第16卷 第11期
2006年11月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol.16 No.11
Nov. 2006

Ajax模式在异步交互Web环境中的应用

徐 驰
(同济大学 软件学院,上海 201804)

摘 要:作为Web 2.0标准核心之一的Ajax模式,通过在客户端建构中间层,实现了页面表现与应用逻辑的分离,并且支持B/S环境下用户操作与服务器响应的异步化。文中简要阐述了Ajax模式的核心概念、体系结构和异步交互机制,并通过基于Ajax模式的网上书店实例的设计和开发,体现了其在异步交互Web环境中的框架结构、逻辑组织和在改善网络带宽、优化用户体验等方面相比于传统Web模型的诸多优越性。

关键词:Ajax;异步交互;MVC;JavaScript;XML
中图分类号:TP311.5 **文献标识码**:A **文章编号**:1673-629X(2006)11-0228-03

Application of Ajax in Asynchronous Interactive Web Environment

XU Chi
(School of Software Engineering, Tongji University, Shanghai 201804, China)

Abstract: Compared with traditional Web models, Ajax pattern, the core of Web 2.0 Standard, establishes a browser-end middle layer, which separates page representation and application logic effectively and also realizes an asynchronous response of server towards client's operations. Introduces core concepts of Ajax and presents its architecture, logical organization and unique superiority in reducing network and server load through the design and development of an online bookstore case.

Key words: Ajax; asynchronous interaction; MVC; JavaScript; XML

0 引 言

随着互联网技术的飞速发展,网络信息交互愈发频繁。在传统的Web应用程序模型中,B/S信息交互采用同步方式:用户在提交请求后,被迫中断当前工作,等待页面的刷新、交换和重载。完整页面的传输会加重网络载荷和服务器工作量;与此同时,用户在等待服务器响应的过程中面临较长时间的空白页,响应实时性降低。对于这一问题,较理想的解决策略是将客户端的页面表现和应用逻辑进行"拆分",使应用逻辑部件按照需求,独立地与服务器实现信息交互,实现对页面控制的解耦,从而建立更加面向服务的Web应用程序结构。

作为Web2.0标准核心之一的Ajax(Asynchronous JavaScript and XML)模式正是在这样的背景下出现的。它在客户端加入一个沟通用户界面与服务器的中间层,实现页面呈现与应用的分离以及用户操作与服务器响应的异步化。这样,一方面可以利用客户端闲置的处理能力承担一部分服务器的工作,减轻带宽和服务器的负担;另一方面降低了页面重载的频率,给予Web用户更好的使用体验。目前,Ajax模式已应用于Google Maps[1],Google Suggest[2]等领域。文中将探讨Ajax模式的核心概念、体系结构以及异步交互机制,并通过实例体现其在Web应用程序开发中的特点和优越性。

收稿日期:2006-02-28

作者简介:徐 驰(1981-),男,陕西乾县人,硕士研究生,研究方向为软件工程、图像处理等;导师:徐燕凌,博士,讲师,研究方向为图像处理、图像通信等。

1 Ajax模式概述

1.1 Ajax的核心概念

Ajax是JavaScript,CSS,DOM,XMLHttpRequest等多项Web技术按照规范进行组合,并在基于B/S的Web应用中整合信息交互的设计模式。其核心要素包括:

(1)JavaScript:它是内嵌在页面程序中的通用脚本编程语言,为多种浏览器广泛支持。在Ajax模式中,是绑定数据和处理业务逻辑的主体。

(2)CSS(Cascading Style Sheets):它是一种为Web页面元素提供可重用样式定义的语言。在Ajax模式中,可通过CSS定义和修改用户界面的外观风格。

(3)DOM(Document Object Model):它是一组对Web文档对象的节点结构进行操作的API。在Ajax模式中,可通过DOM动态地改变用户界面的布局层次。

(4)XMLHttpRequest对象:它为客户端程序提供了在后台与服务器交换数据的能力,是异步交互式Web应用开发的关键。

1.2 Ajax的体系结构

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"[3]),如图 1 所示。

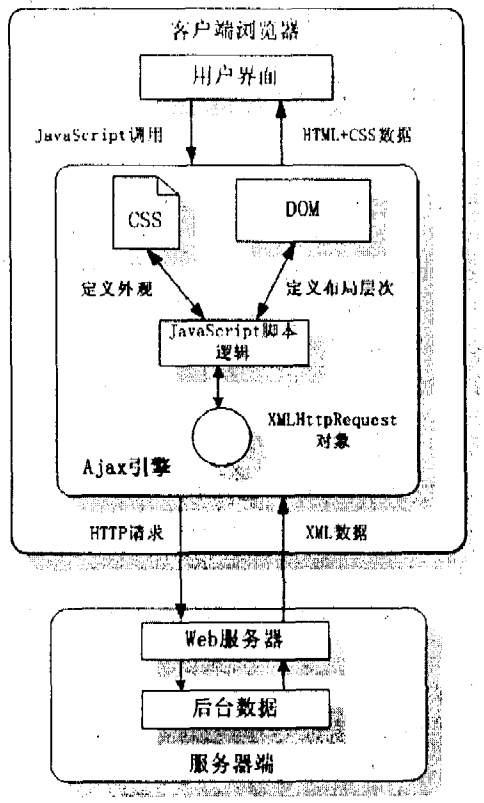

图 1　Ajax 引擎模型

Ajax 引擎的作用是分离 Web 页面的呈现和应用,实现浏览器和服务器交互的异步化。一方面,通过 JavaScript 脚本编程语言灵活地处理数据验证、数据编辑等用户请求,并依靠 DOM 和 CSS 实现用户界面的动态显示和格式化呈现;另一方面,通过 JavaScript 在无需刷新和重载浏览器页面(页面呈现完全独立于与服务器端的交互)的条件下,利用 XMLHttpRequest 对象与服务器进行线路化的数据交换。

这种引擎模型的优势在于:

(1)中间层引擎分担了一部分服务器处理数据的工作,减轻了用户请求对网络和服务器的负担;

(2)客户端与服务器端异步交互的特点使得页面更新无需重载,减少了用户的等待时间,给用户更好的使用体验;

(3)JavaScript,XMLHttpRequest 等已为浏览器广泛支持,具有很好的通用性。

1.3　Ajax 的异步交互机制

传统 Web 应用"提交请求—页面等待—页面重载"的同步机制已难于应对频繁的 Web 交互,而 Ajax 的异步交互机制无疑具有更好的健壮性。一个典型的 Ajax 异步交互过程如下:

(1)用户提交请求,页面程序调用 JavaScript 脚本,进行简单的数据处理。

(2)如果需要与服务器进行交互,则通过 JavaScript 创建 XMLHttpRequest 对象:为兼容不同的浏览器,可创建一个通用的 XMLHttpRequest 对象获取方法,定义如下:

```
function getXMLHttpRequest(){
  var xmlReq = null;
  if(window.XMLHttpRequest){
    //在非微软浏览器中创建 XMLHttpRequest 对象
    xmlReq=new XMLHttpRequest();
  }
  else if(window.ActiveXObject){
    try{
    //在微软新版浏览器中创建 XMLHttpRequest 对象
      xmlReq=new ActiveXObjext("Msxml2.XMLHTTP");
      }
      catch(e1){
        try{
        //在微软旧版浏览器中创建 XMLHttpRequest 对象
        xmlReq=new ActiveXObject("Microsoft.XMLHTTP");
          }
        catch(e2){}
      }
  }
  return xmlReq;
}
```

(3)通过 XMLHttpRequest 对象向服务器发送请求:使用 XMLHttpRequest 对象的 open 方法,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。如果将是否支持异步模式的参数设置为真,则 JavaScript 脚本语言将继续运行而不用同步等待服务器的响应。此时,浏览器页面无需刷新,便可以继续与用户进行交互。

(4)与此同时,服务器端的程序对 XMLHttpRequest 对象请求进行处理,返回结果并改写 XMLHttpRequest 对象的请求状态。

(5)通过 JavaScript 回调函数处理服务器端响应:在(3)中,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,将触发回调函数对 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。

(6)最后,通过 JavaScript 控制 DOM 动态地更新用户界面(例如,在页面的局部添加或更新服务器返回的数据)。

2　基于 Ajax 模式的 Web 应用实例研究

2.1　需求分析

文中使用网上书店这一典型实例来说明如何设计和开发基于 Ajax 模式的 Web 应用程序。在该系统中,要求用户在浏览书刊选项时,可随时在购书车中添加和删除书刊,并动态地显示当前的购书记录。在传统的购书页面中,每次单独提交定购或撤销请求,都需要刷新页面,等待

页面重载,才能显示购书信息,显然难以满足"动态显示"的用户需求。在基于 Ajax 模式的网上购书系统中,将浏览器前端变为无需刷新的单页界面,通过 XMLHttpRequest 对象线路获取数据更新,并即时地呈现在页面上,给予了用户流畅的视觉效果和更好的浏览体验。

2.2 结合 Ajax 模式的 MVC 框架设计

利用 Ajax 模式对已在 Web 开发中得到广泛应用的 MVC 设计模式(Model - View - Controller,模型 - 视图 - 控制器[4])进行扩展,形成网上订书系统的设计框架,如图 2 所示。

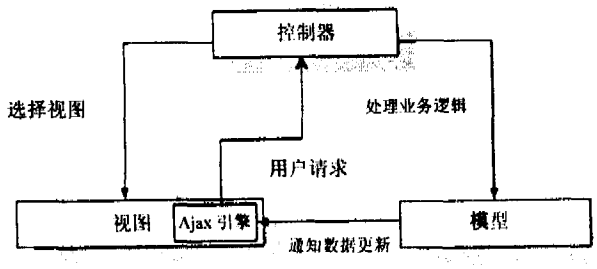

图 2 基于 Ajax 扩展的 MVC 框架

在该设计框架中,View 代表用户视图,可通过内置的 JavaScript 脚本向 Controller 发送请求。Controller 接收来自 View 的请求(GET 或者 POST 方式),传递给相应的业务/数据模型执行,并根据执行结果选择合适的 View 展现给用户。Model 用于处理业务逻辑流程,通过 XMLHttpRequest 对象建立与 View 的线路联系,可直接向 View 发送数据更新通知,触发 View 模块内置的 Ajax 回调函数,以控制 DOM 对页面实行动态的更新。

2.3 实现细节

2.3.1 视图表示层

网上书店订书系统在浏览器端使用的 HTML 代码片断如下:

```
<! -图书列表->
<th>Book Name</th> <th>Description</th> <th>Price
</th><th></th>
<tr> <! -图书详细信息->
  <td>Gone with the Wind</td> <td>Saga Novel</td> <
td>48.50</td>
  <td> <! -加入购书车->
    <button onclick="addToCart('Gonewiththewind','Cart.
handle')">Add to Cart</button>
  </td>
</tr>
……
<! -购书车信息列表(异步更新) ->
<ul id="Bookcart-contents">
    <! -在这里动态显示已购书刊信息->
</ul>
<! -动态显示已购书刊总价->
Total price:<span id="total">0.00</span>
```

2.3.2 Ajax 引擎发送用户请求

名为 addToCart 的 JavaScript 函数是 Ajax 引擎的主体,它负责将页面请求映射到一个 Servlet,并提供了响应服务器数据更新的处理函数。主要代码如下:

```
function addToCart(BookCode,url){
var request = getXMLHttpRequest();//创建 XMLHttpRequest 实
例
var callFunc = onReadyState(request,update);
request.onreadystatechange = callFunc;//设置回调处理函数
request.open("POST",url,true)//打开 HTTP POST 异步连接
request.setRequestHeader("Content-Type","application/x-www
-form-urlencoded");
request.send("action=append&book=" + BookCode);//传递数据
}
```

2.3.3 Servlet 处理请求

Servlet 作为控制器,对 XMLHttpRequest 对象请求进行处理,方法与处理普通的 HTTP 请求相同。它利用 HttpServletRequest 类的 getParameter()方法得到 Ajax 引擎发送的请求数据,并传递给 Bookcart JavaBean 实体模型进行状态更新。

2.3.4 JavaBean 序列化生成 XML

用 JDOM API 将 Bookcart 序列化成 XML 文档格式并写入 ServletResponse 返回给 Ajax 引擎。

2.3.5 JavaScript 回调函数响应服务器回传数据

XMLHttpRequest 对象的 readyState 属性代表 Ajax 请求的生命周期,是一个数值,从 0 到 4 分别代表"未初始化"、"装载中"、"已装载"、"交互"和"完成"。每当 readyState 属性变化,都会触发 readystatechange 事件,从而使 onreadystatechange 指定的回调处理函数被调用。onReadyState 回调函数将根据 readyState 属性检查 XMLHttpRequest 是否已完成请求,并调用 update 函数动态更新页面,呈现 XML 文档的内容。onReadyState 函数定义如下:

```
var READY_STATE_COMPLETE 4 //XMLHttpRequest 请求完
成
var STATUS_COMPLETE 200 //已接收到服务器端响应
function onReadyState(request,XMLHandler){
return function(){
    if(request.readyState==READY_STATE_COMPLETE){
      if(request.status==STATUS_COMPLETE){
      XMLHandler(request.responseXML)//将 XML 文档传递
给处理函数
        }else{
          alert("Waiting… HTTP Status:" + request.status);
        }
        }
  }
}
```

2.3.6 用 XML DOM 动态更新 Web 页面

update 函数使用 XML DOM 的 getElementsByTagName 方法提取 XML 文档中各节点元素信息,添加

(下转第 233 页)

络训练成本,提高网络认知能力和准确性 。网络输出为28d碳化深度;网络学习误差设为0.01。样本数据共有9组,随机地分成2部分,第一部分7组数据作为训练集,剩下的2组不参与训练,作为测试集,用于检测网络的泛化能力。

由表2可知,神经网络碳化深度计算误差在0.06%~2.1%,计算结果很理想,这表明神经网络计算模型归纳了这7组实验数据;表3预测两组配比的28d碳化深度,误差在4.1%~7.4%,这里考虑到实验数据较少,采用交叉检验(cross validation)方法进行测试,结果不一一列述,表3为其中一组预测结果,预测误差都在10%之内。综上所述,该模型计算及预测误差能满足实际工程的允许误差要求。

表2　多因子输入向量网络计算混凝土碳化深度(训练集)

实验组号	C (kg)	FA (kg)	W (kg)	B (kg)	FA (%)	W/B	碳化深度实测值(mm)	BP网络计算值(mm)	相对误差 (%)
1	458	0	162	458	0	0.35	2.8	2.86	2.1
2	368	92	160	460	20	0.35	6	5.93	1.18
3	276	184	138	458	40	0.3	7	7.08	1.14
4	340	70	160	410	17	0.38	8.88	8.81	0.79
5	276	184	184	458	40	0.4	11.5	11.51	0.08
6	228	152	150	380	40	0.395	15.94	15.93	0.06
7	184	276	162	458	60	0.35	16.5	16.45	0.3

表3　多因子输入向量网络预测混凝土碳化深度(非训练集)

实验组号	C (kg)	FA (kg)	W (kg)	B (kg)	FA (%)	W/B	碳化深度实测值(mm)	BP网络预测值(mm)	相对误差 (%)
8	400	70	162	470	15	0.345	3.5	3.76	7.4
9	276	184	162	458	40	0.35	10.5	10.93	4.1

3　结　论

采用神经网络计算以及预测混凝土碳化深度是研究混凝土碳化的新道路。神经网络具有自适应、自学习、自调整的特点,它根据对以往经验的学习和适应,形成对同类问题的总结,采用神经网络建立的计算和预测模型十分适合在工程中使用。实验结果显示神经网络模型的计算误差最大为2.1%,预测误差最大为7.4%,满足实际工程的允许误差要求。随着实验进展,收集到更多样本供网络学习,网络的预测能力也会进一步加强。

参考文献:

[1] 朱安民.混凝土碳化与钢筋混凝土耐久性[J].混凝土,1992(6):18-22.

[2] Smolczyk H G. Carbonation of Concrete - Written Discussion [C]//Proceedings of 5th international symposium on chemistry of cement. Tokyo:[s.n.],1968:343-368.

[3] CENGIZ D A. Accelerated carbonation and testing concrete made with fly ash[J]. Construction and Building Materials, 2003,17(3):147-152.

[4] Khanml, Lynsdee C J. Strength, permeability and carbonation of high performance concrete[J]. Cement and Concrete Research, 2002,32(1):123-131.

[5] Hagan M T.神经网络设计[M].戴葵等译.北京:机械工业出版社,2002.

[6] 飞思科技产品研发中心.MATLAB6.5辅助神经网络分析与设计[M].北京:电子工业出版社,2003.

(上接第230页)

到Bookcart - contents列表中,并将document.getElementByID("total")的innerHTML属性设置为更新后的购书总额。最后,通过CSS修改控件的外观风格,效果如图3所示。

Book Name	Descpritien	Price	
Gone With the Wind	American Saga Novel	48.50	Add to Cart
Golden Autumn	French Essay	23.40	Add to Cart
A Brief History of Time	British Science Fiction	62.80	Add to Cart

Your Book Cart
1 copy : Gone With the Wind
1 copy : Golden Autumn

Total price: 71.90

图3　Web页面动态更新效果图

3　结　语

Ajax模式通过客户端中间层实现了页面表现和应用逻辑的分离,并且支持Web页面内容的"无刷新"式更新。实验表明[5],Ajax异步交互模式相比于传统的整页刷新模式,节省网络带宽超过60%。与此同时,服务器端的处理负担也得到相应的减轻。可以预见,结合诸多优势的Ajax模式将会在Web分布式程序开发中得到更广泛的应用。

参考文献:

[1] Google公司. Google Map[EB/OL]. 2006. http://maps.google.com.

[2] Google公司. Google Suggest[EB/OL]. 2006. http://www.google.com/webhp? complete=1&hl=en.

[3] Garrett J. Ajax: A New Approach to Web Applications[EB/OL]. 2005. http://www.adaptivepath.com/publications/essays/archives/000385.php.

[4] 许　鑫,费祥林.基于MVC模式的应用软件开发框架研究[J].计算机工程与应用,2005,41(30):102-104.

[5] Merrill C L. Performance Impacts of AJAX Development: Using Ajax to Improvethe Bandwidth Performance of Web Applications[EB/OL]. 2006. http://webperformanceinc.com/library/reports/AjaxBandwidth/index.html.