

企业网上年检系统^①

付学敬, 张 熔

(上海长安信息技术咨询开发中心, 上海 200032)

摘 要: 针对企业网上年检系统的建设和应用要求, 提出了基于 J2EE 技术平台和分层设计的思想, 采用业界成熟的 SSH (Struts+Spring+Hibernate) 应用开发框架完成系统设计与实现的方法, 对系统的表示层、业务逻辑层和数据持久层的主要功能设计做了阐述, 对基于 PKI 技术的应用安全设计以及系统实施中的应用服务器配置做了概要说明。基于 J2EE 技术平台的 SSH 应用开发框架的应用, 提高了系统开发的效率, 增强了系统的可靠性和可维护性, 系统开发和应用达到了预期效果。

关键词: J2EE; Struts; Spring; Hibernate; PKI

Enterprise Online Annual Inspection System

FU Xue-Jing, ZHANG Rong

(Shanghai Chang'an Information Technology Consultation & Development Center, Shanghai 200032, China)

Abstract: Aiming at the demands of construction and application of Enterprise Online Annual Inspection System, solutions based on J2EE platform and layered-design methodology is presented along with the detailed design and implementation process with a well-rounded application development framework set of SSH(Struts+Spring+Hibernate), including presentation layer, business layer and data persistence layer, then the brief introduction of the application security design is given based on PKI as well as the configuration of application server. The application of SSH framework architecture based on J2EE improves the development efficiency and enhances the liability and maintainability, therefore acquires favorable effect in the system's development and application.

Key words: J2EE; struts; spring; hibernate; PKI

企业网上年检系统是工商行政管理部门依托互联网为企业申报和接受年检提供便捷服务、为企业登记事项监管提供有力支撑的重要工作平台。建立企业网上年检系统, 实现企业年检的网上申报和网上审查, 有助于降低企业商务成本, 提高政府行政效率。

1 基本需求分析

企业年检申报、工商年检审查、年检资料归档是企业网上年检系统的三个主要功能。

(1) 企业年检申报: 参加网上年检的企业通过营业执照电子副本(含 CA 数字证书)进行身份认证, 然后填写相应类型的年检报告书(主要包括生产经营

情况及其它相关信息); 选择提交后, 系统对相关数据进行校验, 通过后调用企业的数字证书对所填写的年检报告书内容进行签名并保存, 同时提供经工商机关 CA 数字证书签名的《网上年检申报凭证》。

(2) 工商年检审查: 工商年检审查人员选择待检企业, 系统显示该企业提交的年检报告书和系统产生的年检信誉档案, 根据有关法规作出年检审查意见。

(3) 年检资料归档: 网上年检流程结束后, 系统对审查结果符合规定的网上年检申报书和相关文书凭证进行归档, 以 JPG 文件的形式生成电子文书档案。

在系统的非功能需求方面, 由于系统用户分布广泛, 访问时间集中, 且处于互联网的开放环境中, 这

^① 收稿时间:2011-11-12;收到修改稿时间:2011-12-14

对系统的安全性、可靠性、易用性、可扩展能力等都提出了较高要求。这需要通过系统的总体架构设计、应用设计、相关产品选型等予以综合平衡实现。

2 系统设计与实现

2.1 主要技术应用

(1) 多层体系结构: 分层系统架构在系统设计时支持逐层抽象, 可扩展性强并支持软件复用^[1]。

(2) J2EE 技术平台: J2EE 技术平台的开放性和标准化使得其产品选择更加丰富, 应用服务器组件的互操作性强, 系统可扩展性高, 适应需求变化能力强。

(3) 框架开发技术: J2EE 平台体系庞大, 结构复杂; 轻量级的 WEB 框架提供了预置的软件架构和软件包, 在保持 J2EE 优势的同时简化了相关开发工作^[2]。

2.2 总体架构设计

系统采用基于 J2EE 的多层分布式体系结构, 利用应用服务器的水平和垂直扩展能力以及数据库连接的管理功能, 有效提高应用逻辑的并发处理能力和数据库的访问性能。

系统的 WEB 应用分为表示层、业务逻辑层和数据持久层, 贯穿这三个层面的是业务对象。表示层用于数据的采集与展现, 数据采集的结果通过业务对象提交给业务逻辑层; 业务逻辑层负责对表示层提交的用户请求进行业务逻辑处理, 同时管理系统的核心资源; 数据持久层提供对数据访问的支持, 并对上层隐藏数据访问细节。

2.3 系统分层设计

系统采用基于 J2EE 平台的开源框架 Struts+Spring+Hibernate (简称 SSH) 进行分层设计^[3], 业务对象在各层间进行数据传递; 同时, 利用面向方面的程序设计(AOP)进行单独的异常处理、日志记录和权限控制。

以下以企业网上年检系统的年检申报处理为例, 对系统的分层设计做以阐述。

2.3.1 表示层设计

系统的表示层采用 Struts 1.2 框架以 MVC 设计模式进行构建, 实现系统的页面展示与数据采集, 并利用其控制器完成页面的请求和转发。具体来说, JSP 对应于视图, 系统主要通过 JSP 与外界进行交互; Servlet 对应于控制类, 作为 JSP 与业务逻辑层的中间枢纽; 业务逻辑层负责数据业务的处理^[4]。

Struts 框架与 Spring 框架的集成建立在对动作 (Action) 控制权转交的基础上, 因此需要在 Struts 框架的配置文件 struts-config.xml 和 Spring 框架的配置文件 ApplicationContext.xml 中进行协同配置。

企业年检申报表单类 ApplyForm.java 是表示层的核心类, 作为表示层数据交互的基本载体, 与其它页面功能文件共同构成表示层的整体, 应用设计概要说明如下:

在 ApplyForm.java 类中, 通过函数 copyToBo 将企业年检采集信息、企业生产运营信息、企业违法违规情况和数字签名信息等复制到业务对象 alBasic 中用于后台处理:

```
public void copyToBo(AlBasic alBasic) throws
IllegalAccessException, InvocationTargetException {
    /** 年检采集信息 */
    AlGather alGather = (AlGather) gathers.
convertToBO();
    /** 企业生产运营信息 */
    AlRun alRun = (AlRun) runinfo.convertToBO();
    /** 年检企业违法情况 */
    ApplyUtil.copyFromVOToBO(alBasic.getIlglSet(
), allIlglList, etpsPlanCode, etpsId, planId);
    /**年检数字签名信息*/
    AlSign alSign = (AlSign) sign.convertToBO();
}
```

通过函数 copyFromBO 将 alBasic 相关信息转换到 BeanUtils 中用于数据展现:

```
Public void copyFromBO(AlBasic alBasic) throws
IllegalAccessException, InvocationTargetException {
    BeanUtils.copyProperties(this.basic, alBasic);
    AlGather alGather = alBasic.gather();
    AlRun alRun = alBasic.runInfo();
    AllIlglSub allIlglSub = alBasic.ilglSub();
    Set ilglSet = alBasic.getIlglSet();
    BeanUtils.copyProperties(this.sign, alSign);
}
```

在企业年检申报数据录入页面 annlMaterial.jsp 中, 建立年检申报表单对象 applyForm, 引入年检企业基本信息视图对象 alBasicVO 和年检企业签名信息视图对象 alSignVO, 分别用于各自的数据存取:

```
ApplyForm applyForm = (ApplyForm) request.
```

```
getAttribute("applyForm");
```

```
AlBasicVO alBasicVO = applyForm.getBasic();
```

```
AlSignVO alSignVO = applyForm.getSign();
```

引入页面校验规则文件用于页面级的数据校验, 如必填项检查、数据类型匹配、数字平衡关系等:

```
<script type="text/javascript" src="js/validator_1.1.0.js"></script>
```

定义年检申报表单对象的后台处理以存储年检申报信息:

```
<form name="applyForm" id="applyForm" action="apply.do?method=saveAllInfo" method="post" >
```

通过以上业务对象, 表示层年检相关信息就得到了有效地组织并将传递到业务层进行核心的业务逻辑处理。

在系统的表示层设计中, 采用 Ajax 技术的动态加载模式和内容分块模式^[5], 通过异步交互的方式为用户提供诸如企业关联信息输入、工商信用状况查询等 WEB 交互功能, 减少了冗余数据对服务器和网络带宽的压力, 以提高系统性能、改善用户体验。

2.3.2 业务逻辑层设计

系统的业务逻辑层采用 Spring 2.0 框架实现, 通过配置文件对相关的业务逻辑对象、数据访问对象、安全及事务控制服务进行统一的组织和管理, 实现与表示层框架和持久层框架的集成; 利用 Spring 的控制反转 (IOC) 和面向方面的程序设计 (AOP) 机制实现数据源的有效引入和声明式的事务控制, 提高系统的开发效率和代码质量。

Spring 的相关配置在应用环境配置文件 applicationContext.xml 中完成, 包括与 Struts 框架和 Hibernate 框架的集成; Spring 与 Struts 之间通过 Action 对象建立关联关系, 而与 Hibernate 之间则通过数据访问对象 (DAO) 建立关联关系。具体来说, 在 Spring 框架的配置文件 applicationContext.xml 中, 为表示层的企业年检申报 Action 类配置一个 bean 属性, 利用 Spring 的控制反转机制为该 Action 类注入业务逻辑实现类 applyService, 这样表示层的 Action 类就可以象使用自身方法一样来调用业务逻辑层定义的业务处理方法; 同时, Spring 对 Hibernate 进行了封装, 并引入模板 HibernateTemplate 整合事务操作, 通过对 SessionFactory 的引用, 即可利用 AOP 机制实现对持久层的操作。

企业年检申报处理业务通过定义 ApplyService 类

实现相关的业务逻辑处理功能。当表示层的 Action 需要时, 即可调用 ApplyService 类中的方法进行业务逻辑处理, 处理完成后将处理结果返送表示层; 将需要持久化的数据发送持久层, 或者把表示层需要的数据通过持久层读取出来后返送表示层。

作为业务逻辑实现类, applyService 通过对 HibernateDaoSupport 的继承实现数据访问对象 DAO。HibernateDaoSupport 是 Spring 框架的一个对象关系映射支持类, 它使用 setSessionFactory() 方法接受 Spring 配置文件 applicationContext.xml 中依赖注入的 SessionFactory 实例; 而 applyService 则使用 getSessionFactory() 方法获得该 SessionFactory。同时, HibernateDaoSupport 类提供 getHibernateTemplate() 方法返回一个 HibernateTemplate 实例, 通过该实例所提供的方法即可实现对数据的持久化操作。

企业年检申报业务逻辑处理类 (ApplyService.java) 应用设计概要说明如下:

引入 Spring 框架的日志记录和异常处理功能:

```
import org.apache.log4j.Logger;
```

```
import com.framework.core.exception. ServiceException;
```

建立并引入企业年检基本业务处理对象 AlBasic 和企业年检基本数据存取对象 AlBasicDAO:

```
import com.sabic.netannl.apply.bo.AlBasic;
```

```
import com.sabic.netannl.apply.dao.AlBasicDAO;
```

在企业年检申报业务处理类 public class ApplyService 中, 获取框架的日志记录器:

```
private static Logger logger = Logger. Get Logger (ApplyService.class);
```

建立核心业务处理功能 (以年检数据存储为例), 利用 Spring 框架的 AOP 功能设置日志记录和异常处理:

```
public void saveAlBasicAndRelatedSet(AlBasic alBasic) {
    try {
        alBasicDAO.update(alBasic);
    } catch (Exception ex) {
        logger.error(ex.getMessage());
        throw new ServiceException("年检数据保存出错!");
    }
}
```

```
}
```

2.3.3 数据持久层设计

系统的数据持久层采用 Hibernate 3.0 框架实现, 具体负责管理 Java 类到数据库表的映射并提供数据存取的方法; 通过和 Spring 框架的集成, 利用其控制反转 (IOC) 和面向方面的程序设计 (AOP) 机制实现数据库操作的事务管理, 利用数据访问对象 (DAO) 对数据源访问的抽象和封装实现数据访问和实体管理功能。

ORM(对象/关系映射)是 J2EE 数据持久层的一种经典解决方案, 程序员能以完全面向对象的思维来分析和设计系统, 并能支持多种不同的数据库平台^[6]。在系统的数据持久层设计中, 根据数据库表的结构生成普通的 Java 对象 POJO, 系统对数据库表的操作转换为对这些 POJO 对象的操作。当需要处理数据库表中的元素时, 系统将相应的 POJO 对象传给数据访问对象 (DAO) 进行业务逻辑处理, 最后 Hibernate 再通过封装的 JDBC 将上述操作同步到数据库表。每个数据库表和对应的 POJO 类都设置有一个映射文件; 系统运行时, Hibernate 将利用这些映射文件生成相应的 SQL 语句用于数据库操作。

在系统的年检申报基本业务类映射文件 AlBasic.hbm.xml 中, class name 标识数据库表对应的 POJO, table 标识对应的数据库表, id 标识数据库表的主键, property 标识数据库表中的其他字段和 POJO 对象中属性的对应关系。

数据存取对象类 AlBasicDAO.java 继承自 BaseDAO, BaseDAO 又继承自 HibernateTemplate, 用于对年检业务对象 AlBasic 中的数据进行持久化处理。

2.4 应用安全设计

系统基于 SSL 协议和 PKI/CA 实现业务双方的身份认证, 实现数据通信的机密性、完整性和业务行为的不可抵赖性。具体的应用安全机制如下^[7]:

通过 SSL 协议完成双向身份认证后, 企业用户端对年检申报数据原文进行计算处理, 形成数字摘要; 然后用企业私钥对数字摘要进行加密形成数字签名; 用随机生成的会话密钥 (对称密钥) 对年检申报数据和数字签名进行加密形成年检申报数据密文, 然后用工商机关的公钥对上述会话密钥进行加密形成数字信封, 最后将该数字信封和年检申报数据密文一同发送给工商机关。

工商机关收到企业用户传送的相关数据后, 用自

己的私钥打开数字信封, 还原出用于加密数据的会话密钥; 利用该会话密钥将年检申报数据密文恢复成年检申报数据原文和相应的数字签名; 用企业用户的公钥解密数字签名得到数字摘要, 再用同样的算法对年检申报数据原文计算生成新的数字摘要, 若两段数字摘要一致, 则可确认相关信息来自该企业用户, 并且在网上传输过程中没有被篡改。

2.5 应用服务器配置

Web 应用服务器能够为 Web 应用提供强大的运行时服务和相应的工具集。系统的应用服务器采用 IBM WebSphere ND V6.0, 具有良好的安全管理、事务监测、负载均衡能力, 能够提供集群服务、高速缓存、失效恢复和系统的动态配置, 提供了一系列的监控、分析与配置工具, 增强了系统的可管理性和灵活性。

基于系统安全性的考虑, 将 WEB 服务器 (IHS) 部署在 DMZ 区中, 以有效隔离恶意的攻击; 通过配置 SSL (Secure Sockets Layer) 模块, 实现 SSL 3.0 技术规范, 提高系统的安全性。

为提高系统性能, 将一般的静态页面放在 IHS 中以减轻应用服务器负担, 通过配置 AFPA (Adaptive Fast-Path Architecture) 模块实现内核级 TCP/IP 高速缓存, 通过 WebSphere Plug-in 实现应用服务器的负载均衡。对 Web 服务器、Java 虚拟机、传输通道 (TCP 传输通道、HTTP 传输通道、Web 容器传输通道)、数据源与连接池的相关参数进行优化以提升系统性能^[8]。

3 结语

本文对企业网上年检系统的业务需求做了简要分析, 针对系统的建设和应用要求, 提出了基于 J2EE 开发平台和分层设计的思想, 采用 SSH 应用开发框架完成系统的设计与实现的方法, 对系统的总体架构设计以及表示层、业务逻辑层和数据持久层的设计做了阐述, 并对基于 PKI/CA 的应用安全设计以及系统实施过程中应用服务器配置做了概要说明。

基于 J2EE 技术平台的 SSH 应用开发框架的使用, 提高了系统开发的效率, 增强了系统的可靠性和可维护性, 系统的开发达到了预期的效果。

参考文献

- 1 李龙澍. 软件体系结构风格综述. 安庆师范学院学报(自然

(下转第 159 页)

据,通过不同数据字典定制数据转换元数据能够实现不同数据库的数据迁移,具有通用性。(2)扩展性好:使用基于数据映射的数据转换元数据,不同数据模型之间的数据迁移过程中,数据映射模式繁多,当出现一种新的映射模式只需要编写独立的映射函数,在元模型中添加函数及参数类型信息,定制之后确定使用函数及参数值即可实现新映射的数据转换。(3)易维护,使用元数据驱动的方式实现 ETL 功能,维护工作大部分为修改元数据,使维护简化。

ETL 系统在“大庆油田公司数据中心建设项目”中得到了较好应用,数据中心数据模型构建完成后,需从 A2 数据库、勘探开发数据库、压裂曲线数据库、计划统计数据库、经营管理数据库中迁移业务数据,以保证数据中心的正常启动运行,该项目使用该系统完成了 A2 数据库 62 张数据表 1223 万条生产数据、勘探开发数据库 37 数据表 413 万条基础数据、压裂曲线数据库 16 张数据表 369 万条压裂数据、计划统计数据库 22 张数据表 160 万条统计数据、经营管理数据库 18 张数据表 35 万经营数据的数据迁移任务,应用证明系统具有较好的实用性和较高的应用价值,目前系统正准备在中石油推广。

(上接第 138 页)

- 科学版),2006,12(4):3.
- 2 陈尚松,杜旭英,俞欢军.基于 Struts+Hibernate+Spring 框架的毕业设计管理系统.计算机工程与设计,2008,29(15):54-57.
 - 3 陈刚.基于 SSH 的 J2EE 开发平台研究与应用[硕士学位论文].成都:四川师范大学,2007.
 - 4 陆荣幸,郁洲,阮永良,王志强.J2EE 平台上 MVC 设计模式的研究与实现.计算机应用研究,2003,20(3):145.

3 结语

本文重点研究了数据映射,对数据映射进行抽象,定义了映射函数与参数类型,并通过数据映射的分析,给出了基于数据映射的数据转换元数据的描述形式,并构建了通用的元数据定制元模型,技术与系统在实际应用中证明具有良好的通用性、适应性、可移植性、可扩展性,能够对数据迁移需求做出快速反应,减轻用户的负担。

参考文献

- 1 宋杰,王大玲,鲍玉斌.一种元数据驱动的 ETL 方法的研究.小型微型计算机系统,2007,28(12):2167-2173.
- 2 孙伟,张忠能.ETL 架构研究.微型电脑应用,2005,21(3):13-15.
- 3 胡晓鹏,李晓航,李岗.一种基于 xml 映射规则的数据迁移方法设计和实现.计算机应用,2005,25(8):1849-1852.
- 4 熊辉,刘彦峰,郭大庆.分布式异构数据库迁移系统的设计与实现.计算机工程,2008,34(4):57-59.
- 5 Zhao XF, Huang ZQ. A formal framework for reasoning of Meta-data based on CWM.The 25th International Conference on Conceptual Modeling. 2006.371-384.
- 5 周杨.AJAX 应用的典型设计模式.计算机系统应用,2011,20(1):128-132.
- 6 田珂,谢世波,方马.J2EE 数据持久层的解决方案.计算机工程,2003,29(22):93-95.
- 7 肖竟华,何洁.PKI 技术在网上报税系统中的应用.电脑与信息技术,2006,14(4):40-42.
- 8 李一鸣,张剑,李哲,黄鑫.WebSphere 性能问题的发现及其处理对策.电脑知识与技术,2009,5(6):93-94.