

基于 JSF 框架的权限管理系统的研究与实现

胡 乔

(武汉大学,湖北 武汉 430072)

摘 要:JavaServer Faces(JSF)是 Java 用来构建 Web 应用的新技术新标准,给开发者提供非常丰富的用户界面组件库,同时也使得表现层与后台逻辑层的结合十分容易,可以说 JSF 是一种开发网络应用的理想技术。介绍了 JSF 的特点及运行机制,并运用该技术开发了一个权限管理系统,重点解释了如何利用 JSF 技术根据权限系统的需要去实时动态地生成 Web 页面。工程实践表明,JSF 能使 Web 用户界面开发具有灵活性和扩充性,有助于 Web 应用程序结构的清晰明了,为开发者带来更高的工作效率。

关键词:JavaServer Faces;权限管理系统;动态实时;表现层

中图分类号:TP302.1

文献标识码:A

文章编号:1673-629X(2010)11-0005-05

Research and Implementation of Authority Management System Based on JSF Framework

HU Qiao

(Wuhan University, Wuhan 430072, China)

Abstract:JavaServer Faces is a new technology and new standard to develop Web application in the field of Java, which provides developers with a rich library of user interface components, and shows an easy way to combine the presentation layer with the back end business logic layer. These virtue make Java Server Faces is the ideal technology to develop a Web application. The features and working mechanisms of JSF are introduced, a system about authority management has been developed based on JSF, make a special effort to explain how to generate Web pages dynamically in real time according to the need of the authority system by using the advantage of JSF. The project shows that JSF has a better flexibility and extensibility to develop Web user interface, and it helps to make the structure of Web application clarity so as to improve the developing efficiency.

Key words:Java Server Faces;authority management system;dynamically in real time;presentation layer

0 引言

在开发一个 Web 应用程序的时候,有 2 大主流技术可以应用:Microsoft 的 .Net 和 Java。对于使用过 ASP.Net 来开发网站的开发者都知道,ASP.Net 提供非常丰富的可重用的用户界面(以下简称 UI)组件,而且使用这些组件也非常的方便快捷。这就使得 ASP.Net 在开发快速小型的 Web 应用程序上比传统的 JavaServlet, JSP 技术更胜一筹。JSF 技术的出现,可以说是对 ASP.Net 在 UI 组件上的优势提出了强有力的挑战,再加上 Java 技术在跨平台和企业级应用方面的固有优势,使得 JSF 技术具有非常广泛的应用价值。

计算机技术应用的普及和企业信息化步伐的加快,以及因特网的快速发展,使得开发基于 Web 的管理信息系统成为软件发展的一个方向。基于 Web 的管理信息系统因为摆脱了地域上的限制,使信息系统的管理更加方便。Internet 所及的地方,就是 Web 管理系统可以发挥作用的地方。远程办公、远程管理因为 Internet 成为可能。但是当把管理信息系统移植到 Web 上时,权限问题便成为影响管理信息系统安全性的最重要的问题。

文中以 JSF 表现层框架为技术支持,考虑权限管理对于 Web 环境下的各种企业信息系统的重要性,以权限管理在专题数据维护中的应用为例,研究并实现基于 JSF 技术的权限管理系统。

1 JSF 技术概览

JSF(Java Server Faces)是由 Java 社区进程 JCP(Ja-

收稿日期:2010-03-05;修回日期:2010-06-24

基金项目:国家科技支撑计划课题(2007BAH16B03);空间数据挖掘与信息共享教育部重点实验室开放基金(200808)

作者简介:胡 乔(1986-),男,湖北武汉人,硕士研究生,研究方向为电子地图与网络信息系统。

va Community Process) 制定的一个 Web 应用框架标准,也是首个 Web 应用程序开发的标准框架。JSF 的目的是结束当前 Web 应用开发中框架众多混乱的局面,简化 Web 应用程序的开发。JSF 不是一门开发语言,而是由 Sun 公司推出的一种用来开发 Web 应用程序的技术,同时也是对诸多实现 Web 应用程序的规范的一种补充^[1,2]。

JSF 从核心上来看是构建 Web 应用程序界面的标准 Java 框架,JSF 的关键优势是它简化了用户界面的开发,而用户界面开发通常是 Web 应用程序开发中较为困难和繁琐的部分,虽然不用专门针对企业 Web 应用程序设计的综合框架,只用基本的 Java Web 技术(例如 Java Servlet 和 Java 服务器页面)也能构建用户界面,但是这些核心技术通常会带来各种开发和维护问题。JSF 在以前许多 Java 开发框架经验之上,提供了健壮、优良、具有制作良好开发模式的框架,从而避免了这些问题^[3]。

JSF 和传统的 Web 技术有着本质上的差别,在传统的 Web 技术需要用户自己对浏览器请求进行捕捉,保存客户端状态,并且手工控制着页面的转向等等。而 JSF 的出现,无疑给这些问题带来了新的解决办法,JSF 提供了事件驱动的页面导航模型,该模型使应用程序开发人员能够设计应用程序的页面流。与 Struts 的方式类似的是,所有的新页面流信息都定义在 JSF 配置文件(faces - config. xml)中,而非硬编码在应用程序中。这很大程度简化了开发人员开发难度,简化了应用程序的开发。同时 JSF 也是一种遵循模型 - 视图 - 控制器(MVC)模式的框架。实现了视图代码(View)与应用逻辑(Model)的完全分离,使得使用 JSF 技术的应用程序能够很好地实现页面与代码的分离。所有对 JSF 页面的请求都会通过一个前端控制器(Faces Servlet)处理,系统自动处理用户的请求,并将结果返回给用户。在 JSF 的 Backing Bean 中,可以把视图所需要的数据和操作放进一个 Backing Bean 中。得益于 JSF 使用的 DI 技术,可以在配置文件中初始化 Managed Bean,同时也可以通过这样的技术很方便地使用类似技术的 Spring 进行整合^[4]。

图 1 代表 JSF 的基本工作流程。

2 权限管理系统的设计

本系统利用 J2EE 架构技术框架,以城市专题数据更新为研究内容,构建了在 B/S 结构下的基于权限控制的专题数据更新系统,系统的权限管理方式以角

色的访问控制 RBAC(Role - Based policies Access Control)模型为基础,提出根据用户的实时权限,利用 JSF 技术动态生成页面的新思路^[5,6]。

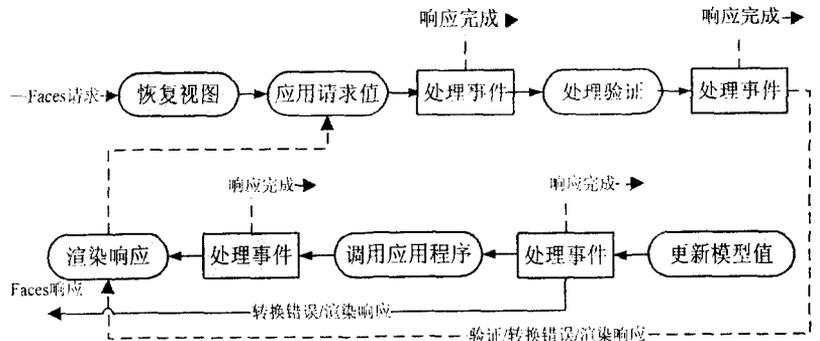


图 1 JSF 的工作流程

2.1 权限管理系统

基于角色的权限管理系统的原理是划分出不同的角色,每个角色具有符合自身特点的唯一权限。根据整个管理信息系统的权限自身的特点,每个角色可以具有一个或者多个的系统用户。如果对应角色的权限比较复杂,该角色下的用户可以有多个,但是需要保证用户的权限不能交叉^[7]。例如文中实现的数据更新权限系统中,系统的角色包括管理员、数据操作人员以及数据审核人员。对于一个大型的数据更新系统而言,数据包含的内容是非常广泛的,很多用户同属于数据操作这一角色,但是每个用户的权限是不同的,可能用户 A 具有的是对房地产方面数据的维护权限,而用户 B 具有的是对环境方面数据的维护权限。

2.2 权限管理系统的权限体现方式

基于 Web 的信息系统的体系结构一般分为三层,处于第一层的是客户端表示层,第二层是指应用服务层,而处于第三层的是数据层,通常是指后台的数据库系统。而这些组成部分中唯一能与用户联系起来的就是客户端表现层,客户端表现层通常是由一些能实现与用户交互的文件组成,它们可以是 html 文件、jsp 文件或 asp 文件等。用户是通过这些文件中的各种控件同后台进行交互。而权限管理的权限控制的最小单位就是这些控件,所有的权限都可以表现为表达权限的控件是否显示^[8]。还是以用户 AB 为例,同样是进入相同的数据操作页面,页面上显示的内容却不同,用户 A 显示的是操作房地产专题数据的链接,而用户 B 显示的却是操作环境专题数据的链接。简而言之,区别于传统方法的“预先生成而后隐藏”,文中所提出的实现方式是“动态自动生成”。

2.3 权限数据表的设计方法

根据上述的权限管理的思想,在采用这种方法的实际系统开发中使用了一个用户对应一个角色,角色

由 ROLEID 控制。有关的数据表主要包括:

(1)操作员、审核员信息表 USERINFO_U, 数据表主要记录系统数据操作人员和审核人员的基本信息,用来判断用户是否有权限进入系统以及根据用户 ID 匹配角色对照表以判断其角色。

(2)系统用户角色对照表 USER_ROLE, 数据表记录用户 ID 以及用户角色 ID,角色对照表的 USERID 与操作审核员信息的 USERID 对应。

(3)数据权限表 DATA_AUTHORITY, 数据表主要用于记录系统人员具有的权限,表中的一条记录代表一个权限,记录具体是哪位系统人员对哪个专题具有相应的何种权限。表中专题视图是数据表的主键为视图名,代表一个视图只能具有唯一的数据操作人员,不允许有多个人员同时具有对同一视图的相同权限。其中为了保证对用户登录表的安全性,采用加密算法对用户 ID、登录日志、登录密码进行了加密处理。

3 基于 JSF 框架的权限系统的具体实现

在整个权限系统的组成中,功能模块的实现是由 JSF 标签(RichFaces)组成的 JSP 页面程序、数据库程序(主要包括权限数据的存储过程)和处于应用服务层的 JavaBeans 程序的调用来组成,各部分之间的连接通过程序接口规范的文档来说明。下面以系统中的权限数据显示模块的实现作为示例来说明如何利用 JSF 技术实现权限系统的开发。

图 2 为权限系统的基本工作流程。

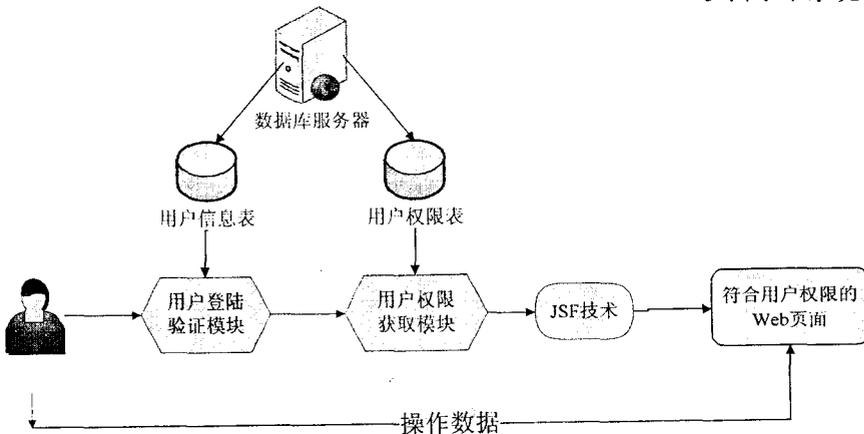


图 2 权限系统的基本工作流程

权限数据显示的含义是指由于系统每位数据操作人员的权限不同,所以能够管理和查看的数据只是符合自己权限的那一部分^[9]。主页面包括:左边是包含该用户权限的可折叠的权限目录,这样就可以保证用户只可能看到符合自身权限的数据。右边部分包含用户选择的数据表格和具有相应功能的多个按钮供用户

对数据表格进行操作,主要指对数据的增删查改。本页面是通过包含大量 RichFaces 标签的动态 JSP 页面文件和对应该页面文件的后台 JavaBean 共同组织实现。

RichFaces 是一个比较新颖的概念,简单来说, RichFaces 是一个拥有 Ajax 功能的 JSF 组件库^[10]。JSF 和 Ajax 的应用已经比较普遍,二者结合起来就衍生出了 Ajax4JSF,其目的就是将 Ajax 的功能集成到 JSF 组件中去,而 RichFaces 就是 JBoss 在收购 Ajax4JSF 后集成出来的产品, RichFaces 为构建 RIA 和支持 Ajax 的 Web 应用程序提供了大量的 JSF 组件,不但丰富了控件库,还添加了很多新功能,它的最大的优势是可以使开发者不再关注于繁琐的 JavaScript 实现, RichFaces 将这一切都变成透明的,开发人员只要按照 JSF 的组件规范来组织页面,免去了书写和调用庞杂的 JavaScript 代码或库,就能达到梦寐以求的 Ajax 效果,增强了用户体验^[11]。

由于系统采用的是折叠型的面板条控件,而为了实现系统用户只能处理符合自己权限的数据,所以这里的折叠型的列表所包含的内容不是固定的,而是根据每个系统用户的权限的不同生成的。而如何根据登录系统的用户的权限去实时动态生成列表则是整个权限系统最为关键的技术难点。

RichFaces 关于折叠型的面板条控件是 panelBar, panelBar 中必须包含多个面板条容器 (panelMenuGroup), 而 panelMenuGroup 中的内容是 panelMenuItem。由于本系统的数据是以表格的形式进行组织,

每一个表格代表一个权限,所以 panelBarItem 中放置的内容就必须严格地体现出用户的权限,举个例子,如果用户 A 具有对房地产和综合经济的表格数据操作的权限,这里就有且只能有二个 panelMenuItem,分别对应这两个权限。上面已经介绍过 JSF 的基本工作原理,如果权限系统的前台 JSP 页面按照固定组件的写法应该写为如下形式:

```

<rich:panelMenu>
<rich:panelMenuGroup>
<rich:panelMenuItem label="房地产" style="font-size: 10px;
font-family: Arial"
action="#{SZUserNameSelect_bean.opItem}"
reRender="superview">

```

```

</rich:panelMenuItem>
< rich:panelMenuItem label = “综合经济” style = “font - size:
10px;font - family:Arial”
action = “# {SZUserNameSelect - bean. AddOpUser}
“reRender = ”superview”>
</rich:panelMenuItem>
</rich:panelMenuGroup>
</rich:panelMenu>

```

这里的 action 绑定到后台 JavaBean 中的函数,代表点击相应条目后触发的动作,即显示该专题下的数据表格。但是这种方式显而易见是不合理的。因为可以看出在该 JSP 中所有 panelMenuItem 的数目和 label 都是固定的。对于一个权限系统中的多个用户,每个用户的权限是完全独立的,是不可能为每一个用户各自设计单独服务于其自身的 JSP 页面。尽管一般的 JSP 页面都是采用这种方式,但是同样 JSF 提供另外一种方式去生成前台 JSP 页面中的组件,就是在 JavaBean 中提供一个函数,而利用 JSF 组件提供的绑定功能将所需要生成的组件绑定到这个函数上,由于每次在请求该 JSP 页面时,都会实例化与该 JSP 页面上组件相关的所有 JavaBean,所以初始化绑定组件的函数也会相应执行。

根据权限系统的流程图,首先需要得到的是正确登录用户的权限。然后根据所得到的权限去动态生成相应的组件。具体的 JSP 页面如下:

```

<rich:panelMenu mode = “ajax” binding = “# {SZAtlasMainten-
bean. optreePanelMenu}”></rich:panelMenu
>

```

代码中表示将一个 panelMenu 组件绑定到名为 SZAtlasMainten-bean 的后台 JavaBean 中的 optreePanelMenu 函数上。另外 mode = “ajax”代表将 Ajax 功能赋予组件。

再来看看 optreePanelMenu 函数是如何实现动态生成权限列表的:

```

public HtmlPanelMenu getOptreePanelMenu()
{
    JdbConn MenuDB = new JdbConn();
    ResultSet MenuRS = null;
    String getTheMaticListSQL = “select dis-
tinct thematic - name from
data - authority where operator - name = “ + sUserName + ””;

```

```

(首先获得数据库连接,根据 sUserName(进入权限系统的用户名)建立 Sql 语句,从权限数据表中查出符合该用户权限的 thematic - name(专题数据名))
optreePanelMenu. getChildren(). clear();
MenuRS = MenuDB. getRs(getTheMaticListSQL);

```

```

HtmlPanelMenuGroup myPanelMenuGroup = new HtmlPanel-
MenuGroup();
(将 optreePanelMenu 清空,然后生成 HtmlPanelMenuGroup 作为
HtmlPanelMenuItem 的容器)
myPanelMenuGroup. setLabel(“权限列表”);
try {
    int j = MenuDB. getRsRowsNum(MenuRS);
    for (int i = 1; i <= j; i++) {
        MenuRS. next();
        String authoriName = MenuRS. getString(1);
        HtmlPanelMenuItem myPanelMenuItem = new HtmlPanelMenu-
Item();
        myPanelMenuItem. setLabel(authoriName);
        myPanelMenuItem. setId(“MMG” + i);
        (循环得到权限,对每一次得到的权限都生成一个
新的 HtmlPanelMenuItem,设置其 Label 为权限的名
字,另外需要保证每一个 Item 的 ID 不同)
myPanelMenuGroup. getChildren(). add(myPanelMenuItem);
        (每次循环都将新生成的 HtmlPanelMenuItem 加
入到前部分生成好的 HtmlPanelMenuGroup)
optreePanelMenu. getChildren(). add(myPanelMenuGroup);

```

通过这样的方式,可以根据进入系统的用户名的不同实时地从后台的权限数据库中读出相应的权限动态地生成正确的列表控件。例如分别利用数据操作人员 A 和 B 帐号进入系统,由于 A, B 用户权限的差别,系统会在左边的目录树中详细体现权限的差别。

权限系统界面如图 3 和图 4 所示。

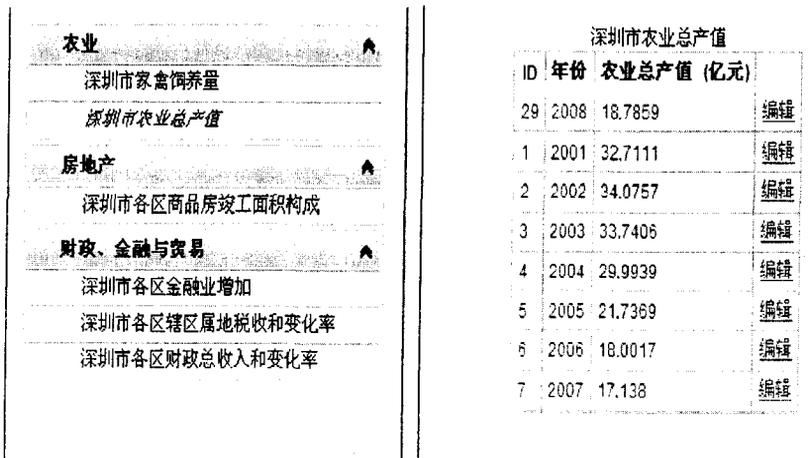


图 3 系统数据操作人员 A 页面

4 结束语

JSF 技术使得企业级的 Web 开发变得有章可循,可以提供给开发人员更高的开发效率和更好的软件质量。虽然目前仍然存在不足,但相信以后的应用将会越来越广泛。

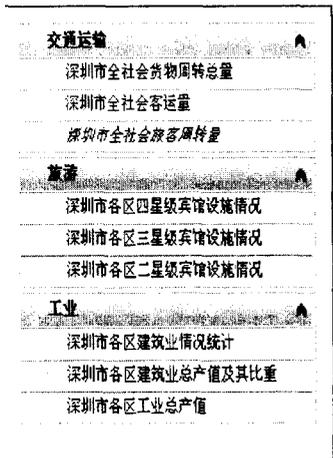


图 4 系统数据操作人员 B 页面

ID	年份	铁路(公里)	公路(公里)	水路(公里)	民航(公里)	编辑
1	2006	36.62	77.15	150.97	0.95	编辑
2	2007	46.31	82.4	180.11	1.08	编辑

文中借助 JSF 技术,研究并构建了基于 JSF 框架的权限管理系统,提出并实现了利用 JSF 表现层技术实时获得权限进而动态生成满足权限要求的页面。能够高效地保证系统权限管理更为科学、快捷和准确。利用本方案已完成某市城市专题数据维护系统的开发。现在该系统正在使用,并得到用户好评。

参考文献:

- [1] Burns E, Horwat J. An Introduction to JavaServer Faces [EB/OL]. 2004-08-02. <http://java.sun.com/j2ee/javaserver-faces/jsfintro.html>.
- [2] McClanahan C, Burns E, Kitain R. JavaServer Faces Specification 1.1 [S/OL]. 2004-08-02. <http://java.sun.com/>
- [3] 汤琪亮, 高海锋. JavaServer Faces 技术在网络开发中的应用[J]. 计算机工程与设计, 2006, 27(10): 1895-1898.
- [4] 侯雅楠. Sun JSF 的诱人特性[J]. 中国计算机报, 2004, 1304: 2-4.
- [5] 邝文清, 郭跟成. 基于 JSF 框架 Web 应用开发的研究[J]. 计算机应用研究, 2007, 24(12): 272-275.
- [6] 李 仲, 杨宗凯, 刘 威. 一种基于 RBAC 的实现动态权限管理的方法[J]. 计算机技术与发展, 2006, 16(10): 1-4.
- [7] Sandu R S, Coyne E J, Feinstein H L. Role based access control mcxtels[J]. IEEE Computer, 1996, 29(2): 38-47.
- [8] 杨振宇, 耿玉水, 王新刚. 基于 B/S 模式可复用的用户权限管理的设计与实现[J]. 计算机应用与软件, 2007, 24(6): 123-125.
- [9] 王 峰, 江勤绕, 俞欢军. 基于 JSF 框架的信息管理系统的设计和实现[J]. 计算机工程与设计, 2007, 28(21): 5221-5224.
- [10] 暴志刚, 胡艳军, 顾新建. 基于 Web 的系统权限管理实现方法[J]. 计算机工程, 2006, 32(1): 169-170.
- [11] JBoss Community. RichFaces Developer Guide [EB/OL]. 2009. <http://www.jboss.org/richfaces/docs/referenceguide.html>.
- [12] 董 平, 秦雅娟, 张宏科. 支持普适服务的一体化网络研究[J]. 电子学报, 2007, 35(4): 599-606.
- [13] 杨 冬, 周华春, 张宏科. 基于一体化网络的普适服务研究[J]. 电子学报, 2007, 35(4): 607-613.
- [14] 高文宇, 陈松乔, 王建新. IP 网络接纳控制研究评述[J]. 计算机科学, 2004, 31(9): 6-10.
- [15] Farooq M O, Aziz S. Admission Control and Multipath Routing Algorithm for Differentiated Services Based Networks [C] // 2009 International Conferences on Complex, Intelligent and Software Intensive Systems. New York: IEEE, 2009: 359-366.
- [16] Van Mieghem P, De Neve H, Kuipers F. Hop-by-hop quality of service routing [J]. Computer Networks, 2001, 37: 407-423.
- [17] De Neve H, Van Mieghem P. TAMCRA: a tunable accuracy multiple constraints routing algorithm [J]. Computer Communications, 2000, 23: 667-679.

(上接第 4 页)

较好。

参考文献:

- [1] 杨 勇, 王雪晶, 陈良臣. QoS 在 IP 中的研究和应用[J]. 计算机技术与发展, 2007, 17(5): 33-36.
- [2] 林 闯, 王元卓, 任丰原. 新一代网络 QoS 研究[J]. 计算机学报, 2008, 31(9): 1525-1535.
- [3] 张宏科, 苏 伟. 新网络体系基础研究——一体化网络与普适服务[J]. 电子学报, 2007, 35(4): 593-598.
- [4] 李 茹, 王春峰, 黄晓璐, 等. IPv6 流标签提供的服务质量支持[J]. 计算机科学, 2004, 31(4): 8-13.
- [5] 易 侃, 王志坚. 基于 IPv6 的标签交换体系结构的研究[J]. 计算机技术与发展, 2006, 16(6): 227-229.
- [6] Sun Qiong, Huang Xiaohong, Ma Yan. IPv6 end-to-end QoS provision in heterogeneous networks using aggregated flow label [C] // in Proceedings of WRI World Congress on Computer Science and Information Engineering (CSIE 2009). Los Angeles, United States: IEEE Computer Society, 2009: 438-441.