◎数据库与信息处理◎

发布关系数据为 XML 文档

姚全珠,赵朋飞
YAO Quan-zhu,ZHAO Peng-fei

西安理工大学 计算机学院,西安 710048
College of Computer,Xi'an University of Technology,Xi'an 710048,China
E-mail:qzyao@xaut.edu.cn

YAO Quan-zhu,ZHAO Peng-fei.Method of publishing relational data as XML document.Computer Engineering and Applications,2007,43(15):160-162.

Abstract: In this paper,a describing language,R2XL (Relational to XML Language),is defined to express mappings of relational data into XML that conforms to arbitrary DTDs.In the end,a middle-ware is designed based on R2XL to integrate and publish relational data as XML document.
Key words: relational database;XML document;publishing;data integration

摘 要: 提出了一种描述语言 R2XL(Relational to XML Language),它可以根据任意的 DTD 将关系数据映射为 XML 数据。并设计了一个基于 R2XL 的中间件,用于将多个关系型数据库数据集成发布为 XML 文档。
关键词: 关系数据库;XML 文档;发布;数据集成
文章编号: 1002-8331(2007)15-0160-03 **文献标识码:** A **中图分类号:** TP311

1 引言

随着 Web 的不断发展,XML 逐渐成为 Web 上数据表示和交换的标准,它的嵌套、结构自描述等特点为数据交换提供了简单、灵活的手段。然而由于关系数据库系统的高性能、高可靠性以及丰富的软件工具等原因,大多数商业数据仍会继续存储在关系数据库系统中,而且大多数企业也不会固守一个平台,商业数据可能存储在不同的数据库中。为了充分有效地利用企业的宝贵资源——数据,并实现 XML 的潜能,需要将这些孤立的数据集成发布为 XML 文档,以 XML 文档集成发布关系数据是可行的技术。

XML 作为数据交换的标准,对于应用来说,需要从关系数据中产生 XML 文档并传送给合作者。事实上,在电子商务、卫生保健等应用领域,基于 XML 文档的标准化 DTD(Document Type Definition)已经被建立,这些标准化的 DTD 用来描述在本领域内用来进行数据交换的 XML 文档的格式,以便于能被机器有效地识别。因此,需要一种工具将关系数据库中的数据转换为符合 DTD 要求的 XML 文档,并且这种工具应具有足够的一般性来表示复杂的映射,可以将关系数据转换为符合任意 DTD 要求的 XML 数据。

把关系数据发布为 XML 数据,需要两方面的技术:一是需要一种语言来描述从关系数据到 XML 文档的转换;二是需要一种实现技术来有效地实现这种转换[1]。

针对上述问题和功能要求,本文首先介绍了现有的将关系数据转换为 XML 数据的主要方法,然后提出一种映射语言 R2XL(Relational to XML Language)来描述从关系数据到 XML 数据的映射,并设计一个基于 R2XL 的中间件,实现了以 DTD 为指导将多关系数据库数据集成发布为 XML 文档。

2 现有将关系数据发布为 XML 数据的方法

2.1 由数据库直接产生

在 Microsoft SQL Server 中,为了从关系数据库中检索 XML 数据,SQL Server 为 Transact-SQL 语句提供了一个 FOR XML 关键字形式的扩展,通过在 SELECT 语句后添加 FOR XML,可以使 SQL Server 查询处理器将结果返回为一个 XML 流。此外,还必须指定一种模式来指定返回的 XML 的格式,可以将这个模式指定为 RAW、AUTO 或者 EXPLICIT。RAW 模式只能返回以属性为中心的 XML 结果;AUTO 模式可以返回以属性为中心或者以元素为中心的 XML 结果;EXPLICIT 模式根据一个通用表来定义 XML 片断,可以更好地控制返回的 XML 结构。

除 Microsoft SQL Server 外,其它三种最主流的关系型数据库 Oracle、Sybase 和 DB2 也对从关系数据向 XML 数据的转换提供了类似很好的支持。

2.2 由 Web 中间件产生

XML 能描述各种各样的数据,可以在标记语言中嵌入脚本程序动态生成包含多个数据源数据的 XML 文档,当然也可以根据要求生成特定标准的 XML 文档。通过在标记语言中嵌

基金项目: 国家自然科学基金(the National Natural Science Foundation of China under Grant No.52079041)。
作者简介: 姚全珠(1960-),男,教授,主要研究方向为数据库与信息系统集成,形式化软件开发方法;赵朋飞(1981-),男,硕士研究生,主要研究方向为数据库与信息系统集成。

入脚本程序动态生成 XML 文档从本质上讲和动态生成 HTML 文档没有区别，因为两者都是文本格式且都使用标记形式。这种方式可以很方便的插入或者删除数据源，即在脚本程序中增加新数据源的描述部分或者去掉被删除数据源的描述部分即可。

除了上述两种常用的主要方式，还有其它一些系统也支持将关系数据发布为 XML 文档，像 ROLEX[2]、PRATA[3]、SilkRoute[4,5]等，更多的发布工具见参考文献[6]。本文用 R2XL 语言定义映射文件，并设计一个基于 R2XL 的中间件实现了多关系数据库数据到 XML 文档的集成与发布。

3 描述语言 R2XL

3.1 R2XL 语言的形式化定义

本文先对文法 G 进行定义，然后再用此文法对 R2XL 进行定义。

定义 文法 G(grammar)是一个四元组：

$G=(V,T,P,S)$

其中，

V——变量(variable)的非空有穷集，$\forall v\in V$，v 叫做一个语法变量(syntactic variable)，简称为变量。文法 G 中包含两类变量：一类为元素变量，图形化表示为一个椭圆；一类为属性变量，图形化表示为一个矩形。

T——终结符(terminal)的非空有穷集，$\forall t\in T$，t 叫做终结符，为一个字符串，用来表示元素内容或属性值，图形化表示为一个圆。

P——产生式(production)的非空有穷集合。P 中的元素具有形式 $\alpha\rightarrow\beta$，称为产生式，α 称为产生式的左部，β 称为产生式的右部。本文对产生式 $\alpha\rightarrow\beta$ 有以下四种定义：(1)$\alpha\rightarrow elem(\beta)$：父元素 α 有子元素 β；(2)$\alpha\rightarrow attr(\beta)$：元素 α 有属性 β；(3)$\alpha\rightarrow value(\beta)$：元素 α 的值为 β 或者属性 α 的值为 β。在产生式的右部可以跟一个结构符号“？”或者“*”(A？表示 A 可以出现 0 次或者 1 次；A* 表示 A 可以出现 0 次或者多次；没有任何结构符号时表示 A 必须出现且只能出现 1 次)。

S——$S\in V$，文法 G 的开始符号(start symbol)，图形化表示为有一个开始箭头指向的椭圆。

根据所定义的文法 G，R2XL 的形式化定义为：

$G=(V,T,P,S)$

V={elements，element，attribute，value，data，nested，resultset，type}

$T=\{t|t\in T\}$

P={elements→elem(elements)*，

elements→elem(elememt)*，

elements→elem(attribute)*，

elements→elem(value)?，

elements→attr(data)?，

elements→attr(resultset)?，

elements→attr(nested)?，

elements→value(t)?，

element→elem(attribute)*，

element→elem(value)?，

element→value(t)，

attribute→elem(value)?，

attribute→value(t)，

value→attr(type)?，

value→value(t)，

data→value(t)，

nested→value(t)，

resultset→value(t)，

type→value(t)}

S={elements}

图 1 为 R2XL 的图形化表示：

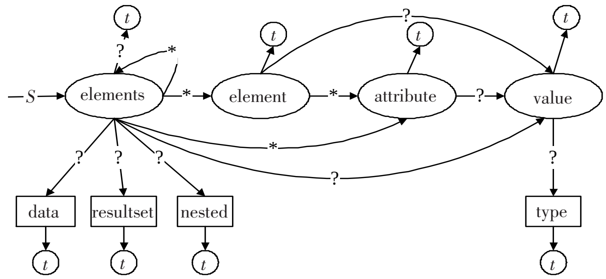

图 1　R2XL 的图形化表示

R2XL 语言为该文法产生的所有句子的集合，一个映射文件即为该语言的一个句子。

3.2 用 R2XL 定义映射文件

发布关系数据为 XML 文档，需要在扁平的关系数据和嵌套的 XML 数据之间建立映射，用 R2XL 语言定义的映射文件为一个 XML 文档，映射文件包含 4 种元素：

(1)elements 元素，用来定义元素，其值定义元素名称。elements 元素之间可以形成嵌套，最外层的 elements 元素定义目标 XML 文档的根元素。elements 元素有 3 个属性：data 属性的值为 sql 语句，定义从关系数据库中抽取的数据；resultset 属性定义返回结果集的别名，结果集别名为一个以$开头的变量；nested 属性用于嵌套查询。

(2)element 元素，和 elements 元素一样，用来定义元素，其值定义元素名称。element 元素总作为 elements 元素的子元素存在，相应 element 元素定义的元素也为 elements 元素所定义元素的子元素。

(3)attribute 元素，总作为 elements 元素和 element 元素的子元素存在，用来描述元素属性，其值用来定义属性名。

(4)value 元素，总作为 elements 元素、element 元素及 attribute 元素的子元素存在，用来定义元素或者属性的值。value 元素的 type 属性用来描述值类型，默认为字符串。

这 4 种元素分别用来描述目标 XML 文档的元素、属性及元素属性值，它们之间的嵌套及次序关系描述了目标 XML 文档的结构。

为了能更好地控制 XML 文档的结构，elements 元素可以嵌套另外一个 elements 元素，形成如下嵌套查询：

```
<elements data="select * from class"
 resultset="$c">class
  <elements data="select * from student"
     resultset="$s" nested="$s.classid = $c.id">student
   <element>name
          <value type="string">$s.name</value>
          <attribute>id
      <value type="string">$s.id</value>
          </attribute>
```

```
        </element>
        <element>address
        </element>
      </elements>
    </elements>
```

生成的 XML 片断如下：

```
<class>
  <student>
    <name id="…">…</name>
    <address></address >
  </student>
  …
</class>
<class>…</class>
…
```

外层的 elements 元素为外层查询返回结果集中的每一行创建一个 class 元素。内层 elements 元素的 nested 属性用来实现嵌套查询，组合 nested 属性和 data 属性形成新的内层查询，然后根据内层查询返回结果为每个相应的 class 元素创建零个或者多个 student 子元素。在映射文件中，因为值为 address 的 element 元素没有 value 子元素，所以最后生成的 address 元素为一个空元素。

4　基于 R2XL 的异构数据集成框架及中间件实现

4.1　异构数据集成框架

将上文定义的 R2XL 语言进行扩展，即在 elements 元素的属性中添加数据源的描述信息：driver(驱动信息)、url、usename 和 password，就可以将多关系数据库中的关系数据映射到一个 XML 文档中。系统管理人员根据 R2XL 语言定义映射文件，这些映射文件是虚的，以源代码的形式存在，映射文件对应的数据并不被物化，仍存放在各个数据库中。映射文件的集合构成了一个虚拟数据库，用户对信息的访问和操作，并不直接作用于各数据源，而是通过 XML-Based 的虚拟数据库(VDB，Virtual DataBase)来访问，用户并不需要了解具体查询数据所在的位置和平台类型，从而实现用户对数据的透明访问。

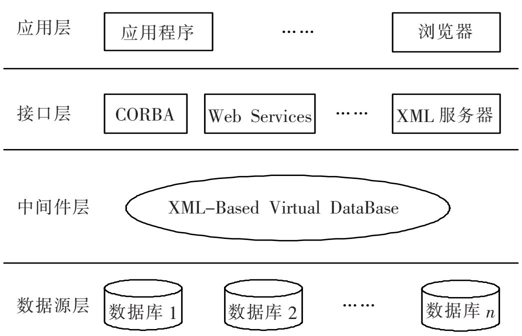

图 2　基于 XML 的数据集成框架

基于 XML 的数据集成框架如图 2 所示，系统自下而上分为数据源层、中间件层、接口层和用户层，各层功能如下：

(1)数据源层：处于最底层，是系统的数据提供者，包括各种类型的数据库、文件、多媒体等信息。本文主要对各种关系数据库数据进行集成。

(2)中间件层：维持各数据源与 XML 数据之间的映射关系，并提供必要的数据转换功能和工具，进行数据到 XML 格式的转换，并将数据存储到 XML 数据空间中。

(3)接口层：依据特定的协议或协作模型，负责不同应用组件请求格式的信息发布。

(4)应用层：即用户界面层次，根据具体的应用和用户计算环境，采用合适的信息访问技术或应用软件。

4.2　中间件实现

在图 2 所示的 4 层结构中，应用层和数据源层从实现角度来看，相对比较简单。而接口层，根据具体的应用需求，可以采用多种手段实现用户对数据的访问。中间件层中的转换器是这个模型中最重要的部件，它根据用户查询和映射文件从数据库中抽取数据，并将抽取出的数据转换为 XML 文档传递给用户使用，其结构如图 3 所示。

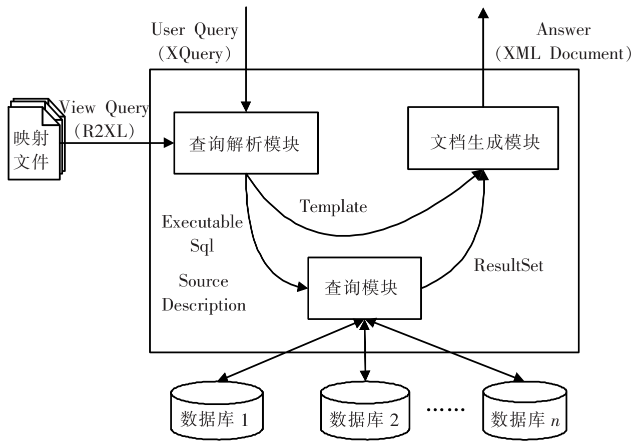

图 3　转换器系统结构

转换器主要组件及功能介绍如下：

(1)查询解析模块，本系统中最复杂的一个模块，该模块接受用户的 XQuery 查询，根据用户查询读入相应的映射文件，将用户查询和映射文件中的查询结合，生成最终的可执行查询，并生成 XML 模板，最后将可执行查询和相应的数据源描述信息传给查询模块，将 XML 模板传给文档生成模块。

(2)查询模块，该模块根据数据源描述信息连接数据库，将可执行子查询分派给各个数据库执行，本文使用多线程，为每个查询建立一个线程以提高查询速度。

(3)文档生成模块，该模块接收查询模块在各个数据源的查询结果集，根据 XML 模板生成符合用户要求的目标 XML 文档，返回给用户系统。

在本系统中，映射文件以源代码的形式存在，对应数据在用户查询之前并不被物化，因此每次查询到的数据都是最新的；因为各数据库系统一般都具有针对其自身存储体系模式的查询优化策略，查询被分配到各个数据库，可以充分利用它们各自的查询能力，从而效率得以提高；同时只有符合用户要求的数据被物化，也减少了数据库与查询转换引擎之间的数据流量；因此本系统具有实时性，高效性和动态性。考虑到在实际应用中，用户的查询请求可能具有一定的相似性，本文利用缓存优化关系数据的 XML 发布[7]。

5　结束语

本文利用 XML 元语言的特点，定义了基于 XML 的映射语言 R2XL 来描述从关系数据到 XML 文档的映射，基于 R2XL

(下转 175 页)