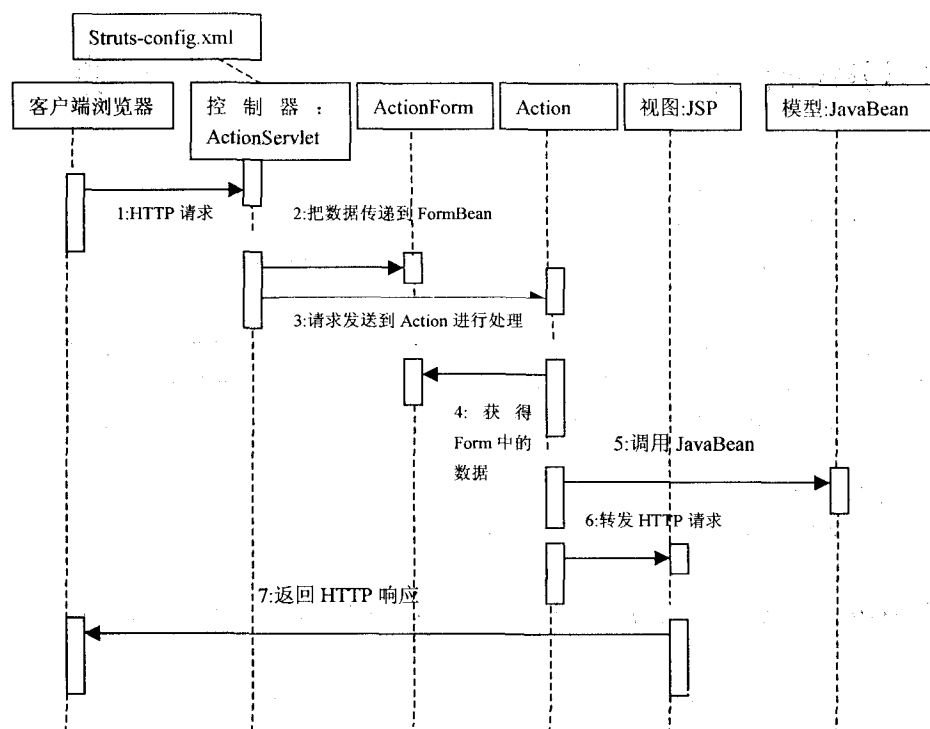


图 1 Struts 业务逻辑处理时序图

从图 1 可以看出, Struts 框架的按时间顺序响应事件的基本步骤如下:

第一步, 客户端浏览器是系统的入口, 用户通过界面中的表单输入系统必需的信息, 也就是 HTTP 请求, 当用户单击表单中的提交按钮后, HTTP 请求被提交到系统的控制器^[6]中;

第二步, 控制器中的 ActionServlet 对象封装表单信息, 将数据传递到具体的 Action 类(由特定的 FormBean 实现);

第三步, 根据 struts-config.xml 中的配置信息, ActionServlet 获取该配置信息所指示的映射关系^[7], 决定 HTTP 请求该由哪个 Action 类来处理, 随即请求发送到指定的 Action;

第四步, 在将 HTTP 请求发送到特定的 Action 类, 接下来要将处理请求所需要的数据, Action 类从 FormBean 中获取需要的数据^[8];

第五步, 在 Action 类处理 HTTP 请求的过程中, 除了获取客户端的表单数据外, 如果要与数据库打交道, 那么 Action 调用 JavaBean 来实现具体的对数据库的访问操作^[9,10];

第六步, Action 类处理 HTTP 请求, 如果不涉及到数据库操作, 则转发 HTTP 请求到 JSP 页面;

第七步, 将 HTTP 响应返回到客户端浏览器。

1.3 Hibernate 框架配置文件的构建

Hibernate 框架^[11]中的核心文件配置文件, Hiber-

nate 通过配置文件 hibernate.cfg.xml 定义数据库基本配置属性、XML 文件定义数据库字段结构, POJO 为数据库记录的对应类。

关键代码如下:

```
<? xml version="1.0" encoding="utf-8"? >
```

.....

```
<hibernate-configuration>
```

```
<session-factory>
```

```
<!-- 数据库连接 URL -->
```

```
<property name="hibernate.connection.url">
```

```
Jdbc: mysql://localhost/hibdemo? useUnicode = true&characterEncoding = utf-8
```

```
</property>
```

```
<!-- 数据库驱动类 -->
```

```
<property name="hibernate
```

```
.connection.driver.class">
```

```
Com.mysql.jdbc.Driver
```

```
</property>
```

```
<!-- 数据库用户名和密码 -->
```

```
<property name="hibernate.connection.username">root</property>
```

```
<property name="hibernate.connection.password"></property>
```

```
<!-- mapping resource = "com/jspdev/hibdemo/User.hbm.xml" -->
```

```
</session-factory>
```

```
</hibernate-configuration>
```

由于是在 Eclipse 集成开发环境下开发本应用程序, 可以通过 Hibernate 配置文件向导来生成配置文件 Hibernate.cfg.xml, 接下来使用 Hibernate Synchronizer 从数据库表自动生成 hbm 映射文件。在集成环境下, 利用向导生成 Hibernate 配置文件、User 类文件及映射文件, 同时生成了封装持久方法的 UserDAO。

2 应用实例

在学校对学生的日常管理中, 作业是一个必不可少的环节, 任何一门课程都会要求学生交作业, 学生查看作业和老师批改作业等程序, 设计作业管理系统来规范作业管理, 为了实现作业提交系统的实时、可靠、安全等特性, 在设计时根据易用性原则, 高效性原则, 可扩展性原则。将 Struts 和 Hibernate 框架整合, 实现

了表示层基于 MVC 模式的设计,解决了数据库访问技术的安全性,同时可以提高应用程序的开发效率。

2.1 系统功能

在线作业提交系统共有四个功能模块,通过这四个模块,达到了应用的基本要求。

(1)登录模块。该模块的主要功能是实现用户登录功能,为两类用户即学生和教师提供良好的登录界面,若输入信息正确则转到用户的详细信息页面,否则转到错误页面。

(2)公告管理模块。该模块主要包括查看公告,发布公告和维护公告。

(3)作业管理模块。包括老师布置作业,学生提交作业和修改作业。

(4)用户管理模块。包括对学生管理子模块,老师管理子模块和课程管理子模块。

2.2 系统总体架构

本系统基于 Struts 和 Hibernate 框架,采用 B/S 结构,总体架构如图 2 所示。

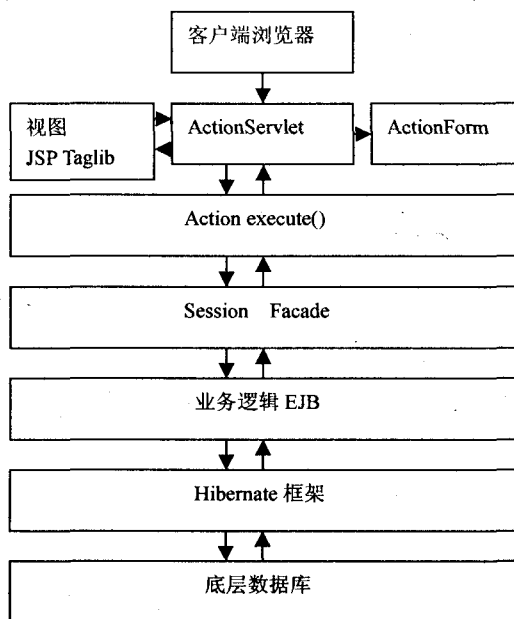


图2 系统总体架构图

对于图2所示的系统,各层的功能如下:

(1)客户层:由客户端浏览器担当,负责处理与用户的交互,用户通过客户层输入系统必需的信息;

(2)表示层:运行在 Web 容器中,处理客户端 HTTP 请求并作出响应;

(3)业务逻辑层:完成系统的业务功能,为表示层提供所需的业务方法;

(4)数据持久层:由 Hibernate3.0 实现。Hibernate 利用数据源生成持久对象,供业务逻辑层处理具体业务时调用;

(5)数据源层:Oracle 数据库,用于存放系统的全部应用数据。

2.3 构建系统表示层配置文件

为了说明如何使用 Struts 实现 JSP 页面的请求及响应过程,构建配置文件时以学生用户登录模块为例。当用户需要登录本系统时,需要一个简单的表单提交页面,这个表单提交页包含了两个表单域即用户名和密码。登录页面 StudentLogin.jsp,当用户输入正确的用户名和密码时,转到学生用户基本信息页 StudentInformation.jsp,否则转到错误 Error.jsp。配置文件 Struts-config.xml 是 Struts 框架的核心部分,它负责协调系统表示层各个组件的工作。下面给出该配置文件的详细配置信息。

Struts-config.xml 首先建立 FormBean 映射,代码如下:

```

<form-bean>
    <form-bean type="Stulogin.controller.StudentLoginForm"
name="StudentLoginForm">
</form-bean>

```

每一个 FormBean 都必须直接或间接地继承 ActionForm 类,在 FormBean 类中,用 set 方法将客户端的请求封装进 FormBean 对象里,用 get 方法将 FormBean 对象的数据取出来供商业逻辑调用。

接下来 Struts-config.xml 定义 Action Mapping,将 ActionServlet 映射事件到 Action 类:

```

<action-mappings type="org.apache.struts.action.ActionMapping">
</action-mappings>
<action path="/login/studentlogin" parameter="log" type="login.controller.LoginAction" scope="request" name="StudentLoginForm" validate="true">
    <forward name="success" path="login.studentinfo.jsp"/>
</action>

```

根据 request 相应的请求动作找到处理该事件的动作类即 action 类,然后由该动作类取出 formbean 里面封装的数据学生用户名和密码,传给业务对象类,在业务对象类中调用业务方法完成相应的业务要求。最后将处理的结果在 JSP 页面中显示。

```

Public class LoginAction extends extends Action{

```

```

    Public ActionForward execute(ActionMapping mapping, ActionForm form,HttpServletRequest request,HttpServletResponse response){

```

```

        ActionForward loginforward=null;
        String loginaction=mapping.getParameter();
        If("Log".equalsIgnoreCase(loginaction)){
            Loginforward=performlog(mapping,form,request,response);
        }
    }

```

3 系统涉及的关键技术

3.1 多页面间共享数据

在 B/S 模式下, 用户发送请求及接受服务器端的请求都是基于 HTTP 协议的, 由于 HTTP 协议是无状态的, 因此请求信息和响应信息无法通过 HTTP 协议本身进行传递。为了跟踪用户的操作状态, 使用 Struts 中的 HttpSession 对象实现对会话状态的跟踪。具体实现代码如下:

```
HttpSession session = request.getSession();//创建与当前请求相关联的会话
```

```
DynaActionForm login = (DynaActionForm)form;
String userid = (String) login.get("userid");
String password = (String) login.get("password");
List userlist = module.login(userid, password);
if (userlist.size() > 0) {
    TbUser user = (TbUser) userlist.get(0);
    session.setAttribute("classid", user.getClassid());
    session.setAttribute("departid", user.getDepartid());
    .....
    return mapping.findForward("index");
} else {
    return mapping.findForward("login");
}
```

3.2 数据访问的实现

由于使用了 Spring + Hibernate 的机制, 这种机制主张接口编程, 充分利用 Java 中的继承机制, 基于这种情况笔者特地编写了一个操作数据库的类 BaseDao, 这样在后面的要操作数据库的类中, 只要用 Spring 的依赖注入机制就可以实现其它类操作数据库的功能。

BaseDao 类的主要方法如下:

```
public class BaseDao extends HibernateDaoSupport implements
IBaseDao {
    public void addObj(Object obj) {
        getHibernateTemplate().save(obj);
    }
    public void delObj(Class clazz, Integer ID) {
        Object obj = findById(clazz, ID);
        getHibernateTemplate().delete(obj);
    }
    .....
}
```

(上接第 90 页)

- [2] WfMC. Workflow management coalition terminology and glossary [EB/OL]. 1999. http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf.
- [3] VanderAast WMP, Hee K. 工作流管理—模型、方法和系统 [M]. 王建民, 闻立杰等译. 北京: 清华大学出版社, 2004.

4 结束语

通过实践证明, 整合框架可以提高系统代码的可重用性和控制层的灵活性, 大大提高了系统的稳定性、可扩展性和可维护性。文中的创新点在于: 把基于 MVC 模式的 Struts 框架和 Hibernate 框架灵活地应用到了一个在线作业管理系统, 并深入分析了系统涉及的关键技术部分 Struts-config.xml 配置文件及 Hibernate 框架的数据封装和 O/R 映射技术, 深入分析了该框架的核心配置文件 hibernate.cfg.xml。同时详细介绍了 Struts 框架的详细工作流程。后续研究目标是实现基于 Struts 和 Hibernate 整合框架的事务管理机制并提出一种合理的安全控制措施。

参考文献:

- [1] 李刚. Struts 2 权威指南: 基于 WebWork 核心的 MVC 开发 [M]. 北京: 电子工业出版社, 2007.
- [2] 计磊, 李里, 周伟. 精通 J2EE-Eclipse. Struts. Hibernate. Spring 整合应用案例 [M]. 北京: 人民邮电出版社, 2006.
- [3] 王嘉. Ajax 经典案例开发大全 [M]. 北京: 人民邮电出版社, 2007.
- [4] Malinescue F. EJB Design Patterns [M]. 北京: 机械工业出版社, 2004.
- [5] Allamaraju S, Buest C, Davies J. Professional Java Server Programming J2EE 1.3 Edition [M]. 北京: 机械工业出版社, 2002.
- [6] 郭荷清, 林拉, 张为. 现代软件工程 [M]. 广州: 华南理工大学出版社, 2004.
- [7] Deitel H M, Deitel P J, Santry S E. Advanced Java 2 Plat2 form How to Program [M]. 北京: 电子工业出版社, 2004.
- [8] 随永, 周家纪. MVC 在 J2EE 框架中的应用研究 [J]. 计算机技术与发展, 2006, 16(12): 119-121.
- [9] 吴寅斐, 何钦铭, 吴大珊. Struts 分析及其应用研究 [J]. 计算机工程, 2006, 31(16): 42-44.
- [10] 吴昊, 李志蜀, 杨正海. 基于 Struts 框架构建 GSM 电子运行维护系统 [J]. 计算机应用与软件, 2005, 22(5): 63-65.
- [11] 杨涛, 周志波, 凌力. 基于 Struts 和 Hibernate 的 J2EE 快速开发框架的设计与实现 [J]. 计算机工程, 2006, 32(10): 82-85.

- [4] 王维平. 离散事件系统建模与仿真 [M]. 北京: 清华大学出版社, 1999.
- [5] 卢光松, 葛运建. 随机时间 Petri 网综述 [J]. 计算机科学, 2005, 32(3): 26-30.
- [6] 林闯, 田立勤, 魏丫丫. 工作流系统模型的性能等价分析 [J]. 软件学报, 2002, 13(8): 1472-1480.