

## • 信息化技术 •

## 基于 JavaEE 的固定资产管理系统的设计与实现

张 俐<sup>1</sup>, 张维玺<sup>2</sup>

(1. 江苏技术师范学院 计算机科学与工程学院, 江苏 常州 213001;

2. 江苏技术师范学院 电气信息工程学院, 江苏 常州 213001)

**摘 要:** 结合企业固定资产管理信息化建设的实践和需求, 引入 JavaEE 架构体系, 提出了一种在 Spring 和 Struts 平台上开发企业固定资产管理系统的方法。对 Spring 和 Struts 架构下固定资产管理系统的技术实现进行了详细说明, 并给出了相关构造功能描述和部分核心代码及其说明。现场运行表明, 该技术的新特性提高了实际系统的安全性和稳定性。该架构不但使资源得到最大限度的节省和利用, 也使得项目开发简洁、结构清晰, 并且具备了更好的可扩展性和可维护性。

**关键词:** 固定资产管理; Spring 框架; Struts 框架; Java 数据库连接; façade 模式

**中图分类号:** TP311.12 **文献标识码:** A **文章编号:** 1000-7024 (2009) 16-3797-04

Design and implementation of fixed assets management system  
based on JavaEEZHANG Li<sup>1</sup>, ZHANG Wei-xi<sup>2</sup>

(1. College of Computer Science and Engineering, Jiangsu Teachers College of Technology, Changzhou 213001, China;

2. Department of Electronic and Information Engineering, Jiangsu Teachers College of Technology,  
Changzhou 213001, China)

**Abstract:** Based on the practice and requirement of the enterprise fixed assets management, JavaEE architecture is introduced, the web-based the enterprise fixed assets management is researched and implemented. Detailed the enterprise fixed assets management based on framework of Spring+Struts is given. At the same time, function descript of correlative constitution section and partial core code are also given. By practical application, it shows that new features of technique used in system improve security and stability of system. This framework technique not only make a good use of resources but also make the project developing to be compact and clear in flame. Besides good expansibility and maintainability also are provided by this kind of system.

**Key words:** fixed assets management; Spring; Struts; JDBC; façade model

## 0 引 言

固定资产是企业资产的重要组成部分, 在企业经营过程中扮演着越来越重要的角色。如何提高企业固定资产的使用率, 实行科学有效的动态管理, 成为企业固定资产管理工作的重点。虽然目前大部分企业的固定资产管理实现了一定程度的信息化, 但多数系统实现技术落后, 在实际应用中都或多或少地存在一些问题, 如数据共享性不高、系统安全性差、资产盘点工作繁琐等, 因此有必要引入新的技术手段来进一步实现企业固定资产管理的“标准化、信息化、网络化”。

本文采用 Spring+Struts 框架技术对企业固定资产管理系统进行了研究和应用。此方案解决了以下问题:

(1) 业务层和数据持久层的逻辑分离, 便于数据持久化操作的重用, 有利于数据持久化操作的重用和改动, 并进一步减

少重复代码, 同时达到了有利于代码的长期维护;

(2) 减少了网络负载, 使数据库底层结构信息不暴露, 实现了可移植性;

(3) 部署、开发和维护的有效性;

(4) 系统运行的健壮性和可靠性;

(5) 简化组件重用、直观的编程模型、支持行业标准和通用编程接口。

## 1 系统的分析与设计

## 1.1 固定资产管理业务流程

具体部门是固定资产的保管和使用部门, 负责本部门资产业务的申请提交和所使用的固定资产实物的管理; 资产管理科是固定资产管理的职能部门, 管理固定资产物、卡、实物台账, 负责资产的分类与编码; 会计科是固定资产的资金核算

收稿日期: 2008-08-25; 修订日期: 2009-03-13。

基金项目: 江苏省高校自然科学基金研究基金项目 (08KJD520005)。

作者简介: 张俐 (1977—), 男, 陕西汉中, 硕士, 讲师, 研究方向为软件工程、基于 JavaEE 的系统开发; 张维玺 (1952—), 男, 陕西南郑人, 教授, 硕士生导师, 研究方向为计算机理论。E-mail: zhangli\_3913@yahoo.com.cn

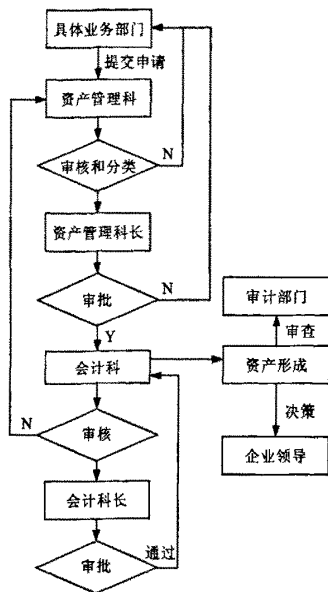


图1 固定资产管理业务流程

部门,负责固定资产财务凭证的制作,对固定资产的管理起审核和监督作用;各级领导负责固定资产业务的审批及监督,如图1所示。

## 1.2 系统功能设计

本系统是基于Web的企业固定资产管理系统。围绕固定资产的“进、出、用”各个环节进行科学管理。全面反映固定资产的增加、减少及相关变动情况,提供全生命周期跟踪管理。主要包括以下几个功能:

- (1) 资产管理包括:资产增加,资产台账管理,资产调拨,资产借出归还,资产维修管理,资产销账。
- (2) 基础信息维护:操作人员管理,部门维护,职员管理,资产档案初始化,数据备份与恢复。
- (3) 固定资产查询:资产变动情况查询,资产增加单查询,个人借用资产查询。
- (4) 数据分析报表:资产分类分析表,资产分类明细表。
- (5) 系统管理:个人信息管理,数据初始化管理。

## 1.3 固定资产管理系统模型<sup>[1-4]</sup>

系统是在JavaEE平台上开发的,采用MyEclipse6.0为开发工具、MySQL为数据库、Tomcat为Web服务器。框架上采用了一个整合策略,在职责上被分成3层,分别是:描述层 presentation 采用 Struts 框架,持久层 persistence 采用 Spring JDBC 框架,业务层 business 采用 Spring 框架,系统模型如图2所示。

## 2 系统的架构设计与实现

### 2.1 系统架构设计

在设计固定资产管理系统平台时,我们参照了JavaEE分层设计模型和图2,将整个系统设计分为数据访问层(即DAO操作具体业务数据和Façade完成对具体DAO的封装)、业务逻辑层和页面显示层,如图3所示。

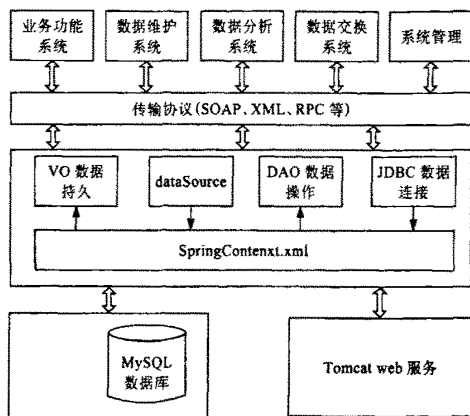


图2 系统模型

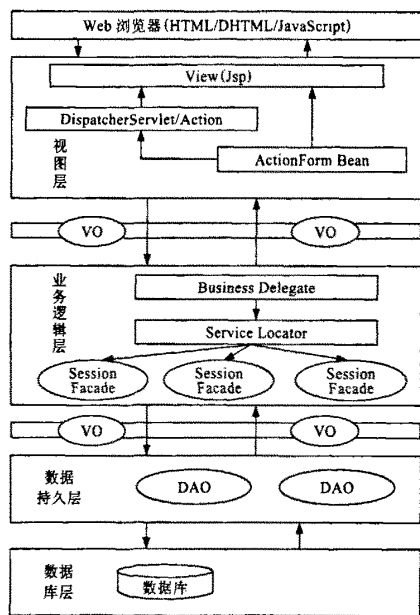


图3 固定资产管理系统的系统结构

#### 2.1.1 页面显示层<sup>[5]</sup>

页面显示层就是客户所看到的视图,它通常包含当前页面所用到的功能。它主要是通过jstl和Tiles模板来完成的。因为页面显示层在整个系统中处于关键位置,它的作用是由两件事情构成的。其一是向用户提供友好的界面,其二是要完成用户与系统的交互功能,在这一部分我们采用Struts来实现MVC(model-view-controller)模式,因为通过MVC模式可以把应用分为输入、处理、输出3个部分,从而减弱了Action控制类和Service业务逻辑接口之间的耦合。

#### 2.1.2 业务逻辑层

在业务逻辑层的设计过程中,我们使用了JavaEE设计模式中的Façade模式对业务逻辑进行了封装。因为业务逻辑层是以DAO为基础的,而我们对这些DAO进行了Façade封装详细如图4所示。因此对于每个具体的业务组件可能就是多个具体DAO的Façade封装的调用,同时利用了Spring框架对

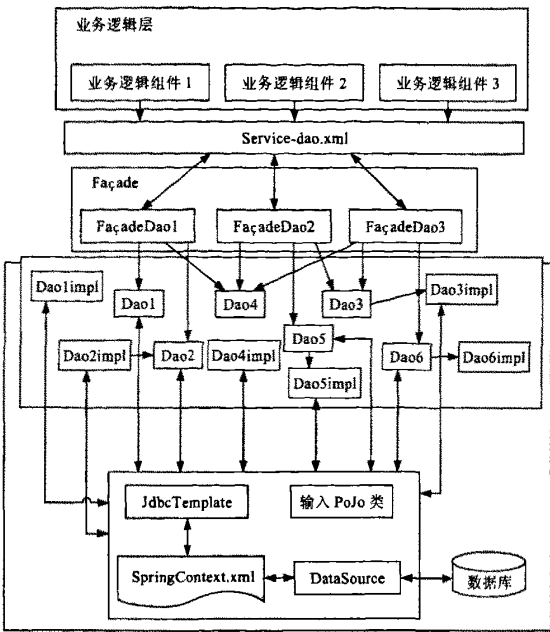


图 4 JavaEE 的数据持久层架构模型

对象进行统一的配置和管理, 既简化了业务实体对象的具体配置, 同时, 我们也对每个业务逻辑进行了 Façade 封装。从而为 Action 提供了清晰的业务逻辑服务并且通过 Spring 框架的配置降低了 Action 对 Service 的耦合度。

2.1.3 数据访问层

数据访问层我们使用 Spring JDBC 框架, 因为从文中知道在一个具体 Service 业务模块中是使用 Façade 模式进行业务逻辑和数据访问的细节相分离的, 而在一个具体 Façade 类中有一个或者多个具体的 DAO 类的。DAO 主要作用是用来封装数据库访问过程, 即就是完成具体 POJO 类的 CRUD 操作。通过 Spring JDBC 框架可以帮助我们完成两件事情: 其一是实现与具体数据库无关的操作, 其二是实现 Façade 模式的大数据粒的封装, 从而达到一个与数据源无关的数据存取, 并实现了数据的轻松操作。

2.2 数据持久层的设计与实现

2.2.1 Spring JDBC 框架的设计<sup>[4-10]</sup>

(1)从图 4 中, 我们可以看到 Façade 模式实现了数据访问层与业务逻辑层之间的弱耦合。因为在业务逻辑层中一个具体 Service 组件仅需访问在它管理的一个 Façade 类或者多个 Façade 类, 在由其访问 service-dao.xml (主要是由 Spring 框架支持的) 管理下的 FaçadeDao 即可。这样做的好处是一个具体 Façade 类和它所管理的 Dao 类在同一层中 (数据访问层中), 而在 Façade 类调用某个具体的 DAO 类进行 CRUD 操作时是不用与它所对应的业务逻辑组件进行通信的, 它们之间的通信关系只是在某个具体 Façade 类的结果完成后才会发生, 这样做有两个好处。其一可以减少网络传输的次数, 提高了系统的性能; 其二降低业务逻辑层与数据访问层之间的耦合度。

(2)降低各个具体业务逻辑组件的耦合度: 因为业务逻辑

组件的分类是按照具体的功能来划分的, 每个业务逻辑组件是对多个 DAO 类的 Façade 封装的调用, 从而为业务逻辑层提供了一个清晰的业务逻辑服务。

(3)进行事务处理: 因为 Façade 类包含数据持久逻辑的操作即就是数据的 CRUD, 因此数据访问层是非常适合进行数据安全验证和数据事务处理。

2.2.2 Spring JDBC 框架应用

首先需要编写它的配置文件: 再根据此文件编写 POJO 对象并与关系数据库数据之间进行数据传递, 同时生成 POJO 对象; 最后编写 DAO 类, 以实现具体的 CRUD 操作, 并对其进行封装。本文以资产管理数据模块为例, 简要地介绍一下 Spring JDBC 框架在系统中的应用。在资产管理数据模块中, 有包括资产信息表 (assetinfo)、资产卡片表 (assetcard)、附属设备表 (accessory)、资产类别表 (assettype)、领用记录表 (applyhistory) 等。

(1)Spring JDBC 数据源配置

Spring JDBC 的数据库连接信息是从配置文件中加载的。在这里, 我们主要对数据库连接的各种参数进行配置和根据项目的进展以供事务管理、产生 DAO 等。在本模块中, 配置文件 SpringContext.xml 如下:

```
<bean id = "dataSource" class = "org.apache.commons.dbcp.
BasicDataSource" destroy-method = "close">
<property name = "driverClassName"
value = "com.mysql.jdbc.Driver"/>
<property name = "url"
value = "jdbc:mysql:///localhost:3306/EAM"/>
<property name = "username" value = "root"/>
<property name = "password" value = "123"/></bean>
```

(2)创建数据库表的 POJO 对象类

要使用 JdbcTemplate 进行数据库操作, 就必须要有与数据库对应的 POJO 类, 以共 JdbcTemplate 使用。在 Spring JDBC 框架中, POJO 对象类可以使用标准的 JavaBean 来编写。利用 Myeclipse 分别生成 AssetInfo.java、Assetcard.java、Accessory.java、Assettype.java、Applyhistory.java 等持久化对象。

(3)创建数据库操作类

我们现在以 AssetInfoDAO.java 为例, 该类使用 JdbcTemplate 来实现数据库的访问逻辑。要进行数据库的访问, 就要为该类定义一个数据源 dataSource。

首先需要为该类添加一个 dataSource 数据源变量, 并添加 getter/setter 的方法, 该数据源的对象将会在 SpringContext.xml 通过 Spring 的 loc 容器配置进行注入, 配置如下:

```
<bean id = "AssetInfoDAO" class = "com.erp.dao.AssetInfo-
DAO">
<property name = "dataSource">
<ref local = "dataSource"/></property></bean>
```

其次, 在 AssetInfoDAO.java 类中操作数据, 我们现在以查找资产的 getAssetInfo() 方法为例。该方法首先创建一个 JdbcTemplate 对象, 然后创建一个具体的 SQL, 最后调用 queryForObject() 方法并根据某个具体 assetInfo 来查找具体的资产详细信息。伪代码如下所示:

```
public AssetInfo getAssetInfo(String assetInfoId) {
```

```
JdbcTemplate jdbcTemplate = new JdbcTemplate(dataSource);
String sql = "SELECT * FROM assetinfo WHERE assetInfoId = "
+assetInfoId"";
return (AssetInfo) jdbcTemplate.queryForObject(sql, assetInfoId);}
```

### 2.3 业务逻辑层的实现

业务逻辑层负责处理由 Façade 模式封装一个或者多个 DAO 类,并且业务逻辑层为 Action 提供某项具体业务功能。在这里我们需要确认业务逻辑层的数据来源于数据持久层的 POJO 类和表现层的 FormBean,因此说业务逻辑层是表现层与数据持久层进行数据通信的中间层。我们在业务层主要要完成下面 3 项工作。

#### (1) 在 ApplicationContext.xml 中配置事务

为了避免在业务逻辑层中重复编写大量的事务处理代码。我们在固定资产管理系统中,把事务管理定义在 transactionManager 这个配置上,这样开发人员只须关注业务逻辑,而将事务管理完全交给配置文件,这样做的好处是即可以节约开发时间又可以提高开发效率。

#### (2) 使用 Façade 进行 DAO 类的管理

在 service-dao.xml 进行 Façade 类和一个或者多个 dao 类的配置。这样做的好处是组件间就不存在硬编码的关联,任何组件都可以最大程度地得到重用。当我们在一个业务逻辑组件的开发时只需要在使用 setter 方法进行依赖注入,便可轻松实现组件间的引用。

(3) 编写具体业务组件和使用 Façade 模式进行基于 Façade-DAO 类的配置管理,以供页面显示层中的 Action 类使用。

### 2.4 Struts 显示层的流程设计以及实现<sup>[11-12]</sup>

#### 2.4.1 用户界面层的流程设计

基于 Struts MVC 的用户界面层架构模型如图 5 所示。

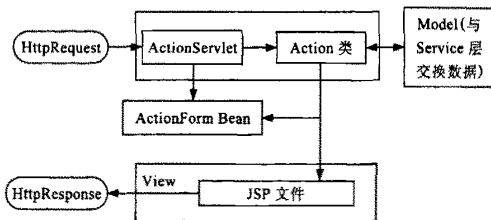


图 5 基于 Struts MVC 的用户界面层架构模型

(1) 从图 5 中,我们可以看到 Client(浏览器)发出 HTTP 请求。Web 容器将用一个 Response 作出响应。

(2) Controller(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。这个控制器可以用来管理导航。在 Struts 中控制器的具体是以 org.apache.struts.action.ActionServlet 实现的,其中 struts-config.xml 是配置请求处理的文件。

(3) Action(业务逻辑)是用来更新模型的状态,并帮助控制应用程序的流程,并且 Action 是调用业务层的 Services 实现的。当处理完毕使用 ActionForward 类指向下一个要显示的页面。

(4) Model(模型)主要是以 JavaBean 的形式存在的。主要

为: ActionForm 和 Action。

(5) View(视图)视图就是一个基于 jstl 和 Tiles 模板构成的 JSP 文件,负责页面的显示。

#### 2.4.2 用户界面层的实现

限于篇幅,这里只给出与资产管理页面相关的 AssetInfoAction 类的部分实现:

```
protected void process(ActionForm form)
{AssetInfoVO uct = new AssetInfoVO();
//1.set AssetInfo fields from the form
//2.call AssetInfo method to do something
//3.return mapping 所指的页面}
```

从以上代码可以看出,表现层中的数据表示是 FormBean,表现层与业务逻辑层进行通信时使用 POJO 进行。

### 3 结束语

基于 Spring+Struts 框架的固定资产管理系统,解决了传统 J2EE 架构中,开发难度大、数据访问效率低、难以进行单元测试等问题。并证明结合使用这个框架确实可以:使用 Façade 模式有效地降低网络之间的开销;使用 DAO 模式实现了具体的业务逻辑和数据的读取分开,从而减少逻辑之间的依赖性,并提高了开发的效率;通过此框架,简化了系统编程代码,提高软件开发效率,同时提高系统的性能和维护性,使得系统在质量上有了大幅度的提高。

#### 参考文献:

- [1] 秦敏.利用 Hibernate 简化 Java 数据库访问[J].计算机应用与软件,2006,23(6):46-47.
- [2] 林青,许锁坤.基于 J2EE 的企业级系统持久性框架的设计与实现[J].计算机工程与设计,2007,28(7):1732-1734.
- [3] 郭文生,杜军平,尹伯欣.基于 J2EE 的旅游电子商务平台的设计研究[J].计算机应用与软件,2007,24(2):118-120.
- [4] 吴立春.基于 Façade 模式的业务逻辑层框架设计[J].微计算机信息,2007(24):245-246.
- [5] 窦东阳,赵英凯,许文稼.利用 Myeclipse 简化基于 Hibernate 的数据库访问[J].计算机工程与设计,2007,28(4):773-774.
- [6] 彭宝琴,罗晓沛.基于 J2EE 轻量级框架组合的消费信贷系统的实现[J].计算机工程与设计,2007,29(3):647-649.
- [7] 余腊生,叶楠.基于 J2EE 三层架构的远程答疑系统的研究与实现[J].计算机工程与设计,2007,28(13):3216-3215.
- [8] 陈平,邢桂芬.Façade 模式在 JDBC 编程中的应用[J].计算机工程与设计,2006,27(1):60-69.
- [9] 刘和洋,王健华,黄永红,等.基于 Web 的政务系统通用框架的研究与实现[J].计算机工程,2006,32(14):263-265.
- [10] 李光俊,华庆一,吴海松.基于 AOP 技术的 Composite 模式的改进[J].计算机工程,2008,34(10):73-74.
- [11] 孙小锥,上官右黎,文福安.基于轻量级 J2EE 框架的网络教学系统[J].计算机工程,2008,34(6):266-267.
- [12] 李守振,张南平,常国锋.Web 应用分层与开发框架设计研究[J].计算机工程,2006,32(22):274-276.

作者: [张俐](#), [张维玺](#), [ZHANG Li](#), [ZHANG Wei-xi](#)  
作者单位: [张俐, ZHANG Li \(江苏技术师范学院计算机科学与工程学院, 江苏常州, 213001\)](#), [张维玺, ZHANG Wei-xi \(江苏技术师范学院电气信息工程学院, 江苏常州, 213001\)](#)  
刊名: [计算机工程与设计](#) **PKU**  
英文刊名: [COMPUTER ENGINEERING AND DESIGN](#)  
年, 卷(期): 2009, 30 (16)  
被引用次数: 11次

## 参考文献(12条)

1. 秦敏 利用Hibernate简化Java数据库访问[期刊论文]-[计算机应用与软件](#) 2006 (06)
2. 林青;许锁坤 基于J2EE的企业级系统持久性框架的设计与实现[期刊论文]-[计算机工程与设计](#) 2007 (07)
3. 郭文生;杜军平;尹怡欣 基于J2EE的旅游电子商务平台的设计研究[期刊论文]-[计算机应用与软件](#) 2007 (02)
4. 吴立春 基于Facade模式的业务逻辑层框架设计[期刊论文]-[微计算机信息](#) 2007 (24)
5. 窦东阳;赵凯凯;许文稼 利用Myeclipse简化基于Hibernate的数据库访问[期刊论文]-[计算机工程与设计](#) 2007 (04)
6. 彭宝琴;罗晓沛 基于J2EE轻量级框架组合的消费信贷系统的实现[期刊论文]-[计算机工程与设计](#) 2007 (03)
7. 余腊生;叶楠 基于J2EE三层架构的远程答疑系统的研究与实现[期刊论文]-[计算机工程与设计](#) 2007 (13)
8. 陈平;邢桂芬 Facade模式在JDBC编程中的应用[期刊论文]-[计算机工程与设计](#) 2006 (01)
9. 刘和洋;王健华;黄永红 基于Web的政务系统通用框架的研究与实现[期刊论文]-[计算机工程](#) 2006 (14)
10. 李光俊;华庆一;吴海松 基于AOP技术的Composite模式的改进[期刊论文]-[计算机工程](#) 2008 (10)
11. 孙小维;上官右黎;文福安 基于轻量级J2EE框架的网络教学系统[期刊论文]-[计算机工程](#) 2008 (06)
12. 李守振;张南平;常国锋 Web应用分层与开发框架设计研究[期刊论文]-[计算机工程](#) 2006 (22)

## 本文读者也读过(8条)

1. 穆丹 集成JavaEE框架构建MIS系统的研究与实现[学位论文]2008
2. 吴兵, 宋菲, 赵志强, 武文芳, WU Bing, SONG Fei, ZHAO Zhi-qiang, WU Wen-fang 基于Web技术的固定资产管理系统的设计与实现[期刊论文]-[实验技术与管理](#)2007, 24 (1)
3. 姚云 基于JavaEE的在线考试系统研究与实现[学位论文]2007
4. 管文杰 固定资产管理系统对物流管理的促进和发展[期刊论文]-[商场现代化](#)2009 (16)
5. 梁勇, LIANG Yong 固定资产管理系统分析与设计[期刊论文]-[科技与管理](#)2005, 7 (5)
6. 张冬梅, 肖淑兰, 谭燕 用友ERP-U861固定资产管理系统操作常见问题解析[期刊论文]-[中国管理信息化](#)2010 (8)
7. 张俐, ZHANG Li 基于JavaEE的电信CRM数据持久层的实现[期刊论文]-[计算机工程](#)2009, 35 (6)
8. 张艳伟 河南财院固定资产管理系统的设计与实现[学位论文]2009

## 引证文献(11条)

1. 郑明涛 浅析企业固定资产管理中如何实现信息化管理模式[期刊论文]-[软件](#) 2013 (10)
2. 宋昕, 朱瑞龙 企业资产管理系统设计[期刊论文]-[科技广场](#) 2013 (1)
3. 薛淑敏 企业资产管理系统中通信模型的研究与实现[期刊论文]-[电子设计工程](#) 2012 (11)
4. 王浩 固定资产的研究[期刊论文]-[经济视野](#) 2013 (19)
5. 宋成明, 李彦军 基于轻量级Java EE架构的教师考核系统的设计与实现[期刊论文]-[铁路计算机应用](#) 2011 (4)
6. 张俐, 张维玺, 陆冰峰 在线B2C充值卡销售系统设计与实现[期刊论文]-[实验室研究与探索](#) 2013 (7)
7. 张俐, 张维玺 基于B/S的实验室信息系统的分析与设计[期刊论文]-[实验室研究与探索](#) 2012 (5)

8. [张俐](#) [基于ibatis和设计模式的数据持久层的应用](#)[期刊论文]-[安徽大学学报（自然科学版）](#) 2010(3)
9. [聂常红](#), [唐远强](#), [冯相忠](#) [基于Java EE多层架构的进销存管理系统](#)[期刊论文]-[计算机与现代化](#) 2012(10)
10. [冉崇善](#), [蒋中强](#) [电力EAM系统中数据库中间件的研究与设计](#)[期刊论文]-[计算机工程与设计](#) 2010(22)
11. [张俐](#), [张维玺](#), [陆冰峰](#) [在线B2C充值卡销售系统设计与实现](#)[期刊论文]-[实验室研究与探索](#) 2013(7)

引用本文格式: [张俐](#), [张维玺](#), [ZHANG Li](#), [ZHANG Wei-xi](#) [基于JavaEE的固定资产管理系统的设计与实现](#)[期刊论文]-[计算机工程与设计](#) 2009(16)