

文章编号:1001-9081(2005)07-1590-02

## 基于 XDR 纲要的空间元数据存储策略

张 涛,于雪芹

(武汉大学 遥感信息工程学院,湖北 武汉 430079)

(ztao\_00@163.com)

**摘要:**提出一种基于 XDR 纲要的空间元数据存储体系结构,建立 XDR 纲要,将以 XML 表达的空间元数据映射到 SQL Server 2000 关系数据库中,可以使用加注释 XDR 纲要的 XML 视图对应到数据库的查询,并以 XML 文件的形式返回查询结果。

**关键词:**XDR 纲要;XML;空间元数据;

**中图分类号:**TP311.12    **文献标识码:**A

## Storage strategy of spatial metadata based on XML data reduced schema

ZHANG Tao, YU Xue-qin

(School of Remote Sensing and Information Engineering, Wuhan University, Wuhan Hubei 430079, China)

**Abstract:** A storing systematic configuration of spatial metadata based on XDR Schema was proposed and a XML data reduced schema was created. The spatial metadata expressed by XML was mapped to SQL Server 2000 RDBMS. The annotated XDR schema corresponded with XML view, so we could query database using annotated XDR schema and get result in XML form.

**Key words:** XML data reduced schema; XML; spatial metadata

### 0 引言

空间元数据是地理空间数据集的描述性信息,它形成了逻辑上元数据子集、实体和元素嵌套的复杂结构。XML 作为一种可扩展标记语言,其自描述性和半结构化的特点使得它非常适用于表达空间元数据的复杂结构。不仅如此,XML 的核心作用主要体现于数据的交换与共享,利用 XML 的样式语言可以方便地实现元数据间的转换与显示等,以满足用户的多种需求。

虽然 XML 有诸多优势,但并没有表现在存储上。XML 数据模式具有海量存储、并发性差、安全控制不灵活等缺点,如何有效地存储 XML 空间元数据,以方便用户查询是急需解决的问题。对 XML 数据的存储,目前有三种方案:1) NXD<sup>[1]</sup>。NXD 是特为存储和查询 XML 文档而构建的,它保留了文档的原始结构和 XML 原有的优点,存储简单,同时有利于对文档的进一步的数据挖掘,但信息格式、内容相对繁琐,索引建立庞大复杂。2) XEDB。XEDB( XML Enabled Database)的基本数据存储单位是 XML 文档中的数据,主要是通过增加一个映射层来管理 XML 数据的存储,它是 XML 与数据库之间转换的桥梁。数据首先要与一个明确的格式相匹配,符合要求的才能根据预先定义好的规则映射到数据库中。XEDB 方法将 XML 文档的数据进行重新组织,存储相对规范,但破坏了原文档的结构,在存入关系数据库中时要进行分解、映射等预处理。3) HXD。HXD( Hybrid XML Database)根据应用的需求,该数据库既可看作是 NXD 也可看作 XEDB,如 Excelon, Ozone 就采用了该思想。

由于 XEDB 可以通过指定的 DTD 或 Schema 规范化 XML

文档并转存在关系数据库中,利用关系数据库管理系统来查询处理 XML 数据是一种可行而有前景的方式,因此本文采用这种思想进行 XML 空间元数据存储策略研究。

### 1 基于 XDR 纲要的空间元数据存储体系结构

XDR 纲要(XML Data Reduced Schema)基于 W3C 所定义的 DTD 或 XML Schema,并对其进行了扩展,它描述了 XML 文档的结构以及对文档中数据的不同约束。Microsoft SQL Server 2000 采用加注释的 XDR 纲要来描述 XML 文档的逻辑结构,它指明 XML 数据是如何映射到关系数据库中的,由此设计了一种基于 XDR 纲要的空间元数据存储体系结构,如图 1 所示。

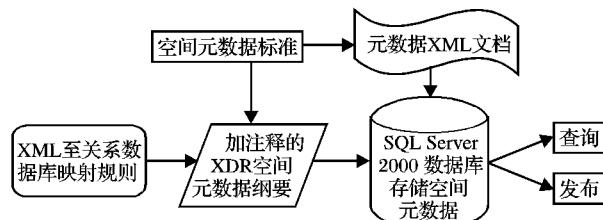


图 1 基于 XDR 纲要的空间元数据存储体系结构

首先,按照一定的空间元数据标准描述并依据 XML 至关系数据库的映射规则构建加注释的 XDR 空间元数据纲要,读取此纲要创建 SQL Server 2000 数据库的表、字段及主键/外键等相关约束,将元数据存储于关系数据库中。这种加注释的 XDR 纲要也称映射架构,它实质上是关系型数据的一种 XML 视图,可以用 XPath 去查询映射架构并且以 XML 文件的形式返回查询结果。

收稿日期:2005-01-05;修订日期:2005-03-16

作者简介:张涛(1978-),男,黑龙江富裕人,硕士研究生,主要研究方向:地理信息系统、元数据; 于雪芹(1977-),女,山东昌邑人,硕士研究生,主要研究方向:网络地理信息系统。

## 2 XML 文档到关系数据库的映射规则

XML 空间元数据文档作为一种层次数据格式,它从一个根元素开始,然后是一些嵌套的元素结构,元素的数据可以是属性,也可以是子元素。文档类型构造不论采用 DTD 描述还是采用 Schema 描述,其映射实质都是树状层次结构向关系数据库二维表的转换<sup>[2,3]</sup>,由此建立以下映射规则:

1) 建立根元素表 A(0 层),将其根元素 ID 设置为主键 PK。

2) 对于根元素子节点(1 层)的每个元素节点类型依次创建表 B1,B2,B3…在每个表中将子节点对应的 ID 设置为主键 PK,同时将与根元素表对应的 ID 设置为外键 FK。对于包含下一层子元素节点 i,将其所包含的下一层子元素映射为其父节点对应表(Bi)的属性列,同时将下一层子元素中还包含子元素的属性 ID 设置为该表的外键 FK。

3) 对于包含子元素的 2 层元素节点 j,创建表 Cj…同样,在每个表中将节点对应的 ID 设置为主键 PK,并将与其父表对应的 ID 设置为外键 FK。同理,对于包含下一层子元素节点 j,将其所包含的下一层子元素映射为其父节点对应表(Cj)的属性列,同时将下一层子元素中还包含子元素的属性 ID 设置为该表的主键 FK。

4) 依次类推直至树状层次元素节点的底层。

图 2 所表示的是空间元数据层次结构的一个实例,它是地理信息共享元数据中标识信息的部分内容。

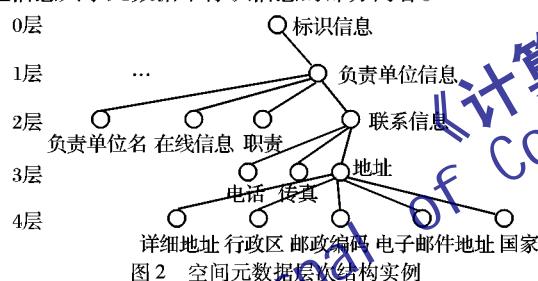


图 2 空间元数据层次结构实例

依据以上的映射规则将图 2 表达的层次结构文档映射成如图 3 的二维表形式。

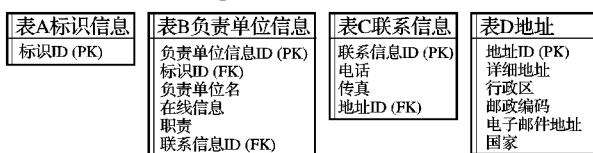


图 3 映射成的二维表结构

## 3 加注释 XDR 纲要的建立

SQL Server 2000 为 XDR 纲要语言提供了一系列的注释<sup>[4]</sup>,它们可以使用于 XDR 纲要以声明 XML 文件与关系数据的对应关系,在默认的情况下,加注释纲要中的一个元素对应到数据库中的一个表,而属性则对应到列。

1) 建立 XDR 纲要正确的命名空间:

```

<?xml version = "1.0" encoding = "GB2312"?>
<Schema xmlns = "urn: schemas-microsoft-com: xml-data"
  xmlns: dt = "urn: schemas-microsoft-com: datatypes"
  xmlns: sql = "urn: schemas-microsoft-com: xml-sql" >
</Schema>
  
```

其中 xmlns = "urn: schemas-microsoft-com: xml-data", 它指定本文档是一个 XML Schema 文档; xmlns:dt = "urn: schemas-microsoft-com: datatypes", 它定义了在本文档中可以使用的数

据类型; xmlns:sql = "urn: schemas-microsoft-com: xml-sql", 它定义本文档中可以使用的注释声明。

2) 依据 XML 至关系数据库的映射规则为每个表创建一个元素,对于表中的每个字段创建一个属性。采用 sql:relation 和 sql:field 来说明这种关系。如果元素/属性名与映射地名相同就不需要用 sql:relation/sql:field 来显式说明这种映射关系了。

```

<ElementType name = "标识信息" sql: relation = "dataIdInfo" >
  < AttributeType name = "标识 ID" />
  < attribute type = "标识 ID" />
</ElementType>
  
```

3) 用 sql: key-fields 设置每个表的主键,并用 <sql: relationship> 来表明表与表之间主键/外键之间的关系。<sql: relationship> 与 Transact-SQL 语言主键/外键约束类似,用来存放表明元素之间关系的各种键值。

```

<ElementType name = "地址" sql: relation = "cntAddress" >
  < AttributeType name = "详细地址" />
  < AttributeType name = "行政区" />
  < AttributeType name = "邮政编码" />
  < AttributeType name = "电子邮件地址" />
  < AttributeType name = "国家" />
  < AttributeType name = "地址 ID" />
  < attribute type = "详细地址" />
  < attribute type = "行政区" />
  < attribute type = "邮政编码" />
  < attribute type = "电子邮件地址" />
  < attribute type = "国家" />
  < attribute type = "地址 ID" />
<element type = "联系信息" >
  < sql: relationship
    key-relation = "cntAddress"
    key = "地址 ID"
    foreign-relation = "rpCntInfo"
    foreign-key = "地址 ID" />
</element>
</ElementType>
  
```

key-relation 声明表的主键关系;key 声明 key-relation 的主键;foreign-relation 声明表的外键关系;foreign-key 指定 foreign-relation 中外键,在描述主键和外键关系时,foreign-key 代表主键。

4) 其他注释的使用。用 dt:type 与 sql:datatype 来控制 XDR 数据类型与 SQL Server 2000 数据类型间的对应关系。用 sql:constant 新建一个不对应到任何表格的 XML 元素,该元素会出现在查询输出中。用 sql:limit-field 和 sql:limit-value 以数值来限制返回的内容。

## 4 实例应用

本文针对地理信息共享领域的元数据标准创建了加注释 XDR 纲要 (GeoMetadataXDR.xml)<sup>[5]</sup>,扫描该纲要创建 SQL Server 2000 的表和字段及相关约束,并存储了全国 1:100 万和 1:400 万基础地理信息共享平台数据库的元数据。应用 Microsoft SQL Server 2000 的 IIS 虚拟目录管理实用工具定义并注册管理空间元数据的 SQL Server 2000 的虚拟目录,注意在设置选项卡上选中“允许 URL 查询”、“允许模板查询”、“允许 XPath”和“允许 POST”选项。加注释 XDR 纲要相当于关系数据定义虚拟的 XML 视图,可以进行 XPath 查询。

(下转第 1597 页)

$$\left( \bigwedge_{b_i \in D} f(x_i, b_i) = \tau_i \right)$$

$\wedge$  表示并算子,  $RED$  为 VPRS 中的  $\beta$  约简。直观地说, 支持数表示在整个论域  $U$  中支持该规则的元素个数, 可信度表示运用该风险规则进行推理正确的概率, 覆盖率表示该规则的支持数在相应的决策类中的比重。在规则过滤中一般选取可信度和覆盖率都相对较高的有效规则。同时结合实际情况和专家经验选取风险规则。

## 4 应用实例

利用文献[4]中建立的专家数据库, 提取100例系统风险评估数据, 离散化处理后如表1所示。

表1 风险评估数据

风险事件	威胁性 $T$	脆弱性 $V$	影响 $F_1$	影响 $F_2$	恢复费用 $E$	风险等级
1	(0, 0.3]	(0, 0.3]	(0, 0.3]	(0.3, 0.6]	(0, 0.3]	低
2	(0.3, 0.6]	(0.3, 0.6]	(0.3, 0.6]	(0.6, 1]	(0.3, 0.6]	中
3	(0.6, 1]	(0.3, 0.6]	(0.6, 1]	(0, 0.3]	(0.3, 0.6]	高
4	(0, 0.3]	(0.3, 0.6]	(0.3, 0.6]	(0, 0.3]	(0, 0.3]	低
5	(0.6, 1]	(0, 0.3]	(0.6, 1]	(0.3, 0.6]	(0, 0.3]	中
6	(0.6, 1]	(0.6, 1]	(0, 0.3]	(0.6, 1]	(0.3, 0.6]	高
7	(0.6, 1]	(0, 0.3]	(0, 0.3]	(0.3, 0.6]	(0.6, 1]	高
8	(0.6, 1]	(0, 0.3]	(0.6, 1]	(0.3, 0.6]	(0, 0.3]	高
9	(0.6, 1]	(0.6, 1]	(0, 0.3]	(0.6, 1]	(0.3, 0.6]	中
10	(0.6, 1]	(0.3, 0.6]	(0.3, 0.6]	(0.6, 1]	(0, 0.3]	中
...	...	...	...	...	...	...

采用 VPRS 模型, 选取误分类  $\beta = 0.3$  对表1数据进行  $\beta$  约简。在此基础之上定义风险过滤规则:  $accuracy(\alpha \rightarrow \beta) \geq 0.75$ ,  $coverage(\alpha \rightarrow \beta) \geq 0.2$ ,  $support(\alpha \rightarrow \beta) \geq 3$ , 对提取出来的风险规则再进一步通过专家筛选, 过滤出 15 条强有效的风险规则, 如表2 所示。

(上接第 1591 页)

XPath 查询可直接在 URL 中指定, 也可在 URL 指定的模板中指定, 参数可传递到直接在 URL 指定的 XPath 查询, 也可传递到使用 XPath 变量的模板中指定的 XPath 查询。例如下面是查询“地址”的模板文件(a.xml):

```
<?xml version="1.0" encoding="GB2312" ?>
<ROOT xmlns:sql="urn:schemas-microsoft-com:xml-sql">
  <sql:xpath-query mapping-schema="GeoMetadataXDR.xml">
    /地址
  </sql:xpath-query>
</ROOT>
```



图4 执行查询模板的结果集

表2 风险规则

规则	威胁性 $T$	脆弱性 $V$	影响 $F_1$	影响 $F_2$	恢复费用 $E$	风险等级
1	—	—	(0, 0.3]	(0, 0.3]	(0, 0.3]	低
2	—	—	(0.3, 0.6]	(0.6, 1]	(0, 0.3]	中
3	(0.6, 1]	(0.3, 0.6]	(0.6, 1]	—	—	高
...	...	...	...	...	...	...

表2 中每一行表示一条风险规则, “—”表示运用此风险规则时无需考虑此属性。例如, 规则 1 表示为: 当风险事件发生后对系统的影响及其恢复费用都较低时, 无论威胁性和脆弱性有多大, 系统所受到的风险等级低。

## 5 结语

通过运用 VPRS 模型对信息系统风险评估数据的挖掘得到了较为有效的风险评估规则, 借助这些风险规则不但可以帮助一般的管理人员做出专家级的风险决策, 而且也大大简化了进一步数据分析的数量。但在实际运用中, 对于风险规则的过滤和提取还需反复做一些调整, 以期得到最符合实际最有效的风险规则。例如 VPRS 中的  $\beta$  值取多少最好, 可信度、覆盖率和支持数怎么选取等。

### 参考文献:

- [1] 孙清. Rough 集及 Rough 推理[M]. 北京: 科学出版社, 2001.
- [2] PAWLAK Z. Rough sets[J]. International journal of information and computer science, 1982, 11(5): 314–316.
- [3] ZIARKO W. Variable precision rough set model[J]. Journal of computer and system sciences, 1993, 46(1): 39–59.
- [4] 肖龙, 戴宗坤. 信息系统风险的多级模糊综合评判模型[J]. 四川大学学报(工程科学版), 2004, 36(3): 98–102.
- [5] QHRN A. Discernibility and rough sets in medicine: tools and application[D]. Norwegian University of science and Technology, 1999.

## 5 结语

基于 XDR 纲要的空间元数据存储策略较完整地保存了 XML 空间元数据文档的信息, 它不仅利用了关系数据库数据存储完整性、存取的高效性, 强大的安全机制、并发访问控制机制等优点, 而且在查询方面既可以使用 SQL 语句进行内容查询, 也可以使用 XPath 进行结构查询, 为促进信息的交换与共享提供了一条新思路。

### 参考文献:

- [1] BOURRET R.. XML Database Products middleware[EB/OL]. <http://www.rpbourret.com/xml/XMLDatabaseProds.htm># middleware, 2004–10.
- [2] FLORESCU D, KOSSMANN D. Storing and Querying XML Data using an RDBMS[J]. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering Special Issue on XML, 1999, 22(3): 29–34.
- [3] 邓芳. XML 文档到数据库数据转换研究[J]. 北京邮电大学学报, 2004, 27(1): 84–88.
- [4] 沈兆阳, 李劲. SQL Server 2000 与 XML 整合应用[M]. 北京: 清华大学出版社, 2001.
- [5] 地理信息共享领域元数据专用标准[EB/OL]. <http://nfgis.nsdi.gov.cn/>, 2005–02.