

基于 Ajax 的远程教育系统的研究与应用

Research and Application of Distance Education Based on Ajax

杨晓燕 吴桂芹 (西安工程大学管理学院 陕西西安 710048)

摘要: Ajax (Asynchronous JavaScript and XML) 采用异步消息通讯机制实现浏览器端脚本代码和服务器快速和轻量交换, Web 页面无需重载即可实现动态更新, 从而可在 B/S 架构下实现跨平台的富客户端网络应用。本文对 Ajax 的工作原理和机制以及关键技术进行分析, 并采用 Ajax 技术设计和实现一个在线教学系统。

关键词: Ajax DOM CSS JavaScript XML

1 引言

近年来, 随着 Internet 技术、多媒体技术等现代信息技术的发展, 教育方式已产生很大变化。教育形式由单一面授模式走向多元化。基于 Web 的远程教育系统以计算机网络技术、卫星通信技术为基础, 以多媒体技术为主要手段, 将远程教育的水平和效果推向一个前所未有的新高度。基于 Web 的远程教育系统通过网络可以实现 Face to Face 的交流, 赋予教与学概念新的内涵, 从而提供一个具有真实感觉的虚拟学习环境。

然而目前的基于 Web 的远程教育的效果并不令人满意。当前对于远程教育的研究重点集中在远程教育系统的建设、评价管理与应用、远程教学模式的探索等方面, 忽略了用户在使用远教育系统进行教与学过程中的交互体验。根据传统 Web 应用架构实现的远程教育系统采用同步“请求/响应”交互模式, 客户端对于所需信息种类或获取时限完全没有自主权。仅当用户做出页面请求操作后, 才能向 HTTP 服务器发出请求, 此时客户端界面出现白屏, 用户只有等到服务器处理完毕传输到客户端后才能做进一步的操作, 导致用户使用体验断续。这种交互模式严重挫伤学习者的学习积极性, 增加不满意度, 降低学习效果。

Ajax (Asynchronous JavaScript and XML), 即异步 JavaScript 与可扩展标记语言。它是一种新兴的 Web 应用富客户端构建技术。Ajax 没采用传统 B/S 同步请求/响应模式, 而是使用异步消息通讯机制, 从而浏览器不必等待用户请求, 也不需要刷新整个 Web 页面即

可实现从远端服务器上获取和显示数据。因此, 采用 Ajax 技术设计和实现的系统更加方便灵活, 页面呈现和应用数据相分离, 用户可以获得更佳的使用体验。Ajax 技术对于远程教育系统这种需要和服务器频繁交互的 Web 应用来说, 具有很大的优越性。

2 Ajax 的工作原理

2.1 传统 Web 应用

在传统 Web 应用中, 浏览器和服务器采用同步通讯模式: 根据用户操作, 浏览器首先向 HTTP 服务器触发一个行为或请求, HTTP 服务器接收浏览器请求后, 对请求内容解码分析, 执行相应处理, 如需要则调用后台应用服务器或者访问数据库服务器, HTTP 服务器形成要传送给浏览器的 HTML 代码, 然后将此代码返回浏览器, 浏览器将其显示, 以供用户查看。

如果用户需要同服务器交互以更新 Web 页部分内容, 服务器也必须发回整个页面, 浏览器也必须重新绘制整个页面。

这种传统同步模式是一种不连贯的用户体验: 当服务器处理用户请求时, 用户多数时间只能等待, 同时屏幕一片空白。同时存在完全刷新限制, 浪费带宽, 系统反应不够灵敏, 不能实现按需传递。传统 Web 应用结构如图 1 所示。

2.2 Ajax Web 工作原理

与传统 Web 应用架构不同, Ajax 采用异步通讯模式。Ajax 在浏览器和服务器之间引入中间层——Ajax 引擎层。浏览器在执行操作时安装 Ajax 引擎, Ajax 引

擎实现浏览器和服务器的异步通讯,独立于用户与服务
器进行交互。

待服务器更新。

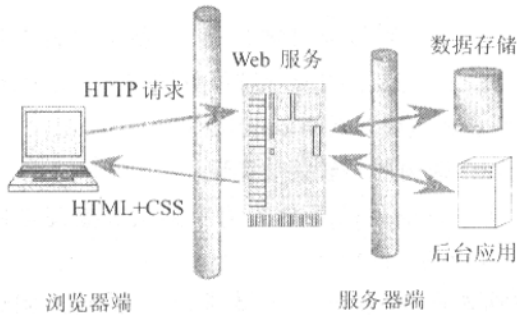


图 1 传统 Web 应用架构

用户请求首先提交给 Ajax 引擎,部分请求 Ajax 引擎直接就可处理,其余请求 Ajax 代为向服务器提交,此时浏览器不用等待服务器的响应,因此用户可以继续别的操作,用户屏幕也不会白屏。服务器处理完用户请求后,返回处理结果并改变 HTTP 就绪状态, Ajax 引擎调用回调函数处理结果并更新浏览器页面指定部分,而浏览器不用刷新整个页面,从而用户可获得和使用及时响应的桌面程序一样的体验。采用 Ajax 的 Web 应用架构如图 2 所示。

3 Ajax 的关键技术

术语 Ajax 描述蓬勃发展并以强大方式组合成一体的一组技术。Ajax 的关键技术包括 JavaScript、DOM、XMLHttpRequest 以及 CSS。

在 Ajax 应用中,使用 JavaScript 操作 DOM 改变和刷新用户界面,重绘来自 HTTP 服务器显示给用户的数据,处理用户基于鼠标和键盘的交互。CSS 提供一致可重用样式,并为以程序模式操纵 DOM 提供捷径。XMLHttpRequest 和服务器实现异步通信。四种技术的配合关系如图 3 所示。

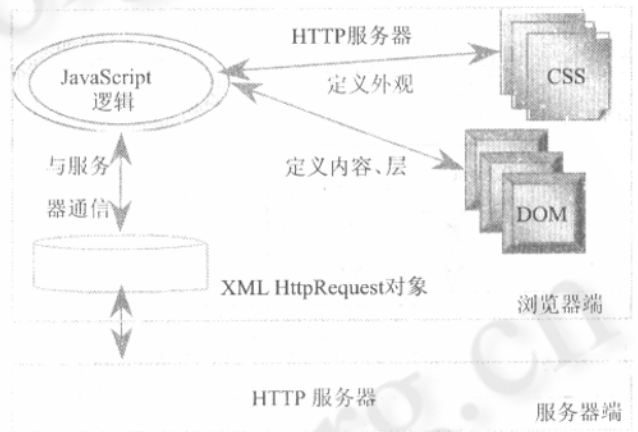


图 3 Ajax 技术配合关系

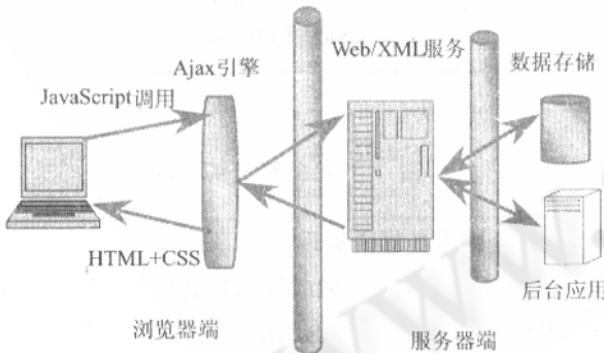


图 2 基于 Ajax 的 Web 应用架构

基于 Ajax 构建的富客户端,服务器的部分处理工作转嫁到客户端,利用客户端闲置处理能力来实现,使服务器和浏览器交互数据量减少,HTTP 服务器处理时间缩短,降低带宽要求,系统响应性更佳和页面重载更为流畅。通过 Ajax 提高 Web 页面更新速度,用户可以连续和远程教学服务器进行交互,而不用刷新页面,等

3.1 JavaScript

JavaScript 是一种在浏览器端大量使用的脚本语言。嵌在 Web 浏览器中的 JavaScript 解释器允许通过程序与浏览器进行交互。JavaScript 将 Ajax 各种技术聚合在一起,并定义工作流和业务逻辑,操作 DOM 改变和刷新用户页面,重绘显示给用户的数据和处理与用户的交互。

3.2 DOM

DOM 是访问和维护 HTML 和 XML 的 API。它定义了文档的逻辑结构以及存取和维护文档的方法,并可改变其中的内容和可见物。脚本语言通过 DOM 实现和页面的交互。Web 开发人员可操作及建立文件的属性、方法和事件并都以对象来展现。

3.3 XMLHttpRequest

XMLHttpRequest 是 Ajax 的核心技术,支持异步请求。XMLHttpRequest 对象在后台运行采用异步通讯从 HTTP 服务器获取数据,不用中断用户操作,自动和服务端进行数据层面的交互。

3.4 CSS

CSS 属于一种样式语言,以可重用、一致的方式定义 HTML 标记的可视化样式,提供从内容中分离样式和设计的机制。CSS 可以定义明显的样式元素、元素之间的相互布局和简单用户交互功能,以较少代码动态地为元素设置预先定义的外观。

4 基于 Ajax 的远程教育系统框架

基于 Ajax 的远程教育系统框架如图 4 所示。该系统由用户端浏览器、Web 服务器、应用服务器和数据库服务器构成。

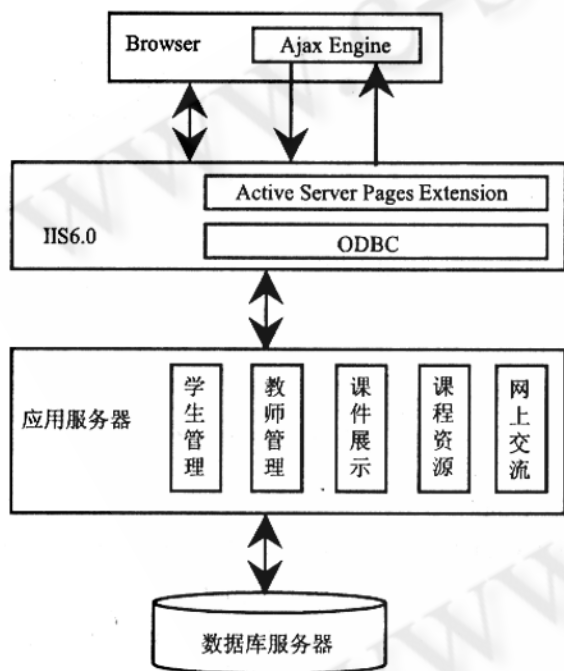


图 4 基于 Ajax 的远程教育系统框架图

系统中,利用 Ajax 技术,服务器端返回一个纯文本流,客户浏览器负责进行处理。在浏览器中,使用 XMLHttpRequest 对象来请求页面,服务器在页面内写入结果文本。客户端在异步获取结果后,不是直接显示在页面上,而是由客户端的 JavaScript 脚本处理之后对页面对应部分进行更新。

5 Ajax 在远程教育系统的应用实例

本文以远程教育系统树形导航菜单为例说明 Ajax 的应用。

树形导航菜单可以展开或折叠,方便学习者在各个功能模块之间跳转。树形导航菜单的样式可以采用 CSS 来控制。其实现方式有以下几种。

5.1 完全客户端树形导航菜单实现

通常实现树形导航菜单的简单办法就是完全由 JavaScript 进行菜单的控制。

客户端向服务器发送一次请求,服务器端返回该树的所有数据。客户端通过 JavaScript 实现菜单的展开与折叠。原理如图 5 所示。此模式要求服务器端要处理大量的数据。对于用户第一次打开需要等待较长时间。另外若用户不需要获得树的所有数据,而只关心某一节点下的数据,服务器端会产生大量的无用数据。

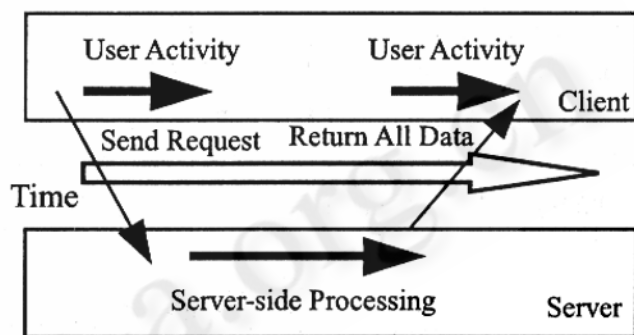


图 5 完全客户端实现

5.2 完全由服务器端实现

此种模式下,每次的展开与折叠操作,都先发送请求,再根据请求来判断返回给树形菜单的显示内容。服务器端要随时记录菜单的展开与折叠状态。原理如图 6 所示。这种方法会导致服务器端数据处理量的增大,客户端需要频繁和服务器联系,影响服务器端的处理速度。与服务器的频繁交互会产生大量的重复数据,浪费网络资源。

5.3 基于 Ajax 的实现

以上两种模式都存在明显的缺陷,采用 Ajax 方案可以将客户端和服务器端结合起来。当客户端提交请求后,并不将所有数据取回,而是返回该菜单的顶级菜

单,当展开某一菜单时在发送请求,服务器端处理完毕后,只返回该类菜单下的子菜单,即请求的内容。在客户端,利用 JavaScript 将数据显示到客户端,所有数据获取和处理都在后台完成。原理如图 7 所示。

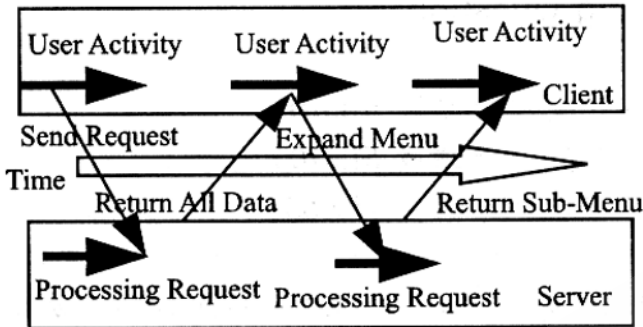


图 6 完全服务器端实现

基于 Ajax 技术的实现,避免了完全基于客户端或服务器端实现的种种弊端,用户既能够在展开菜单时及时获取最新数据,又可避免服务器端频繁处理数据而影响系统性能。

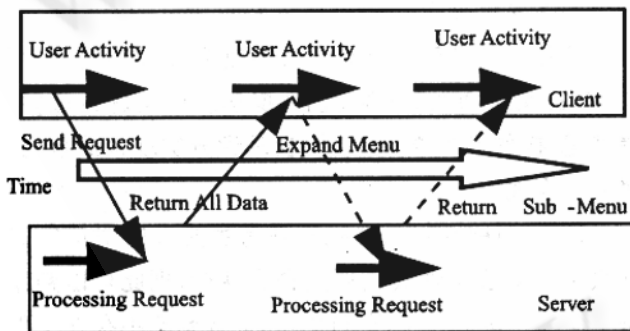


图 7 基于 Ajax 的实现

在客户端的实现逻辑如下:

```
<%
//创建 XMLHttpRequest 对象
Var xmlhttp = creatxmlrequest();
Try{
    xmlhttp.onload = handler;
}
Catch(e){
    Try{
```

```
        xmlhttp.onreadystatechange = handler; }
    Catch(e){
        Alert("您的浏览器无法正常浏览本网页!");
    Return false; } }
// 具体处理函数
Function handler() {
    Try{
        If(xmlhttp.readyState != 4)
            Return;
    } catch(e) {
        xmlhttp.overrideMimeType('text/xml');
        //获取服务器的响应
        Var xslldoc = xmlhttp.responseXML;
        If(! xslldoc.documentElement || xslldoc.docu-
            mentsElement.getElementsByTagName('onclick')) { re-
            turn; }
        //获取树形菜单的顶级菜单
        For(i=0;i < top.length;i++)
        {
            //获取点击菜单的名字和可扩展性
            //并对每个节点田间 onclick 事件处理函数
            Node_name = top[i].getNodeName();
            Node_id = top[i].getID();
            bExpand = top[i].canExpand();
            f_file = top[i].getFile();
            //将从服务器获取的数据显示在页面上
            str += ' <tr onclick = " document.sform.name =
            \'Node_name + \' ;expfile = \' + f_file + \' \' ;process
            ();"
                onclick = " this.bgcolor = \' ; > <td >... </td >
                <td > \' +s_expand + \' </td > </tr >
            if(i == 0) { return false; }
            ...
        }
    }
```

客户端主要工作就是根据用户选择的子菜单向服务器请求数据,并将所有的子节点递归显示在对应的父节点处。同时当从服务器获取数据时,给用户一个提示。

(下转第 105 页)

服务器端的处理逻辑主要是:

```
//判断节点是否可以扩展
```

```
Public Boolean canExpand(id) {
```

```
//查询数据库,判断查询结果个数,若个数 > 0
```

```
//那么返回可扩展
```

```
}
```

```
//当用户单击节点是返回对应的子节点
```

```
Public getChildren(id) {
```

```
//查询数据库,如返回记录个数 > 0 则有子节点
```

```
//否则,没有子节点。
```

```
//将结果返回
```

```
}
```

在服务器端, `getChildren` 当接收到 Ajax 请求后, 根据 `id` 查找数据库中对应表 `CouseList`, 然后将结果返回给客户端。 `canExpand` 用于判断是否在用户页面上显示可扩展标记符号。

6 小结

采用 Ajax 技术构建远程教育系统可以用户等待时间,降低带宽要求,提高响应速度,给学习者更好的学习体验。

参考文献

- 1 Garrett J J, Ajax: a new approach to web Applications.
- 2 Asleson R, Schutta N T. Ajax 基础教程[M], 金灵译, 北京: 人民邮电出版社, 2006.
- 3 Ryan Asleson, Nathaniel T Schutta. Foundations of Ajax. Apress L. P. 2005.
- 4 Rohit Khare. Beyond AJAX: Accelerating Web Applications with Real Time Event Notification. August 2005.