

# 基于 XML 网站的设计与实现

## On the Technique Realization of XML-based Website

魏武华 (西安陕西省行政学院计算机系 710068)

**摘要:** 本文对传统WEB网站和XML网站的机制进行了比较,指出XML网站利于客户与服务器进行大量数据传输以及易于实现异构多数据源信息访问的优势,并以ASP为例阐明了XML网站的技术实现。

**关键词:** 计算模式 XML ASP

### 1 客户 / 应用服务器 / 数据库服务器三层计算模式的基本概念

一般客户/服务器的体系结构可划分为二层结构和三层结构。而目前流行的三层结构的英文名称是Client(Browser)/Server/Database Server。在网络环境中,客户向服务器端发出服务请求,服务器端接收并处理客户的请求,然后把处理结果返回给客户端。

Client和Server系统不单指硬件和软件,通常是指软、硬件相结合的环境。如果从硬件层面上看问题,“客户”指的是用户使用的计算机,“服务器”指的是提供服务的服务器计算机。如果从软件层面看问题,“Client”和“Server”指的是软件。“Client”软件和“Server”软件可以在一台计算机上运行,也可以在网络环境中的两台或多台计算机上运行。

通过客户机访问web的结构如图1所示。在三层结构中,客户机通过Internet/Intranet网络访问应用服务器,应用服务器根据客户请

求自动完成对数据库数据的操作,然后把处理结果返回客户机。

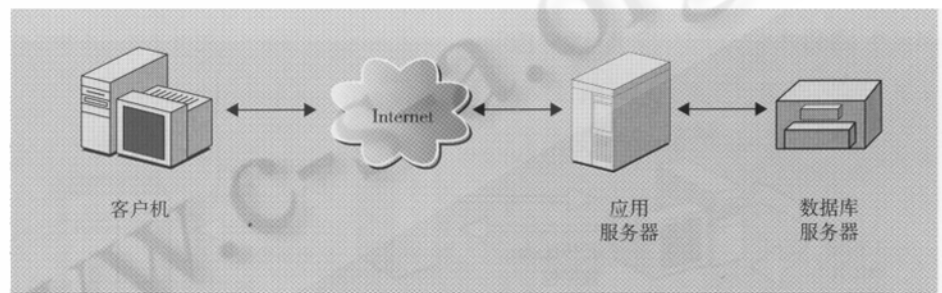


图1

### 2 传统的 web 网站的基本工作过程

web系统是一种基于超链接(hyperlink)超文本(hypertext)和超媒体(hypermedia)系统,由于提供媒体的多样性,也称为超媒体环球信息网。

Web的工作过程如图2。它的工作步骤如下:

- (1) 用户启动客户端浏览器,在浏览器中确定将要访问页面的URL地址。浏览器软件使用HTTP协议,向该URL地址所指向的web服务器发出请求。
- (2) web服务器根据浏览器送来的请求,把URL地址转换成页面所在服务器上的文件路径名,找到相应的文件。
- (3) 如果URL指向HTML文档,web服务器使用HTTP协议将该文档直接送给浏览器。在

HTML文档中可能包含有JavaScript和VBScript等脚本程序段，随HTML一起下载的还可能有Java Applet和ActiveX等小程序，把结果送到浏览器。

(4) 浏览器解释HTML文档，在客户端屏幕上向用户展示结果。

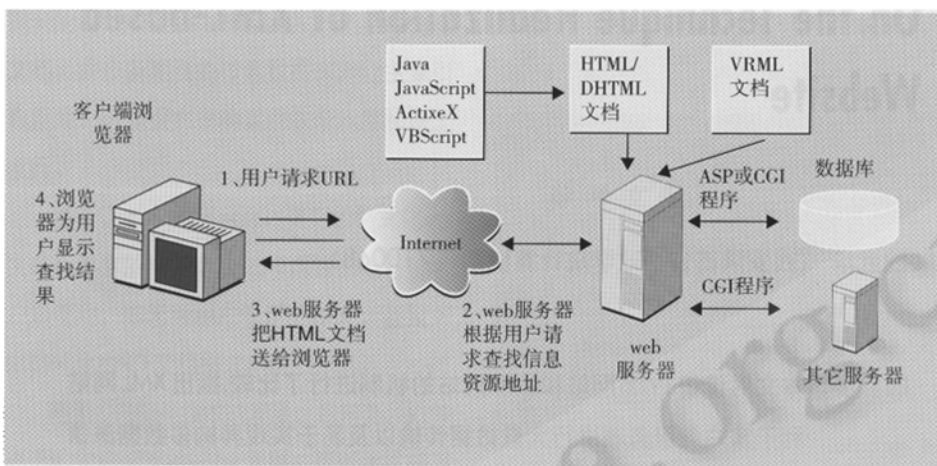


图 2

### 3 XML 网站架构

一个完整的XML网站架构如图3所示，不管是服务器还是客户端送出的数据都是XML。它和传统的web网站有很大的效率差别，传统的web网站由于客户端无法一次送太多数据到服务器，因此，若要更新数据库的数据，必须分成很多次将所有的信息传送到服务器。如此一来，对于网络和服务来说是非常没有效率的。新的XMLHTTP对象则允许我们传送完整的XML包到服务器，因此，可以大幅提升网站服务的效率。

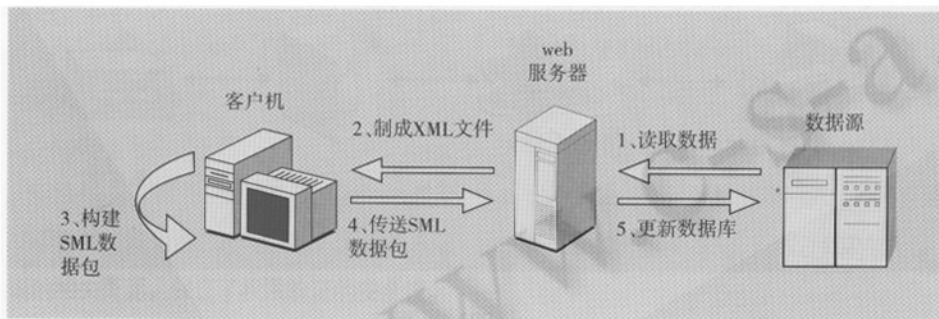


图 3

步骤一：服务器端的ASP程序从数据库中取出数据，并制作成XML文件。

步骤二：服务器将XML文件传送到浏览器，为了显示数据，我们可以附带一个XSL以便于在浏览器端将XML转换为HTML网页。

步骤三：客户端浏览器通过VBScript或JavaScript,将要送到服务器端的数据制作成XML数据包。

步骤四：客户端浏览器通过XMLHTTP对象将XML数据包送到服务器端做进一步的处理。

步骤五：服务器端的ASP通过DOM的协助取出XML包内的相关数据，并使用这些数据来更新数据库的内容，完成整个过程。

## 4 基于XML的系统架构实现

### 4.1 服务器端的实现机制

一个基于XML的网站中，服务器端可能返回静态的XML文件也可能通过与数据库的交互，动态产生一个XML文件。可通过使用ASP或PHP等服务器端的Script语言来编写程序代码，访问数据库（案例库）的资源。将其组织成XML文件，传送到客户端。本文以ASP为例，讨论利用ASP实现XML应用的机制。

ASP是通过ADO来访问数据库资源的。在通过ADO连接数据库之前，首先要声明一个对象作为与数据库连接的接口。采用如下的声明方式：

```
Set 变量名称=Server.CreateObject
("ADODB.Connection")
```

该变量是一个与数据库连接和交换数据的对象，称为Connection对象。然后，可利用如下的语法连接到数据库中：

```
Connection.open ConnectString,UserID,
Password,Options
```

其中open是Connection对象的一个函数，可以用来打开数据库与Connection对象的通道；ConnectString是open函数所需要的一个参数，可包括provider, Data Source, Initial Catalog, UesrID, Password等五种信息来分别说明该Connection是哪一种服务所提供，Connection数据源名称，所需要访问的数据库名称，连接数据库所需要的用户帐号及用户密码。

Recordset对象是用来存储查询数据库的结果。对Recordset必须先声明，再通过其Open 函数或Execute函数访问数据库，而Recordset对象的Open函数可以通过Connection的帮助获取数据库查询的结果。

Recordset的声明：

```
Set 对象变量=Server.createObject
("ADODB.Recordset")
```

通过Open获取数据：

```
Set 对象变量A=Server.createObject
```

("ADODB.Recordset")

对象变量A.Open "SQL命令", Connection对象变量

通过Execute获取数据:

Set 对象变量A=Server.createObject ("ADODB.Recordset")

Set 对象变量A=Connection对象.Execute ("SQL访问")

获取Recordset对象中的数据,方法有很多种,常用的方式有:

(1) 使用字段名称读取:

Recordset对象变量.Fields("字段名称")

(2) 使用字段索引读取字段数据:

Recordset对象变量.Fields(字段索引值) < !-从0开始- >

为了节省内存,使用完毕Recordset和Connection对象后,使用Close对象来关闭这两个对象。其形式为:

Recordset对象变量.Close

Connection对象变量.Close

ASP中的脚本程序对Recordset对象中存储的查询结果以XML文件形式传送到客户端。

用ASP返回XML文件时,至少需要确认以下三件事:

(1) 是否声明了XML版本信息以及编码方式。

(2) 是否所有的标记都有开始与结束标记,或者以空标记形式出现。

(3) 是否含有唯一的一个根标记。

使用ASP返回XML的标准模板如下:

```
<% Response.ContentType="text/XML"%>
<?xml version="1.0"
encoding="GB2312"?>
```

<根元素>

..... XML内容

</根元素>

使用ASP返回XML文件的时候,应考虑XML标记与属性的意义。可编写相应的DTD或Schema,以该语言规定的标记构成相应的

XML文件。由于在DTD或Schema中对标记进行了定义,因而,使用该标记的XML文档是符合格式的(Well-Formed)且有效的(Validated)XML文件。

服务器将XML文件传送到浏览器,为了显示数据,可以附带一个XSL以便于在浏览器端将XML转换为HTML网页。

同样,可以通过ADO与数据库获取连接并将XML数据存入数据库中。

服务器要处理来自客户端的XML包,首先必须在服务器端建立起XML DOM对象来接收XML数据包。使用ASP建立一个服务器端的XML DOM对象的语法范例如下:

```
<% Set ReceivedDoc=CreateObject
("Microsoft.XMLDOM") %>
```

随后,应设置以同步或异步加载,并可通过Load函数直接指定要加载的对象为对象,也就是所接收到的XML数据包。语法范例如下:

```
<% ReceivedDoc.synchronous=false
```

```
ReceivedDoc.load Request %>
```

最后,我们可以读取该XML DOM树内的任何节点,并做应用。

```
<% Set rootNode= ReceivedDoc.
documentElement
```

```
.....
```

```
%>
```

#### 4.2 客户端的实现机制

客户端浏览器通过VBScript或JavaScript,将要送到服务器端的数据制作成XML数据包。而要建立一个XML数据包,首先要建立一个XML DOM对象,然后将要传送的数据存入放入这个对象中。

声明一个XML DOM文件对象的方法:

```
Set XML DOM文件对象名=CreateObject
("Microsoft.XMLDOM")
```

然后,通过该对象的loadXML函数指定XML数据包内容(包括用户通过表单输入的内容)。

传送XML数据到服务器主要是通过XMLHTTP对象来完成,它同样使用HTTP协议传送数据到服务器端。为了简化流程,微软制作了一个COM对象,称为XMLHTTP对象,通过该对象传送XML数据到服务器端。首先,必须声明XMLHTTP对象,其语法范例如下:

```
Set xmlhttp=CreateObject("Microsoft.
XMLHTTP")
```

同样,该对象提供许多的属性和函数,通过这些函数的帮助,我们可以将XML数据传送到服务器。在送出XML数据之前,首先必须通过open函数设置相关传送细节,然后通过Send函数来执行传送XML的操作。

## 5 结束语

基于XML的网站解决了传统的web网站的客户端无法一次送太多数据到服务器的问题,实现了XML数据的双向传递,即可将客户端的数据组织成XML文件传送到服务器端,也可将服务器端的数据库数据转换成XML文件传送到客户端供不同应用。利用XML文档作为访问数据库的中间层,将服务器端传统数据库和其它文档数据转换为XML的数据源,然后基于XML数据源在客户端进行查询,以屏蔽掉多数据源的异构性和分布性。

## 参考文献

- 1 段友祥、岳后光,基于XML的异构多数据源信息访问,《微型机与应用》,2002,(6): 30-31。
- 2 陈锦辉,XML与ASP网站实作大全,中国铁道出版社,2001。