

基于 Ajax 和 MVC 架构的研究与应用*

林剑雄, 周晓慧*

(杭州电子科技大学 自动化学院, 浙江 杭州 310018)

摘要: 现有的 Ajax 和 MVC 结合方案都存在着增加系统结构和实现的复杂性、带来额外的工作和降低运行效率等缺点, 不适合小型甚至中等规模的应用程序。为了解决这个问题, 提出了一种改进的基于 Ajax 和 MVC 的架构。并以网上聊天室为例, 实现了基于 Ajax 和 MVC 架构的 Web 应用。研究表明, 该架构适合中小型 Web 项目的开发, 达到了减轻服务器的负担、提高系统的响应速度、提供流畅的视觉浏览等效果。

关键词: Ajax; MVC; 异步请求; 聊天室

中图分类号: TP311.52

文献标识码: A

文章编号: 1001-4551(2010)04-0090-03

Study and application based on Ajax and MVC structure

LIN Jian-xiong, ZHOU Xiao-hui

(Institute of Information and Control, Hangzhou Dianzi University, Hangzhou 310018 China)

Abstract: All the existing schemes combining Ajax with MVC are not suitable for small or medium-scale applications. The complexity of system is increased, additional work is brought and the operational efficiency is reduced. Aiming at solving this problem, an improved structure based on Ajax and MVC was discussed. Finally, the Internet chatting room was illustrated as an example to carry out the Web application based on the Ajax MVC structure. The results indicate that it is suitable for small and medium-sized Web projects, and the burden on the server can be reduced the system response speed is improved and smooth visual browsing is proved.

Key words: Ajax; model-view-controller(MVC); asynchronism request; chat room

0 引言

MVC(model-view-controller)是 Xerox PARC 在 20 世纪 80 年代为编程语言 Smalltalk-80 发明的一种软件设计模式。尽管 MVC 设计模式很早提出,但在 B/S (Browser/Server)结构的项目中引入 MVC 却是步履维艰。随着 Sun 公司 J2EE 的发展, MVC 最近几年在 Web 开发中得以广泛应用。MVC 设计模式实现了网页和业务逻辑的分离,提高了系统的可维护性、可扩展性、可移植性和组件的可复用性^[1]。

传统 Web 开发以同步方式发送请求。客户端发送一次请求,就必须刷新整个页面,提交整个页面的数据,降低了 Web 应用的执行速度和响应时间,增加了服务的负担。特别地,在对实时性要求较高的应用中,传统 Web 技术根本解决不了问题。Ajax 的出现改变

了这种局面,实现了以异步方式发送请求,减轻了服务器的负担,大大加快了响应速度,缩短了用户等待的时间^[2]。随着 Web2.0 的发展, Ajax 技术逐渐兴起,在 Google、微软的大力支持下,得到迅速的推广和应用。但是,易读性差、代码混乱和难以调试却成了制约 Ajax 发展的瓶颈。

在传统的 Web 开发中已经能够很好地运用 MVC 设计模式来实现所需的应用,但是引入 Ajax 后,随着应用模式的改变,对异步通讯模型支持的缺乏,使客户端开发、与 Web 框架的集成、标记库的使用等方面都受到了影响。因此,本研究探讨一种中小型 Web 应用的架构—基于 Ajax 的 MVC 架构。首先分别介绍 Ajax 技术和 MVC 开发模式的基本原理;然后总结现有的整合方案,再对这两种技术进行整合,详细分析了该构架工作原理;最后以网上聊天室为例,实现了该架构,并实现了较为理想的效果。

收稿日期:2009-08-31

基金项目:浙江省科技厅攻关项目(2008C11010)

作者简介:林剑雄(1984-),男,广东揭西人,主要从事计算机控制方面的研究, E-mail: linj0123@163.com

通信联系人:周晓慧,男,教授,硕士生导师. E-mail: zhouxh@hdu.edu.cn

1 基于 Ajax 的 MVC 架构的研究

1.1 Ajax 基本原理

其实, Ajax (Asynchronous JavaScript Technology And XML)并不是一种新技术,而是 JavaScript、XHTML、CSS、DOM、XMLHttpRequest 等技术的综合^[3]。其基本原理是:由页面事件触发,调用 JavaScript 创建一个 XMLHttpRequest 对象,通过 Ajax 引擎向服务器发送请求,从服务器获得数据,然后用 Javascript 来操作 DOM 而更新页面。XMLHttpRequest 是 Ajax 的核心机制,是一种支持异步请求的技术^[4]。简单的说,也就是 JavaScript 可以及时向服务器提出请求和处理响应,而不阻塞用户,达到无刷新的效果。其基本原理如图 1 所示。

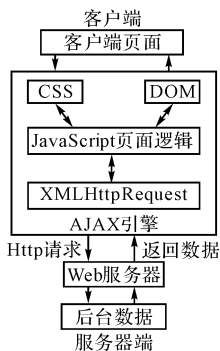


图1 Ajax 基本原理图

1.2 MVC 基本原理

MVC 包括模型、视图和控制器 3 个组件。当 MVC 设计思想应用于 Web 时,视图就表现为页面表示层,负责有关页面数据的显示。模型是应用程序的主体部分,负责有关业务逻辑和业务数据的处理,一个模型能被多个视图重用,提高了应用的复用性。控制器主要用来控制数据流向,接受用户的请求,并把请求分配给模型,最后调用视图响应用户的请求^[5]。MVC 的基本原理如图 2 所示。

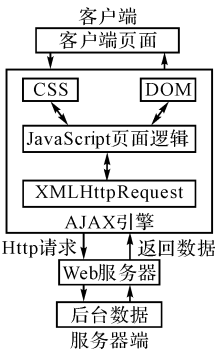


图2 MVC 基本原理图

在 MVC 模式中,各层各司其职,降低了系统间的耦合,有利于开发中的分工及组件的重用。MVC 模式结构清晰,易于扩展和维护。

1.3 基于 Ajax 的 MVC 构架

目前,Ajax 在 Web 应用开发中,与 .Net、PHP 的结合已经有了很多的实例,并且有比较成功的开发框架出现,比如 Altas(与 .Net 结合的开发框架)。而相对于其他平台,J2EE 的应用范围更为广泛。比较流行的 J2EE 开发框架有 Struts 和 Spring。

Struts 是一个典型的 MVC 设计模式开发框架,其优点主要体现在两个方面:Taglib(Struts 的标记库)和页面

导航。Struts 2 内置了对 Dojo 工具包的支持,它的 Ajax 标签基于 Dojo Widget,为在应用程序中使用 Ajax 提供了快捷的方式。同时,Struts2 也支持 DW2 和 JSON 插件,可以利用 JSON 作为 Ajax 请求和响应数据的载体,也可以利用 DW2 实现 Ajax 验证。但是,Struts 中的 Ajax 应用还存在着一些 Bug,例如跨浏览器和传值问题。

Spring 是一个非常优秀的轻量级框架,由于 Spring 具有的优点,在 Spring 中整合其他应用变得十分简单。Spring 与 Ajax 的整合可以通过 Spring 跟 DWR、JSON 等框架的整合来实现。通过 IOC 就可以把具体的 Ajax 注入到业务应用中,而 AOP 则能动态地决定切入点。可以说,Spring 跟 Ajax 的整合可以达到无缝结合。然而,由于过于注重设计原则、高内聚和低耦合,Spring 存在着复杂、配置文件繁琐、运行效率降低等缺点。

在众多的 Ajax 和 MVC 模式结合的方案中,虽然都基本解决了在 MVC 模式下方便使用 Ajax 的问题,但是都存在着增加系统结构和实现的复杂性、带来额外的工作和降低运行效率等缺点,不适合小型甚至中等规模的应用程序。本研究结合 Ajax 和 MVC 的特点和优点,模仿 Struts 的 MVC 和 Hibernate 的数据层处理,介绍一种中小型 Web 开发的架构。其模型如图 3 所示。

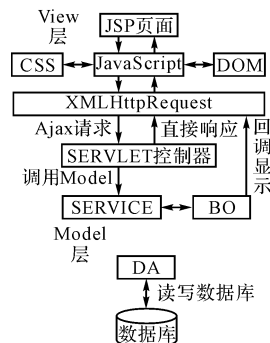


图3 基于 Ajax 的 MVC 构架原理图

(1) 视图设计。

由于 JSP 具有显示动态信息、一次编写、到处运行的优点,采用 JSP 作为视图,显示页面数据。同时,引进 Ajax 技术,使得页面数据能够以异步的方式发送,达到实时刷新、减轻服务器负担的效果。

(2) 模型设计。

在 MVC 思想中,模型负责业务数据和业务逻辑的处理,而业务逻辑又跟数据库紧密联系在一起。所以,业务数据从模型中抽象出来,成为一个独立层。该层主要是用 Javabean 来封装数据,称为“Bessiness Object”,简称“BO”。业务逻辑也设计为一个独立层,专门负责业务逻辑的处理,称为“Service”。为了降低数据库跟 DA 的耦合和提高组件的重用性,在数据库跟

“Service”之间设计了一个数据库接口层,该层称为“DataAccess”,简称“DA”。即用数据库连接池连接数据库,提供数据库查询和执行接口。考虑到数据库的一致性和完整性,该架构使用了事务管理策略。同时,为了保证每次只有一个数据库在运行,又使用了设计模式中的单件模式(Singleton Pattern)^[6]。

(3) 控制器设计。

控制器通过 Servlet 实现。Servlet 是用 Java 写的服务端程序,它与平台无关,可以动态扩展 Server 的能力,并用请求—响应模式提供 Web 服务。通过配置 web.xml 文件,把页面请求分配给相应的 HttpServlet 程序。在 HttpServlet 程序中调用模型,对请求进行处理,并把结果数据进行封装,响应给视图。为了理清类与类之间的关系,采用了设计模式中的外观模式和代理模式^[6]。

由以上分析可知,该架构的总体流程如下:首先,在 JSP 页面中触发事件,例如鼠标从一个表格移到另外一个表格,Ajax 引擎就把由这次事件触发产生的请求发送;Servlet 控制器通过 web.xml 捕捉该次请求,分配给相应的控制程序;接着,控制器把请求数据封装为 BO,并调用“Service”层进行逻辑处理,如果需要进行数据库操作,则通过 DA 层进行数据库读写;最后控制器把结果发送给视图进行显示。

2 基于 Ajax 和 MVC 的 Web 应用

2.1 需求分析

以在线聊天室为例来说明基于 Ajax 和 MVC 的 Web 应用。该系统主要分为用户管理、房间管理和聊天三大模块。这里主要讨论的是聊天模块。在聊天模块中,用户在网页上输入聊天内容,以私聊或者公聊的方式发送给指定的某个人或在线的所有用户。私聊时,只有指定的用户能够看到发送的内容;公聊时,所有在线用户都能看到聊天内容。该模块的难点在于,聊天室的在线用户较多,接收和发送信息十分频繁。若以传统方式定时刷新,将导致视觉不流畅和服务器负载过大。

2.2 实现细节

2.2.1 视图表示层

在聊天的 JSP 页面上用 JS 定义了一个发送信息函数 send()。该函数首先创建 XMLHttpRequest 对象,然后利用 XML DOM 动态获取节点 Content 的内容,最后调用 XMLHttpRequest 对象的 send()方法,把信息发送给服务器。由于采用 Ajax 技术,当点击发送时,触发 send()函数,以异步方式传送数据,仅仅刷新网页

中的发送部分。

相应地,在该页面上定义了一个接收信息的函数 get()。当服务器响应完毕时,利用 XMLHttpRequest.responseText 回调响应内容,并赋值给节点 Chat 的 innerHTML,以异步方式动态更新页面,获取聊天内容。

2.2.2 业务逻辑处理

根据需求,定义一个 BO 对象 Content。该对象包括是否私聊、用户昵称、聊天时间以及聊天内容等属性。因为用户需要查找历史聊天记录,定义了一个“Service”对象 ContentService。在 DA 层中有两个方法 executeUpdate 和 executeQuery,分别是根据 sql 语句对数据库进行写操作和读操作。ContentService 调用这两个方法,向数据库中添加 Content 和读取 Content。

2.2.3 Servlet 处理请求

Servlet 作为控制器,用来控制整个数据的流向。当页面触发 send()时,Servlet 根据 web.xml 的配置,将请求分配给 Servlet 程序 ChatServlet。ChatServlet 把页面数据封装成 Content 对象,然后调用 ContentService 中的方法添加到数据库中。同时,ChatServlet 还调用 ContentService 中的读数据库方法 getContent(),获取 Content 对象,通过过滤和解析,获取符合条件的数据,最后封装在 Content 对象中,通过 IO 流输出到 Ajax 的 XMLHttpRequest 对象中,在页面显示。

3 结束语

通过网上聊天室的应用实例的分析可知,采用基于 Ajax 和 MVC 架构,适合中小型 Web 项目的开发,达到减轻服务器的负担、提高系统的响应速度、提供流畅的视觉浏览等效果。而且,代码结构清晰,各层分工明确,耦合性低,可重用性高,易于维护和扩展。

参考文献 (References):

- [1] 何成万,余秋惠. MVC 模型 2 及软件框架 Struts 的研究[J]. 计算机工程,2002,28(6):274-275.
- [2] CRAME D. PASCARELLO E, JAMES D. Ajax in Action[M]. Manning Publications Co.,2006.
- [3] PAULSON, L D. Building rich web applications with Ajax[J]. Computer,2005,38(10):14-17.
- [4] 吕林涛. 基于 AJAX 的 Web 无刷新页面快速更新数据方法[J]. 计算机应用研究,2006,23(11):199-201.
- [5] 唐奕,杨楚华. 基于模式设计的 MVC 框架原理研究[J]. 软件教育技术导刊,2008(6):3-4.
- [6] ErichGamma. 设计模式[M]. 北京:机械工业出版社,2000.