

文章编号 1000-5013(2003)03-0325-06

XML 的关系数据库技术与应用

骆炎民 张全伙

(华侨大学信息科学与工程学院, 福建 泉州 362011)

摘要 分析 XML 的数据库应用的系统模型, 研究 XML 技术在数据库领域的优势、应用和实现技术. 同时, 以 SQL Server 2000 的相关 XML 技术为基础, 通过网上选课系统实例, 给出解决方案和具体实现.

关键词 XML, Internet, 数据库, 映射, XQL, Schema

中图分类号 TP 311.132.3

文献标识码 A

在信息技术与网络技术高速发展的今天, 网络已经成为新一代操作平台. 信息的传播正全面地以互联网方式展开, WWW 正成为全球信息传递日益重要和最具潜力的资源. 互联网的信息传播, 极大地加速了人类创新进程. 随着 Web 技术的日益发展及 Web 应用的日益广泛, 人类对信息的描述和获取的手段提出了更高的要求. 它要求具有可扩展的语言描述能力, 以及基于语义的信息搜索特征. 此时, HTML 技术固有的极限性开始表现出来. 可扩展标识语言 XML(eXtensible Markup Language), 正成为 Internet 上数据描述和交换的标准^[1]. 它将会代替 HTML 而成为 Web 上描述数据的主要形式. XML 及其一系列相关的标准已经被广泛地接受和使用, 包括 XML 文档的存储、查询、集成、索引等. 它们为 XML 数据库技术奠定了基础, XML 标准的制订将大大促进 Internet 的应用.

1 XML 概述

XML 是为了用来对信息进行描述而设计的一种新语言. XML 是 SGML(标准通用标记语言)的子集, 它并不是类似于 HTML 的预定义标识语言, 而是用于定义其他标识语言的一种元语言. XML 能够使编程者轻松自如地完成许多有趣的事情, 但 XML 并不是编程语言, 也不是基于对象的平台, 更不是操作系统. 它是一种能够思考、交换和表示数据的, 独立于平台的, 强大而精巧的技术. XML 在数据管理方面^[2]具有如下 5 个特点. (1) 结构化的数据存储格式, 语义清晰、规范严格. XML 提供结构化和集成的数据, 对数据的管理是结构化的. 它不仅可以对数据进行结构化, 而且可以指定元素间的联系. 从而, 可以建立有任何复杂层次的数据模型. (2) 易于实现数据共享与重用. XML 中数据描述与表现形式分离, 它只描述数据, 而与

收稿日期 2002-12-20

作者简介 骆炎民(1974-), 男, 讲师, E-mail: lym@hqu.edu.cn

基金项目 华侨大学科研基金资助项目(01HZR07)

数据的表现形式无关. XML 使不同来源的结构化数据很容易结合在一起. (3) XML 是纯文本. XML 以文本的形式来描述数据, 是非专业的, 且很容易读写. 所以, 它很适合在不同的应用程序之间交换数据, 实现数据共享, 并且速度快、效率高. (4) 具有可扩展性. 用户可以根据需要定义新的标志. (5) 具有开放性标准、平台无关性, 独立于供应商和平台. XML 曾经被称为“Web 上的 ASCII 码”, 与具体平台的选择无关. XML 的出现为 Web 的数据管理提供新的模型. XML 形式的 Web 数据不仅是一种新的 Web 数据组织形式, 而且它的面向数据交换的特性推动了 Web 应用模型的发展. 目前, 很多关于数据存储、查询和系统实现与应用模式等方面的研究已经展开. XML 正成为 Internet 上的新一代通用语言.

2 基于 XML 的数据库应用的系统模型

一个成熟的基于 Web 的数据库应用模型, 应该能够在内部数据库系统中进行很好的结合, 以统一的语言描述交换的数据. 它具有良好的柔韧性, 使用开放的标准, 安全可靠, 能和原有的系统以及其他应用程序建立无缝连接. 利用 XML, Internet 以及基于 Internet 的服务和数据库连接技术, 可以建立符合这些要求的 Web 应用系统框架^[1], 其中 XML 是关键. XML 功能强大, 其实现和利用过程比较复杂. 本文以 Web 应用程序中使用比较普遍的 3 层服务结构(图 1), 即数据库服务器、Web 服务器和 Web 用户浏览器为构架基础. 3 层中任一层都有多种实现方法, 无论采用何种方法, XML 都可在其中发挥数据存储、交换和表现的重要作用.

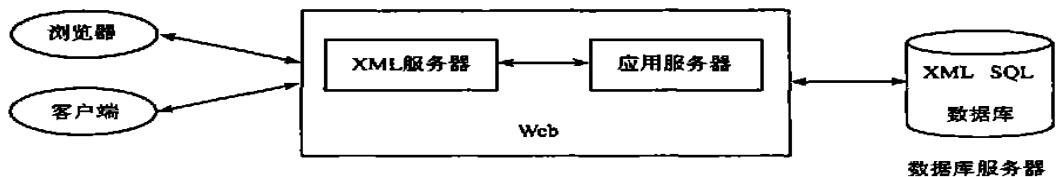


图 1 XML 三层应用系统模型

2.1 数据库服务器

数据库服务器的功能是响应 Web 服务器的数据请求, 执行所需的数据库操作, 并把结果返回给 Web 服务器. 位于数据库服务器层底端是数据库, 它是系统的数据源. 数据库中存储的数据是 XML 格式的, 随着众多数据库提供商增强对 XML 的支持, 未来的主流数据库将可以直接存储、交互和查询 XML 数据. 在数据库服务器端使用 XML 数据, 应注意确保 XML 文件的结构良好性和合法性, 要按照 W3C 制订的 XML 标准和 OGC 的规范严谨地书写和交换 XML 文件. SQL Server 2000 可以采用 XML 架构来描述 XML 数据的逻辑结构, 并且提供了 XML 扩展功能. 它能利用 Transact-SQL 语言进行编程, 开发者可以利用 FOR XML 子句进行 XML 查询. 因此, 我们选择 SQL Server 2000 来支持底端数据库服务器^[1].

2.2 Web 服务器

位于中间的是 Web 服务器, 它包括 XML 服务器和应用服务器. 其主要功能是响应客户端的请求, 进行 XML 数据的识别和转换. 它与用户进行对话, 并根据需要从底层基础数据库中申请数据. 利用 XML 实现的三层 Web 应用结构中, 应用服务器的功能与传统的 Web 应用的应用服务器的功能一样, 在中间层引入了一个 XML 服务器层, 它在体系结构中充当数据连

结器的作用. 因为数据以基于文本的 XML 文档描述, 这就使得客户浏览器端只要能理解 XML 文档中的数据, 就能够从网络中的任何地方读取数据, 便于实现多数据源的共享与集成. 在网上选课系统中, 中间层还负责记录用户临时信息及检查用户提交的数据的合法性、权限等任务.

2.3 客户端

客户端的主要作用是将应用产生的结果显示给用户, 客户端应该能支持广泛的客户平台. 目前, XML 已得到广泛的软件开发商的接受. Microsoft 的 IE 5.0 和 Netscape 的 Navigation 6.0 等浏览器已经可以对 XML 进行解释. 随着 XML 的发展, 更多的网络应用软件支持 XML 将是必然趋势. 在客户端, XML 具有对数据进行个性化和多样化显示的能力, XML 的解释器可以完成 XML 的数据显示. XML 客户端能充分利用浏览器资源, 动态地生成、显示、修改、接收、提交 XML 数据. XML 的内容和表现分离, 客户端展现的数据格式由样式表决定, XSL 是 XML 的样式表语言. 利用上述浏览器内置的 XML 解析器, 可以解析 XML 的结构. 通过浏览器支持的脚本语言(如 JavaScript), 加上 DOM(Document Object Model 即文档对象模型)提供的属性、方法和集合等可以对 XML 文档中的结构及数据进行操作.

3 基于 XML 的 Web 数据库技术

目前大部分应用程序的数据库设计都基于关系数据库, 而 XML 所描述的信息是复杂的半结构化数据. 因此, 需要在 XML 文档与关系数据库之间实现直接的映射关系. 它包括根据 XML 文档设计出存储数据的关系数据库, 以及从关系数据库中映射出 XML 文档两个方面.

3.1 从 XML 到关系数据库的映射

XML 数据模式的描述有两种方法, 它们是 DTD 和 Schema. 我们建议选择 Schema, 因为 Schema 具有 3 方面优点. (1) 支持多种数据类型. 在 DTD 中所有的数据都是字符串, 而数据库中允许多种类型的字段存在, 因此处理起来比较困难. Schema 提供多种不同的数据类型, 因此可以直接对 XML 中的数据类型进行描述, 无需通过对字符串进行类型转换来实现. (2) 支持对数据的限制. 利用 Schema 可以对数据的类型、取值范围等进行限制, 使得基于 XML 的应用具备完整性和健壮性. 与关系数据库的数据完整性约束要求相一致. (3) 支持名字空间. 名字空间是解决多义性和名字冲突问题的方案. DTD 几乎不能与 XML 名字空间配合使用. Schema 与关系数据库的映射有以下 5 个规则. (a) 将该元素映射为关系模型数据库中的表 A, 并且在此表上建立主码约束. (b) 针对每一个子元素, 在数据库中建立对应的表 B_n . (c) 将表 B_n 与表 A 建立主码-外码参照关系. (d) 对每个 PCDATA 项, 在数据库中建立对应的表 C_n . (e) 将表 C_n 与表 A 建立主码-外码参照关系. 例如以简化的 XML-Data 为实现方式在网上选课系统. 为了描述某一选修班级(包括选修课的信息和选修的所有学生), 简化的 Schema 为

```
<!-- 指定 URI 的默认名字空间及数据类型名字空间 -->
```

```
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
```

```
xmlns:dt="urn:schemas-microsoft-com:datatype">
```

```
<!-- 分定义班级元素 class, 学生子元素 student, 课程信息子元素 course. -->
```

```
<Element type name="class" content="mixed" mode="closed">
```

```
< AttributeType name= " classID" dt: type= "id" required= "yes"/>
< attribute type= "classID"/>
< element type= "student"/>
< element type= "course"/>
< /Element Type>
< ElementType name= "student" content= "mixed" mode= "closed">
< AttributeType name= "stuID" dt: type= "id" required= "yes"/>
< attribute type= "stuID"/>
..... ( 学生的其他信息)
< /Element Type>
< ElementType name= "course" content= "mixed" mode= "closed">
< AttributeType name= "courseID" dt: type= "id" r required= "yes"/>
< attribute type= "courseID"/>
..... ( 课程的其他信息)
< /Element Type>
< /Schema>
```

按照上述的转换规则, 针对这个 Schema, 我们可以设计如图 2 所示的数据库. 上述数据库设计中 Farther -ID 是为建立主码-外码参照关系而引入的. 限于篇幅, 本文对一些实现的细节部分就不进行完整的讨论.

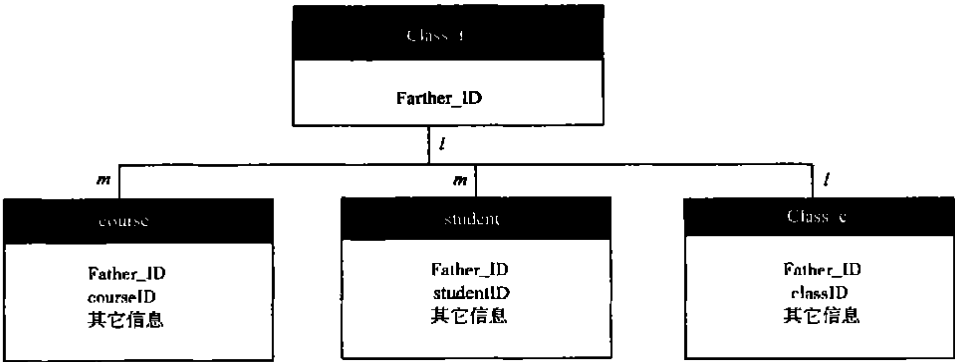


图 2 根据 Shema 设计的选修班级数据库

3. 2 关系数据库到 XML 的映射

基于 XML 的 Web 应用程序中, 遇到的另一个问题是以所需的 XML 格式从数据库中提取数据, 即如何建立从关系数据库到 XML 的映射. 这种映射的实现可以利用软件开发的支撑环境提供 ‘映射方案’ 实现. 本文以 Microsoft SQL Server 2000 为例, 说明 SQL Server 支持使用 ‘映射方案’, 我们可以用它来从数据库中检索适当的 XML 片断. 设 XML 的数据结构为简单的二维表结构, 即与关系数据库的表结构完全匹配, 则可以把数据库中的表映射为 XML 的元素, 数据库中的列映射为 XML 中的属性. SQL Server 2000 采用 sql: relation 批注建立元素和表的映射, 采用 sql: field 批注建立特性和列的映射. 例如针对数据库中的学生表 STU-

DENT (stuid: 学号, stuname: 姓名, stuage: 年龄, studep: 系别,), 将其映射为 XML 结构的方法为

```
< ElementType name= "学生" sql: relation= "STUDENT ">
< AttributeType name= "学号"/>
< AttributeType name= "姓名"/>
< AttributeType name= "年龄"/>
< AttributeType name= "系别"/>

< attribute type = "学号" sql: field= "stuid"/>
< attribute type = "姓名" sql: field= "stuname"/>
< attribute type = "年龄" sql: field= "stuage"/>
< attribute type = "系别" sql: field= "studep"/>

< /ElementType>
```

若 XML 的数据结构比二维表复杂时, 可以采用 sql 的其它批注来建立嵌套的子元素与列的映射, 在这里就不再举例子进行说明. 在从数据库提取出 XML 数据时, 需要注意不同的列可能对应于 Schema 的元素属性, 也可能对应于元素值. 为了区别, 我们可以通过字段名的不同来简单地加以区分.

4 XML 数据库查询

XML 数据与关系数据和面向对象数据不同, 因此传统的查询语言不能直接用于 XML. 但 XML 数据模型与近来数据库界研究的半结构化数据模型很相似, 一些处于研究阶段的查询语言已被设计并运用于半结构化数据. 目前, 针对 XML 文档和页面的查询多采用 XQL (XML Query Language) 实现^[6]. XQL 是 AT&T 实验室提出的一种基于 XML 的半结构数据的查询语言. 它能够通过查询从 XML 文档中抽取信息, 能够在不同的 DTD 之间翻译 XML 数据. 同时, 它也能够从多个 XML 文档中进行数据结合, 对 XML 文档进行查询、构造、转换和集成. XQL 方案的作者按照数据库的思路来解决问题, 采用了用于访问层次型数据库的相同技术. XQL 与 SQL 语言很相似, 有类似 SELECT-FROM-WHERE 的查询结构. 但是, XQL 有一些很重要的区别于 SQL 语言的特点. XQL 的 WHERE 子句由模式和条件表达式两部分组成, 这意味着满足条件的数据项不仅要满足条件表达式, 而且数据项的类型 (或 schema) 和值必须与指定的模式匹配. 引入模式匹配是 XQL 区别于 SQL 的最大特点. 下面列举一些 XQL 的查询例子.

要在 lym.dnx.hqu.edu.cn/xml/stu.xml 页面上查询“电脑系”的所有学生的姓名, 使用的 XQL 代码段为

```
CONSTRUCT < students> {
WHERE
< student>
< dept> 电脑系< /dept>
```

```
< name> $t< /name>  
< /student> IN http://www.lym.dnx.hqu.edu.cn/xml/stu.xml  
CONSTRUCT  
< name> $t< /name>  
}< /students>
```

上述代码假设系别是元素,因此“电脑系”是对元素值的匹配.查询中,每找到一个符合条件的< student> 元素,将其姓名即< name> 属性赋给 \$t,并最终将得到所有值赋给 \$t 的元素内容.因此查询结果是生成由所有电脑系的学生的名字所组成的 XML 文档.

XQL 能够解决利用 XML-QL 所产生的一些问题,但它还存在缺陷. XQL 的设计是以节点操作,而不是依靠信息,不能提供信息区分,当两个元素具有相同的内容时,它就没有自身解决办法进行鉴别.同时它不能处理关系型信息.为了实现更完善的查询,可以使用 W3C 现有的 XSLT 和 Xpath 两种推荐方案.两者的结合将允许开发者随意而方便地对源文件的信息进行操作和重组.

5 结束语

在科学技术迅速发展的今天,我们的迫切任务已经不是建立满足自己要求的专用系统.我们应充分利用互联网的技术,实现计算机及其应用技术多方面功能的全面集成,实现资源共享、数据共享. XML 技术的开放性、平台独立性、语言独立性、数据的结构化自描述性及数据的可读性等优势,无疑将成为下一代网络应用的基石.

参 考 文 献

- 1 Deitel H M, Deitel P J 著. XML 编程技术大全[M]. 康 博译. 北京:清华大学出版社, 2002. 1~32
- 2 陈 兵, 万 晖. 基于 XML 的 Web 数据交换[J]. 计算机工程, 2002, 28(2): 112~114
- 3 张素智, 卢正鼎, 李春林. XML 数据库及其应用研究[J]. 计算机工程与应用, 2002, (8): 32~36
- 4 苟凌怡, 陈晓波, 熊光楞. 基于 SQL Server 2000 的关系数据与 XML 的集成[J]. 计算机工程与应用, 2001, (10): 10~13
- 5 戚克涛, 汪良生, 张申生等. 基于 XML 的 Web 应用研究[J]. 计算机工程, 2002, 28(1): 65~66

Technology of XML-Based Relational Database and Its Application

Luo Yanmin Zhang Quanhua

(College of Info. Sci. & Eng., Huaqiao Univ., 362011, Quanzhou, China)

Abstract A system model applying XML-based database is analysed at first, a study is then made on the superiority of XML technology in the field of database and its application and technology for its implementation. Taking related XML technology of SQL server 2000 as basis and a net curricula-variable system as example, plan for solving and concrete implementation are given at last.

Keywords XML, Internet, database, mapping, XQL, schema