

Web 开发中的多语言混合编程^①

Mixed Programming in Web Development

李海洋¹ 胡兴华¹ 田乾乾² (1.中国矿业大学 计算机科学与技术学院 江苏 徐州 221008;
2.中国矿业大学 资源与地球科学学院 江苏 徐州 221008)

摘要: 多语言混合开发具有很好的易用性,开发效率高成本较低,可以充分发挥各种语言的优势。本文就 Web 开发中的多语言混合编程从 HTML、CSS、Javascript 在表现层的应用,PHP 与 XML 的混合开发以及 AJAX 技术三个方面做了分析,并预计混合编程在未来将有更多的应用。

关键字: web 开发 多语言 混合编程 AJAX web2.0

1 概述

近年来,随着计算机技术的迅速发展,计算机的应用日趋广泛,软件系统的功能也越来越丰富,但相应的其规模也越来越庞大。在高级语言出现以后软件开发中不同模块使用不同语言编程已有应用,但应用很少,开发方法也不成熟。2000 年以后,网络不断普及,web 应用的不断增多,功能也日益复杂。基于所谓的动态技术的 web 应用如传统的 BBS 以及 web2.0 中的博客、互动社区、在线点播、在线地图等服务也不断涌现。web 开发也成为软件开发中的重要组成部分。web 开发中的多语言协作编程比在其它方面的应用更多也更成熟。

当前在 web 开发中,多语言混合编程已经十分普遍。在 web 表现层使用 html、css、javascript,行为层使用 ASP、JSP、PHP/Perl/Python 等脚本语言实现。最近以 AJAX 为代表的多语言组合开发更使多语言开发在 web 开发上的应用达到了新的高峰。

2 HTML、CSS、JavaScript在Web表现层的静态混合开发

HTML(HyperTextMark-upLanguage)即超文本标记语言,是由 W3C 开发的 WWW 的描述语言^[1]。HTML 是网络的通用语言,一种简单、通用的全置标记语言。和一般文本不同的是,一个 HTML 文件不仅包含文本内容,还包含一些 Tag,中文称“标记”。一个 HTML

文件的后缀名是.htm 或者是.html。用文本编辑器就可以编写 HTML 文件。编写完成后可在浏览器内直接执行。

CSS(Cascading Stylesheet)即层叠样式表,作为网页设计制作的一门基础语言,是 HTML 的控制器,现在越来越得到网页设计师的重视。CSS 不可单独运行,仅作为 HTML 的补充。CSS 按其位置可以分成三种:①内嵌样式(Inline Style) ②内部样式表(Internal Style Sheet) ③外部样式表(External Style Sheet)。CSS 的定义是由三个部分构成:选择符(selector),属性(properties)和属性的取值(value)。

JavaScript: Javascript 是一种解释性的,基于对象的脚本语言(an interpreted, object-based scripting language),其主要是基于客户端运行的,网页里的 Javascript 由浏览器作处理,在客户端完成不需要和 Web Server 发生任何数据交换。

以下是一段简单的综合应用上述几种语言的代码:

```
<html><head>
<title>一个简单的例子</title>
<!--下面这一句是用 css 来设定标题 H1 的表现样式-->
<Style type="text/css">H1.layout {border-width:1; border: solid; text-align: center;color:red} </style>
</head><body>
```

^① 收稿时间:2008-11-22

```

<H1>这是标题一，没有使用样式。</H1>
<H1 class="layout">这也是标题一，使用了 CSS 样式。</H1>
<!-- 下面用 JavaScript 来输出一句文本，用 document.URL 函数来得到当前文件的路径 -->
<script type="text/javascript">
document.write("这是用 JavaScript 输出的。<br>")
document.write(document.URL)
</script>
<p>上面这一行显示的是用 document.URL 得到的本网页的路径。</p>
</body></html>

```

上面这段代码的显示效果如图 1:

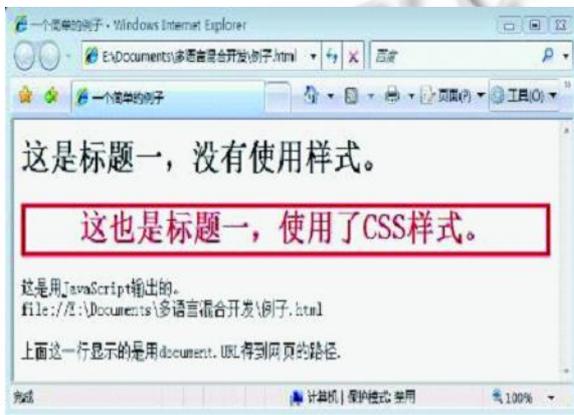


图 1 代码显示效果图

在这个例子中，第二行使用了 CSS 样式为标题一加了边框并设置了颜色。第三行用 JavaScript 的内置函数来输出，虽然和直接输出表面效果一样，但在绝大多数情况下可以做到 HTML 根本不可能完成的任务。比如接下来用 document.URL 得到路径后就只能用 document.write 输出。

在上面的例子中采用的是“内部样式表(Internal Style Sheet)”。而 CSS 在大量网页的应用中更多的是采用“外部样式表(External Style Sheet)”:

```

<link href="xxx.css" rel="stylesheet" type="text/css">

```

其中 xxx.css 代表该样式表文件的路径，上述代码要写在<head>...</head>中。采用外部样式表的

好处是便于大量网页风格的修改，后台开发人员定义好接口后前台设计人员可以十分方便的进行界面的重新设计。

上面只有两行 Javascript 代码，使用了 document.write 和 document.URL 来输出和得到网页路径。利用 Javascript 可以写出很多功能强大的脚本程序。对于多次重复利用的脚本可在 HTML 里引用外部文件里的 Javascript，应在 Head 里写一句<script src="xxx.js"></script>，其中 src 的值，就是 Javascript 所在文件的文件路径。

上述的代码包含在一个 html 文件里直接在客户端(浏览器)里解释执行。也正是由于其解释执行的特点使其能够紧密地混合使用，开发出拥有多种特效的网页。

现在的一般开发模式都是用一种脚本语言进行行为层的开发，用 CSS 控制样式。Javascript 的功能更多的体现在写特效上，而“实用”的更能则由后台程序来实现(AJAX 的出现正在改写这一现状)。

3. PHP与XML在web开发中的混合应用

PHP 是超级文本预处理语言(PHP:Hypertext Preprocessor)的递归缩写。PHP 是一种在服务器端执行的嵌入 HTML 文档的脚本语言，语言的风格有类似于 C 语言，现在被很多的网站编程人员广泛的运用。用 PHP 做出的动态页面与其他的编程语言相比，PHP 是将程序嵌入到 HTML 文档中去执行，执行效率比完全生成 HTML 标记的 CGI 要高许多；与同样是嵌入 HTML 文档的脚本语言 JavaScript 相比，PHP 在服务器端执行，充分利用了服务器的性能。PHP 具有非常强大的功能，所有的 CGI 或者 JavaScript 的功能 PHP 都能实现，而且支持几乎所有流行的数据库以及操作系统。

XML 是 eXtensible Markup Language 的缩写。XML 是一种简单的数据存储语言，使用一系列简单的标记描述数据，而这些标记可以用方便的方式建立，虽然 XML 占用的空间比二进制数据要占用更多的空间，但 XML 极其简单易于掌握和使用。同 HTML 一样，XML(可扩展标识语言)是通用标识语言标准(SGML)的一个子集，它是描述网络上的数据内容和结构的标

准。尽管如此，XML 不像 HTML，HTML 仅仅提供了在页面上显示信息的通用方法(没有上下文相关和动态功能)，XML 则对数据赋予上下文相关功能，它继承了 SGML 的大部分功能，却使用了不太复杂的技术。

作为一般的应用，HTML 应该够用了，但是在处理特殊符号、双字节或者多国文字的支持不够。例如中文信息页面在不同的平台下会出现无法显示等问题。XML 的出现很好的解决了这些问题。

PHP 对 XML 提供了强大的支持。它使用了一个 XML 的“解析器”，并且为了支持这个解析器，它提供了很多 XML 的解析函数。

以下是用 PHP5 实现读取一个以 XML 1.0 格式编写的通讯录 address.xml，并显示其内容的示例。

第一部分：主要定义几个要用到的函数

文件读取函数：

```
function read_file($filename)
```

起始元素处理函数：

```
function start_element($parser, $name, $attrs)
```

终止元素处理函数：

```
function stop_element($parser,$name)
```

字符数据处理函数：

```
function char_data($parser,$data)
```

由于代码内容比较简单。仅对函数功能进行说明；`read_file()`；函数读取 xml 文件内容以便进行处理；`start_element()`；函数是 `xml_set_element_handler` 在开始时调用的函数，在这个例子中主要是对读取内容进行输出样式的设置；`stop_element()`；函数是 `xml_set_element_handler` 在结束时调用的函数，在此主要是输出一条内容后设置水平分割线。`char_data()`；函数对读取的字符数据进行输出。

//第二部分：处理 xml 的代码段

```
<?php
```

```
//xml 文件名
```

```
$file="address.xml";
```

```
//读取文件内容
```

```
$data=read_file($file);
```

```
//产生解析器的实例
```

```
$parser = xml_parser_create();
```

```
//调用起始和终止元素处理函数
```

```
xml_set_element_handler
```

```
($parser, "start_element","stop_element");
```

//当解析器在文件中找到字符数据时调用字符处理函数

```
xml_set_character_data_handler
```

```
($parser,"char_data");
```

```
// 解析文件，
```

```
if(!xml_parse($parser,$data,1)
```

```
{
```

```
//如果解析不成功就输出错误信息
```

```
die(sprintf("XML error: %s at line %d",
```

```
xml_error_string(xml_get_error_code($pa
```

```
rser)),
```

```
xml_get_current_line_number($parser));
```

```
}
```

`xml_parser_free($parser);`//释放解析器所占内存?>

这个例子中 PHP 与 XML 混合应用实现了一个简单的通讯录，并未使用数据库(对于小规模应用也没必要使用数据库)。PHP 的功能十分强大，在 web 开发中结合其他语言可以解决很多棘手的问题，比如与 Javascript 协作通过读写 cookie 实现高级页面缓存。在现代 web 系统中将越来越多的用到这种混合开发技术。

4 AJAX简介

AJAX 全称为“Asynchronous JavaScript and XML”(异步 JavaScript 和 XML)，是指一种利用多种语言进行开发创建交互式网页应用的网页开发技术。

AJAX 主要包含了以下几种技术：基于 web 标准(standards-based presentation)XHTML+CSS 的表示；使用 DOM(Document Object Model)进行动态显示及交互；使用 XML 和 XSLT 进行数据交换及相关操作；使用 XMLHttpRequest 进行异步数据查询、检索；使用 JavaScript 将所有的东西绑定在一起。

传统的 web 应用允许用户填写表单(form)，当提交表单时就向 web 服务器发送一个请求。服务器接收

并处理传来的表单, 然后返回一个新的网页。这样浪费了许多带宽, 因为在前后两个页面中的大部分 HTML 代码往往是相同的。由于每次应用的交互都需要向服务器发送请求, 应用的响应时间就依赖于服务器的响应时间。这导致了用户界面的响应比本地应用慢得多。与此不同, Ajax 应用可以仅向服务器发送并取回必需的数据, 它使用 SOAP 或其它一些基于 XML 的 web service 接口, 并在客户端采用 JavaScript 处理来自服务器的响应。因为在服务器和浏览器之间交换的数据大量减少, 结果我们就能看到响应更快的应用。同时很多的处理工作可以在发出请求的客户端机器上完成, 所以 Web 服务器的处理时间也减少了。Ajax 应用程序的优势在于: 通过异步模式, 提升了用户体验优化了浏览器和服务器之间的传输, 减少不必要的往返, 减少了带宽占用 Ajax 引擎在客户端运行, 承担了一部分本来由服务器承担的工作, 从而减少了大用户量下的服务器负载。

与传统的 Web 开发不同, Ajax 并不是以一种基于静态页面的方式来看待 Web 应用的。从 Ajax 的角度看来, Web 应用仅由少量的页面组成, 其中每个页面其实是一个更小型的 Ajax 应用。而一些简单的 Ajax 应用, 例如一个简单的 RSS 阅读器, 甚至只有一个页面。每个页面上面都包括有一些使用 JavaScript 开发的 Ajax 组件。这些组件使用 XMLHttpRequest 对象以异步的方式与服务器通信, 从服务器获取需要的数据后使用 DOM API 来更新页面中的一部分内容。因此 Ajax 应用与传统的 Web 应用的区别主要在 3 个地方: ①不刷新整个页面, 在页面内与服务器通信。②使用异步方式与服务器通信, 不需要打断用户的操作, 具有更加迅速的响应能力。③应用仅由少量页面组成。大部分交互在页面之内完成, 不需要切换整个页面。由此可见, Ajax 使得 Web 应用更加动态, 带来了更高的智能, 并且提供了表现能力丰富的 Ajax UI 组件。

下面实现一个基本的 AJAX 实例:

(1) XMLHttpRequest 对象的获取

```
/*以支持多种浏览器的方式创建 XMLHttpRequest 对象*/
var xmlhttp = false;
```

```
/*@cc_on @*/
/*@if (@_jscript_version >= 5)
try {/*尝试使用 Msxml2.XMLHTTP 对象创建它*/
xmlhttp=new ActiveXObject("Msxml2.XMLHT-
TP");
} catch (e) {
try {/*如果失败, 尝试 Microsoft.XMLHTTP*/
xmlhttp=new ActiveXObject("Microsoft.XMLH-
TTP");
} catch (e2) {xmlhttp = false;}
}@end @*/
/*如果仍然没有建立 xmlhttp, 则以非 Microsoft
的方式创建该对象*/
if (!xmlhttp && typeof XMLHttpRequest != 'und-
efined')
{ xmlhttp = new XMLHttpRequest();}
```

(2) 发送请求

```
function callServer() {
//从 web 表单获取数据
var city = document.getElementById("city").val-
ue;
var state= document.getElementById("state").va-
lue;
//获取不为空的地区的数据
if ((city == null) || (city == "")) return;
if ((state == null) || (state == "")) return;
//建立连接所用的 url
var url="/scripts/getZipCode.php?city="+
escape(city) + "&state=" + escape(state);
//建立一个新连接到服务器
xmlhttp.open("GET", url, true);
//当建立完成为服务器设置一个可用的函数
xmlhttp.onreadystatechange = updatePa-
ge;
//发送请求
xmlhttp.send(null);
}
```

(3) 处理服务器响应

```
function updatePage() { //当准备好时就处理处理  
    响应  
    if (xmlHttp.readyState == 4) {  
        var response = xmlHttp.responseText;  
        document.getElementById("zipCode").value =  
        response;}  
    }
```

AJAX 并不是一种新的技术,它只是灵活的使用多种语言。用 JavaScript 将一些已有的技术(XHTML、CSS、DOM、XML、XMLHttpRequest)绑定在一起,形成一个有机的技术混合体。相信随着多语言技术应用的的发展,还会出现很多类 ajax 的神奇技术。AJAX 的缺点主要是浏览器后退功能的失效。因为在 AJAX 下,页面的动态更新并不被浏览器认为是进入另一个网页。

5 结束语

现代应用系统建设复杂程度指数级增加,传统的单一语言编码已经无法满足现有应用系统的开发。越来越多的系统也在不自觉的采用了多语言混合编程,在 web 开中更是这种趋势更是明显。在表现层,行为

层都已经使用了多种语言的混合编程。而且随着 web 的发展组合技术的使用越来越多也越来越成熟(如 AJAX)。多语言混合开发具有很好的易用性,开发效率高成本较低,可以充分发挥各种语言的优势使开发变得更轻松。在相同的工作时内能开发出更多的应用,并且提高系统的性能和安全性。相信软件开发(不仅是 web 开发)即将进入混合开发的时代。

参考文献

- 1 Musciano C, Kennedy B. 陈菊明译.HTML 与 XHTML 权威指南(第四版).北京:中国电力出版社,2001.
- 2 莫里.陈黎夫等译.CSS 禅意花园.北京:人民邮电出版社,2007.
- 3 美亚沃斯基.邱仲潘,等译.JavaScript 从入门到精通.北京:电子工业出版社,2002.
- 4 威利等.武欣等译.PHP 和 MySQL Web 开发.北京:机械工业出版社,2005.
- 5 袁鹏飞.XML 编程起步.北京:人民邮电出版社,2001.
- 6 哥特兰,高伯瑞斯,艾米亚,徐锋,胡冰译.Ajax 修炼之道:Web2.0 入门.北京:电子工业出版社,2006.