

Design and Implementation of Internal Bank System Scheme

Wenming Wang, Ran Qi

School of Computer, Beijing Institute of Technology, Beijing
Email: wenmingwang2004@yahoo.com.cn

Received: Apr. 29th, 2012; revised: May 17th, 2012; accepted: May 28th, 2012

Abstract: In view of present situation and problems of building internal bank system for the domestic and foreign enterprises, this paper has studied and analyzed the design idea based on Struts framework, designed a general scheme of enterprise internal bank management system, and realized the system functions. According to the functional requirement, we make full use of Struts framework characteristics and complete the system design efficiently, so the system has high logicity, rationality and developmental controllability. This scheme can serve as a reference for domestic and foreign enterprises.

Keywords: MVC-Model; JSP/Servlet; Struts Framework; Internal Bank; EIBS

企业内部银行系统方案的设计与实现

王文明, 齐 然

北京理工大学计算机学院, 北京
Email: wenmingwang2004@yahoo.com.cn

收稿日期: 2012 年 4 月 29 日; 修回日期: 2012 年 5 月 17 日; 录用日期: 2012 年 5 月 28 日

摘 要: 针对国内外企业构建内部银行系统的现状及问题, 研究和分析了基于 Struts 框架的设计理念, 设计了企业内部银行管理系统的通用方案, 实现了系统的功能。根据功能需求, 充分利用了 Struts 框架的特点, 高效地完成了系统设计, 因此系统具有较高的逻辑性、条理性及开发可控性。该方案可供国内外企业参考。

关键词: MVC 模式; JSP/Servlet; Struts 框架; 企业内部银行; EIBS

1. 引言

随着我国经济的高速发展, 各大商业银行之间的竞争日趋激烈。在激烈的竞争中, 谁能以更小的代价赢得更多的客户, 谁就能赢得最后的胜利。这就需要银行拥有更贴近用户的服务手段, 更全面的服务体系以及更低廉的运营成本。

企业内部银行定义: 企业内部银行是指引进商业银行的信贷与结算职能和方式于企业内部, 来充实和完善企业内部经济核算的办法。

企业内部银行的职能: 内部银行引进商业银行的信贷、结算、监督、调控、资讯反馈职能, 发挥计划、组织、协调作用, 并成为企业和下属单位经济往来

结算中心、信贷管理中心、货币资金的资讯反馈中心。

适用范围: 一般而言, 企业内部银行适用于具有较多责任中心的企事业单位。一般不推荐给小企业或责任实体少的企事业单位使用。

组建内部银行的原则和策略: 要与现行财务制度相衔接遵守企业财务通则、会计准则的基本要求, 与现行财务会计核算体制相结合, 发挥事前、事中控制作用。

企业内部银行系统是集客户信息服务与转账结算业务于一体的多功能, 高效率的现代化金融服务系统。可利用企业内部网络, 进行企业内部范围的金融交易。

2. 企业内部银行系统结构

2.1. 运行环境

本系统使用 Eclipse 3.0 平台开发上的 J2EE 应用系统, 采用了 MVC 模式进行模块化设计, Web 和 Application 服务器采用 Tomcat 5.0, 数据库采用 MYSQL。

前台系统的运行环境: windows98/XP/Me/2000 IE 5.0 以上。

2.2. 系统网络拓扑结构

本系统(图 1)采用 B/S(Browser/Server)架构。它是随着 Internet 技术的兴起, 对 C/S(Client/Server)结构的一种变化或者改进的结构。在这种结构下, 用户工作界面是通过 WWW 浏览器来实现, 极少部分事务逻辑在前端(Browser)实现, 但是主要事务逻辑在服务器端(Server)实现, 形成所谓三层(3-tier)结构。这样就大大简化了客户端电脑载荷, 减轻了系统维护与升级的成本和工作量, 降低了用户的总体成本(TCO)。以目前的技术看, 局域网建立 B/S 结构的网络应用, 并通过 Internet/Intranet 模式下数据库应用, 相对易于把握、成本也比较低。它是一次性到位的开发, 能实现不同的人员, 从不同的地点, 以不同的接入方式(比如 LAN, WAN, Internet/Intranet 等)访问和操作共同的数据库; 它能有效地保护数据平台和管理访问权限, 服务器数据库也很安全。

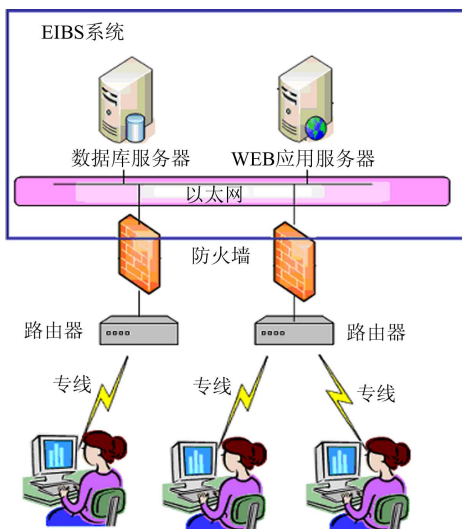


Figure 1. Network structure of the system
图 1. EIBS 系统网络拓扑结构

2.3. 通用系统业务流程描述

一般情况下, 通用系统业务流程如图 2 所示。

首先由系统管理员打开系统, 切换工作日; 其他业务操作人员可以通过用户名密码进入系统进行各项操作。包括账户管理, 操作员管理, 科目管理, 凭证管理, 企业活期账户管理; 在不同性质的科目下可开立企业活期分账户, 不同分账户之间可以进行内部结算交易, 在交易中会用到入库的凭证; 各类账户信息, 科目信息, 凭证信息, 交易情况都可以生成报表, 并且可以从系统中下载打印。

3. Struts 设计模式

3.1. Struts 简介

Struts 是一个基于 Sun J2EE 平台的 MVC(Model, View and Controller)框架, 主要是采用 Servlet 和 JSP 技术来实现的^[1]。由于 Struts 能充分满足应用开发的需求, 简单易用, 敏捷迅速, 在近几年中备受关注。Struts 把 Servlet、JSP、自定义标签和信息资源(message resources)整合到一个统一的框架中, 开发人员利用其进行开发时不用再自己编码实现全套 MVC 模式, 极大的节省了时间, 所以说 Struts 是一个非常不错的应用框架。

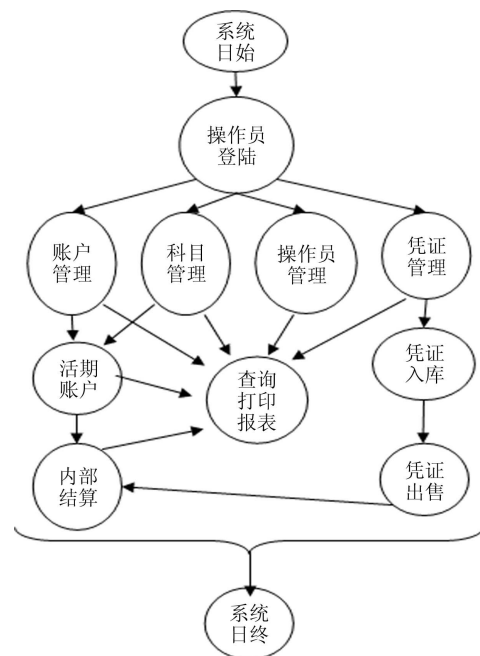


Figure 2. System business process
图 2. 系统业务流程

Struts 是一个比较好的 MVC 框架,提供了对开发 MVC 系统的底层支持,它采用的主要技术是 Servlet、JSP 和 custom tag library^[2]。作为一个 MVC 的框架,Struts 对 Model、View 和 Controller 都提供了对应的实现组件。

3.2. Struts 的实现——模型 - 视图 - 控制器(MVC)

Struts 是一个框架,而不是一个库,但 Struts 也包含了丰富的标记库和独立于该框架工作的实用程序类。图 3 是 Struts 的基本结构。

1) Client browser(客户浏览器)

来自客户浏览器的每个 HTTP 请求创建一个事件。Web 容器将用一个 HTTP 响应作出响应。

2) Controller(控制器)

控制器接收来自浏览器的请求,并决定将这个请求发往何处。就 Struts 而言,控制器是以 servlet 实现的一个命令设计模式。由 struts-config.xml 文件配置控制器。Controller 本质上是一个 Servlet,将客户端请求转发到相应的 Action 类。当客户端与服务器有交互动作时,都由 Controller 来控制。Controller 将 HTTP 请求封包并转发到框架中相应的对象,这些对象可能是一个 JSP 页面或一个 Action。

Controller 在 web.xml 中设置为 org.apache.Struts.action.ActionServlet 的一个实例。在一个完整的控制过程中,也就是处理一个 HTTP 请求时,在控制过程之初,这个 Servlet 会从一个配置文件 Struts-config.xml 中获取请求与控制动作向对应的配置信息,Controller 通过这些配置信息来决定 HTTP 请求该往何处转发,而这些 Action 在接收到转发来的请求后,实现真正的商业逻辑。映射转发用 return mapping.findForward("success");实现。

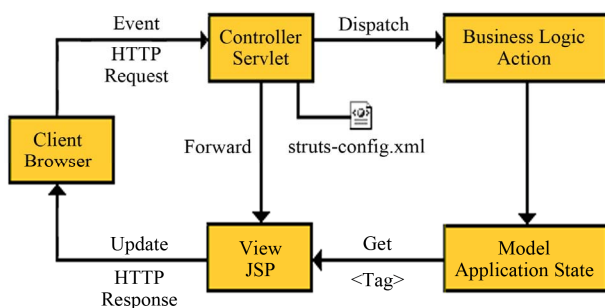


Figure 3. The basic structure of Struts
图 3. Struts 的基本结构

非常重要的一点是 Action 对象要能够调用这个 ActionServlet 的若干方法,当 Action 对象在控制过程中将请求再向别的 Action 对象转发时(最初的请求是由 ActionServlet 获取,向 Action 对象转发,而 Action 对象还可以再转发到别的对象),可以将一些需要共享的数据对象放入这个 Servlet 相关的一些标准容器(例如 session)中捎带过去。这些共享的数据对象通过 session 名称可直接用在其他 Action 或 JSP 页面上,非常方便。这种从其他 Action 取数据的 JSP 页面可以有效防止非法用户的强行页面访问,还可以在其他页面及时取到最新录入的数据。装入 session 的数据过一段时间会失效,此时必须重新调用 Action 再次激活 session。

3) 业务逻辑

业务逻辑更新模型的状态,并帮助控制应用程序的流程。就 Struts 而言,这是通过作为实际业务逻辑“瘦”包装的 Action 类完成的。

4) Model(模型)的状态

模型表示应用程序的状态。业务对象更新应用程序的状态。ActionForm bean 在会话级或请求级表示模型的状态,而不是在持久级。JSP 文件使用 JSP 标记读取来自 ActionForm bean 的信息^[3]。

Struts 中 Model 是一个 Action 类,开发者通过其实现商业逻辑,同时用户请求通过控制器(Controller)向 Action 的转发过程是基于由 Struts-config.xml 文件描述的配置信息的。

在对用户请求的整个控制过程中,真正处理用户请求并保存处理结果的对象,在整个过程中,利用 JavaBean 把一些信息保存起来以便在各个对象之间传递。通常的应用中,一般是一个 Model 对象和一个请求页面对应的关系,但也可以一个 Model 对象对应多个页面请求,本系统都是一对一关系。系统的 Struts-config-qi.xml 配置文件是整个框架的主心骨。web.xml 文件定义了一个请求到来应向何处转发后,后面的工作就全权由 Struts-config.xml 管理控制了。整个系统只使用一个这样的配置文件,保持整个系统的模块化和可维护性都非常的轻松。在应用系统有变动时,只用修改配置文件就行了,而不是再重新编译发布程序了。

5) View(视图)

视图就是一个 JSP 文件。其中没有流程逻辑，没有业务逻辑，也没有模型信息，只有标记。标记是使 Struts 有别于其他框架的因素之一^[4]。

本系统采用 Tiles 框架对 JSP 页面进行优化。在 Struts 中通过标签定义方式的 JSP 页面能够将用户通过表单输入的数据存入一个 JavaBean 中，也就是 ActionForm Bean。用户提交表单的数据的预处理能在 ActionForm 中完成。通过 ValidatorActionForm 类的校验方法来检查用户输入的数据，如果发现不合法的数据，再通过 Struts 的一个通用机制将错误信息返回给用户显示。

本系统的数据预处理由 validation-qi.xml 文件和 resource_zh.properties 文件进行控制，其中 validation-qi.xml 进行字段格式验证，resource_zh.properties 中定义中文提示。

4. 设计与实现

4.1. 系统功能架构设计

系统应用框架如图 4 所示。

4.2. 系统开发环境配置

本系统的开发要建立如图 5 所示的项目文件目录结构。

项目下的 EIBS 文件夹中结构如图 6 所示。

项目下 src 中存放编写的各种类文件；EIBS 中存放 JSP 文件、配置文件及编译文件；其中 WEB-INF 中的 web.xml、Struts-config-qi.xml 和 validator-qi.xml 属于重要的系统配置文件。

使用 Struts 之前，必先设置好 JSP 服务器，以便让服务器在用户请求时，知道该如何将指定后缀的请求转到相应的 Controller-Struts ActionServlet 处理。本系统服务器使用 Tomcat 5.0。相关配置信息都一般在服务器启动时通过 web.xml 文件读入的。所以需要在 web.xml 定义多个 Controller，为每一个应用定义一个。

4.3. 系统数据处理设计

4.3.1. 系统数据传输层次

数据处理层次如图 7 所示。

从客户端开始，在 Web 应用服务器启动的时候，

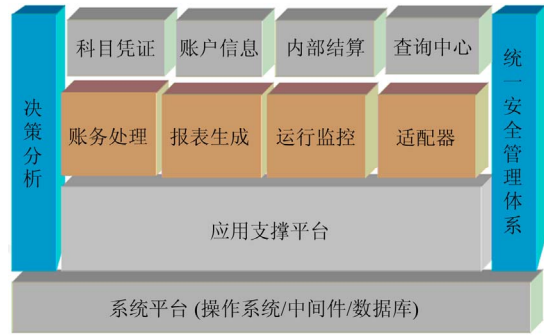


Figure 4. System application framework
图 4. 系统应用框架

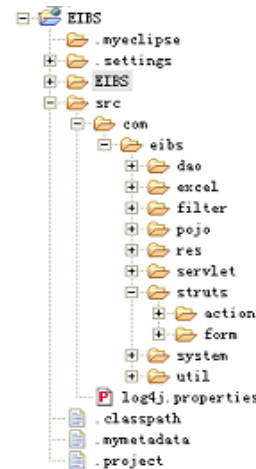


Figure 5. Project file directory
图 5. 项目文件目录

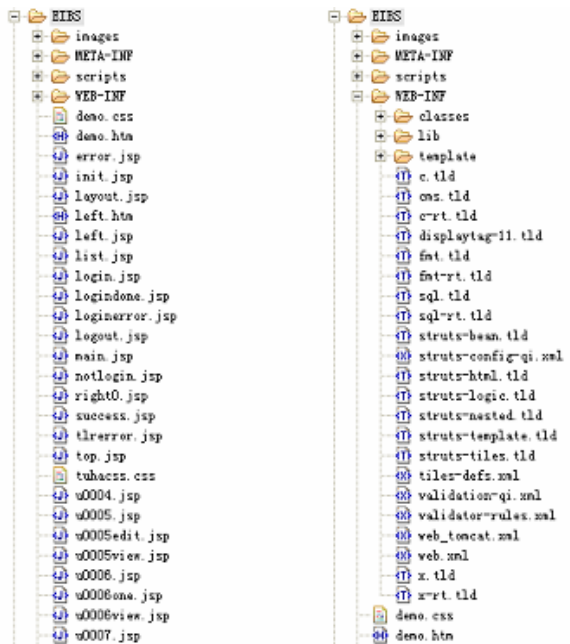


Figure 6. The EIBS folder structure
图 6. EIBS 文件夹结构

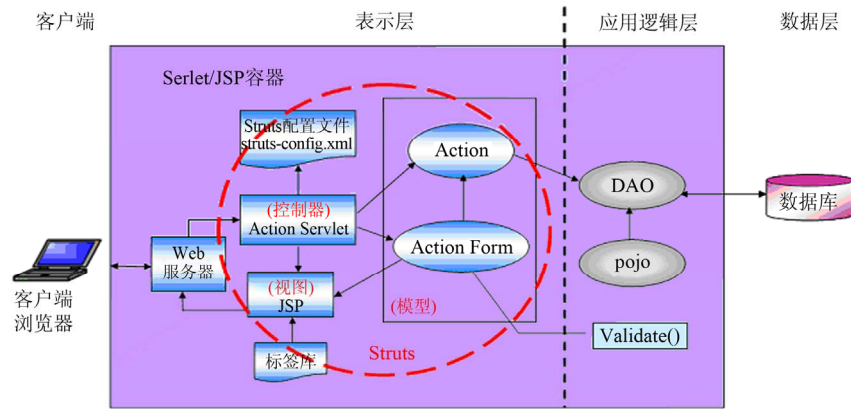


Figure 7. System development framework
图 7. 系统开发架构

系统会自动加载并初始化控制器中 ActionServlet 类，控制器接收到 HTTP 请求时，从 Struts 配置文件 struts-config.xml 中读取相关的配置信息，将 HTTP 请求转发给相应的 Action 对象，同时填充请求页面所对应的 ActionForm Bean，实现对用户请求的处理。Action 对象中定义了处理 HTTP 请求的逻辑流程。这些逻辑功能的实现独立于 Action 对象之外，在应用逻辑层，从而提高了组件的可重用性以及可扩展性。在 DAO 层中构造并执行访问数据库的 SQL 语句，通过应用逻辑层与数据库进行连接，将访问数据库的结果集以 DAO 方法调用的形式返回给调用的 Action 对象。在 pojo 中写入 getXXX 或 SetXXX 方法，然后由 DAO 自动对数据库的操作。在 DAO 中手工完成对数据库 JDBC 的各种调用，并且使用 Datasource 获得一个数据库资源以及连接，从而对数据库直接进行增加 删除修改。ActionForm Bean 由控制器组件负责创建，并提供给 Action 调用。它接收并临时保存用户提交的信息。可以将 ActionForm Bean 的属性设置成视图信息的默认值，也可以在 ActionForm Bean 中通过调用 validate()方法验证属性的合法性。Struts 提供的特殊 HTML 标签库保证了 ActionForm Bean 被正确地初始化并从页面的表单中获取数据。Action 对象内部处理完请求逻辑流程以后，控制器组件将处理返回的结果转交给视图组件。视图组件根据得到的结果，使用标签库中的信息生成相应的 JSP 页面，并以 HTTP 响应的方式返回给用户。

4.3.2. 数据库设计

1) 系统管理类

系统管理类表单如图 8 所示。

2) 基础数据类

基础数据类表单如图 9 所示。

3) 账户管理类

账户管理类表单如图 10 所示。

4) 交易管理类

交易管理类表单如图 11 所示。

5) 日志类

日志类表单如图 12 所示。

4.4. 数据源配置

Struts 中的数据源负责建立和数据库的连接，Java 语言提供了 javax.sql.DataSource 接口，所有的数据源都必须实现该接口^[5]。在应用程序访问数据库时不必编写连接数据库的代码，可以直接从数据源获得数据库连接。因此在使用数据源之前，只要修改 Struts 配置文件 Struts-config.xml 中的<data-source>元素即可。<data-source>元素，代码片段如下：

系统管理表 (SystemInfo)		菜单信息表 (MenuInfo)	
SysId	系统标识	MenuNo	菜单编号
SysName	系统名称	MenuLevel	菜单级别
LastDate	上一工作日	TxCode	交易代码
CurDate	当前工作日	TxName	交易名称
Status	系统状态	SupMenuNo	上级菜单
		MenuTxFlg	查询标志
序号控制表 (SeqCtrl)			
SeqType	序号类型		
BrNo	机构号		
SeqNo	序号		

Figure 8. System management form
图 8. 系统管理类表单

<p>客户信息表(CustomerInfo)</p> <p>CusmID 客户标识 CusmName 客户名称 CusmProp 客户性质 Principal 财务负责人 Phone 联系电话 Address 地址 Status 开立状态</p>	<p>操作员信息表(TellerInfo)</p> <p>TlrNo 操作员编号 TlrType 操作员种类 TlrName 姓名 StartDate 起始日期 EndDate 终止日期 IdNo 身份证号 Status 工作状态 OpenFlg 有无这人 Privileges 权限 BrNo 机构号 tlrPwd 操作员密码 menuinfos 菜单信息</p>
<p>科目信息表(AccountTitleInfo)</p> <p>ActlNo 科目号 SupActlNo 上级科目号 ActlName 科目名称 CurCode 币别 Property 科目性质 ActType 账户类型 BalDir 余额方向 ActlLevel 科目级别 Status 科目状态 ActSeqNo 账号序号</p>	<p>凭证种类表(VoucherTypeInfo)</p> <p>VocTypeCode 凭证种类代码 VocTypeName 凭证种类名称 IsImportant 是否重要凭证 IsManual 是否手工凭证 Status 状态 ShtNoPerBok 每本张数</p>

Figure 9. Basic data type form
图 9. 基础数据类表单

<p>分户账表(AccountInfo)</p> <p>ActlNo 账号 CusmID 客户标识 ActlNo 科目号 ActType 账户类型 Status 账户状态 CurCode 币别 Opentdate 开户日期</p>	<p>ClearDate 销户日期 EnterTlrAct 开户操作员 LstBal 账户余额 IntrFlg 记息标志 OvedFlg 透支标志 lactbal 录入时余额 BrNo 机构号</p>	<p>DepWftFlg 通兑标志</p>
---	--	-----------------------

Figure 10. Account management form
图 10. 账户管理类表单

<p>凭证入库登记表(VocRegInfo)</p> <p>BrNo 机构号 VocTypeCode 凭证类型 VocStartNo 凭证起始号码 VocEndNo 凭证终止号码 VocShtNum 张数 Status 状态 RegDate 入库日期</p>	<p>存放凭证出售的信息(VocSaleInfo)</p> <p>BrNo 机构号 VocTypeCode 凭证类型 ActlNo 账号 VocStartNo 凭证起始号码 VocEndNo 凭证终止号码 VocShtNum 张数 SaleDate 出售日期 Status 使用状态</p>
---	---

Figure 11. Transaction management form
图 11. 交易管理类表单

<p>结算交易日志表(SettlTransJour)</p> <p>TxsSeqNo 交易流水号 Txdate 交易日期 BrNo 交易机构 EnterTlr 录入操作员 BusiTypeCode 业务类型 CrdB 借贷标志 ActlNo 账号 ActlName 账号名称 ActlNo 科目号 ActType 账户类型 VocTypeCode 凭证种类代码 VocNo 凭证号 TxAmt 交易金额 TxStatus 交易状态</p>

Figure 12. Journal form
图 12. 日志类表单

```
<Context path="/EIBS" docBase="EIBS" debug="5" reloadable="true" crossContext="true">
  <Logger class=
    Name="org.apache.catalina.logger.FileLogger"
    prefix="localhost_eibs_log." suffix=".txt" timestamp="true"/>
  <Resource name="jdbc/eibs" auth="Container"
    type="javax.sql.DataSource"/>
  <ResourceParams name="jdbc/eibs">
    <parameter>
      .....
    </parameter>
  </ResourceParams>
</Context>
```

4.5. 系统运行实例

以本系统的一个典型模块(0004 账户信息增加)为例, 来说明基于 Struts 的解决方案的实现。登录页面如图 13 所示。

操作员输入正确的操作员号和相应密码即可进入系统。如图 14 所示。

接下来, 用户可以使用系统提供的各项功能, 完成企业内部银行各项任务。

5. 结论

论文研究的目的在于研究设计一个企业银行通用方案, 为企业提供方便、快捷、高效的实现手段。基于此, 本文从实际应用角度出发, 设计并实现了一个完整的企业内部银行管理系统, 交互机构为企业内部结算中心和不同的二级单位。本系统可设在一台服务器上, 供结算中心内部操作员在不同主机上访问。保额系统已经交付有关机构使用, 并且正在迅速推广。

本系统灵活性强, 可以在多方面进行扩展。首先允许创建多种类型登陆界面, 操作员可以刷卡进入系统也可以输入操作员号与密码进入。系统可加入对银行的接口, 进行各种银行与企业结算中心之间的交易, 例如: 外部结算交易, 银企对账等。企业内部管理模块也可进行扩展: 目前系统中账户管理模块只实现了企业活期账户管理, 可通过加入不同的账户类型进行更复杂的交易。内部结算模块中可扩展企业存款



Figure 13. System initial login page
图 13. 系统初始登录页面



Figure 14. The page after entering the system
图 14. 进入系统后的页面

与贷款的结算交易。还可以考虑加入预算模块，让系统更加实用，为企业带来更大的效益。本系统设定企业中只有一个结算中心，一个机构。若企业存在多个结算中心多个机构，系统可通过扩展多个新的机构号以满足需求。

6. 致谢

本文对参考文献作者以及项目合作单位北京理工大学，北京方正数码公司，新疆乌鲁木齐建设银行，北京工商大学等单位的支持与协作表示衷心地感谢。

参考文献 (References)

- [1] 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.
- [2] M. Hall, L. Brown. Servlet 与 JSP 核心编程(第 2 版)[M]. 北京: 清华大学出版社, 2004.
- [3] J. Falkner, B. Galbraith and R. Irani. JSP web 编程指南[M]. 北京: 电子工业出版社, 2002.
- [4] B. Eckel. Think in Java [M]. 北京: 机械工业出版社, 2003.
- [5] C. Cavaness. Programing Jakarta struts [M]. 北京: 中国电力出版社, 2005.