

# 基于 Ajax 的 DataWindow 框架分析

王东, 孙彬

(新疆财经大学计算机系, 乌鲁木齐 830012)

**摘要:** 借鉴 Ajax 和 DataWindow 技术风格, 对 MVC 应用模型和 J2EE 相关技术进行归纳, 总结出一个典型的 Ajax DataWindow 模型。对 XMLHttpRequest、DIV 容器对象、典型业务抽象、用户操作封装和 URL 参数合成等技术进行分析, 构造出开发效率高、移植性强和响应速度快的 B/S 应用模型。

**关键词:** Ajax 技术; DataWindow 技术; MVC 模型

## Analysis of DataWindow Frame Based on Ajax

WANG Dong, SUN Bin

(Dept. of Computer Science, Xinjiang Institute of Finance & Economy, Urumqi 830012)

**【Abstract】** This article profits from Ajax and DataWindow, the technical style, carries on the induction to MVC using the model and the J2EE correlation technology, then summarizes a typical DataWindow model. It analyzes XMLHttpRequest, the service vessel object, the abstract of service, the backstage control, the URL parameter synthesis and so on. Then it builds the B/S application model with high efficiency, strong transplantation and quick response.

**【Key words】** Ajax; DataWindow; MVC

在传统B/S结构的应用中,存在程序可重用程度低、维护工作繁琐困难、程序应变能力较弱、网页设计者和软件开发者职责不分和开发效率低下等不尽人意的地方,采用MVC设计模式,可以较好地解决上述问题<sup>[1]</sup>。Ajax技术的应用,再次打破了MVC模式的格局,使上述问题可以被更好地解决<sup>[2-4]</sup>。本文设计一种Ajax程序框架来探讨上述问题的解决方法。

### 1 Ajax 开源框架

目前,已经有多种Web开源框架结构,应用效果较好,最杰出的有DWR(Direct Web Remoting),它是一个开源的解决方案,给那些想要以一种简单的方式使用Ajax和XMLHttpRequest的开发者提供了一种便捷的途径。DWR利用一套JavaScript功能集,把Web应用服务器上的对象调用的方法简化了,通过操控不同类型的参数,既保持了程序代码的可读性,又解决了Web应用的异步交互的问题,但也暴露出许多新问题<sup>[4-7]</sup>。

常规的MVC(Model-View-Controller)模型把一个应用的输入、处理、输出流程按照Model, View和Controller的方式进行分离,即应用被分成3层:视图器,模型器和控制器。试图建立业务和技术等多层分离的体系,每个层面都有不同的技术支持,所以MVC技术应用比较复杂,不易推广<sup>[1,3,8-9]</sup>。

在Web应用开发过程中,多层架构已经深入人心。Ajax技术可以使B/S程序结构完全拥有C/S程序结构的所谓“事件灵活性”,是B/S融合C/S技术的最佳解决方案,Web页面的数据交互不与页面整体刷新捆绑在一起,大多数的数据交互都通过后台执行Ajax组件来实现,这种方法极大地提高了Web应用的稳定性和执行效率<sup>[2]</sup>。

### 2 视图器

MVC的视图器,可以认为是用户看到的Web界面。在

Web应用程序中,HTML扮演着基础性的标记的角色,例如:Ajax, Flash, XHTML和XML/XSL等技术,一般总要将标记命令兑现为HTML标记,才能适应Web应用的需要。

在Ajax页面中,视图器的本质发生了变异。技术核心是利用div, span和iframe等块标记(简称DIV块,可以是隐式的)和id标识,应用JavaScript方法,结合用户事件,实现XMLHttpRequest对象的后台数据交换。Ajax使Web页面在不“整体刷新”的情况下,有效地获得数据库数据,减少重复数据的交互,使操作者的业务逻辑更直观。在Ajax技术下的视图器应当改称DIV布局器,如图1所示。

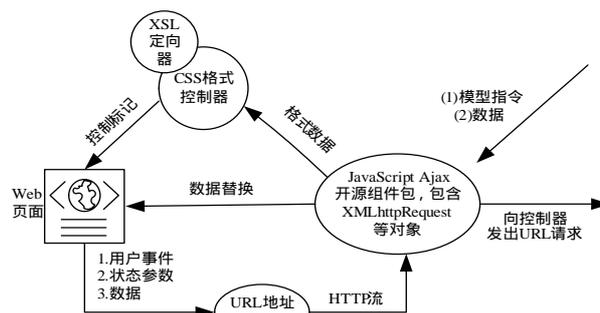


图1 DIV 块布局器示意图

#### 2.1 Ajax 组件包

XMLHttpRequest 等对象组件是 Ajax 组件包的主要内涵,需要进一步丰富 Ajax 组件包,以适应 DIV 布局器的需要。Ajax 的 DIV 布局器采用完全的分层结构,将页面的数据结构、格式表现、参数行为 3 个部分分层处理,将商务业务的对象

**作者简介:** 王东(1966-),男,副教授、在职硕士研究生,主研方向:计算机网络模型与应用;孙彬,副教授、在职硕士研究生

**收稿日期:** 2007-04-09 **E-mail:** wd40@163.com

实例封装在块标记(div)对象的属性中,建立业务处理对象,以响应 XML 业务控制。丰富 DIV 布局器组件的过程,实际上就是通过 JavaScript, XSL 和 CSS 等标记,把业务规则落实到 Web 页面的 DIV 中的过程,它形成含业务逻辑的 HTTP 流数据,以便后台数据交互,完成数据和格式控制的职责。

例如:“req.onreadystatechange=uf1”,其中,uf1()属于用户自定义函数,它根据服务器状态,向浏览器返回业务数据。uf1()主要内容是:

```
if( req.readyState == 4 && req.status == 200) {
    if(业务规则 1)
        document.getElementById(bb).innerHTML= req.responseText;
    或 if( bb=="属性值" )
        document.form1.text1.value = filter(req.responseText); }
```

## 2.2 URL 地址合成

当用户事件(onclick, onchange, onblur 和 oncontextmenu 等)发生的时候,JavaScript 通过 getFormAsString()方法来生成含参数的 URL。控制信息可以由控件的 id.innerHTML 或 name.value 等方法获得,由 JavaScript 调用产生 JSP Servlet 请求,再与保存在数据库中的 XML 密码对照,最终获得完整的业务状态。req.response 对象接收数据的过程与 IE 浏览器的地址栏没有关系。

隐 URL 地址合成的主要难点是不能直接使用“?”和“&”等符号,容易形成 URL 参数串含义的扭曲。考虑到“?”字符和“&”字符在 HTTP 传输中的特殊性,脚本中需要模式匹配技术,进行部分符号替换和映射,方能达到较好的效果。

需要把随机数混合加入到 URL 控制参数中,用以控制 Web 页面的 DIV 块的实时性和有效性,其优点是:(1)防犯非法页面镜像。(2)强制后台实时接收与发送,这对系统稳定性很重要,因为相同 URL 地址的页面,布局不一定相同。(3)隐藏商务业务秘密。出于对安全性的考虑,不能将完整的业务流过程暴露,须使用随机参数构造动态业务过程。

## 3 类 DataWindow 对象

借鉴著名编程工具 PowerBuilder,其 DataWindow 中封装了大多数用户操作,在该技术启示下,结合 Ajax 技术完全可以构造动态的 DIV 标记快,其中封装常见的各种用户操作和用户方法。

### 3.1 Ajax 模型器

结合 Ajax 技术和 MVC 模型,将 Model 部分变异为 Ajax 模型器(AM),使它拥有 Web 处理对象实例化的职责。AM 根据控制器发来的参数信息,自动产生适应商务规则的数据处理对象的实例。例如:把包含了数据结构和组织规则的 DataWindow 对象实例化,即把数据处理方式与数据格式分离,甚至只要提供数据库中的 Table 名,系统就能提供典型的 Web 数据查询和修改模块,如图 2 所示。

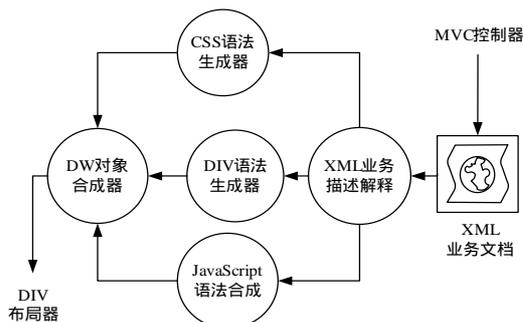


图 2 AM 模型器示意图

描述 AM 结构和基本性能的代码使用 XML 语言。鉴于 XML 语言的特点,控制描述标记可以由用户自定义,并同时自定义 DIV 控制接口规则。AM 是独立于业务控制逻辑和页面表现逻辑之外的处理模型,常常通过 Java Bean 进程长时间驻留于内存中,分布于 J2EE 服务器端,其最终运行结果是动态产生包含商务规则的 DIV 标记格式、方法和参数的代码块,构造内部包容 XML/XSL, CSS 和 JavaScript 标记命令的类似 DataWindow 的容器块对象实例,把业务逻辑合理地布局到 Web 页面的 DIV 上。

类 DataWindow 风格的 DIV 块的设计完成后,在任何一個数据集上进行的数据录入、修改、删除、备份和打印等模块都可以迅速产生,极大地提高了编程效率和程序复用性能。

### 3.2 业务抽象

业务抽象的质量可以表现 AM 的复用性能的好坏,而具体业务表现在 DIV 布局器中,两者有着控制语言和操作平台的距离。业务抽象过程对 AM 以外的其他层来说,属于黑箱操作。AM 模型器接收控制层的数据和控制参数,并返回最终的处理结果对象,例如 DataWindow 等。从应用技术实现的角度看,AM 对业务模型的实现技术做了进一步的探索,以便充分复用现有的组件,强化业务抽象过程。对一个开发者来说,应当专注于业务模型的抽象设计,通过典型过程调用,简化编码设计难度,应用模型按一定的规则抽取出来。抽取的层次很重要,是判断设计方法是否优秀的依据<sup>[9]</sup>。

有了类 DataWindow 对象的 Ajax 处理对象模型后,就可以实现 Web 页面的 DIV 块部分的重用,到达典型业务的比较理想的高开发效率的阶段。例如:鼠标单击一个 Web 数据项,则立刻可以进行单项数据查询,鼠标右击一个数据项,则立刻可以进行属性修改;鼠标双击一个数据项,则可以进行 DIV 对象的切换。这里把 DataWindow 风格的 DIV 块称作重量级胖客户端技术。

## 4 实践模拟结果

将上述 Ajax 对 MVC 模式的改造思想,应用到系统设计中,得到了较好的效果,编程效率得到极大提高,页面响应速度提高 60%,性能比较稳定。

在实现实验室调度管理应用系统的过程中,通过 XML 进行业务描述,产生了通用型业务对象 diss,类似 DataWindow 对象,通过 Java bean 长驻服务器内存。当用户在 Web 页面上的菜单树中单击“材料类型”项时,AM 通过 diss 对象产生对象实例:diss4,然后布局到 Web 页面相应的 DIV 块上,用户可以对数据库的数据表“hua”进行插入、删除、Item 修改和存盘等常见操作;当单击“状态定义”项时,又通过 diss 对象产生对象实例:diss2,再次布局到 Web 页面的 DIV 块上,可以对数据表“sad”进行插入、删除、Item 修改和存盘等常见操作。同时,表格中的任何一个 Item 数据都是一个 Span 标记块,如果用户按下鼠标右键,则立刻能获得修改光标,修改完毕,光标离开时,自动存储修改数据到数据库中,页面并没有被整体刷新,只是该 Span 进行了内容替换,界面非常直观和高效。因为 diss.java 程序被编译,所以 diss 对象实例化的执行速度极快。

AM 通过 Java bean 长驻服务器内存,构造类似于 DataWindow 的 diss 对象,提高了编程效率和系统的稳定性,其主要技术概括为:(1)设计制作 diss.java 程序,使 Web 服务器能产生 diss 对象。(2)对数据库中的数据表进行 XML 登记,

(下转第 83 页)