

基于 ADO 技术的 XML 数据库信息处理

Dealing with XML RDBS Based on ADO

梁莹 陈正铭 (韶关学院 计算机系 512005)

摘要:XML 已经成为万维网重要的信息发布标准和表示技术之一,越来越多的应用之间采用 XML 进行数据交换。ADO 是一种常用的数据访问技术,它与 XML 建立了一种新型的 Web 上数据交换模型。本文最后给出了一个具体的实例,验证了这种信息处理方法的有效性。

关键词:ADO XML DOM XSL

1 引言

随着 XML 及其相关技术的应用发展,XML 已经成为万维网重要的信息发布标准和表示技术之一,越来越多的应用之间采用 XML 进行数据交换,例如电子商务、远程教育等新的 Web 应用在要求高容量、高信息量的数据传输的同时,还要求数据能以不同形式在不同的平台间交换,因为这些应用程序要交换以许多不同的方式形成的集合。XML 的内在特点以及它与 ADO 技术的结合,刚好可以满足这样的要求。

2 XML 简介

2.1 XML 的产生

XML (eXtensible Markup Language, 可扩展的标记语言) 是互联网标准化组织 W3C 发布的通用标记语言 SGML (Standard Generalized Markup Language, 标准通用标记语言) 的一个大大简化了的子集,即是 SGML 的简化版本,并取了其中最常用部分,以一种可理解的方式将它包装起来。它提供了运行更智能、更快的内部网应用程序所需的東西。有了 XML, 文档的组成部分像在 HTML 中那样用标志和属性将它们标记出来。XML 与 HTML 的不同之处在于它不具备内嵌的标志编排能力,因为没有内嵌的标志。因此要将其内容显示出来还要编写样式表 XSL (eXtensibal Stylesheet Language, 可扩展样式语言)。XML 是 HTML 的扩展,有能力处理元数据。因此,对 XML 文档而言,专业的文档管理和检索极其容易。SGML 的过于复杂,HTML 的过于简单,使它们都不能适应新的 Web 应用的要

求,所以推出了 XML。XML 以强大的数据描述功能,简单的语法吸引了人们,并得到了迅速的推广和广泛的应用^[1]。

2.2 XML 数据存贮机制

通常用户会有各种各样的应用需要使用 XML 进行数据传输。XML 数据源有很多,大体可分为下面三种:一种是 XML 格式的纯文本文档,这种来源是基本的也是最为简单的,存储于文件中的数据可以直接读取,然后通过 XSL 对 XML 文档进行转化并在浏览器中显示,或者通过 DOM 接口编程同其他应用相连,DOM 即文档对象模型,它能读取 XML 文档中的节点,是最基本也是最底层的 XML 存取技术,它实际上是提供一组 API 来存取 XML 数据。第二种是关系型数据库,这种方法通过中间件在数据库和 XML 文档之间建立桥梁关系,便于开发各种动态应用。它通过数据库系统对数据进行管理,然后再利用服务器段应用 (ASP、JSP、Servlet) 对其动态的存取 (如图 1)。第三种则来源于其它各种应用数据,如邮件、目录清单、商务报告等。

3 ADO 与 XML 构成的数据交换模型

3.1 ADO 技术

ADO (ActiveX Data Objects) 是一种应用于 Windows 分布式网络应用程序体系结构中的数据访问技术。ADO 利用 OLE DB 技术可以访问多种数据库源,除了与传统的数据源一起协调的工作,还可以与非关系型的数据源协调工作,如目录服务、活动目录服务接口 (Active Directory Service Interface ADSI) 等。

ADO 由几个分别独立的对象组成,每个对象及其可以完成的操作被封装起来,以后就可以对许多不同的数据库重复使用该对象,以这种方式组织起来的模型为开发人员提供了更大的灵活性。**Connection** 对象主要的功能是建立客户端与服务器端数据库的连接;**Command** 对象是对数据存储执行命令的对象,可以用它执行 SQL 命令;**RecordSet** 对象含有从数据存储中提取的数据集,由客户端直接操作。

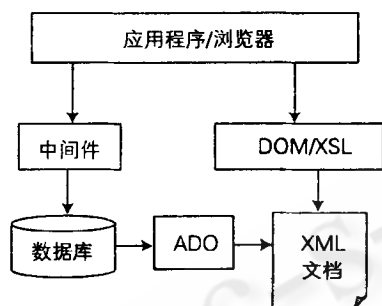


图 1 关系型数据库的数据存取机制

3.2 构成原理

基于 ADO 技术的 XML 数据交换模型其结构分别是数据存储层、中间层和数据显示层^[2]。

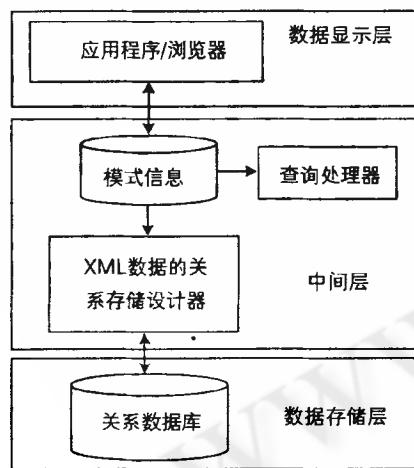


图 2 XML 数据交换模型

各层完成的功能如下:

(1) 数据存储层。本层主要实现数据的存储管理。因为目前大多数的关键数据都放置于数据库中进行管理,数据库技术已经相当成熟而且其管理功能非常大,所以此层由后台数据库系统实现,如 SQL Server、Oracle 等。存储层也可以由其他的数据源构成。

(2) 中间层。这一层实现的功能是对数据库的数据进行操作,将其转换为 XML 数据。主要通过 ADO 的 **Connection** 对象和 ODBC 与数据库建立连接,执行指定的 SQL 语句,生成一组记录。生成 XML 文档的方法有两种,一是将数据库的结果直接保存为 XML 文档。另外 DOM 提供了一组与平台和语言无关的接口,利用 DOM 提供的方法读取数据也可动态的生成 XML 文档。

(3) 数据显示层。利用 DOM 将预先定义好的 XSL 文件中的样式信息添加到 XML 数据中,使其变成可以在浏览器上显示的样式化数据。由于 XML 的内容与形式可分离,XML 的样式信息被放置于 XSL 文件中,当它要在浏览器上显示时,才添加进去。这样只改变 XSL 文件就可以改变 XML 数据的前台显示。

4 基于 ADO 技术的 XML 数据交换的实现

4.1 实例实现的功能

XML 是一种文件格式,它没有规定与数据库的连接方法,需要用传统的方法连接数据库,进行数据库查询,然后将查询的结果转化为 XML 格式。下面将结合一个实例说明 ADO 技术实现 XML 转换的实现过程。实例完成的功能如下:

(1) 客户端提交请求给 Web 服务器,Web 服务器访问数据库。

(2) 将数据库返回的数据转换为 XML 文档。

(3) 完成对 XML 文档样式化,然后输出到浏览器。

4.2 代码分析^[3]

本例使用 SQL SERVER 2000 建立的数据库 sample,其中有一张表 book。

(1) 使用 ADO 的 **Connection** 对象建立与数据库的连接

```

Function GetRecordSet (
Set conn = Server.CreateObject("
Adodb.Connection")
Conn.Open "pp", "", ""
Set rs = conn.Execute("select * from book")
End Function
  
```

其中 pp 是使用 ODBC 建立的数据源的名称。

(2) 动态生成 XML 文档,加上样式信息在浏览器上显示

```
<% = TransformXML ( RecordSettoXMLDoc ( GetRecordSet() ," book" ), " book. xsl" ) % >
```

‘RecordSettoXMLDoc() 将记录保存在 XML 文档中, 通过如下循环实现:

```
Function RecordSettoXMLdoc (
```

```
‘建立 DOM 对象
```

```
Set objXML = Server. CreateObject( "
```

```
Microsoft. XMLDOM" )
```

```
Set objnodecol = addXMLnode( objXML,
```

```
objXML, nodename&"s", "" )
```

```
While not rs. eof
```

```
‘动态生成 DOM 树的节点
```

```
Set objnodecol = addXMLnode(
```

```
objXML, objnodecol, nodename, "" )
```

```
For each x in rs. fields
```

```
addXMLnode objXML, objnode, x. name, trim ( x, value)
```

```
next
```

```
rs. MoveNext
```

```
Wend
```

```
Set recordsettoXMLdoc = objXML
```

```
End Function
```

```
‘addXMLnode() 添加节点
```

```
Function addXMLnode( )
```

```
Set addXMLnode = addXMLnodeex( domXML, parent, name, value, "" )
```

‘TransformXML() 利用 xsl 将 XML 文档转化

```
Function TransformXML ( )
```

```
‘加载 xml 和 xsl 文件
```

```
Set objxml = GetXMLdoc ( xml)
```

```
Set objxsl = GetXMLdoc ( xsl)
```

```
TranFormXML = objXML. Transformnode ( objxsl)
```

5 结语

XML 的需求正在扩大, 新的应用包括采用 XML 标签的 Internet 搜索引擎、必须快速输出结果的电子商务系统、带 XML 标签的电子数据交换、数据重复使用和内容个性化。作为处理上述应用的一环, XML 数据库的需求也将快速增长。这种基于 XML 结构的数据交换体系可以大大的减少 Web 上的数据传输, 对于当前数据传输要求较高的 Web 应用, 有较大的吸引力。

参考文献

- 1 XML 中国论坛, XML 使用进阶教程 [M], 北京 清华大学出版社, 2001。
- 2 宋善德、何栋、梅雪莲, 在 J2EE 体系结构上构建于 XML 的数据交换系统 [J], 小型微型计算机系统, 2003, 24(4): 778 - 780。
- 3 邵敏、李力鸿、郑震坤等, XML 编程实践 [M], 北京 清华大学出版社, 2002。