

利用逆规范化实现关系模式向 XML 模式的语义转换

张 梅, 吉根林, 朱颖雯

(南京师范大学 数学与计算机科学学院, 江苏 南京 210097)

[摘要] 目前将关系模式转换为 XML 模式的方法往往不能保留数据库中原有的语义信息, 为此对基于语义的关系模式向 XML 模式转换方法进行研究, 提出了利用逆规范化实现基于语义的关系模式到 XML 模式转换方法. 该方法映射关系模式到关系模式图, 由关系模式图得到符合连接条件的表集合, 根据关系类别在存在关联的关系模式之间建立连接, 用一个结构复杂的关系取代几个结构简单的关系, 按照不同的数据依赖关系将连接表和非连接表转换为不同结构的 XML schema 将所有 XML schema 整合成一个 XML schema. 实验表明该方法对于保留语义约束、加强数据耦合可行有效.

[关键词] XML, 关系模式, 逆规范化, 语义

[中图分类号] TP311 [文献标识码] A [文章编号] 1672-1292(2006)01-0043-04

Semantic-based Relational Schema to XML Schema Translation through Denormalization

ZHANG Mei, JI Genlin, ZHU Yingwen

(School of Mathematics and Computer Science, Nanjing Normal University, Nanjing 210097, China)

Abstract Now most of methods of translating relational schema to XML schema cannot remain intrinsic semantic information of relational database. This paper is based on the research for this problem and presents a semantic based translating algorithm through denormalization. Relational schema are mapped to relational chart. Table sets in accordance with the joining conditions are obtained from the chart. Associated tables are joined according to the classification, and several simple structure relationships are replaced by a complex one. According to the data dependent relationship, an XML schema is created for every joined table and disjointed table. All schemas are integrated into one. Experiments show that the algorithm is valid and efficient for keeping semantic restriction and strengthening coupling.

Key words XML, relational schema, denormalization, semantic

0 引言

目前 XML 已成为 Web 数据交换标准, 而现有许多应用系统则以关系数据库存储数据, 若将关系数据库转换为 XML 文档, 则可以充分利用 XML 的特性, 通过 CSS (Cascading Style Sheets) 或 XSL (Extensible Stylesheet Language) 方便地实现 XML 文档在万维网上的直接发布, 并能直接使用基于 XML 的查询和数据交换. 关系数据库存在一些语义约束, 如依赖关系、外键引用等. 目前国内外学者对关系数据模式向 XML 模式转换做了大量研究工作, 产生很多转换工具和算法. 如: XPERANTO^[1] 模型能实现简单的数据转换, 但不能保留数据中原有的语义; FT^[2] 实现关系模式到 XML 模式的松耦合映射; RTS^[3] 算法结合 XML 树型结构特点, 提取转换模式, 分析转换方法; KRR^[4] 方法基于 Key 关系模式重构, 采用有向图作为抽象数据模式实现整体数据模式向 XML 模式的转换.

本文提出利用逆规范化的方法实现关系模式向 XML 模式的转化, 映射关系模式到关系模式图, 对存在特定约束关系的表进行连接, 根据分类类别为连接表创建 XML Schema, 整合 XML Schema 形成有效的转换. 在存在外键约束、全键关系等其他语义的情况下, 使用逆规范化的方法实现从关系数据库向 XML 的

收稿日期: 2005-06-18

基金项目: 江苏省高校自然科学基金资助项目 (04KJB520075, 03KJD520117).

作者简介: 张 梅 (1976-), 女, 讲师, 主要从事 XML 信息集成和数据库系统等方面的教学与研究. E-mail: zhangme@njnu.edu.cn

转换能体现原有数据库中的语义约束. 本文将介绍利用逆规范化实现关系数据模式向 XML 模式转换的方法, 包括逆规范化思想、转换模型、模型的主要算法以及模式转换的实现.

1 基于语义的关系模式向 XML 模式转换的一般模型

关系模式中显见的语义通常理解为两点: 一是同一表中各字段之间存在着语义关联, 二是表间外键引用具有某种语义关联. 基于语义实现关系模式向 XML 模式转换的一般模型定义了数据库的统一模式. 根据数据库中每张表的相互关系建立关系模式图, 表对应表结点, 字段对应字段结点, 表和字段之间有从表指向字段的有向线段; 为每一个具有 PK 属性的字段 c 建立主键结点 pk_i , 由 c 引出指向 pk_i 的有向线段; 为每一个具有 FK 属性的 c 建立外键结点 fk_i , 由 c 引出指向 fk_i 的线段, 再由 fk_i 指向对应的主键结点. 关系模式转化为关系模式图, 通过深度优先算法实现关系数据库模式向 XML 模式的转换.

上述方法实现了关系数据模式向 XML 模式的整体转换, 保证了关系模式中信息的完整性, 较好地保持了数据库中原有的语义约束, 但转换后 XML 模式表达的仍然是两个表的平面松耦合结构. 为此本文试图通过逆规范化的方式进一步实现关系数据库数据向 XML 模式的转换, 促使数据紧耦合, 更好地实现语义映射, 使得信息更易为用户接受和识别.

2 通过逆规范化实现关系模式向 XML 模式的语义转换

规范化是指将一个关系 R 分解为一组关系变量 R_1, R_2, \dots, R_n 来表示, 这一组关系是由 R 投影生成的, 对于 R 的任何可能值 r , 由 R_1, R_2, \dots, R_n 的相应值 r_1, r_2, \dots, r_n 连接都与 r 相等. 规范化的目的是通过对 R 的投影, 将一个关系等价拆分成若干个关系, 用高级别的范式取代低级别的范式实现规范化, 减少冗余.

逆规范化是指有一组关系 R_1, R_2, \dots, R_n , 用这些关系的连接去取代这些关系. 对于一组关系 R_1, R_2, \dots, R_n 的每一个可能的值 r_1, r_2, \dots, r_n 通过 R_i 的属性值投影 R 的相应值 r 时将产生 $r_i (i = 1, 2, \dots, n)$.

规范化操作通常促使关系在逻辑上分离, 而逆规范化操作加强了关系之间逻辑紧密程度. 数据库表间逆规范化, 能够在关系模式向 XML 模式转换中, 起到保留语义关联的作用, 由此产生的 XML 文档呈现数据紧耦合, 对关系数据库的结构信息、外键约束表达完整.

通过逆规范化实现关系数据库向 XML 数据的转换, 首先根据用户要求找到数据库中相关关系表, 逆规范化的操作在存在关联的关系模式之间建立连接, 按照不同的数据依赖关系将连接表转换成不同结构的 XML 模式. 即按照数据库规范化的逆方向进行, 通过连接运算转化关系表, 用一个结构复杂的关系取代几个结构简单的关系. 具体可以按照数据库关系表的不同情况进行转换.

2.1 具有外键约束的关系

设表 1 表 2 为关系数据库中的两个关系表, 表 1 中 C2 为外键, 表 2 中 C2 为主键. 表 1 中的 C2 引用表 2 中的主键 C2. 两个表之间具有外键约束的语义关联. 当存在这种关联时, 对表 1 表 2 做连接运算生成表 3. 表 2 中的第三条记录在表 1 中没有对应引用, 但在连接时为保证信息完整性, 仍然保持这条记录, 其它字段置空. 根据这种约束关系的特点和存在的语义关联将其转化为以 C2 为元素, C3 为 C2 元素的属性, C1 为 C2 元素的子元素的 XML 模式, 表现语义关系.

表 1 C ₂ 为外键的表		表 2 C ₂ 为主键的表		表 3 表 1 表 2 连接生成的表		
C1	C2	C2	C3	C1	C2	C3
C11	C21	C21	C31	C11	C21	C31
C12	C22	C22	C32	C12	C22	C32
		C23	C33	NULL	C23	C33

2.2 具有相同键的关系

设表 4 表 5 为关系数据库中两个关系表, 具有相同主键 C1, 两表具有语义关联. 对两表进行连接操作生成表 6. 由于具有相同主键, 存在这种关系的表中非主键字段将被转化为主键的子元素.

表 4 C ₁ 为主键的表		表 5 C ₁ 为主键的表		表 6 表 4 表 5 连接生成的表		
C1	C2	C1	C3	C1	C2	C3
C11	C21	C11	C31	C11	C21	C31
C12	C22	C12	C32	C12	C22	C32

2 3 多表连接的转换

多表连接情况较为复杂, 下面以 3 个表的情况为例描述存在键约束的两种情况.

(1) 多个表之间存在外键约束的关系.

设表 7 表 8 表 9 为关系数据库中 3 个关系表, 表 7 中 C₂ 为外键, 对应表 8 中的主键 C₂ 表 8 中 C₃ 为外键, 对应表 9 中的主键 C₃ 这 3 个表连接生成表 10 连接时首先连接表 7 和表 8 再将连接生成的表与表 10 连接, 连接方式与 2 1 类似. 转换为 XML 模式后, 每条记录的发布以 C₂、C₃、C₄ 为属性, C₁ 为子元素.

表 7 C₂ 为外键的表

C ₁	C ₂
C11	C21
C12	C22

表 8 C₂ 为主键 C₃ 为外键的表

C ₂	C ₃
C21	C31
C22	C32
C23	C33

表 9 C₃ 为主键的表

C ₃	C ₄
C31	C41
C32	C42
C33	C43
C34	C44

表 10 表 7 表 8 表 9 连接生成的表

C ₁	C ₂	C ₃	C ₄
C11	C21	C31	C41
C12	C22	C32	C42
NULL	C23	C33	C43
NULL	NULL	C34	C44

(2) 多个表的键为某个表的全键键集中的元素

设表 13 表 14 表 15 是关系数据库中 3 个关系表, C₁ 为表 13 的主键, C₃ 为表 14 的主键, C₁、C₃ 恰为表 15 的全键集中的元素. 这 3 个表连接生成表 16 C₁ 和 C₃ 为表 16 的主键, 转换以主键为记录属性, 其它字段为子元素进行转换.

表 13 C₁ 为主键的表

C ₁	C ₂
C11	C21
C12	C22

表 14 C₃ 为主键的表

C ₃	C ₄
C31	C41
C32	C42

表 15 C₁C₃ 为主键的表

C ₁	C ₃
C11	C31
C12	C32

表 16 表 13 表 14 表 15 连接生成的表

C ₁	C ₂	C ₃	C ₄
C11	C21	C31	C41
C12	C22	C32	C42

上述讨论的关系模式向 XML 模式转换方法 (算法 1) 描述如下:

① 如关系模式向 XML 模式转换的一般方法所述, 由关系数据库生成关系模式图;

② 扫描关系模式图, 找到符合连接条件的表集合, 表间连接生成 RD;

③ switch (RD):

{ case1 (关系表个数为 2 且表 1 的外键是表 2 的主键):

将各字段分为两种类型:

类型 A: 表 2 所有字段; 类型 B: 表 1 除外键外的其它字段;

RS2= "< xsd:element name= \"table\" type= \"tableType\" />

< xsd:complexType name= \"tableType\">

< xsd:element name= \"row\" type= \"rowType\">

< xsd:complexType name= \"rowType\">

< xsd:element name= \"col\" type= \"colType\" minOccurs= \"0\" maxOccurs= \"unbounded\">

< xsd:complexType name= \"colType\">

< xsd:element name= \"类型 B 的字段名\">

< xsd:element name= \"类型 B 的字段名\">

⋮

< /xsd:complexType>

< xsd:attribute name= \"类型 A 的字段名\" />

< xsd:attribute name= \"类型 A 的字段名\" />

⋮

< xsd:element name= \"类型 B 的字段名\" />

< xsd:element name= \"类型 B 的字段名\" />

⋮

< /xsd:complexType>

< xsd:attribute name= \"tablename\" />

< /xsd:complexType>

< /xsd:element> \"; break;

case2 (关系表个数为 2 且表 1 和表 2 主键完全一致):

将各字段分为两种类型: 类型 A: 主键字段; 类型 B: 主键外的其它字段; 生成字符串 RS2 模式同上; break;

case3 (关系表个数为多个, 且多表之间均存在外键约束, 但不产生约束循环):

将各字段分为两种类型: 类型 A: 非类型 B 的字段; 类型 B 不被引用的表中除外键外的其它字段; 生成字符串 RS2 模式同上; break

case4 (关系表个数为多个, 且有一张表的全键恰好为其他各表的主键):

将各字段分为两种类型: 类型 A: 主键字段; 类型 B 非类型 A 的字段; 生成字符串 RS2 模式同上; break

default

```
RS2= "< xsd: element name= \"table\" type= \"tableType\" />
      < xsd: complexType name= \"tableType\">
        < xsd: element name= \"row\" type= \"rowType\" minOccurs= \"0\" maxOccurs= \"unbounded\">
          < xsd: complexType name= \"rowType\">
            < xsd: element name= \"字段名\" />
            < xsd: element name= \"字段名\" />
            :
          < /xsd: complexType>
        < xsd: attribute name= \"tablename\" />
      < /xsd: complexType>
    < /xsd: element> "
```

}

(3) 由关系模式图得到数据库名 str1, 自动生成一个临时表名 str2, 合并已生成的 XML Schema 模式:

```
<? xml version= \"1. 0\" encoding= \"gb2312\"? >
< xsd: schema xmlns: xsd= \"http://www. w3. org/2001/XMLSchema\">
  < xsd: element name= \"str1\">
    < xsd: complexType>
      < xsd: sequence>
        < xsd: element name= \"str2\">
          RS2+
        < /xsd: element>
      < /xsd: sequence>
    < /xsd: complexType>
  < /xsd: element>
< /xsd: schema>
```

3 实验和结论

我们实现了逆规范化实现关系模式向 XML 模式转换的两种方法. 实验表明, 两种方法都能实现数据间的模式转换. 通过对比可以看出经过逆规范化的方法, 关系数据库当中的关系表呈现数据紧耦合, 这对于数据的语义映射非常有利; 同时逆规范化建立在键的基础之上, 保留了一般方法中键约束的语义. 但逆规范化会增加冗余和消耗^[5], 如何消除冗余将是下一步研究的问题.

[参考文献] (References)

[1] CAREY M, KERNAN J, SHANMUGASUNDARAM J, et al. XPERANTO: A middleware for publishing object-relational data as XML documents[C] // In: AMR ELABBADI, MICHAEL L BRODIE, SHARMA CHAKRAVARTHY, et al. Proceedings of the 26th International Conference on Very Large Databases. Egypt: Cairo, 2000: 646- 648.

[2] DONGWON LEE, MURALIMANI, FRANK CHIU, et al. Nesting-based relational-to-XML schema translation[C] // GIANNISALVATORE MECCA, JÉRÔME SMÉON. Proceedings of the Fourth International Workshop on the Web and Databases (WebDB). California: University of California at Santa Barbara, 2001: 61- 66.

[3] 吴文辉, 殷建平, 姚丹霖, 等. 关系模式到 XML 模式的转换研究[J]. 计算机工程与科学, 2004, 26(2): 94- 96.
WU Wenhui, YIN Jianping, YAO Danlin, et al. Research on the transformation from relational schemas to XML schemas[J]. Computer Engineering and Science, 2004, 26(2): 94- 96. (in Chinese)

[4] ZHOU JT. An XML based architecture for semantic integration of heterogeneous relational data sources[C] // LIMING HU, SUN Xianhe. Proceedings of the International Workshop on Grid and Cooperative Computing. Shanghai: Lecture Notes in Computer Science (LNCS), 2003: 465- 474.

[5] JOSEPH FONG, HING KWOK WONG, CHENG Z. Converting relational database into XML documents with DOM[J]. Information and Software Technology, 2003, 45(6): 335- 355.

[责任编辑: 刘 健]