

整合 ESMSH 框架的 Java EE 应用架构

欧阳宏基¹, 宋笑雪¹, 李红^{1,2}

(1. 咸阳师范学院 计算机学院, 陕西 咸阳 712000;

2. 西安电子科技大学 智能感知与图像理解教育部重点实验室, 西安 710071)

摘要: 好的软件架构能够为应用系统提供可靠的稳定性和扩展性, 并简化应用系统的实现过程; 针对基于 Java EE 平台的应用系统, 提出一种整合 ESMSH 框架的 Java EE 应用架构; 该架构基于模型视图控制器 (model view controller, MVC) 设计模式, 利用 Easy UI 框架构建表示层, Spring MVC 框架构建控制层, 使用 JS 对象标记 (javascript object notation, JSON) 在表示层与控制层传输数据; 利用 Spring 框架构建业务逻辑层, 利用 Hibernate 框架构建数据持久层, 通过控制反转 (Inversion of Control, IOC) 容器管理控制层、业务层和数据持久层组件的依赖关系, 采用注解的方式进行 Spring MVC、Spring 和 Hibernate 的整合; 将该架构应用到高等院校目标考核管理系统的实施中, 按照 Java EE 分层开发方法进行系统设计, 详细阐述了各层的实现过程, 通过实践证明了该架构的有效性。

关键词: JavaEE 架构; Easy UI 框架; Spring MVC 框架; Spring 框架; Hibernate 框架

Java EE Application Architecture Integrated with ESMSH Framework

Ouyang Hongji¹, Song Xiaoxue¹, Li Hong^{1,2}

(1. Computer College, Xianyang Normal University, Xianyang 712000, China;

2. Ministerial Key Lab of Intelligent Perception and Image Understanding, Xidian Univ., Xi'an 710071, China)

Abstract: Good software architecture can provide reliable stability and scalability for application systems and simplify the implementation process of application systems. In view of the application system based on the Java EE platform, a Java EE application architecture based on ESMSH framework is proposed. The architecture is based on MVC design pattern, it uses the Easy UI to build the presentation layer, uses the Spring MVC framework to build the control layer, and uses the JSON format to transmit data at the presentation layer and the control layer. using the Spring framework to build the business logic layer, using the Hibernate framework to build the data persistence layer, the components dependencies of the control layer, business layer, and data persistence layer is managed through the IOC container, the method of annotations is used for the architecture's integration and development. The architecture is applied to the implementation of the target management system of colleges and universities, According to Java EE layered development method for system design, and the implementation process of each layer is described in detail. The effectiveness of the architecture is proved through practice.

Keywords: Java EE architecture; easy UI framework; spring MVC framework; Spring framework; Hibernate framework

0 引言

Java EE 是以 Java 为主进行企业级应用开发的技术平台, 由于 Java EE 具有跨平台、稳定性高、兼容性好、技术自由搭配度高等特点, 所以一直受到企业级应用开发的青睐。近年来, 为了进一步解决传统 Java EE 开发所具有的组件依赖性高、开发难度大、部署与维护困难、用户交互性差等问题, 出现了以 Struts2、Hibernate、Spring、MyBa-

ties 为主的轻量级框架, 目的是为了简化 Java EE 服务器端的开发过程。在客户端出现了 Ext JS、BootStrap、Easy UI 等 JavaScript 框架, 目的是为了提 高 Web 应用的富客户端体验。文献 [1-2] 研究了 Struts2+Spring+Hibernate 为主的 Java EE 应用开发方式, 并应用到高校实验室管理和煤炭企业生产统计管理系统的设计开发中, 阐述了其中的核心技术。文献 [3-5] 主要以 Struts2 框架为研究对象, 讨论了在 Java EE 控制层的具体实现过程。文献 [6] 提出了以 Hibernate 框架为对象关系映射 (object relation mapping, ORM) 实现方式的数据持久化层的设计方式。文献 [7-8] 研究了 Ext JS 框架与服务器端轻量级框架的整合, 并以动态树和地图信息加载为例具体描述了 Ext JS 的应用。

虽然上述学者的研究成果在某个系统设计过程中进行了实验验证, 但笔者认为还存在两个主要问题: 1) Struts2 的高危安全漏洞。由于目前 Struts2 在国内经融、政务、大中型互联网公司、高校等领域的应用中广泛存在, Struts2 存在的远程代码执行漏洞^[9]爆出后, 对相关领域带来了不

收稿日期:2017-12-11; 修回日期:2018-01-08。

基金项目:陕西省科技厅自然科学基金基础研究计划面上项目 (2017JM6086); 陕西省教育厅专项科研计划项目 (17JK0828); 咸阳师范学院“青年骨干教师”培养项目 (XS YGG201615)。

作者简介:欧阳宏基 (1982-), 男, 陕西宝鸡人, 硕士, 讲师, 主要从事软件工程、Java EE 应用方向的研究。

宋笑雪 (1967-), 女, 陕西咸阳人, 博士, 教授, 主要从事软件工程方向的研究。

李红 (1976-), 女, 陕西咸阳人, 博士, 副教授, 主要从事图像处理, 软件工程方向的研究。

少的损失。虽然 Apache 官方给出了漏洞的解决方案, 但是用户逐渐丧失了对 Struts2 的信任, 这种情况在 2017 初 Struts2 再次爆出高危漏洞后尤为明显; 2) Ext JS 的执行效率问题。虽然 Ext JS 框架提供了功能强大的客户端组件, 能够简化 Java EE 客户端的开发, 但是它是一个重量级的前端框架, 在某些复杂页面下, 加载速度和执行效果不尽如人意。

本文在前人研究基础上, 以 Easy UI、Spring MVC、Spring、Hibernate 等框架为基础, 提出了一个基于 ES2SH 框架的 Java EE 应用架构。该架构利用 Easy UI 创建表示层, Spring MVC 创建控制层, 采用 JSON 作为表示层与控制层之间传输数据的格式。利用 Hibernate 创建数据持久层, 通过 Spring 管理控制层与业务层、业务层与持久层之间组件的依赖关系, 采用可扩展标记语言 (extensible markup language, XML) 文件 + 注解的方式进行 Spring MVC、Spring 和 Hibernate 的集成。将该架构应用到高等院校目标考核管理系统的设计与实施中, 详细描述了该架构应用的关键技术, 为以该整合框架应用到其他领域的应用开发提供了一定的借鉴。

1 相关技术

Easy UI 是一组基于 JQuery 和超文本标记语言 (HyperText Markup Language, HTML) 5 的用户接口 (user interface, UI) 插件集合体, 只需要一些简单的 HTML 标签 (不需要过多的 JavaScript 和层叠样式表 (cascading style sheets, CSS)), 即可设计出功能丰富且美观的 UI 界面。具有良好的浏览器兼容性、开源、轻量级、易于掌握等优点^[10]。Spring MVC 是 Spring 框架中用于实现 Web 功能的模块, 它根据 DispatcherServlet 和配置文件, 实现视图、控制器和模型之间的分离^[11]。Spring 利用控制反转容器 (Inversion of Control, IOC) 创建并管理对象之间的依赖关系; 通过面向方面编程 (aspect oriented programming, AOP) 实现业务逻辑与公共服务逻辑相分离。Hibernate 是基于 Java 数据库连接 (Java DataBase Connectivity, JDBC) 的 ORM 框架, 能实现对象与数据库表之间的映射^[12-13], 通过 Hibernate 查询语言 (hibernate query language, HQL) 以简化复杂结构化查询语言 (structure query language, SQL) 的编写, 提高数据库开发效率。

2 ESMSH 整合架构

在上述框架原理上, 建立满足 MVC 模式的 5 层 Java EE 分层架构模型, 如图 1 所示。表示层由 Easy UI、视图组件构成, 发送的请求统一由 Spring MVC 的前端控制器 DispatcherServlet 接收, 根据配置文件将请求转发到某个具体的 Controller 组件。Controller 调用 Service 组件执行

业务逻辑, 以 ModelAndView 组件返回逻辑视图, DispatcherServlet 调用 ViewResolver 解析出物理视图并填充 Model 需要显示的数据响应客户端请求。业务逻辑层通过 Service 接口向 Controller 组件提供服务, 具体的业务逻辑

处理和计算由 Service 对象完成, Service 对象由 Spring IOC 容器管理。数据持久层由数据访问对象 (data access object, DAO) 模式与 Hibernate 框架搭建, DAO 接口定义持久化逻辑并作为业务逻辑层访问的入口, DAO 对象封装 Hibernate 通过映射文件完成持久化对象 (persistent object, PO) 与数据存储层之间的交互, DAO 对象由 Spring IOC 容器管理。数据存储层负责存储应用系统所需要长久保存的数据以及之间的关联关系。

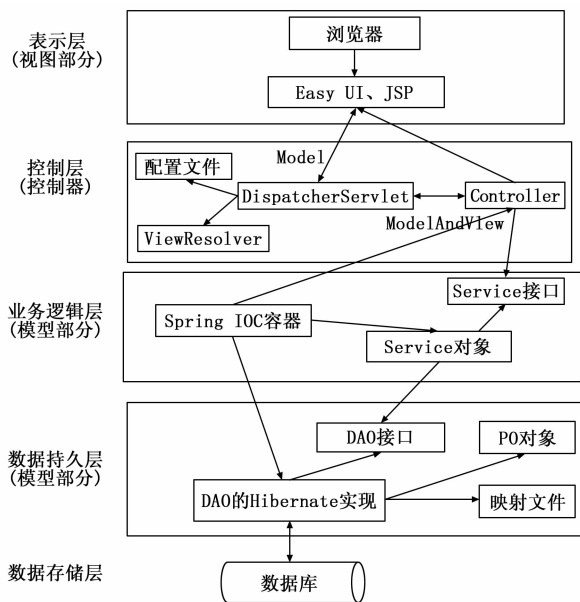


图 1 整合 ESMSH 框架的 Java EE 架构

3 ESMSH 架构在目标考核管理系统中的应用

3.1 目标考核系统简介

目标责任考核是高等院校行政管理的一项重要工作, 通过制定、落实、执行、监督、分析各二级学院、职能部门的各项工作任务 and 进展情况, 针对不足找出差距, 提出改进措施, 从而推动学校又快又好的发展。通过分析目标考核工作的具体流程, 结合上述 ESMSH 框架设计了目标考核管理信息系统, 主要包括基础数据、教工信息管理、指标管理、报表管理、信息查询和系统维护等模块。其中基础数据包括教育教学、科学研究、师资队伍建设和、党团建设、学生工作 5 个大方面覆盖的所有基本数据。例如教育教学包括教务任务、课程设计、毕业设计、教学竞赛、教改项目、教学获奖、大学生科研训练、教育实习基地、“互联网+”大赛等相关数据。教工信息管理是指教工对自己基本信息的维护, 包括工作经历、职称变动、部门调动、岗位聘任结果、个人荣誉、密码修改等操作。指标管理实现对指标的增、删、改、查。指标是考核的一个观测点, 分为一级指标、二级指标和三级指标。例如“英语四级通过率”是一个三级指标, 属于“教学质量”这个二级指标, 对应的一级指标是“教育教学”。报表管理实现对目标任务下达情况、按指标按部门实现指标完成情况、年度考核结

果等数据实现导出显示。信息查询主要针对各二级学院领导查看本单位目标任务完成情况。职能部门领导按照分管的一级指标查看二级学院的完成情况。校领导按照各种组合条件（以学院、工号、学位、职称、聘任岗位、指标等）查看任务完成情况。系统维护实现数据库的备份与还原、系统公告的发布、权限角色信息的管理、相关参数的配置等。

3.2 架构实现的关键技术

3.2.1 利用泛型 DAO+Hibernate 实现持久层

持久层的功能是利用 Hibernate 框架完成关系数据与对象的转换，以面向对象方式访问关系数据库。主要是通过 XML 形式的映射文件或在 POJO 形式的实体类中添加注解，实现对象属性与对应表的字段、对象之间的依赖关系与对应数据表之间的、主、外键约束之间进行映射^[14]。具体实现包括两个核心步骤：第一，创建泛型 DAO 接口及其实现类。由于每个实体类的持久化操作都具有相似的代码，为了提高代码的复用率，定义一个泛型的 DAO 接口—BaseDAO，其中定义所有持久化对象基本的 CRUD 方法^[15]。BaseHibernateDAO 为 BaseDAO 的实现类，该类继承 HibernateDaoSupport 实现 BaseDAO 中的所有方法。HibernateDaoSupport 是 Spring 支持 Hibernate 的 DAO 工具，通过其关联的 HibernateTemplate 模板类可以在无需打开或关闭 Session 对象的情况下，实现数据库的 CRUD 操作，提高了持久层的开发效率。第二，创建并配置每个实体类对应的 DAO 接口及其实现类。某个实体类的 DAO 接口继承 BaseDAO 接口，只需定义该实体类特有的持久化方法，实现类继承 BaseHibernateDAO 完成特有的持久化方法，最后利用@Repository 注解将 DAO 在 Spring IOC 容器中进行配置。上述操作需要的类图如图 2 所示。

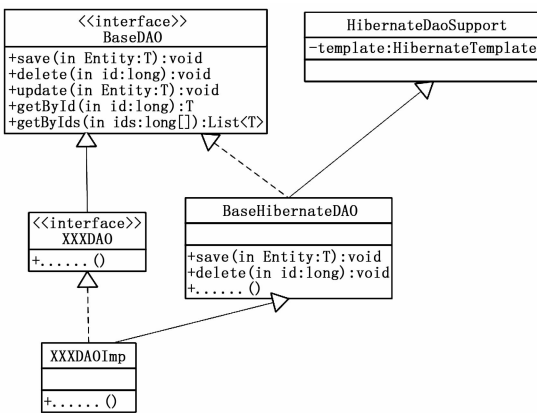


图 2 持久层类图关系

3.2.2 利用 Spring IOC+AOP 实现业务层

业务层完成系统的核心逻辑，包括两类：与业务计算相关的逻辑和与业务计算正交的公共逻辑（例如日志、权限等）。对于业务计算逻辑采用接口分离原则进行设计，首先定义 Service 逻辑接口定义组件的业务功能，作为控制层 Controller 组件的调用入口；定义 ServiceImp 提供该组件的

实现并依赖持久层的若干 DAO 组件获取业务计算需要的持久化数据，DAO 对象由 Spring IOC 创建并注入。然后通过@Service 注解将 Service 组件纳入 Spring IOC 容器管理。最后通过声明式注解为业务逻辑方法添加事务控制。启用声明式事务需要在 Spring 配置文件中添加如下代码：

```

<bean id="txManager" class="org.springframework.orm.hibernate3.HibernateTransactionManager">
    <property name="sessionFactory" ref="sessionFactory">
</property>
</bean>
<tx:annotation-driven transaction-manager="txManager"/>
    
```

针对与业务计算正交的公共逻辑采用 AOP 方式实现。将每个公共逻辑作为一个切面而定义，对于需要公共逻辑的业务逻辑采用正交织入的方式执行，这样便于公共逻辑的修改与维护，节省代码量。目标考核系统中的登录验证、权限验证、日志记录等操作都采用 AOP 的方式来实现，以权限验证为例进行说明。首先，在基于角色的访问控制（role-based access control, RBAC）理论^[16]基础上定义教师账户类、部门账户类、角色类、权限类。系统所设计的登录帐户包括教师账户和部门账户两种类型，教师账户对应教师角色具有教师信息管理模块的所有权限；部门账户对应教师以外的相应角色，具有教师信息管理模块以外的其它相应权限。这样设计的优点有两个：（1）减少了教师账户的角色数量，从而也就减少了教师账户的权限数，便于操作与管理；（2）部门账户不再和某个具体的教师关联，同一部门里的任何一个教工只要知道部门账户，就可以执行相关的功能，便于考核工作的开展。然后，通过实现 HandlerInterceptor 接口定义拦截器，重写 preHandle 方法，该方法在执行 Controller 之前执行。具体的权限验证流程如图 3 所示，对应的代码如下所示：

```

public class PermissionInterceptor implements HandlerInterceptor {
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object obj) throws Exception {
        User user=(User)request.getSession().getAttribute("user");
        String currentURL=getURL(request);
        if (user == null) {
            if("user/userController_login".startsWith(currentURL))
                return true;
            else {
                response.sendRedirect("login.jsp");
                return false;
            }
        }
        // 用户登录后判断权限
        else {
            //对退出操作对应的 Controller 进行放行 if("user/userController_loginOut".startsWith(currentURL))
                return true;
            else {
                
```

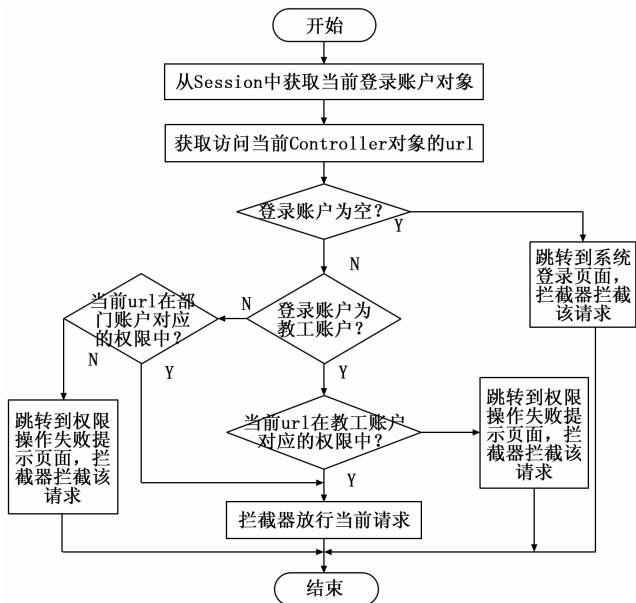


图 3 preHandle 方法执行流程图

```

if(user instanceof UserVO) {
    UserVO loggedUser = (UserVO)user;
    if (loggedUser.hasPermission(currentURL)) {
        //用户拥有权限就放行
        return true;
    } else {
        //如果没有权限就拦截
        response.sendRedirect("noPermission.jsp");
        return false; } }
else {
    // 如果当前登录用户是部门账户,则判断
    // 部门账户的权限,判断逻辑同上。
    .....
}
}
}

```

3.2.3 利用 Spring MVC 实现控制层

控制层充当 MVC 模式中的控制器, 负责接收表示层数据交由业务层进行数据处理, 根据请求结果调用对应的视图数据进行响应。具体实现过程包括以下步骤: 首先在 web.xml 文件中配置前端控制器, 由 Web 容器在启动时创建前端控制器实例, 它负责给 Controller 分派表示层请求, 并执行视图解析。然后定义 spring-mvc.xml 文件, 完成注解扫描、处理器映射器、处理器适配器、视图解析器的配置。注解可以简化 XML 的配置, 处理器映射器和处理器适配器负责映射前端控制器转发给 controller 的映射策略。视图解析器负责解析 Controller 返回的 ModelAndView 组件, 形成特定形式的视图组件。执行顺序如图 4 所示^[17]。最后创建 Controller 组件调用业务层组件。本系统采用无侵入式定义, 将控制器当作普通的 JavaBean, 通过@Controller 对控制器进行标注, 在相关方法上标注@RequestMapping 并注明访问该方法的 URL 后缀。

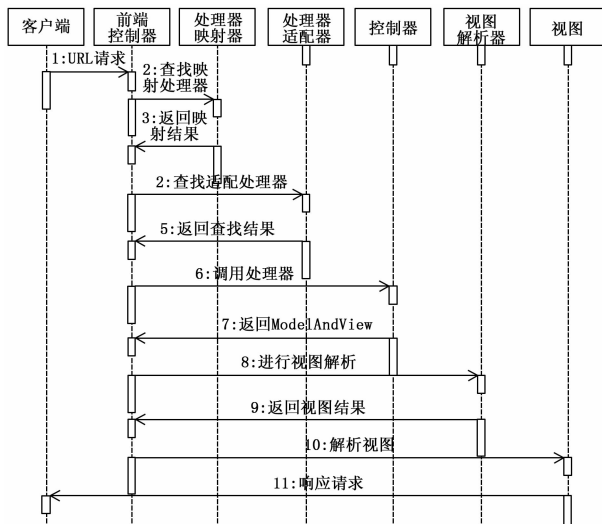


图 4 Spring MVC 执行流程

3.2.4 利用 Easy UI+JSP 实现表示层

表示层为用户提供交互操作并展示 Web 应用的界面。为提高用户交互体验, 便于模块化开发, 本架构以 Easy UI + JSP 作为表示层开发的核心组件。首先, 在 JSP 页面引入 Easy UI 核心 JS 文件和 CSS 文件, 包括 jquery.min.js、jquery.easyui.min.js、中文提示信息库、核心 UI 文件和图标文件等。然后, 根据模块功能定义独立的 JS 文件, 在该文件定义定义各种 EasyUI 函数, 这样可以分离 HTML 与 JavaScript 代码, 便于开发与维护。以招生就业处管理员给各二级学院下达综合指标目标任务的面面为例, 如图 5 所示。左侧通过树形控件显示当前登录用户的权限菜单, 右侧主要通过加载 Datagrid 控件以二维表格形式显示各二级学院对应招生就业工作三级指标的任务分值、年份以及审核状态。表格中的数据是在用户点击相应权限菜单后, 利用 Ajax 请求异步访问 Web 服务器获取的, 并且在返回表示层前, 控制层通过 jackson 序列化 Java 对象为标准 JSON 格式。为了提高用户的可操行, 设置 Datagrid 单元格为可编辑性, 在一个页面中即可完成对各二级学院按三级指标下达任务数据, 点击“保存”按钮将数据传送到服务器端保存。按钮触法事件对应的 JavaScript 代码如下所示。

```

function accept(){
    var ids=""; //修改行的指标:指标值
    var index = editIndex;
    if (endEditing()){
        (' dg').datagrid('acceptChanges').datagrid('unselectAll');
    }
    //表字段取出放入数组中
    var data = (' dg').datagrid('getData');
    for(var item in data.rows[index]){
        if(item != "ejxy" && item != "status" && item != "year"){
            ids += item + ":" + data.rows[index][item] + ",";
        }
    }
    .post("assignTask_StudentPartyWork_update",{ "ids":ids},

```

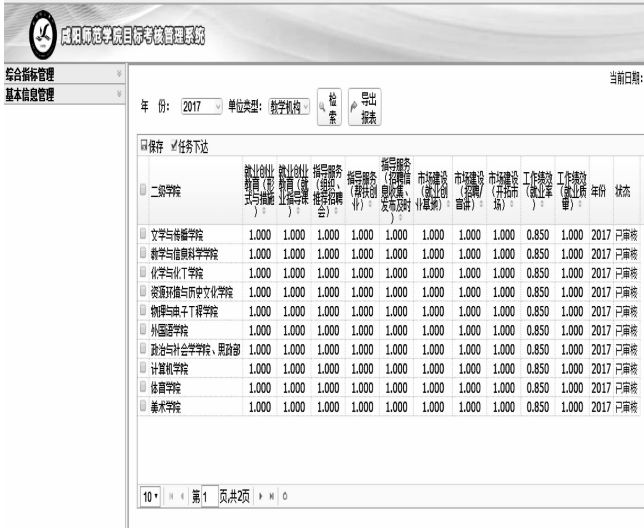


图 5 目标任务下达页面

```
function(data){
    if(data != null){
        (" dg"). datagrid('reload', assignTask_StudentPartyWork_
viewJson);
    }
    }, json);};
```

4 结束语

在对 Easy UI、Spring MVC、Spring 和 Hibernate 框架分析的基础上，提出基于 ESMSH 框架的 Java EE 应用架构。将该架构应用到目标考核管理系统的开发实践中，阐述了架构实现的核心技术。以 ESMSH 架构所实现的目标考核管理系统已在咸阳师范学院各职能部门、二级学院和全体教师之间展开运行，已顺利完成了 2016、2017 年度的目标考核工作，全面提升了学校目标考核工作的信息化水平，取得了较好的满意度，表明本文所提出的 Java EE 架构稳定、可靠。下一步准备在该架构基础上引入微服务思想，细化基础数据部分，并为学校的其他业务工作服务。

参考文献:

[1] 贾松浩, 杨彩, 刘军. 基于 S2SH 框架的个性化实验室管理系统 [J]. 实验室研究与探索, 2014, 33 (8): 232-234.

[6] 杨路, 何萍, 王珊. UFMC 系统中的载波频率同步技术 [J]. 电信科学, 2016, 32 (11): 50-55.

[7] Vakilian V, Wild T, Schaich F, et al. Universal-filtered multi-carrier technique for wireless systems beyond LTE [A]. IEEE GLOBECOM Workshops [C]. 2013: 223-228.

[8] Ali I, Al-Dhahir N, Hershey J E. Doppler Characterization for LEO satellites [J]. IEEE Trans. Commun., 1998, 46 (3): 309-313.

[9] Hou Z, Zhou Y, Shi J, et al. Doppler rate estimation for OFDM based communication systems in high mobility [A]. Interna-

[2] 欧阳宏基, 葛萌. 基于 S2SH 框架的煤炭企业生产统计管理系统 [J]. 计算技术与自动化, 2015, 34 (3): 118-122.

[3] 欧阳宏基, 葛萌. 基于 Struts2 与 Hibernate 框架的 Java Web 应用研究 [J]. 自动化技术与应用, 2016, 35 (2): 48-52.

[4] 甘文丽, 范刚龙. 基于 Struts2 和 DAO 的企业销售服务系统 [J]. 大庆师范学院学报, 2015, 35 (6): 21-24.

[5] 高博, 夏辉. 基于 Struts2 的科研业绩考评系统的设计与实现 [J]. 沈阳师范大学学报 (自然科学版), 2017, 35 (1): 98-102.

[6] 张少应, 程传旭. 基于 Hibernate 持久化层的设计与实现 [J]. 计算机技术与发展, 2014, 24 (12): 101-104.

[7] 唐永瑞, 张达敏. 基于 Ajax 与 MVC 模式的信息系统的设计与实现 [J]. 电子技术应用, 2014, 40 (2): 128-131.

[8] 盛艳, 朱革娇, 张建军, 等. 基于 ExtJS 轻量级框架的教学资源库设计与实现 [J]. 计算机技术与发展, 2015, 25 (4): 185-187.

[9] 郑先伟. Struts2 漏洞已成高校网络安全顽疾 [J]. 中国教育网络, 2017 (10): 52-52.

[10] 苗杰. 基于 Easy UI 框架与 Spring MVC 框架的权限管理系统的设计与实现 [J]. 电脑知识与技术, 2015, 11 (15): 53-55.

[11] 廖福保. 扩展 Spring MVC 模块的 Web 应用 [J]. 实验室研究与探索, 2012, 31 (10): 70-73.

[12] 褚福影, 卫文学. 基于 Web Services 和 Hibernate 技术的数据库中间件 [J]. 计算机与现代化, 2015 (10): 40-44.

[13] 林钰杰, 吴丽贤. 基于整合框架的 Web 应用系统开发平台设计与实现 [J]. 电子设计工程, 2016, 24 (20): 62-65.

[14] 徐金虎, 宋斌, 丁锐. Spring MVC+Hibernate+jQuery 模式开发框架应用研究 [J]. 自动化技术与应用, 2015, 34 (3): 42-46.

[15] 甘文丽, 王岚. 基于轻量级 Java EE 框架的高校招投标管理系统 [J]. 实验室研究与探索, 2012, 31 (11): 182-185.

[16] 贺建峰, 李雅娜, 潘帅涛. 脑卒中信息管理系统权限控制的设计与实现 [J]. 计算机测量与控制, 2017, 25 (8): 183-186.

[17] 薛峰, 梁锋, 徐书勋, 等. 基于 Spring MVC 框架的 Web 研究与应用 [J]. 合肥工业大学学报 (自然科学版), 2012, 35 (3): 337-340.

[10] 谭晓衡, 陈丽. 采用导频子载波的 OFDM 系统频偏跟踪算法 [J]. 重庆大学学报, 2008.

[11] 田由甲, 张冠杰. 基于多普勒频率变化率的固定单位算法研究 [J]. 无线电通信技术, 2016, 42 (4): 61-64.

[12] 郭永娜, 王钢, 郑黎明, 等. 5G 非正交多址接入中干扰消除技术研究 [J]. 无线电通信技术, 2016, 42 (5): 06-09.

[13] 李广达, 孙晨华, 刘刚. 卫星网络与地面融合的 5G 网络架构 [J]. 无线电工程, 2016, 46 (3): 5-8.

(上接第 229 页)

tional Conference on Wireless Communications & Signal Processing [C]. 2013: 1-6.