

基于 Struts 与 Ajax 的 OA 系统的研究与实现

周 浩, 罗立宇, 李 欣

(湖南工业大学 师专校区 数学与计算机科学系, 湖南 株洲 412007)

摘 要: 研究并实现了一个基于 MVC2 模型、Struts 框架与 Ajax 模式的办公自动化系统。讨论了系统的整体设计目标, 以及如何将 Struts 技术与 Ajax 模式相结合, 采用 ERBAC 模式控制系统权限等关键技术。实际应用表明, 该系统设计合理, 功能完善。

关键词: Struts; Ajax; MVC2; OA 系统; ERBAC

中图分类号: TP311

文献标识码: A

文章编号: 1673-9833(2008)03-0046-04

Research and Realization of OA System Based on Struts and Ajax

Zhou Hao, Luo Liyu, Li Xin

(Department of Mathematics and Computer Science, Teachers College,
Hunan University of Technology, Zhuzhou Hunan 412007, China)

Abstract: Having studied and implemented one office automation system based on MVC2 model, Struts frame and Ajax pattern, it discussed the systematic overall design object, adopted key technology of ERBAC pattern control system authority, and how to combine Struts technology and the Ajax pattern. The practical application indicates that system design is reasonable and the function is perfect.

Key words: Struts; Ajax; MVC2; OA system; ERBAC

文章介绍自行设计和开发的基于 Struts 的网络办公自动化系统 (Office Automation)。在开发过程中, 针对传统 OA 系统的不足和客户要求, 系统作出了以下几点技术改进: 1) 针对客户对办公自动化系统的可扩展性、可维护性和可重用性提出了越来越高的要求采用了多层开发; 2) 针对 WEB 开发存在下达不确定性、执行过程交互性等缺点, 采用 Ajax (Asynchronous JavaScript and XML) 技术; 3) 直接使用 J2EE 开发周期长, 使用成熟的开发框架 Struts, 同时把 Ajax 技术融合在 Struts 表示层中。该 OA 系统是利用 Struts 框架将 Ajax 在表示层的魅力与 J2EE 在服务器端的强大支持结合起来所建立的全新的 Web 应用。

1 Struts 框架与 Ajax 技术

1.1 Struts 框架概述

Struts 是一种成熟的 J2EE 框架。Struts 是 MVC2 的一种具体实现^[1], 是一组相互协作的类、servlet 和 JSP 标记, 三者组成一个可重用的 MVC2 设计。这个定义表示 Struts 是框架, 而不是一个类库。Struts 继承了 MVC 各项特性, 并根据 J2EE 的特点, 做了相应的变化与扩展。

Struts 的工作原理, 如图 1^[2]所示:

客户端 (Client browser) 表示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

收稿日期: 2008-03-23

基金项目: 湖南省教育厅优秀青年科研基金资助项目 (06B103)

作者简介: 周 浩 (1981-), 男, 湖南湘潭人, 湖南工业大学教师, 硕士研究生, 主要研究方向为软件与数据库;
罗立宇 (1969-), 女, 湖南茶陵人, 湖南工业大学副教授, 硕士, 主要研究方向为计算机网络与数据库;
李 欣 (1980-), 女, 湖南安化人, 湖南工业大学教师, 硕士研究生, 主要研究方向为数据挖掘。

响应作出响应。控制器 (Controller) 接收来自浏览器的请求 (Request), 并决定将这个 Request 发往何处。就 Struts 而言, 控制器是以 Servlet 和 Action 实现的命令设计模式。控制器中的 Servlet 通过调用 struts-config.xml (文件配置控制器) 中信息, 完成信息验证并选择对应的 Action 类。Action 类完成决定业务逻辑更新模型的状态, 帮助控制应用程序响应的流程。模型 (Model) 完成业务逻辑的实现。ActionForm bean 在控制层或请求层表示模型的状态, JSP 文件使用 JSP 标记读取来自 ActionForm bean 的信息。视图 (View) 就是 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, 通过标记来显示模型层响应的数据。

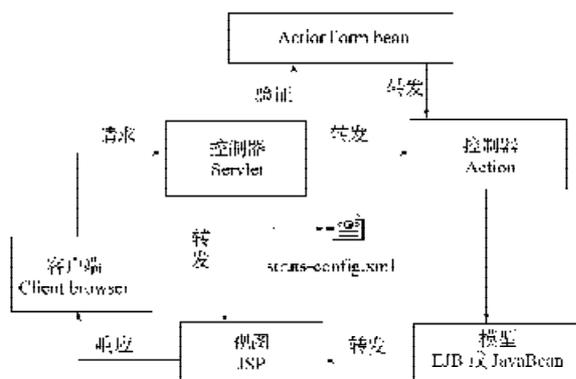


图 1 Struts 工作原理图

Fig. 1 Struts operating principle picture

1.2 Ajax 模式概述

1.2.1 Ajax 模式的技术特征

Ajax 即异步 JavaScript+XML (可扩展标记语言), 是由目前几种相对成熟的技术组合而成的。这组技术包含: 基于 XHTML 和 CSS 标准的表示; 用 DOM (Document Object Model) 动态显示和交互; 用 XMLHttpRequest 与服务器异步通信; 使用 XML 和 XSLT 进行数据交换与处理; 用 JavaScript 绑定一切数据 5 项内容。其中 JavaScript、DOM、XML、XMLHttpRequest 是 Ajax 的核心^[3]。

较之传统的 Web 应用模型, Ajax 模式在客户端加入用于分离页面呈现和应用逻辑的中间层 (也称之为“Ajax 引擎”), 减轻了服务器的负担, 减少了用户操作等待的时间。

1.2.2 Ajax 模式工作原理

Ajax 的工作原理相当于在用户和服务器之间加了一个中间层, 使用户操作与服务器响应异步化。并不是所有的用户请求都提交给服务器, 像一些数据验证和数据处理等都交给 Ajax 引擎自己来做, 只有确定需要从服务器读取新数据时再由 Ajax 引擎代为向服务器提交请求。Ajax 异步交互机制详细工作步骤如下:

1) 用户向服务器提交请求时, 页面程序通过 JavaScript 创建 XMLHttpRequest 对象;

2) 通过 XMLHttpRequest 对象的 open 方法向服务器发送请求, 可以显式设置处理请求的服务器页面地址、HTTP 请求方式, 以及是否支持异步模式等参数;

3) 与此同时, 服务器端程序对 XMLHttpRequest 对象请求进行处理 (一般将服务器端数据结果以 XML 格式返回), 并改写 XMLHttpRequest 对象的请求状态;

4) 通过 JavaScript 回调函数处理服务器端响应: 将 XMLHttpRequest 对象的 OnReadyStateChange 属性设置为用于监听服务器响应的回调函数名; 当服务器端响应到达客户端时, 触发回调函数对 XMLHttpRequest 对象的请求状态进行检查, 并决定是否处理服务器的响应数据; 最后, 在回调函数中 DOM 动态地更新用户界面, 如在页面的局部添加或更新服务器返回的数据。

2 采用 Struts 和 Ajax 模式实现 OA 系统

2.1 OA 系统整体设计结构

本系统采用基于 MVC2 模型的多层体系结构来设计系统的基本框架, 在视图层和控制层采用 Struts 和 Ajax 技术相结合 (文中针对以上技术展开讨论)。此外, 在模型层采用 JavaBean 调用持久层, 通过持久层访问数据库; 在完成系统公文流转功能时, 采用了 OSWorkflow 工作流引擎。按照结构化的系统设计方案, 本网络办公系统从功能上可以划分成几个子系统, 如图 2 所示。

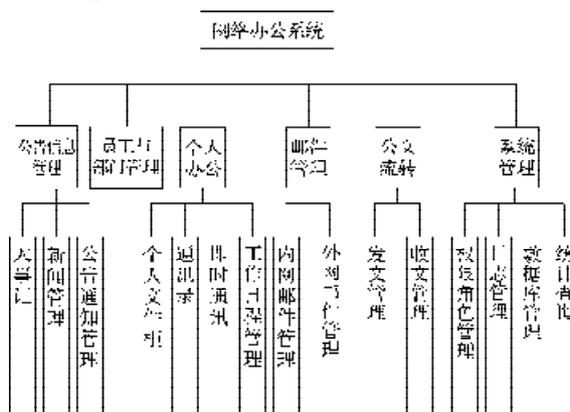


图 2 OA 系统结构功能图

Fig. 2 OA system structure function picture

系统包括公告信息管理 (大事记、新闻管理、公告通知管理)、员工与部门管理、个人办公 (个人文件夹、通讯录、即时通讯、工作日程管理)、邮箱管理 (内、外网邮件收发)、公文流转 (发文管理、收文管理)、系统管理 (权限角色管理、日志管理、数据库管理、统计查询) 等 6 大模块, 是功能强大而又安全的网络办公平台, 拥有统计查询工具、快捷图标工具栏、短信息栏等实用、便捷的工具群。

2.2 表示层采用 Ajax 技术

本 OA 系统在员工管理、日程管理、即时通讯、公告管理、权限分配等模块中,采用了 Ajax 技术和 Struts 技术结合完成控制层和视图层的显示,提高了系统运行的效率,也让系统具备了更好的可维护性和扩展性。系统原理如图 3 所示。

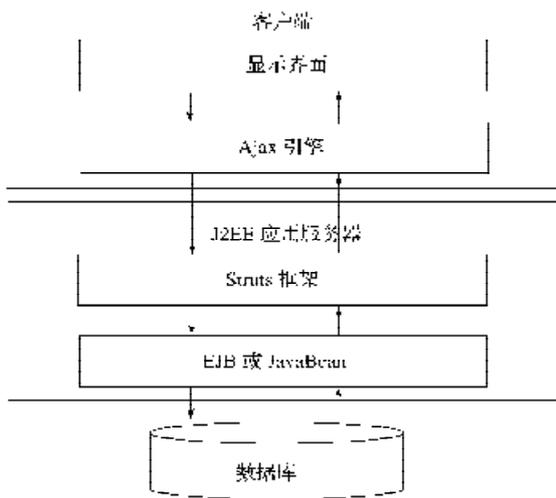


图 3 OA 系统工作原理图

Fig. 3 OA system operating principle picture

日程管理模块的目的是帮助企业员工更加合理、有效地记录、管理日常工作,并对重要日程进行定时提醒,不仅达到工作备忘的目的,同时也方便领导对员工日常工作的指导、监督。下面以日程管理的日程提醒功能为例,详细说明两种技术结合的具体实现。

2.2.1 Ajax 核心对象的生成

XMLHttpRequest 对象是与后台交互的核心,正是有了主流浏览器对 XMLHttpRequest 对象的支持,才使得我们可以使用 Ajax 在 B/S 结构上构建如此强大的网络应用。自定义生成 XMLHttp Request 对象的 JavaScript 函数 newXMLHttpRequest(), 在该函数中,根据不同的浏览器生成对应的 XMLHttpRequest 对象。具体格式为:

```
var http_request=false;
function remind(){
http_request=newXMLHttpRequest();
.....}
```

remind 函数中创建 http_request 对象后,就可以该对象调用 open 方法向服务器请求获得 XML 数据了。

```
http_request.open("post", "/PandaStarOA/remindCalendar.do", true);
```

```
http_request.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
```

```
http_request.onreadystatechange=processRequest;
```

```
http_request.send();
```

```
setTimeout(remind, 300000);
```

open 方法中第一个参数 http-method 指明 HTTP 的通信方式,为向 Struts 请求数据,建议采用 post 方式;第二个参数 url 接收 XML 数据的服务器的 URL 地址;第三个参数 async 是一个布尔标识,它说明请求是否为异步的。如果是异步通信方式(true),客户机就不等待服务器的响应;如果是同步方式(false),客户机就要等到服务器返回消息后才去执行其它的操作。

2.2.2 调用 Struts 配置信息,寻找处理的 Action 类

调用 Struts 配置文件 struts-config.xml,从中获得处理请求 Action 类 RemindCalendarAction,具体配置信息为: <action path="/remindCalendar" validate="false" type="com.struts.action.workplan.RemindCalendarAction"/>。

<action>元素的 path 属性指定请求访问 Action 的路径, type 属性指定 Action 的完整类名, validate 属性以布尔值指定是否验证信息,此处无须验证,值为 false。

类 RemindCalendarAction 继承了 Action 类,重载方法 execute() 完成对 Ajax 客户端数据的响应:

```
InterDaoCalendar idc=new ImpDaoCalendar();
String strXML=idc.getRemindCalendar(employeeid);
response.setContentType("text/xml");
response.setHeader("Cache-Control", "no-cache");
response.setCharacterEncoding("utf-8");
try {response.getWriter().write(strXML);
} catch (Exception e) {
e.printStackTrace(); }
return null
```

2.2.3 调用回调函数,解析 Dom 文档

数据从服务器端转发回来后,就可在 process Request 函数中解析数据了,使用 JavaScript 标准的 DOM 对象可解析返回的 XML 格式的数据:

```
var calendarslist=objDom;
getElementsByTagName("calendars")[0];
var calendarlist=calendarslist;
getElementsByTagName("calendar");
var calCount=calendarlist.length;
for(var i=0; i<calCount; i++){
...window.open(strURL, "_blank", css); }
.....
```

window.open 方法展开新的提示窗口, strURL 参数指定从 Struts 配置文件中获取显示数据信息的 Action 类 ShowRemind-Action,并通过类中 request 对象转发信息到显示数据页面 remindCalendar.jsp, css 参数指定显示窗口样式信息。

2.2.4 渲染客户端数据显示

remindCalendar.jsp 中完成提醒信息的显示:

```
<bean: message key= "com.workplan.title" />
<bean: message key= "com.workplan.content" />
<bean: write name= "title" />
<bean: write name= "content" />
```

<!-- 从 request 对象中, 获取内容提示信息, 并通过 struts 标签显示在页面上 -->

以上实现效果如图 4 所示。

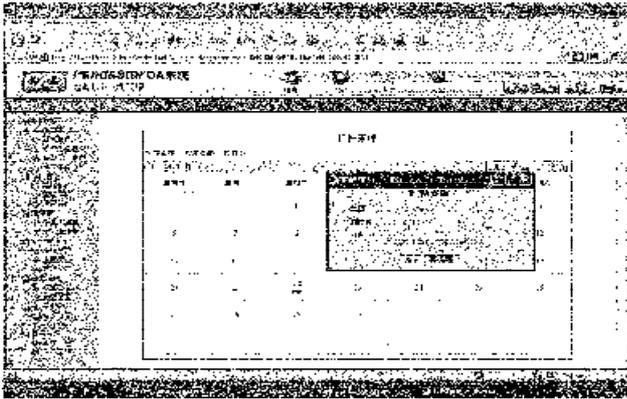


图 4 日程提醒图

Fig. 4 Calendar remind picture

2.3 基于 ERBAC 权限分配模式

企业级 OA 系统对安全问题有较高的要求, 因此系统除了采用 Ajax 提高页面效率外, 在功能权限上采用了 ERBAC 模式。

ERBAC 模型是对传统 ERBAC 模型动态性的扩展, 其基本思想是: 用户只能在允许的时空范围内使用合法的权限对数据进行访问。用户只能使用自己拥有的活跃角色 (active role) 所包含的权限进行相应的操作; 角色响应用户操作请求 (operation request) 或系统事件 (events) 而由休眠状态 (sleeping) 变迁 (transition) 为活跃状态 (activating) 或由活跃状态变迁为休眠状态, 从而实现角色的动态管理。

用户 u_i 在某存取时刻 t 上的带权角色安全系数定义为:

$$\alpha_i^t = \frac{\sum_{r_i=1}^n w_i \times |r_i|}{\sum_{r_i=1}^n w_i \times |r_i|}$$

其中, w_i 为角色 r_i 的安全权重 (weight), n 为用户 u_i 在

时刻 t 拥有的角色总数, p 为 u_i 在时刻 t 拥有的活跃角色数, $|r_i|$ 为角色 r_i 包含的权限数。 α_i^t 值越小, 表示在时刻 t 用户 u_i 拥有的角色中活跃 (重要) 角色越少, 因而用户安全性越高, 这也符合存取控制授权的最小特权原则^[4, 5]。反之, α_i^t 值越大, 表示在时刻 t 用户 u_i 拥有的角色中活跃 (重要) 角色越多, 因而用户安全性越低。角色安全系数可用于分析、比较和评价不同用户的角色授权方案的安全程度。

在 ERBAC 模型中, 通过引入角色环境函数和角色约束规则对传统的 ERBAC 模型进行了扩展, 对用户的角色实行活跃和休眠两种状态管理, 避免了传统 ERBAC 模型中一旦用户拥有某角色, 就可以无限地随时随地享有该角色操作权限而造成的系统不安全问题。使用 ERBAC 的主动数据库功能实现的 ERBAC 具有较好的主动角色管理功能, 该模型在系统中应用对于提高系统安全性, 效果良好。

3 结语

从软件工程及实用性两方面来看, 衡量一个应用系统开发成功的主要标志, 是系统开发完成后能够投入正常的使用。该系统自 2006 年底开发完成后投入试用, 至今已有 1 年余, 工作稳定, 经受了时间的考验, 表明该系统设计合理, 功能实用, 满足了用户需求。

参考文献:

- [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, 2005.
- [2] Malcolm G. Davis. MVC 的一种开放源码实现[J/OL]. [2001-02-10]. http://www.ibm.com/developerworks/cn/java/j-struts/?S_TACT=105AGX52&S_CMP=techccid.
- [3] 陈志平, 徐锡山, 陈玉教. 基于 AJAX 的动态树型结构的设计与实现[J]. 计算机与信息技术, 2005 (12): 8-11.
- [4] Sylvia Osborn, Ravi Sandhu, Qamar Munawer. Configuring role-based access control to enforce mandatory and discretionary access control policies [J]. ACM Transactions on Information and System Security, 2000, 3(2): 85-106.
- [5] Sandhu R, Coyne E J, Feinstein H L, et al. Role-based access control model[J]. IEEE Computer, 1996, 29(2): 38-47.

(责任编辑: 罗立宇)