

# 基于WEB的XML数据交换技术

李寅 西安交通大学 硕士

林宣雄 西安交通大学 副教授

摘要: 随着XML的出现和不断成熟,XML将作为一种标准方法实现WEB上的数据交换。本文首先简要介绍了XML,接着介绍了基于WEB的XML数据交换技术的一般框架和原理,最后结合微软的XML技术框架以实例来演示如何实现基于WEB的XML数据交换技术。  
关键词: XML、XSL、DOM、WEB、ASP、ADO

## XML的出现及特性

XML (Extensible Markup Language) 是基于SGML实现的一个子系统,它保留了SGML主要的实用功能,同时大大缩减了SGML的复杂性。SGML是Standard Generalized Markup Language (标准通用标记语言)的缩写,它是一种元语言 (metalinguage)。所谓的元语言是一种用来定义其他语言的语言,例如目前在Internet上广泛使用的HTML就是一种SGML应用。

随着Web的普及并以爆炸性的速度增长,原来广泛使用的HTML所暴露的不足之处越来越突出,如缺乏可扩展能力、数据的可重用性和结构化程度差、缺乏表达数据语义的能力等,这些缺陷使得HTML面对越来越复杂的网站内容和大型的商业发布应用显得力不从心。于是W3C针对大型商业发布应用需求和考虑未来可扩充性提出了XML,XML在以下三个方面与HTML有显著差别:

1. 可以任意定义新的标签和属性名称
2. 文档结构可以进行任意复杂层次嵌套
3. 任何XML文档可以包含可选语法描述,这些描述可以被需要进行结构合法性验证的应用所使用

## XML的应用范围

XML的基本特性赋予了它最大的可延伸性。按照XML工作组的Bosak的看法,XML的应用范围可分为4大类:

1. 要求Web客户机在两个或多个不同的数据库之间传递信息的应用;
2. 希望将Web服务器的大量处理负荷转移给Web客户机的应用;
3. 要求Web客户机把同一数据以不同的表现方式提供给不同用户的应用;
4. 适应特定用户需求的智能Web工具应用。

目前,XML除了在数据库系统和Web语言等领域

有广泛应用之外,其它许多应用领域也按照XML定义各自的语言标准,如WML、SMIL等。

## XML与DOM、SAX、DTD和XSL

要理解XML的应用,首先要了解DOM、SAX、DTD和XSL。

1. DOM (Document Object Model, 文档对象模型) 就是一个结构化文档编程接口 (API), 它定义了文档的逻辑结构以及访问和操纵文档的方法, XML文档中的元素就是DOM树中的节点对象。使用DOM模型, 程序员可以方便地创建文档、导航其结构, 或增加、修改、删除、移动文档的任何成份。

2. SAX (Simple API for XML) 是一种基于事件的模型, 可用于避免构造文档完整的树状结构。如果文档特别庞大或仅对其中一部分感兴趣, SAX就显得特别有用了。SAX事件由解析器在进入或离开文档、元素、属性、子树时产生。该模型可在不关注文档完整结构的情况下测定其特征或相关数据。

3. DTD (Document Type Definition, 文档类型定义) 规定文档的逻辑结构。它可以定义文档的语法, 而文档的语法反过来能够让XML语法分析程序确认某张页面标记使用的合法性。DTD定义页面的元素、元素的属性以及元素和属性之间的关系。例如, DTD能够规定某个表项只能在某个列表中使用。

4. XSL (Extensible Style Language, 可扩展的样式语言) 是用于规定XML文档样式的语言。XSL能使Web浏览器改变文档的表示法, 例如数据的显示顺序改变, 就不需要再与服务器进行交互通信。通过变换样式表, 同一个文档可以显示得更大, 或者经过叠折只显示外面的一层, 或者变为打印格式。

DOM、SAX、DTD和XSL与XML之间的关系如下:

1. DOM提供了对XML文档模型化 (树结构) 功能,

并对文档树进行操作。

2. SAX 提供的功能与 DOM 类似，只是 SAX 采用事件机制。
3. DTD 是对 XML 文档的合法性进行校验。
4. XSL 规定 XML 文档样式，通过改变 XSL 文件可以改变 XML 文件的显示样式。

### 技术框架和原理

使用 XML 在 WEB 上实现数据的融合、发布和显示的三层框架结构可以用图 1 表示。

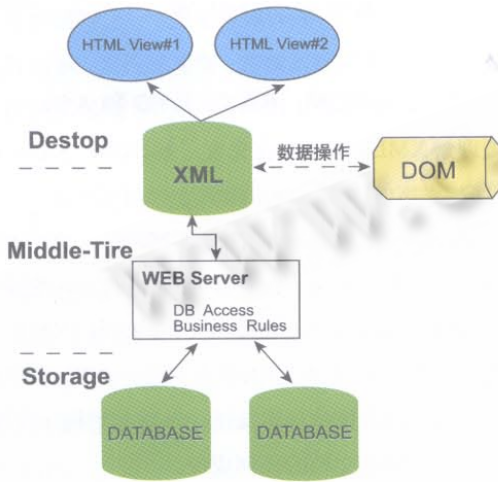


图 1 基于 WEB 的 XML 数据交换结构

基于 WEB 的 XML 数据交换结构主要分为三层，分别是数据存储层 (Storage)、中间层 (Middle-Tire) 和显示层 (Desktop)，各层主要完成的功能如下：

\* Storage (数据存储层): 主要是实现数据的存储管理，一般由系统后台数据库系统实现，存储的数据可以由异种数据源构成。

\* Middle-Tire (中间层): 主要通过数据存储层提供的接口 (如 SQL) 实现数据的访问和操作，并将数据转化为 XML 格式，并实现系统的业务规则，同时完成对 XML 格式数据的处理。中间层可以从多种数据源中提取所需要的数据并进行融合，通过 XML 技术，这种异种数据源的数据提取操作是透明的。通过 DOM 可以对 XML 文档树结构的各个对象节点进行处理，完成对 XML 数据的显示样式处理操作。

\* Desktop (显示层): XML 配合 XSL(Extensible Style Language, 可扩展的样式语言) 可以按照预先定义的样式显示，改变 XSL 就可以改变 XML 数据的前台显示。例如，改变数据的显示顺序，叠折只显示外面的一层，或者变为打印格式。这种显示方式的处理如图 2 所示。

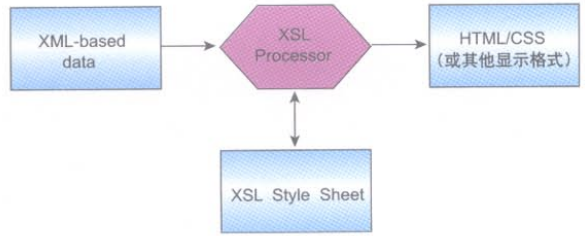


图 2 使用 XSL 显示 XML 数据

### 实例研究

为说明基于 WEB 的 XML 数据交换技术，本文以一个简单的例子来说明整个技术框架的实现过程。这个实例完成以下功能：

1. 客户端 (浏览器) 提交数据请求给 WEB 服务器，WEB 服务器从数据库中提取数据。
2. 将数据库返回的数据格式化为 XML 文件。
3. WEB 服务器通过 XSL 样式文件使用 DOM 接口将 XML 格式化为 HTML 文件并返回给客户端 (浏览器)，由浏览器完成数据的显示功能。

本文的实例运行的服务环境如下：

1. Windows 2000 Server 中文版 + IIS5.0
2. Internet Explorer 5.5
3. Microsoft SQL Server 7.0
4. 在 SQL Server 7.0 建立 test 库，用户名为 test，口令为空，其中有一个客户表 customer，如表 1。

表 1 Customer 表结构

字段名称	字段类型和长度	字段说明
CUST_ID	CHAR(10)	客户编号
CUST_NAME	CHAR(20)	客户姓名
PHONE	CHAR(30)	客户电话
EMAIL	CHAR(30)	客户电子邮件

### 技术实现简介

本文的实例基于微软的 ASP 和 XML 实现技术，技术框架如图 3 所示：

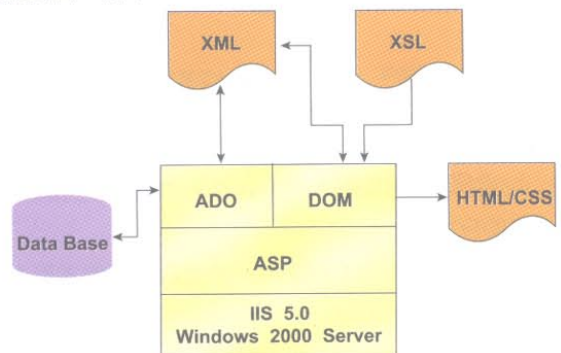


图 3 基于 WEB 的 XML 数据交换技术实现框架

图3所示的技术实现框架包含以下要点:

1. 采用Microsoft的IIS和ASP作为底层WEB服务平台。
2. WEB数据库的访问采用Microsoft的ADO(ActiveX Data Object)组件实现, Microsoft的ADO组件版本从2.1(Windows 2000的ADO版本为2.5)开始提供了直接将数据集保存为XML标准格式文件,这也是本文实例的所利用的实现手段。
3. Microsoft提供了DOM组件(XMLDOM),直接可以对XML文档进行操作,同时结合XSL文档可以将XML格式化为HTML输出。

### 实例代码分析

本文实例的核心代码如下所示(为了使代码更清楚,去掉了一些辅助代码,如出错处理等,以保证代码精简易懂):

```

1. <%@ Language=VBScript %>
2. <!--#include file="adovbs.inc"-->
3. <%
4. dim strConn, strSQL, objRS
5. strConn="Provider=SQLOLEDB.1;Persist
Security Info=False; User ID=test; password=; Initial
Catalog=test; Data Source=ntserver"
6. strSQL = " select
CUST_ID,CUST_NAME,PHONE,EMAIL from
CUSTOMER"
7. set objRS = Server.CreateObject(
"ADODB.Recordset" )
8. objRS.Open strSQL, strConn ,adOpenStatic,
adLockReadOnly, adCmdText
9. objRS.Save "d:\xml\customer.xml",
adPersistXML
10. Dim objDomSource, objDomStyle
11. set objDomSource = Server.CreateObject
("Microsoft.XMLDOM")
12. objDomSource.async = false
13. objDomSource.load("d:\xml\customer.xml")
14. set objDomStyle = Server.CreateObject
("Microsoft.XMLDOM")
15. objDomStyle.async = false
16. objDomStyle.load("d:\xml\customer.xsl")
17. Response.Write(objDomSource.transformNode

```

```
(objDomStyle))
```

```
18. dim objFS
```

```
19. set objFS = Server.CreateObject(
"Scripting.FileSystemObject" )
```

```
20. objFS.DeleteFile "d:\xml\customer.xml", True
```

```
21. %>
```

在实例代码中,第4到第8行主要是利用ADO访问数据库,并生成记录集,第9行是将记录集中的数据存为XML格式,并将内容保存在硬盘文件上。第10到16行代码作用是使用DOM(Microsoft.XMLDOM)装载XML数据文件和XSL样式文件,第17行代码是使用DOM利用XSL文件样式将XML数据文件格式化为HTML文件输出给客户端(浏览器)。第18到20行代码是将临时保存在硬盘上的XML文件删除。

### 总结

XML用文档描述和应用功能分开的原则使描述同一类型所有文档的共同属性成为可能,使用XML,可以在结构上保证文档间的数据传递和处理大大简化,不同数据源的互操作性变得简单。尽管XML从本质上具有许多突出优点,但任何事物的成熟都需要经过时间的考验,在目前阶段XML还有比较长的路要走,还有一些不尽如意的地方(如使用还不够方便、许多标准还不够完善),但我们看到XML正逐渐漫向成熟,很多标准正不断出台,现在,大多数关系数据库厂商都添加了一个XML层,这个XML层提供数据转换。使用XML来实现基于WEB的数据交换是Internet技术发展的一个重大方向。■

### 参考文献:

1. 《XML, Java, and the future of the Web》Jon Bosak, Sun Microsystems  
<http://www.ibiblio.org/pub/sun-info/standards/xml/why/xmlapps.html>
2. 《XML从入门到精通》Ann Navarro等著,周生炳等译,电子工业出版社,2000年8月
3. 《msdn Online Web Workshop》Microsoft Corporation,  
<http://msdn.microsoft.com/workshop/c-frame.htm?workshop/xml/index.asp>
4. 《NT动态站点设计指南》周世雄 编著,大连理工大学出版社,1997年12月