

J2EE 中 SSH 架构的研究及其在 WEB 开发中的应用

周 岚

(芜湖信息技术职业学院 软件工程系, 安徽 芜湖 241003)

摘要: Struts、Spring 和 Hibernate 是目前 J2EE 中流行的三种框架, SSH 框架的使用可以有效提高软件开发效率, 简化程序代码, 实现用户界面、业务处理和数据库访问的分离。文中通过具体实例详细分析了 SSH 框架的设计和实现。

关键词: Struts; Spring; Hibernate; 框架

中图分类号: TP311.1 **文献标识码:** A **文章编号:** 1671-8275(2011)03-0120-03

随着社会信息化和网络化的飞速发展,越来越多的企业使用基于 J2EE 的开发平台来进行 WEB 应用开发。用 J2EE 开发一个 WEB 应用系统时,需要考虑如何建立用户接口,在哪里处理业务逻辑和实现持久化,以及如何实现程序的松散耦合等问题。我们可以通过使用目前业界流行的 Struts、Spring 和 Hibernate 三种框架来实现程序的松散耦合,达到代码的合理分离,从而提高项目代码的复用性、可扩展性、可移植性和可维护性,提高系统的运行效率。SSH 架构中 Struts 用来实现表现层, Spring 实现业务逻辑层, Hibernate 则实现数据持久层。

1 J2EE 中 SSH 架构的研究

1.1 Struts 框架技术

Struts 是现今 WEB 开发技术中运用最广泛的开源框架,可以实现表现层,主要用于管理用户请求,调用业务层逻辑和处理异常。Struts 框架具有组件的模块化、灵活性和重用性的优点。Struts 可以更清晰的把模型、控制器和视图分开,简化了基于 MVC 的 WEB 应用程序的开发,使页面更加简洁,便于后期的维护和升级。

Struts 的核心是一个 MVC 风格的控制器,负责管理请求和响应。Struts 控制器搭起了 Model 和 View 之间的桥梁, Struts 的自定义标签十分有用,并能与 Struts 本身很好的融合。框架也可以包括开发人员想用来开发可伸缩的、先进的应用的其他组件。Struts 是一个“隐藏支柱”的集合,可以帮助开发人员将分散的材料(如数据库和页面)结合成一个整体的应用程序。

1.2 Spring 框架技术

Spring 是用在业务层的框架。Spring 是一个解决了许多在 J2EE 开发中常见问题的强大开源框架,可以处理应用程序的业务逻辑和业务校验,管理程序的执行。它提供

了管理业务对象的一致方法并且鼓励对接口编程而不是对类编程的良好习惯。Spring 提供了一个细致完整的 MVC 框架,为模型、视图和控制器之间提供了一个非常清晰的划分,各部分耦合度极低。Spring 的 MVC 是非常灵活的,它完全基于接口编程,真正实现了与视图无关。Spring 通过 IOC 依赖注入、AOP 应用、面向接口编程,来降低业务组件之间的耦合度,增强系统扩展性。

Spring 能简化企业级开发,大部分情况下可以用简单的 java bean 来代替实现复杂的 EJB。Spring 是一个轻量级的 IOC 和 AOP 框架,可以用 Spring 的 IOC 实现松耦合,而作为一个 AOP 框架它又能分离系统服务,实现内聚开发。Spring 是非侵入式,基于 spring 的系统可以不依赖于 Spring 的类。在程序设计中可以充分利用 Spring 依赖注入的特性,当修改类的属性时不用直接修改类的代码,而只在在配置文件里面进行简单的设置。

1.3 Hibernate 框架技术

Hibernate 主要是用于对数据库的一系列操作,可以实现持久层,完成持久化。Hibernate 是一个十分优秀的也是目前最流行的 ORM 框架,采用低侵入式的设计,用它来进行持久层的开发,可以减少持久层的工作量。相对于使用 JDBC 和 SQI 来手工操作数据库,使用 Hibernate 可以大大减少操作数据库的工作量。

采用 Hibernate 作为持久层技术的最大的好处在于可以完全以面向对象的方式进行系统分析和系统设计。Hibernate 完全采用 Java 对象,它既可以用来在 Java 应用程序中取代大部分 JDBC 代码,也可以整合到 J2EE 系统中作为持久层框架。Hibernate 可以和多种 Web 服务器或者应用服务器良好集成,支持几乎所有的流行的数据库服务器。作为一种流行的 ORM 映射工具, Hibernate 不仅提供了从 Java 类到数据表之间的映射,也提供了数据查询和恢

收稿日期:2011-02-10

基金项目:本文系安徽省青年优秀人才基金项目“面向 IT 产业需求如何培养高职软件技术人才”(项目编号:2009SQRZ224)阶段性研究成果。

作者简介:周岚(1978-),女,安徽绩溪人,芜湖信息技术职业学院软件工程系讲师,硕士,软件技术教研室主任。研究方向:软件技术。

复制制。

1.4 SSH 框架的整合

用 SSH 架构来开发基于 WEB 的应用系统的流程是：通过 JSP 界面来实现与用户的交互，负责发送用户的 Request 和接收 Response；Struts 根据 Struts-config.xml 文件中配置，实现业务处理或跳转切换等动作，把收到的请求发给 Action 处理；在持久层中由 Hibernate 实现数据库与持久对象的交互，更新数据库中的数据或获取数据库中的信息存放在持久对象中；在业务逻辑层中会连接表现层和数据层的中间层，Spring 向 Action 提供具体的业务逻辑实现，同时根据 Action 收到的请求，结合 Hibernate 提供的数据库对象，进行相应的业务处理。

2 SSH 架构在在线项目管理平台中的应用

2.1 在线项目管理平台软件的功能

在线项目管理平台软件是一个部署于 WEB 服务器上的 B/S 架构应用系统，该项目目标是搭建一个在线项目管理协作平台，方便项目负责人对于项目的规划、管理、任务分配和整体的把握，成员之间协作、沟通、分享资料和互相通知，以适应项目和团队的快速变化和远程协作。系统用户可以使用设定的帐号登录系统。系统提供项目管理、人员管理、任务列表和消息板功能。

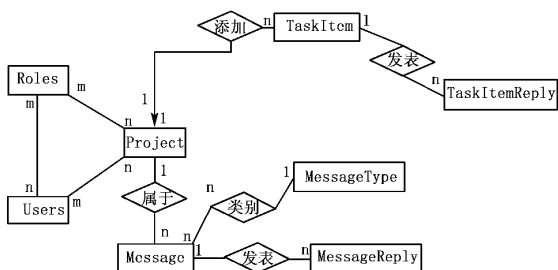


图 1 E-R 图模型设计

2.2 消息板模块

消息板模块主要是实现项目成员间交流和共享信息。成员可以发送消息，并选择通知的成员（是否 email 通知），添加消息评论，按类别查看消息。在线项目管理系统中有三种角色：项目经理（创建人）、技术带头人、项目成员。项目经理具有最大的权限，具有所有的功能。技术带头人和一般项目成员也具有消息板中的所有功能。如图 2：

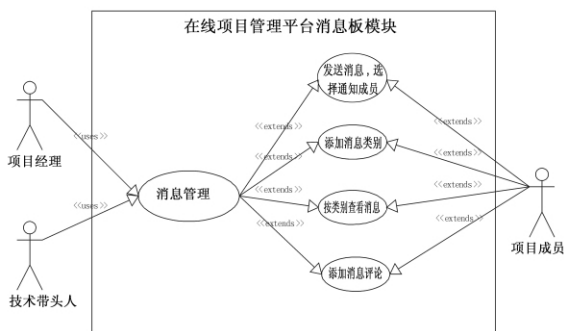


图 2 消息板模块用例图

2.3 SSH 架构在在线项目管理平台中消息板模块的应用

在整个项目中，表现层的实现是基于 Struts 架构。Struts 的核心是控制器组件 ActionServlet，在设计每个模块的控制器组件时，都会定义一个配置文件 Struts.xml 来配置业务流程。业务逻辑层的数据来源于数据持久层 Persistent Objects 和表示层 Formbean，是表现层和持久层进行数据通信的中间层。它负责封装持久层提供的数据库对象，并为表现层提供功能接口，由 Business 组件实现。数据持久层是基于 Hibernate 来实现的。实现过程中，需要创建配置文件、定义持久化类、对持久化对象进行映射定义和实现持久化操作。在系统中，采用 XML 进行定义数据库连接的各种参数以及供 Hibernate 管理事务、产生 SQL 和管理 JDBC 连接时所涉及的数据映射文件等，Hibernate 根据这些属性来连接数据库。

首先要进行消息接口的定义，消息接口的定义如下：

```
public interface MessageDao {
    public void save(Message message);
    public void update(Message message);
```

```
public Message getMessageById(int id);
public List<Message> getMessageByUserId(int userid);
public List<Message> getMessageByProjectId(int projectid);
public List<Message> getMessageByCondition(String hql);
}
```

(1) Struts 与 Spring 集成

对于 Struts 来说，所有的请求最终会被 Action 所接收到，通过 Action 去呼叫业务逻辑层的相应业务逻辑服务对象来处理客户请求。我们需要在 Action 里面去实例化某个具体的业务逻辑服务对象，通过 Spring 来实现二者的松耦合。

第一步，把 Struts2-spring-plugin 和 Spring 相应的 jar 包加到项目中。

第二步，在 web.xml 中配置 Spring 监听器。

```
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/classes/applicationContext.xml</param-value>
</context-param>
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
```

第三步，在 applicationContext.xml 中配置 Struts action。用 Spring IOC 注入引用的业务逻辑类实例。

第四步，在 struts.xml 中配置 action，它的 class 就不是一个类了，而是在 applicationContext.xml 中定义过的类的 ID。

```

<struts>
  <constant name=" struts. objectFactory" value = "
spring" />
  <package name=" onlineproject" extends = " struts -
default">
    <interceptors>
      <interceptor name=" authentication" class = " com.
dragon. online. util. AuthenticateInterceptor" />
      <interceptor - stack name=" authStack">
        <interceptor - ref name=" authentication" />
        <interceptor - ref name = " paramsPre-
pareParamsStack" />
      </interceptor - stack>
    </interceptors>
    <action name = " messagelist" class = " messageAc-
tion" method = " projectMessageList">
      <result name = " success " >/project/message/mes-
sage_list.jsp</result>
    </action>
  </package>
</struts>

```

(2) Spring 与 Hibernate 集成

在 Spring 框架中,只要在 Spring 的配置文件 applicationContext.xml 中配置 sessionFactory 的 bean, Spring 就可以和 Hibernate 联系起来。配置文件中的 class 属性必须是真正的实现类,而不是接口。

第一步,在 applicationContext.xml 中创建数据源。

第二步,在 applicationContext.xml 中创建 sessionFactory,并把相应的 Message.hbm.xml 包括进来。

第三步,修改 dao 包下的持久类实现,采用继承 HibernateDaoSupport 实现。

```

public class MessageHibernateDao implements HibernateD-
aoSupport implements MessageDao {
    public void setMessageType(MessageType message-
type) {
        this.messageType = messageType;
    }
}

```

第四步,在 applicationContext.xml 中配置 dao 类。

```

<bean name=" messageHibDao" class = " com. drag-
on. online. dao. impl. MessageHibernateDao">
  <property name=" sessionFactory" ref=" sessionFac-
tory" />
</bean>

```

第五步,服务层通过 Spring 注入引用的 dao 类实例。

```

public class MessageService {
    private MessageDao messageDao;
    public void setMessageDao ( MessageDao
messageDao) {
        this.messageDao = messageDao;

```

```

    }
    public void save(Message message) {
        this.messageDao. save(message);
    }
}

```

(3) 经过 SSH 整合后的 applicationContext.xml 文件:

```

<bean name=" messageService" class = " com. dragon. on-
line. service. MessageService">
  <property name=" messageDao" ref = " messageHib-
Dao" />
</bean>
<bean name = " messageAction" class = " com. dragon. on-
line. ui. action. MessageAction">
  <property name=" messageService" ref = " messageS-
ervice" />
  <property name = " projectService" ref = " projectSer-
vice" />
</bean>

```

3 小结

用 J2EE 开发一个 WEB 应用系统时,数据的流向是 ActionFormBean 接受用户的数据, Action 将数据从 ActionFormBean 中取出,封装成 VO 或 PO,再调用业务层的 Bean 类,完成各种业务处理后再 forward。而业务层 Bean 收到这个 PO 对象之后,会调用 DAO 接口方法,进行持久化操作。

在 SSH 架构中, Struts 作为前台控制框架简化了程序的开发,使得界面开发人员和 Java 程序员实现了有效的分离,增强了项目的可扩展性,提升了开发效率,降低了开发成本。Spring 则通过控制反转,层层回调,极大地减少了业务逻辑代码。而 Hibernate 则极大地减少数据库相关操作代码,实现一个简单高效的持久化。

SSH 框架的使用有效地缩短了系统开发周期,提高软件开发效率,简化了程序代码,使得程序的层次结构更清晰,提高了编程效率,实现了用户界面、业务处理和数据库访问的分离。

参考文献:

- [1] 谢辛. J2EE 轻量级 SSH 架构无缝集成的研究[J]. 科技信息, 2009(6).
- [2] 郭宏杰, 朱荣. 基于 SSH 的物流信息系统的研究与实现[J]. 计算机与数字工程, 2009(2).
- [3] 王美琴. 基于 J2EE 的轻量级 SSH 架构整合研究[J]. 电脑知识与技术, 2009(21).
- [4] 薄奇, 许林英. Spring 框架中 IoC 的实现[J]. 微处理机, 2008(29).
- [5] 刘乃丽. 精通 JavaEE 项目案例—基于 Eclipse SpringStruts Hibernate [M]. 北京: 人民邮电出版社, 2008.

责任编辑: 孟云玲