

# 基于 DWR 框架的 WEB 应用的设计与实现

张 栗<sup>1</sup>, 张凤元<sup>1</sup>, 危胜军<sup>2</sup>

(1. 北京化工大学 信息科学与技术学院, 北京 100029;  
2. 北京理工大学 计算机网络攻防对抗技术实验室, 北京 100081)

**摘 要:** AJAX 动态技术的广泛应用, 增加了 WEB 项目开发的复杂性和难度。AJAX 框架技术有效地解决了这一问题, 在实现功能的同时提高了开发效率, 具有推广意义。讨论了 AJAX 技术的特点和应用情况, 并针对基于 AJAX 的动态客户端开发中的效率问题, 提出了使用 DWR 框架设计实现 AJAX 应用效果的实例解决方案, 采用客户端脚本语言对服务器端远程调用技术完成动态显示功能, 优化了开发效率, 并给出动态 HTML 页面中部分代码和服务器端的设计方法。

**关键词:** DWR; AJAX 框架; Java; 远程调用

**中图分类号:** TP393.09

**文献标识码:** A

**文章编号:** 1673-629X(2008)08-0084-04

## WEB Application Design and Realization Based on DWR Frame

ZHANG Li<sup>1</sup>, ZHANG Feng-yuan<sup>1</sup>, WEI Sheng-jun<sup>2</sup>

(1. Sch. of Information Science and Technology, Beijing University of Chemistry Technology, Beijing 100029, China;  
2. Lab. of Computer Network Defense Technology, Beijing Institute of Technology, Beijing 100081, China)

**Abstract:** As a result of AJAX technology used universally, WEB projects become more complicated. It is advantageous to use AJAX frame in WEB projects. Discussed the AJAX technology characteristic and the application situation, in view of based on in the AJAX dynamic client side development efficiency question, proposed that uses the DWR frame design to realize the AJAX application effect example solution, uses the client side script language to complete the dynamic demonstration function to the server end long-distance transfer technology, optimized the development efficiency. Gives partial codes in dynamic HTML page, server end design method.

**Key words:** DWR; AJAX frame; Java; remote-call

### 0 引言

作为 WEB2.0 的核心技术, AJAX 技术越来越为人们关注和使用, 它无需刷新整个页面, 从而节省了网络带宽, 减轻了服务器的负担, 并且促进了页面效果和数据的分离, 使网络应用更贴近使用者的需求。最成功的 AJAX 应用当属 Google Suggest、Google Earth 和 Gmail, AJAX 技术给用户带来了更为人性化的用户体验。然而 AJAX 的 XMLHttpRequest 通信问题繁琐复杂, 且客户端脚本语言 Javascript 缺乏良好的开发调试工具, 这两方面的问题加重了开发人员的编程负担, 给 AJAX 实现造成了一定的困难, 影响了开发效率。基于这些问题, 许多旨在简化 Javascript 与服务器之间异

步通信问题的 AJAX 框架出现了。笔者基于 AJAX 框架 DWR, 针对 AJAX 经典应用——无限级树型菜单, 提出了一种新的解决方法。

### 1 AJAX 技术及框架简介

在传统 B/S 交互模型中, 用户在浏览器端向服务器端发送一个 HTTP 请求(如提交表单), 服务器端分析请求内容, 并与数据库进行交互, 然后将响应结果返回浏览器端。当服务器处理请求内容的时候, 用户处于空闲等待状态, 直至浏览器端返回了一个完整的 HTML 页面。在不需要更新整个页面数据的情况下, 这种提交-响应的方式浪费用户时间和网络带宽, 降低了 WEB 应用的执行速度。

AJAX 技术<sup>[1,2]</sup>的特点是可以只更新部分页面信息, 按需提交及获取数据, 它在用户请求和服务器之间建立了一个中间层(AJAX 引擎), 使用户请求与服务器响应异步化, 所有的用户请求都必须通过 AJAX 引擎的分析判断, 把需要更新数据的用户请求发送给服

收稿日期: 2007-11-09

基金项目: 北京市科技计划资助项目(20005268000091); 校青年教师自然科学基金(QN0519)

作者简介: 张 栗(1983-), 女, 硕士研究生, 研究方向为信息安全、图像处理; 张凤元, 博士, 副教授, 研究方向为信息传输与处理、信息安全。

务器端处理,不需要更新的页面数据不做提交,从而节省了带宽资源,并且提高了响应速度,减少了用户等待时间。图 1 是对两种应用模式的比较。

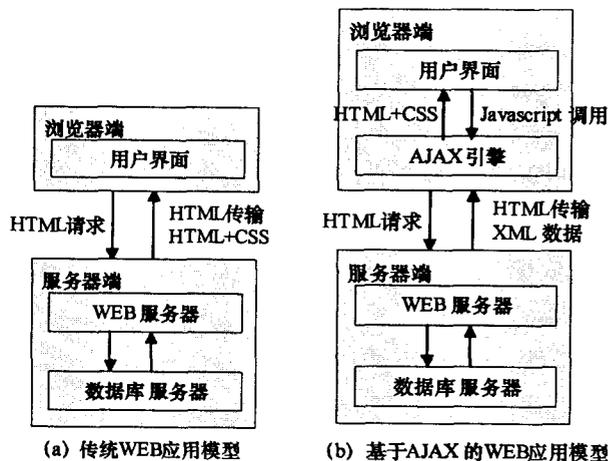


图 1 两种应用模式的比较

AJAX<sup>[3,4]</sup>是 Asynchronous Javascript and XML 的缩写。AJAX 是几项技术的组合,在协作中发挥各自的作用,包括:使用 XHTML 和 CSS 标准化呈现;使用 DOM 实现动态显示和交互;使用 XML 和 XSLT 进行数据交换与处理;使用 XMLHttpRequest 进行异步数据读取;最后用 JavaScript 绑定和处理所有数据。其中 XMLHttpRequest 和 Javascript 对象是 AJAX 技术的核心。

AJAX 框架技术<sup>[5]</sup>简化了 AJAX 实现,它通常分为两类:一类是基于浏览器的框架实现,一类是基于服务器端的框架实现。浏览器端框架把 Javascript 封装起来,实现特定的功能模块,保留用户定义的部分,这种框架的 AJAX 引擎位于客户端,如 Prototype.js。服务器端框架把 XMLHttpRequest 异步通信进行封装到服务器端运行,减轻了客户端的负担,使异步通信过程对开发者透明。两种框架技术各有优势,前者简化了浏览器的动态显示代码,后者封装了 AJAX 复杂的异步通信过程。

DWR 框架是基于 Java 的 AJAX 开源框架,它兼备了浏览器端框架和服务器端框架的优点,提供了常见的 Javascript 工具箱,减轻了编写和调试 Javascript 代码的负担,隐藏了 AJAX 与服务器通信的代码,使不熟悉 AJAX 的开发者也可以实现 AJAX 动态刷新页面的功能。DWR 框架最突出的特点是封装了 WEB 页面和 Java 对象之间的通信,在 WEB 页面通过 Javascript 函数可以直接调用 Java 函数,让开发者把注意力放在如何增强应用程序的交互性以及更好地完成业务功能上。

## 2 DWR 的核心机制

DWR 是一个成熟的 AJAX 框架,它可以让开发人员在不熟悉 AJAX 编程的情况下,实现 AJAX 的功能,轻松地把客户端请求与服务器调用联系起来。DWR 有两个主要作用:第一,对于公开的 Java 类,DWR 动态地生成包含在 Web 页面中的 Javascript 函数。生成的 JavaScript 函数,表示 Java 类上的对应方法并在幕后执行 XMLHttpRequest。当向 DWR 发送请求的时候,它的第二个作用就是把请求翻译成服务器端 Java 对象上的方法调用并把方法的返回值发送回客户端,编码成 JavaScript。原理如图 2 所示<sup>[6]</sup>。

远程 Java 对象方法和对应的 Javascript 存根函数之间的映射简便易用。通用的形式是 JavaScriptName.methodName(methodParams ..., callback),其中 JavaScriptName 是 creator 的 Javascript 属性指定的名称,methodParams 代表 Java 方法的 n 个参数,callback 是要用 Java 方法的返回值调用的 JavaScript 函数。通过这种在 Javascript 代码中调用 Java 函数的方式,就可以实现清爽的用户体验,节省了编写大量 AJAX 代码的时间。此外,DWR 还提供了帮助执行常见的用户界面任务的 Javascript 工具函数,简化了部分 Javascript 脚本语言,并把某些常用功能模块封装起来,比如动态地修改表单、操作 Select 控件等,提高了开发效率。

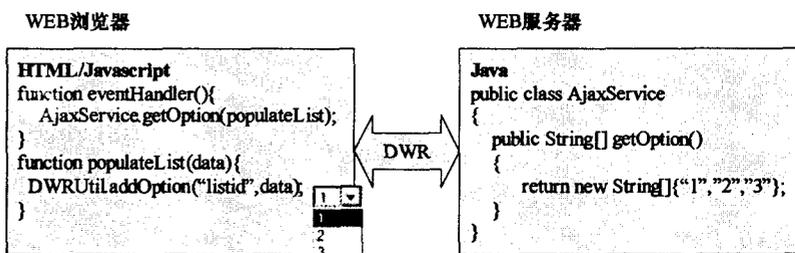


图 2 DWR 框架核心机制示意图

## 3 设计实例

AJAX 技术的一个经典应用是树型无限级菜单,在 WEB 应用中人们能看到很多树型菜单的实例,例如 CSDN 社区网站的 <http://community.csdn.net/>。传统的树型菜单开发模式是一次性读取所有数据。

在使用多级菜单的时候,为了避免每次操作都从后台调用数据,通常采用一次性将树形菜单的所有数据全部读取出来并写入数组,然后根据用户的操作用 Javascript 来控制它的子集项目的呈现,从而解决了反复重载页面以及向服务器频繁发送请求的问题。但是如果用户不对菜单进行操作或只对菜单中的一部分进行操作的话,那读取的数据就会出现冗余而造成浪费用户资源,特别是在菜单结构复杂、数据量大的情况

下,这种弊端就愈发突出。

应用 AJAX 技术后,情况有所改观,在初始化页面时只读出顶级的所有数据并显示,在用户操作顶级菜单其中一项时,会通过 AJAX 向后台请求当前一级项目所属的二级子菜单的所有数据,以此类推。这样就减少了数据下载总量,既得到了正确的结果又避免了数据冗余,相对于重载页面的方式缩短了用户等待时间。

典型的 AJAX 开发纠缠于 XMLHttpRequest 通信代码,而使用 DWR 可以彻底摆脱异步通信过程的复杂局面。DWR 框架让开发工作集中在功能设计上,客户端和服务端之间的异步通信和 Javascript 调试工作的问题得以解决。通过配置 dwr.xml 把客户端的 Javascript 函数和服务端端的 Java 类联系起来,规定了 Javascript 函数和 Java 类的调用关系;通过设计 Java 类定义菜单结构和调用方法;通过编写 Javascript 函数实现数据回传并控制动态显示。功能设计围绕这三部分展开,它们之间的关系如图 3 所示。

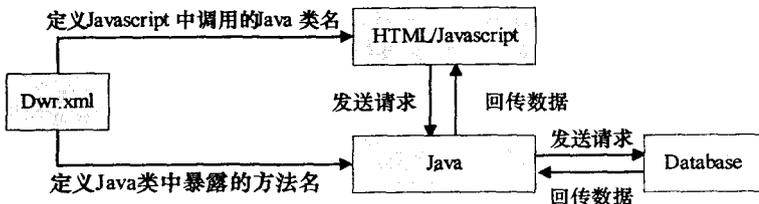


图 3 应用设计实例原理图

### 3.1 定义 Java 类

Java 类是服务器端部分,定义 Menu 类和 MenuManager 类。Menu 类中包含 Menu 对象的成员变量,定义了菜单的逻辑结构。菜单的逻辑结构决定了菜单可以扩展到无限级,开发者可以自主设计。MenuManager 类中定义了 JavaScript 中要远程调用的 Java 方法,即按需求返回菜单数组,只更新用户使用的部分,这部分设计是不产生冗余数据的关键。

```
public class Menu {
    private Long id;
    private String name;
    private Long parent; //上级菜单 id, id=0 说明是第一级菜单
    private List<Menu> submenu; //子菜单
    private int isincludeSubmenu; //是否包含下级菜单
    .....
}
```

getSubMenu 方法从前台获取 menu 的 id,在数据库中查询是否包含子菜单,在用户操作时才获取子菜单列表,显示在页面上。getMainMenu 方法自动获取出 parent 属性为 0 的第一级菜单列表从数据库中读取出来。在第一次加载页面的时候,保证一级菜单的正

确显示。

```
public class MenuManager {
    public List<Menu> getSubMenu(Long id); //获取子菜单
    public List<Menu> getMainMenu(); //获取主菜单
    .....
}
```

### 3.2 配置 DWR

使用 DWR 框架,首先要在 dwr.xml 文件中配置要使用的类和方法。通过 include 和 exclude 来保证代码安全性,它们用来定义是否对 DWR 暴露某个方法。

```
<dwr>
    <allow>
        <!-- 在前台调用时使用的名称 -->
        <create creator="new" javascript="MenuManager" scope="application">
            <!-- 对 DWR 暴露的 Java 类 -->
            <param name="class" value="com.thinkor.menu.MenuManager"/>
            <!-- 对 DWR 暴露的 Java 类中的方法 -->
            <include method="getMenu"/>
        </create>
        <convert converter="bean" match="com.thinkor.menu.*"/>
    </allow>
</dwr>
```

create 元素告诉 DWR 应当公开给 AJAX 请求的服务器端类,并定义 DWR 应当如何获得要进行远程的类的实例。这里的 create 属性被设置为值 new,这表示 DWR 应当调用类的默认构造函数来获得实例。

create 负责公开用于 Web 远程的类和类的方法,convert 则负责这些方法的参数和返回类型。convert 元素的作用是告诉 DWR 在服务器端 Java 对象表示和序列化的 JavaScript 之间如何转换数据类型。

### 3.3 前台调用

在测试成功之后,就可以在前台远程调用 Java 方法,在页面的指定位置显示菜单。程序中还使用了 DWR 的 JavaScript 类库 util.js 中的 \$( ) 方法获取显示菜单的位置,它与 JavaScript 中的 document.getElementById() 方法作用相同,但使用起来更加简洁明了。

```
function initMenu() {
    MenuManager.getMainMenu(showMainMenu); //远程调用 Java 方法
}
```

在 initMenu() 函数中远程调用 Java 方法,并把返回值列表作为参数 menus 传给 showMainMenu() 函数。showMainMenu() 函数在页面的 <ul id = 'menuposi-

tion' > </ul> 位置把菜单显示出来。

```
function showMainMenu(menus){
var menuText = "";
<! - $( ' ' ) 是 DWR 框架里定义的工具函数, 等价于
Javascript 脚本语言里的 document.getElementById( " " ), 显然前者
比后者更简单实用 - >
//var mainmenu = document.getElementById( "menuposition" );
var mainmenu = $( 'menuposition' ); //获取显示菜单位置
for( var i = 0; i < menus.length; i + + ) {
if( menus[ i ].parent == 0 ) {
menuText = menuText + "<li id = 'foldheader' name = '" + menus
[ i ].id + "' >" + menus[ i ].name + "</li>" + "<ul id = 'fold-
inglist' style = 'display:none' ></ul>";
}
}
mainmenu.innerHTML = menuText;
}
```

配置 dwr.xml、定义并访问后台 Java 类和回传 Javascript 函数, 完成了一个 AJAX 动态调用机制的循环, 实现了不提交整个页面也可以刷新信息的卓越用户体验。在无限级的菜单使用中, 每展开一级菜单才从数据库中读取其下级菜单的内容生成并显示出来。当不访问菜单时, 页面只包含并显示第一级菜单的内容。这样避免了前台一次载入数据过大, 造成响应缓慢的问题, 并且在不用时, 不会读取大量数据, 避免

(上接第 83 页)

#### 4 结束语

面向 Agent 技术为复杂、开放、分布式系统的开发和实现提供了新途径, 在开发新型应用中起着重要作用。随着应用系统的日益复杂, 系统需要的数据和信息的规模日益扩大, 以及对应用系统人性化、智能化的日益需求, 面向 Agent 技术, 特别是多 Agent 技术将会受到越来越多的重视。但是 Agent 目前还有一些问题需要解决:

(1) 缺乏有关基于 Agent 的软件系统开发方法学, 这使 Agent 的开发无规范可言。开发高性能 Agent 及多 Agent 系统必须有一套科学化的方法来解决: 个体 Agent 的可靠性和扩展性? 整个系统的 Agent 如何组织和协作? 基于 Agent 软件系统的可扩展性? 等等。

(2) 基于 Agent 软件系统的设计方案缺乏评估、比较的标准。由于缺乏开发方法, 则系统开发自然仁者见仁智者见智了, 对于系统设计方案的评估、比较也就难以有一个统一的标准了。

(3) 与可重用技术和分布式对象技术的结合不够紧密。由于目前基于 Agent 的软件系统在实践中还是

给服务器带来不必要的负担, 提升了页面响应速度。

#### 4 结束语

介绍了 AJAX 技术的优势特点, 以及 AJAX 框架出现的缘由。提出一种使用 AJAX 框架 DWR 来实现无限级树型菜单的方法, 并给出实现代码。此方法既具有 AJAX 动态显示数据的特点, 轻松实现了无限级扩展, 又隐藏了异步通信编程工作, 提高了开发效率, 在实际项目中有普遍推广和深化开发的意义。

#### 参考文献:

- [1] 游丽贞. AJAX 引擎的原理和应用[J]. 微计算机信息, 2006, 22(23): 205-207.
- [2] 徐 驰. Ajax 模式在异步交互 Web 环境中的应用[J]. 计算机技术与发展, 2006, 16(11): 228-233.
- [3] Zakas N C. Javascript 高级程序设计[M]. 北京: 人民邮电出版社, 2006.
- [4] Asleson R. AJAX 基础教程[M]. 北京: 人民邮电出版社, 2006.
- [5] 余翔宇. AJAX 技术及其框架实现[J]. 软件导刊, 2006 (9): 28-30.
- [6] Carneiro C. AJAX made simple with DWR[EB/OL]. 2005. <http://www.javaworld.com/javaworld/jw-06-2005/jw-0620-dwr.html>.

使用面向对象语言作为开发平台, 客观上要求软件系统能够与可重用技术和分布式对象技术的结合紧密结合起来。

(4) 不成熟的面向 Agent 的程序设计语言。虽然目前有众多的面向 Agent 的程序语言, 但都是不太成熟的, 使得目前基于 Agent 的系统还是多以成熟的面向对象语言作为开发语言。

#### 参考文献:

- [1] Wooldridge M J, Jennings N R. Intelligent Agent: Theory and Practice[J]. Knowledge Engineering Review, 1995, 10 (2): 115-152.
- [2] Yang Kun, Liu Da-you. Agents: properties and classifications[J]. Computer Science, 1999, 26(9): 30-34.
- [3] 刘大有, 杨 鲲, 陈建中. Agent 研究现状与发展趋势[J]. 软件学报, 2000, 11(3): 315-321.
- [4] 王汝传, 徐小龙, 黄海平. 智能 Agent 及其在信息网络中的应用[M]. 北京: 北京邮电大学出版社, 2006.
- [5] Wooldridge M. An introduction to multi-Agent systems [M]. New York: John Wiley & Sons, Inc., 2002.
- [6] 蒋云良, 徐从富. 智能 Agent 与多 Agent 系统的研究[J]. 计算机应用研究, 2003(4): 31-34.