

基于 SSH 框架的钢铁交易平台的设计与实现

周庞荣

(湘潭大学 信息工程学院, 湖南 湘潭 411105;
湖南铁路科技职业技术学院, 湖南 株洲 412000)

摘要:为高效、安全构建钢铁现货电子交易系统,文中研究整合了目前流行的 Struts、Spring 和 Hibernate (SSH) 框架技术构建中联钢铁现货电子交易平台。在系统的设计实现中,首先根据钢铁现货交易系统业务流程,采用 SSH 框架技术对系统进行了分层设计,构建了系统的总体框架设计图,然后以合同管理模块中的合同生成为例,详细介绍基于 SSH 框架技术的系统的实现。通过使用 SSH 框架技术,极大程度上提高了系统代码的可重用性和可移植性,并保障了系统安全性,对于其他行业电子交易平台系统实现也有很好的借鉴性。

关键词:钢铁;电子交易平台;框架

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2013)05-0250-04

doi:10.3969/j.issn.1673-629X.2013.05.065

Design and Implementation of Steel E-commerce Platform Based on SSH

ZHOU Pang-rong

(College of Information Engineering, Xiangtan University, Xiangtan 411105, China;
Hunan Railway College of Science & Technology, Zhuzhou 412000, China)

Abstract: In order to build steel stock electronic trading system efficiently and safely, studied the electronic trading platform based on the currently popular Struts, Spring and Hibernate (SSH) framework technology. In the design and implementation of the system, according to the processes of steel spot trading systems, at first, use the technique of SSH framework for hierarchical design system, build the overall framework of the system design, then take the contract generation in contract management module as an example, introduce the achievement based on SSH framework in detail. By using SSH framework, improve the system code's reusability and portability with great extent, protect the system security, and also provide a good reference for other industry electronic trading platform system implementation.

Key words: steel; electronic trading platform; framework

0 引言

随着互联网的快速发展和钢铁贸易的繁荣,钢铁现货电子交易发展如火如荼,如何高效、安全构建钢铁现货电子交易系统,是目前研究的一个重点^[1]。

基于 MVC 模式的 Struts 技术、Spring 技术和 Hibernate 技术的 SSH 技术架构使系统的视图层、业务逻辑层和数据持久层实现了分离,极大程度上提高了代码的可重用性和可移植性,并保障了系统安全性,越来越多 Web 应用程序都采用 SSH 技术架构构建系统。文中介绍了采用 SSH 技术架构开发中联钢现货电子

交易平台^[2]。

1 相关框架介绍

1.1 Struts 框架

Struts 是一个基于 JSP Model2 的 MVC 的表示层框架。从其体系结构分析,它包括控制器、模型和视图三个部分。

(1) 视图:视图部分主要由 JSP 页面组成。在这些 JSP 页面中没有业务逻辑和模型信息,只有标记。标记可以是系统标准的 JSP 标签,也可以是用户自定义标签。

(2) 控制器:控制器由 Action 类和 ActionServlet 类来完成各项任务。Action 类担负调用模型的方法,更新模型的状态,并控制应用程序的流程。ActionServlet 类起着中央调度器的作用,它接收所有来自用户的请

收稿日期:2012-08-30;修回日期:2012-12-05

基金项目:国家自然科学基金资助项目(60773047);湖南省教育重点科研项目(06A074);湖南省科学计划项目(2011FJ3067)

作者简介:周庞荣(1974-),男,高级工程师,主要研究方向为软件工程。

求,并把请求分配给相关的 Action 类处理。在 Struts 中,有一个很重要的配置文件 Struts-con_fig. xml,该配置文件中描述了用户请求路径和 Action 映射关系的信息。

(3)模型:Struts 没有定义具体 Model 层的实现。在目前的商业领域和开源世界,有一些优秀的工具为 Model 层的开发提供支持^[3]。

1.2 Spring 框架

Spring 框架最大的优点就是提供了控制反转 (IoC)和面向方面编程 (AOP)的功能。通过这两大功能,该框架系统解决了 J2EE 开发中的很多问题。该框架能方便地把各种组件和框架整合进来,并形成一个连贯的整体构架。该框架中的 bean 容器为业务对象、DAO 对象和资源对象 (如 JDBC 数据源) 提供了 IoC 类型的装配能力,同时支持事务管理、远程调用等功能^[4~6]。

1.3 Hibernate 框架

Hibernate 框架为数据库提供了数据持久化服务,根据对象/关系映射 (Object/Relation Mapping, ORM) 技术思想,该框架对 JDBC 实施了封装,使开发人员可以采用面向对象的方式来操纵和管理关系数据库。Hibernate 核心组件包括 Configuration 接口、SessionFac-tion 接口、Session 接口、Transaction 接口、Query 接口和 Criteria 接口。通这些接口,不仅可以存储和获得持久对象并且能够进行事务控制^[7,8]。

2 系统分析与设计

2.1 系统功能分析

中联钢现货交易平台是中联钢电子交易市场系统中一个子系统,通过对系统工作过程的充分挖掘,确定子系统和其他系统之间的关系以及子系统内部业务流程,如图 1 所示。

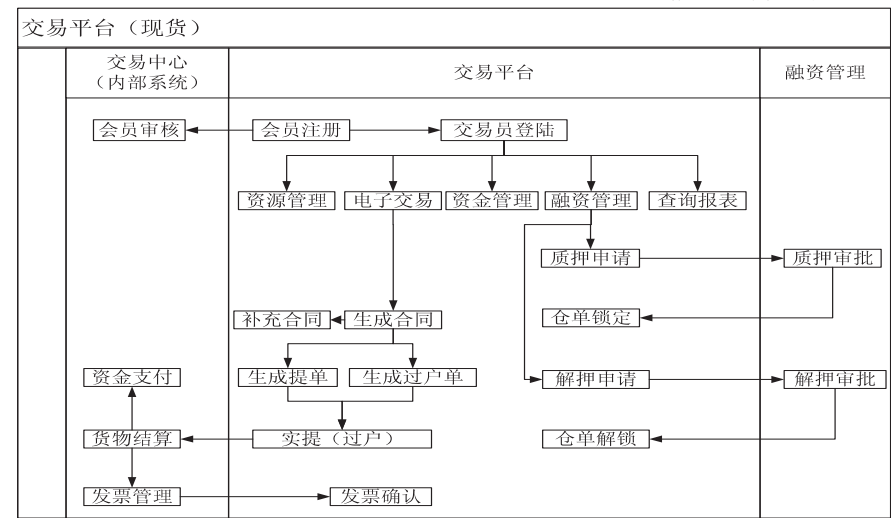


图 1 现货交易系统业务流程图

2.2 系统总体框架

依照 SSH (Struts、Spring、Hibernate) 框架对系统进行分层设计,设计的层次结构图如图 2 所示。

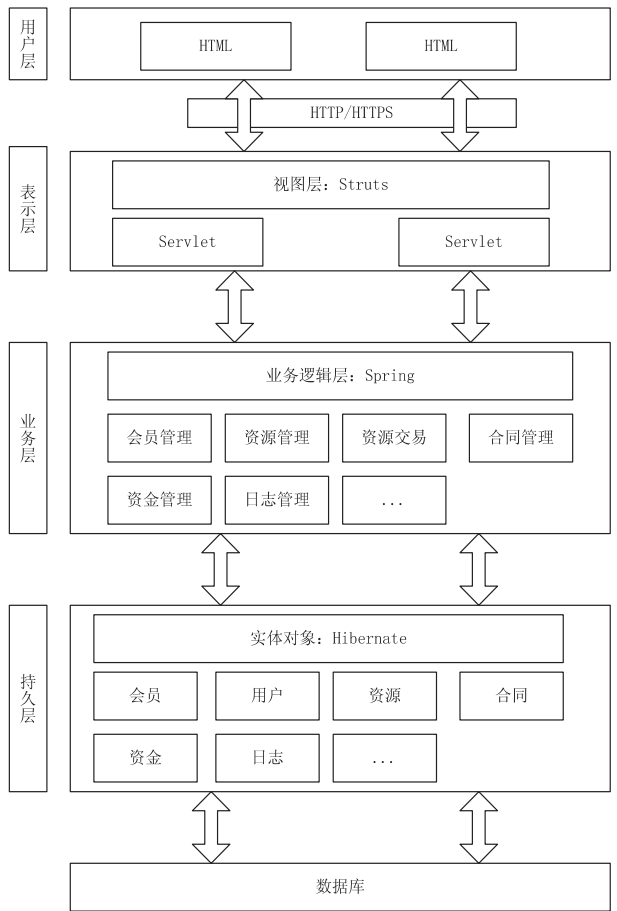


图 2 系统架构层次图

依据图 2,系统架构总体分为四个层次,由上到下依次为:

- (1)用户层:该层用于处理与用户的交互,在浏览器中运行。
- (2)表示逻辑层:该层从用户层接收用户的输入信息并传递给业务逻辑层,业务逻辑层调用相关的组件处理信息并把结果以 HTML 文件的形式传给用户端。表示逻辑层由 Struts 框架来构建,主要包含 Servlet 和 JSP 程序,在 J2EE 的 Web 容器中运行。

(3)业务逻辑层:该层处理从表示逻辑层传递过来的用户请求,处理完后将结果返回给表示逻辑层。业务逻辑层定义了各个子系统的核心逻辑实现,需要通过数据访问层来对数据库进行各种操作。

业务逻辑层是通过 Spring 容器来管理的。

(4)数据持久层:该层利用 Hibernate 将 Java 持久化对象映射到数据库表中,并实现对象的增加、删除、修改、查询等各种操作^[9]。

3 系统实现

以合同管理模块中的合同生成为例,详细介绍基于 SSH 框架系统的实现。

3.1 表示层实现

(1)JSP 页面。

合同生成模块所需的主要页面如表 1 所示。

表 1 合同生成模块的主要页面及描述

组件名	描述
mgpResourceList.jsp	卖挂牌资源查询页面
mgpResource.jsp	卖挂牌资源详细页面
orderEdit.jsp	订货购买页面
htOrderList.jsp	合同列表页面
htOrderDetail.jsp	合同详细页面

(2)Struts Form 表单。

用户通过 JSP 页面提交请求后,由 FormBean 组件封装表单数据。FormBean 中的属性和表单中的字段一一对应。具体代码框架如下:

```
/* *
 * 合同信息 Form 类
 * @ since version1.0
 */
package com.mysteelsoft.etrade.zymgp.web;
import com.mysteelsoft.common.web.BaseForm;
public class MgpOrderSaveForm extends BaseForm {
private String id;
private Stringsorderid;
.....;
//get 方法
//set 方法
}
```

(3)Struts Action 处理类。

控制器 ActionServlet 根据系统的配置文件将表单数据交给 Action 组件。在 Action 组件的方法中,首先获得表单数据,然后用表单数据构造需要持久化的值对象(Value Object),并通过 Spring 提供的工具类,从 Spring 的工厂中取得需要用到的业务逻辑对象进行调用。以保存合同信息为例加以说明,其实现代码框架如下:

```
/* *
 * 合同保存
 * 从请求中取得用户购买的资源信息和购买量等信息,并进行相关数据校验后,调用业务层生成合同
 */
public ActionForward doSaveOrder ( ActionMapping mapping,
```

ActionForm form, HttpServletRequest request, HttpServletResponse response) throws Exception

```
{
.....
}
```

(4)Struts 配置文件。

```
<struts-config>
<! -- ===== Form Bean Definitions =====>
<form-beans>
<! -- 合同生成 -->
<form-bean name="mgpOrderGenerateForm" type="com.mysteelsoft.etrade.zymgp.web.MgpOrderGenerateForm" />
.....
<! -- ===== Action Mapping =====>
<action-mappings>
<! -- 合同生成 -->
<action path="/zymgp/mgpOrderGenerate"
name="mgpOrderGenerateForm"
type="com.mysteelsoft.etrade.zymgp.web.MgpOrderGenerateAction"
parameter="action" validate="false">
<forward name="toEdit" path="/zymgp/orderEdit.jsp" />
</action>
.....
</action-mappings>
</struts-config>
```

3.2 业务逻辑层实现

在业务逻辑层,采用面向接口编程,所有自己开发的类都应该有对应的接口,并且自己开发的接口应该继承 CommonService 接口,CommonService 声明了通用业务方法。在该系统中,所有业务层的类都应该继承 CommonServiceImpl 类,它实现了 CommonService 接口,CommonServiceImpl 类实现了调用相应的数据访问对象进行业务逻辑操作,是表现层与数据持久层进行通信的中介。

业务逻辑层的数据来源于持久层的持久化对象(PO)和表现层的 FormBean。

(1)合同生成业务层接口。

```
/* *
 * 业务层合同生成接口
 * @ since version1.0
 */
public interface MgpIOrderBO extends CommonService{

/* *
 * 保存合同信息到数据库
 */
public String saveOrder(MgpOrderSaveForm form) throws ServiceException,
```

```
ParseException;  
}  
  
(2) 合同生成业务层实现类。  
public class MgpOrderBO extends CommonServiceImpl implements MgpIOrderBO {  
    /* *  
    * 主要是用于调用持久层,生成合同信息  
    */  
  
    public synchronized Strings saveOrder(MgpOrderSaveForm form)  
        throws ServiceException,  
        ParseException {  
        HtOrder order = null;  
        try {  
            .....//尝试生成合同  
  
        } catch (DAOException ex) {  
            // 如果合同生成失败,则将所有的信息还原到初始状态  
            .....  
        }  
        .....  
    }  
}
```

(3) Spring 配置文件。

```
<bean id="mgpOrderServiceTarget" class="com.mysteelsoft.  
etrade.zymgp.bo.MgpOrderBO" singleton="false">  
    <property name="dao"><ref bean="commonDAO"></ref>  
</property>  
</bean>  
.....
```

3.3 数据持久层实现

在数据持久层,主要采用了 DAO 模式进行数据源的封装。下面以生成合同为例说明 DAO 模式的实现和 Hibernate 的配置方法。

(1) DAO 实现。

OracleCommonDAO 是通用的数据访问对象的实现类,它实现了大量的数据访问和操作方法,一般情况下用户可以直接使用即可,如有特殊要求,可以继承该类,以扩展自己的功能^[10]。

```
/* *  
* 根据一个 id 从数据库装载一个持久层对象  
*/  
  
public Object get(Serializable id,boolean lock) throws DAOException {  
    .....  
}  
/* *  
* 根据 HQL 语句,去数据库查询一条记录出来,并返回记录对应的持久层对象,  
* @ param whereHql  
* @ param values 查询语句中的条件值  
* @ return Object
```

```
* @ throws ServiceException  
*/  
  
public Object loadByWhere(String whereHql,String[] values)  
    throws DAOException {  
        .....  
    }  
    .....  
}
```

(2) Spring+Hibernate 配置文件。

因为是 Spring 与 Hibernate 集成环境,所以 Hibernate 原来的 xml 配置文件可以转交给 Spring 来处理,以下同在 Spring 配置文件中配置 Hibernate 需要的数据源的代码片段如下:

```
<bean id="dataSource" class="org.apache.commons.dbcp.  
BasicDataSource" destroy-method="close">  
    .....  
</bean>  
  
<bean id="sessionFactory" class="org.springframework.orm.  
hibernate3.LocalSessionFactoryBean" singleton="true">  
    <!-- 合同管理 -->  
    .....  
</bean>
```

(3) 持久化对象和配置文件。

持久化类为每个属性定义了 getter 和 setter,并实现了 Serializable 接口。

```
public class HtOrder implements Serializable {  
    private String id;  
    private String sorderid;  
    private String itype;  
    .....;  
    //省略 get/set 方法  
}
```

配置文件 HtOrder.hbm.xml 文件确定了持久化类 HtOrder 和数据库表之间的映射关系。具体实现如下:

```
<hibernate-mapping>  
    <class name="com.mysteelsoft.po.ht.HtOrder" table="HT_  
ORDER">  
        <id column="SGUID" name="id" type="string">  
            <generator class="guid"/>  
        </id>  
        <property column="SORDERID" length="30" name="sorderid" not-null="true" type="string"/>  
        .....  
    </class>  
</hibernate-mapping>
```

4 结束语

通过建立钢铁现货电子交易平台,钢铁企业不仅
(下转第 257 页)

帧中的地址是变化的。

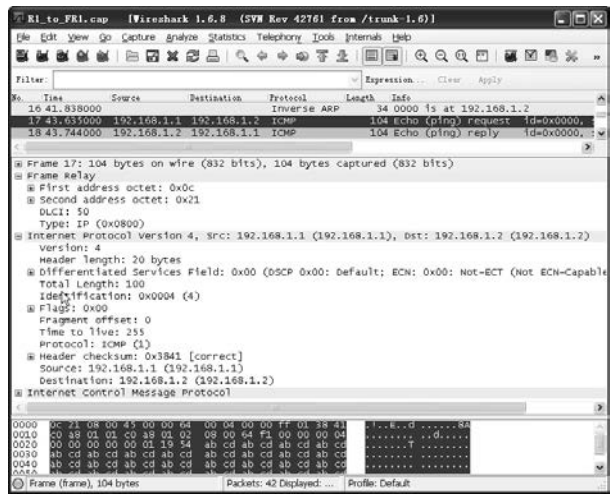


图3 R1-FR1 上的 ICMP 请求报文

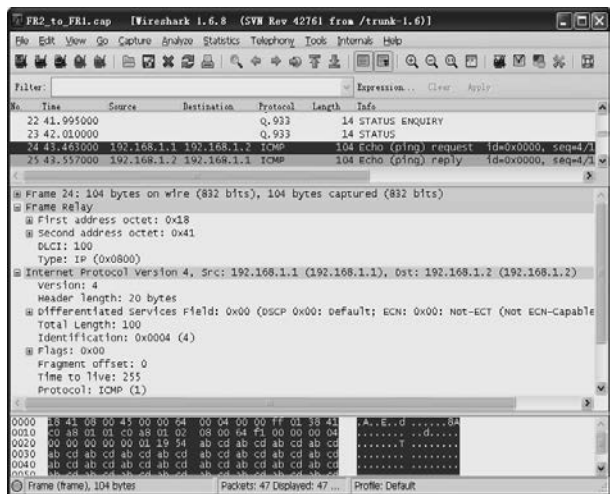


图4 FR1-FR2 上的同一个 ICMP 请求报文

3 结束语

通过在 GNS3 模拟软件中设计并配置帧中继协议

仿真实验,采用协议分析的方法对帧中继数据帧结构、LMI 协议和 InARP 协议工作过程进行分析说明,加深了对理论知识理解,提高了理论联系实际的能力,对其他网络协议的研究和学习有一定的参考价值。

参考文献:

[1] Burns K. TCP/IP Analysis and Troubleshooting Toolkit[M]. Indianapolis:Wiley Publishing, Inc. ,2003.

[2] Chappell L A, Tittel E. TCP/IP 协议原理与应用 [M]. 马海军, 吴 华, 译. 北京:清华大学出版社,2005:18-31.

[3] 周 斌,霍严梅,周春光,等. 网络协议分析实验课的教学改革[J]. 实验室研究与探索,2006,25(1) :56-59.

[4] Orebaugh A, Ramirez G, Burke J, et al. Wireshark & Ethereal Network Protocol Analyzer Toolkit [M]. Rockland: Syngress Publishing, Inc. ,2007.

[5] 潘文婵,章 韵. Wireshark 在 TCP/IP 网络协议教学中的应用[J]. 计算机教育,2010(6) :158-160.

[6] GNS3 OnLine Help[EB/OL]. 2012. <http://www.gns3.net/documentation>.

[7] 梁发洵. GNS3 在网络实验中的应用[J]. 电脑与电信,2010(10) :45-46.

[8] Atkins J, Norris M. Total Area Networking: ATM, IP, Frame Relay and SMDS Explained [M]. 2nd ed. Chichester: John Wiley & Sons Ltd,1999:43-62.

[9] 黄声烈. 帧中继实验的研究[J]. 实验技术与管理,2009,26(12) :47-49.

[10] Solie K. CCIE 实验指南(第一卷)[M]. 李 津,卓 林,译. 北京:人民邮电出版社,2002:261-294.

[11] 曹雪峰. 计算机网络配置、管理与应用[M]. 北京:机械工业出版社,2010:235-246.

[12] Bradley T, Brown C, Malis A G. Multiprotocol Interconnect over Frame Relay[S]. IETF RFC 1490, 1993.

[13] Bradley T, Brown C. Inverse Address Resolution Protocol[S]. IETF RFC 1293, 1992.

(上接第 253 页)

可以扩展销售渠道,提高企业的知名度,还可以获得高效率、高收益,节省企业的交易成本,增强企业的竞争力。该系统采用成熟的 SSH 框架技术设计,结构性好,灵活性强,对于其他行业电子交易平台系统也有很好的借鉴性。

参考文献:

[1] 杨 军. 钢铁物流电子商务系统的研究与应用[D]. 武汉:武汉理工大学,2010.

[2] 邹燕飞,罗鸿伟. 基于 Struts+Spring+Hibernate 缺陷管理系统实现[J]. 计算机技术与发展,2012,22(2) :146-148.

[3] 孙卫琴. 精通 Struts: 基于 MVC 的 Java Web 设计与开发 [M]. 北京:电子工业出版社,2004.

[4] Fowler M. Inversion of Control Containers and the Dependency

Injection Pattern [EB/OL]. 2003-04. <http://www.martinfowler.com/articles/injection.html>.

[5] Johnsonet R. Professional Java Development with the Spring Framework[M]. [s.l.]:Wrox,2005.

[6] 计文柯. Spring 技术内幕-深入解析 Spring 架构与设计原理[M]. 北京:机械工业出版社,2010.

[7] Kruszelnicki J. Persist data with Java Data Objects, Part I[J]. Javaworld,2002(3) :20-25.

[8] 孙卫琴. 精通 Hibernate:Java 对象持久化技术详解[M]. 第 2 版. 北京:电子工业出版社,2010.

[9] 孟 晨,赵春亮,张建国. 泛型 DAO 模式在 Java Web 开发中的应用[J]. 计算机应用与软件,2012,29(1) :175-177.

[10] 李淑芳,胡克寒,张凤丽,等. 基于 SSH2 框架的高校科研网络管理系统的实现[J]. 计算机应用与软件,2010,27(7) :195-196.