

GML 空间数据的对象化存储研究

於 荔, 鲍培明, 张书亮

(南京师范大学 数学与计算机科学学院, 江苏 南京 210097)

[摘要] GML (Geography Markup Language) 作为一种存储和交换空间信息的 XML 编码格式, 已逐渐被普遍接受并广泛使用. GML 文档用于描述地理数据内容, 包括对空间对象的空间几何属性的描述, 如点、线串、面等. 空间数据是极其庞大的, 因而导致 GML 数据类型繁多, 数量巨大. 所以一般地将 XML 文档存入关系数据库管理系统中的方法并不适合 GML 文档. 基于该特性提出并实现了一种将 GML 数据存储到面向对象数据库管理系统中的方法, 该方法更加充分地利用了 GML 中面向对象的思想, 比存入关系数据库更有效, 速度更快, 所占空间更少.

[关键词] XML, GML, 关系数据库, 对象数据库

[中图分类号] TP311.131 [文献标识码] A [文章编号] 1672-1292(2006)01-0067-05

Research in Storage Management Of GML Spatial Data

YU Li BAO Peiming ZHANG Shuliang

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

Abstract GML is gradually accepted as an XML encoding form for the exchange and storage of spatial information. GML is used to describe the geographic data including description of spatial object's spatial attributes, such as points, lines, faces and so on. The amount of spatial data is huge, resulting in the variety and greatness of GML. Therefore, the general method of storing XML into relational databases is not fit for GML. This article puts forward and realizes the method of saving the GML in the Object Oriented databases, and this method makes the best of the object facing ideology in GML, which is better than the method of saving in relational databases in that it uses lesser time and lesser space.

Key words XML, GML, ORDB, OODB

GML (Geographic Markup Language) 是由 OGC (Open Geospatial Consortium) 提出的适用于 Internet 环境的空间信息编码方式, 用于地理信息的传输、存储和发布. GML 不仅可以作为一种有效的空间数据传输、交换的手段, 而且也是一种很好的空间数据存储格式. GML 是严格的按照被广泛采用的 XML (eXtended Markup Language) 标准制定的, 如果使用 GML 来存储管理空间数据, 即各 GIS 软件开发商都使用 GML 作为其数据模型和文件格式, 那么 GIS 的空间数据就可以方便有效地集成与共享.

现有最常用的 GML 存储方式还是文本文件方式和传统关系数据库方式, 也有将 GML 文档存入本源数据库中, 但由于本源数据库技术尚未成熟, 因此该方法并不普及.

本文提出将 GML 存入对象数据库中这一思想方法, 并用实验证明了其优点.

1 GML 数据模型

1.1 GML 数据模型

GML 用来描述地理信息系统的空间数据和非空间数据, 空间数据是极其庞大的, 因而导致 GML 数据类型繁多, 数量巨大^[1]. GML 的作用是提供一种机制让用户来定义这些具体的地理要素. 使用 GML 模型

收稿日期: 2005-09-14

基金项目: 国家自然科学基金资助项目 (040401045) 和江苏省教育厅自然科学基金资助项目 (03KJB520117, 04KJB520075).

作者简介: 於 荔 (1981-), 硕士研究生, 主要从事 XML 数据管理等方面的学习和研究. E-mail: sinao_y@sina.com

通信联系人: 鲍培明 (1966-), 女, 副教授, 主要从事 XML 数据管理等方面的教学与研究. E-mail: pmba@mail.njnu.edu.cn

及其模式组件,用户可以在自己的应用模式中定义其问题领域中的地理要素^[2].

GML数据模型由 GML 模式来表达.和以前版本相比, GML 3 版本增加了很多新的模式组件,但最常用的核心模式仍是:要素模式(定义了抽象地理特征模型,如图 1 中 Feature 文件)、几何模式(定义具体的几何形状信息,如图 1 中 Geometry 文件)、拓扑模式(定义了各种功能链接,如图 1 中 XLinks 文件).这 3 个模式 schen e 关系如图 1 所示.

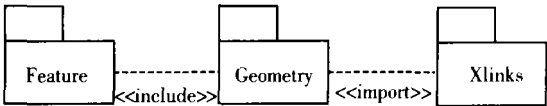


图 1 模式 schema 文件关系图

1.2 GML文档面向对象的特点

```
< SchoolDistrict>
  < gml name> District 28< /gml name>
  < gml boundedBy>
    < gml Box srsName= "http //www. opengis net/gml/srs/epsg xm# 4326">
      < gml coordinates> 0 0< /gml coordinates>
      < gml coordinates> 50 40< /gml coord inates>
    < /gml Box>
  < /gml boundedBy>
  < schoolMember>
    < School>
      < gml name> Alpha< /gml name>
      < address> 100 Cypress Ave < /address>
      < gml location>
        < gml Point srsName= "http //www. opengis net/gml/srs/epsg xm# 4326">
          < gml coordinates> 20 0 5 0< /gml coordinates>
          < /gml Point>
        < /gml location>
      < /School>
    < /schoolMember>
  < /schoolMember>
  < School>
    < gml name> Beta< /gml name>
    < address> 1673 Balsam St < /address>
    < gml location>
      < gml Point srsName= "http //www. opengis net/gml/srs/epsg xm# 4326">
        < gml coordinates> 40 0 5 0< /gml coordinates>
        < /gml Point>
      < /gml location>
    < /School>
  < /schoolMember>
```

GML描述了地理信息,而地理信息无外乎点、线、面、属性等等,其中每个 school 标签就是一个 GML 点. GML 文件调入内存后,通过 GML 解析器的解析成如图 2 所示的树状结构.从图上也可反映出, GML 文档经解析后,其中每个要素 feature 都可以看成是一个对象,如上例子中,每个 school 就是一个 feature 对象,所以 GML 文档中面向对象的特性很强,更适合存储于对象数据库中.

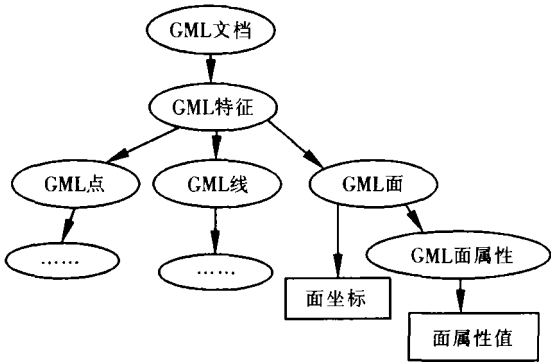


图 2 GML 文档内存结构图

2 GML 常用存储方法

2.1 二进制文件

采用二进制文件格式来存储 GML 空间数据, 这种方法存储容易, 是最简单直接的存储方式, 它与数据被理解的方式一致, 自然地反映了数据之间的嵌套和所属关系. 但是存储粒度较大, 不利于对存储数据的检索和管理, 并且每次查询检索都要解析 XML 文档. 特别是在分布式环境下, 当数据量很大时, 网络通信的代价很高, 数据共享困难. 所以二进制方式的存储限制了 GML 的开放性, 不利于 GML 的应用和发展.

2.2 关系数据库

一种常用的方法是将 GML 存入平面关系数据库, 可以充分利用 RDBMS 关系数据库管理系统进行数据管理. 但由于 GML 的数据模型与传统关系模型有一定的区别, 所以利用关系数据库存储存在一定的不足^[3]. 如关系数据用平面的二维关系表格进行数据的组织, 而 GML 数据表示为嵌套的层次结构, 而嵌套的深度一般是不规则的. 使用结构类型或表间的外键 (foreign keys), 也可以表达嵌套的数据结构, 但是当嵌套的深度未知时, 关系数据库很难查找这种嵌套的数据结构. 例如在 GML 文档或数据库中查找所有总数据属性为 18 的实体, 可以用 XQuery 查询中的路径表达式表示为: `//* [@sum = 18]`, 但是这种查询在 SQL 中很难表达.

另外在关系数据库中, 表中的记录是无序的, 而 GML 文档的元素之间具有内在的顺序. 这就要求 GML 查询语言要能够处理这种顺序, 例如“查找第 4 个实体”, 关系数据库不支持这类顺序查询.

所以用关系数据库存储 GML 并不合适, GML 文档中一些结构信息将被丢失.

2.3 本源数据库

本源数据库是专门为存储 XML 文档设计, 也兼有一般数据库的特性, 例如支持事务、并发控制、查询语言、安全机制、二次开发接口等. 由于本源数据库内部存储采用的是标准的 XML 格式, 不需要进行数据转换, 因而可以充分发挥 XML 的优势. 同样, 将这些数据库创建一些索引, 并将这些索引与 XML 文档一起存到资源库中, 以支持快速搜索资源库来查找包含特定信息的文档. 这样的系统拥有 XML 查询语言、数据存储和各种操作都是专门针对 XML 数据而设计的, 所以在性能上有较好的表现. 但是本源数据库技术尚未成熟, 在数据库系统的安全性、多用户并发、数据的聚合能力等方面还有待提高^[4].

3 GML 与对象数据库

3.1 面向对象数据库技术简介

一个面向对象数据库可以视为一个按类划分并且通过各种关系相互关联的对象的集合^[5]. 它从关系模型中脱离出来, 强调在数据库框架中发展类型、数据抽象、继承和永久性的概念. 它把整个数据库完全建立在新的对象模型之上, 支持对象的所有特性. 所以说对象、类及继承性构成了面向对象数据模型结构特性的基础^[6].

3.2 XL2 对象数据库

XL2 是对象数据库. XL2 数据库是由 java 来实现的一系列的序列化对象, XL2 数据库加强了 Java 中的序列化. 所谓序列化, 其会自动存储必要的信息, 用以反序列化被存储的实例. XL2 数据库其序列化有两大重要改进: 当要获得一个对象时, 并不需要载入整个的数据库, 载入局部的和该对象相关的数据即可; 当要将修改后的某一个对象重新存入 XL2 数据库时, 只需要将修改过的部分再一次保存入 XL2 数据库中即可. XL2 数据库不支持分布式事务, 并且没有其自身的一套查询算法. 因此可结合一些查询工具来实现查询 (例如 JDO).

本文提出了将 GML 存入 XL2 对象数据库的方法, 并用 java 语言实现了这个方法. XL2 对象数据库中提供了 5 个包, java util 包 (提供了 java 的一些底层 API 接口), x2 odb 包 (创建对象数据库), x2 odb collections 包 (实现集合的持久化), x2 transaction 包 (管理, 实现数据库中的事务), x2 util 包 (实现缓冲容器). 存入数据库中的对象必须是经过序列化的对象.

3.3 GML 文档存入 XL2 对象数据库

3.3.1 GML 文档对象化

采用 SAX 事件触发型的解析方式, 在触发到 endElement() 事件时, 进行判断, 是否为所要保存的 GML

中的一个特征对象 (feature), 若是, 则连同该对象的所有子元素 (利用堆栈), 进行序列化, 再调用保存入库函数存入 XL2对象数据库中. GML文档对象化流程如图 3所示, 利用堆栈对象化元素的算法如下所示:

```
public void endElement( String namespaceURI, String localName, String qName) throws
SAXException {
    if (所要保存 feature) {
        父堆栈节点出栈 Parentstacker pop( );
        Child appendChild( txt);
        节点赋予入库对象 featureElement
        对象写出 QName qname= new QName( namespaceURI, lo-
        caName);
        序列化 featureElement ( 3. 3. 2中 BasicObject函数);
        保存对象 ( 3. 3. 3中 savedata函数);
    }
    else if (所要保存 feature的子对象) {
        父节点, 子节点均出栈;
        Child appendChild( txt); }
        子节点链接到父节点;
        父节点压栈;
    }
    else 不保存;
}
```

3. 3. 2 GML文档对象序列化

public class BasicObject implements Serializable{ ... }

该 BasicObject类实现了对对象的序列化, 其实现了 java 中的 Serializable接口, 因为存入 xl2对象数据库中的对象首先必须是序列化的对象, 所以在 3. 3. 1中得到所想要保存的对象后, 要调用 BasicObject类构造函数来将对象进行序列化.

保存对象

```
public void savedata( String dbname, Object O) {
    新建 XL2数据库 db= new XL2Database( );
    以可读可写方式新建事务 db.openReadW rite( dbname). ne-
    wTransaction();
    开始事务 t begin();
    对象入库;
    提交事务;
    关闭数据库;
}
```

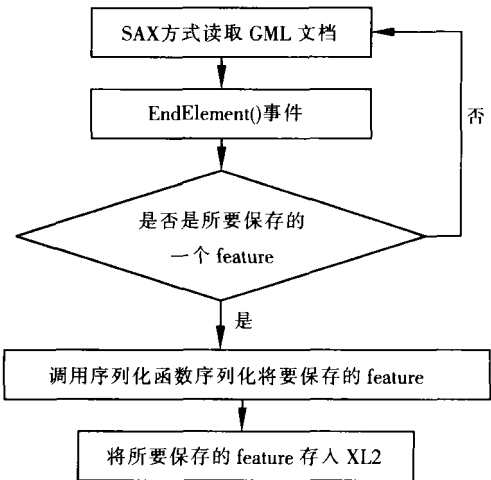


图 3 GML 文档对象化流程图

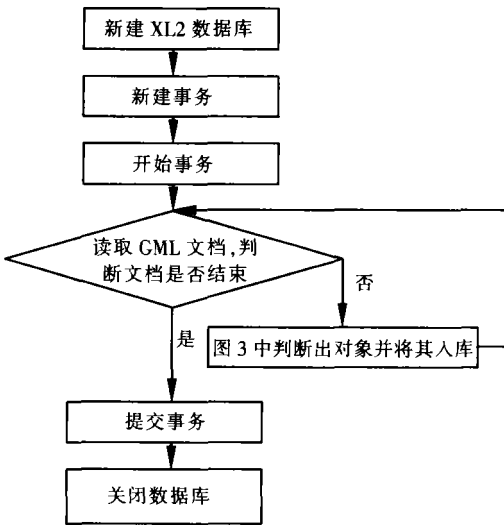


图 4 GML 文档中对象入库流程图

对应 GML 示例文档中, 将 schoolmember 下的每个 school 作为一个 feature 对象, 将其 name, address, location, point, coordinates 作为子对象, 连同 school 父节点一起作为一个对象 O, 调用 BasicObject 函数进行序列化后, 再调用 savedata 函数保存存入 xl2 对象数据库中.

3. 4 实验

3. 4. 1 实验结果

实验分别测试了将 GML 文档存入 xl2 对象数据库和关系数据库中所花费的时间以及结果所占用的存储空间 (分别对应如图 5 左和图 5 右). 在多次对 4k, 500k, 25M 大小的 GML 文档做测试后, 最终得到如下实验结果:

文档大小	所用时间 / s	所占字节空间	文档大小	所用时间 / s	所占字节空间
4k	1. 563	7. 04k	4k	1. 523	10. 584k
500k	2. 031	175k	500k	4. 869	325k
5M	6. 61	3. 50M	5M	8. 637	8. 67M

图 5 试验结果图

3. 4. 2 实验分析

从空间、时间上考虑, 由于地理空间数据类型繁多, 因此所需表格的数量较多, 且空间数据数量巨大, 因此对应表格的元组数目也相对较多, 空间数据之间关联的密切性, 又导致了表格间内外键关联频繁. 由上导致了利用表格存储 GML 文档空间巨大, 当查找某一属性值时, 将表格载入的时间也必将变大. 因此由存储空间又直接影响了读取的速度问题.

从存储数据的关联性上来看, 在存储入 xD 对象库中时候, 将一个 feature(要素) 存为一个对象, 就很容易的得到该 feature 的所有属性. 而在关系表中, 还将通过多表关联来确定哪些是属于该 feature 的属性, 然后再得到相关值和属性. 因此采用存入 xD 对象数据库这样一种方式, 文档特有的图 树结构被线性化 XML 为字节流. 存储整个文档时, 这种方式存储的效率较高.

4 结束语

使用对象数据库存储 GML 数据时, 需要将 GML 模式映射为对象模式, 即先解析 GML 模式, 得到即要保存的对象要素 feature, 然后遍历 GML 文档数据, 所要保存的要素 feature 映射为对象, 其对应的子节点相应对象中的属性.

XD 对象数据库在存储 GML 文档所需花费的时间和空间上比平面关系数据库存在一定的优势, 但由于对象数据库中每个对象是以某一唯一 ID 作为对象标志, 所以在做相关的查询工作时, 在速度上存在一定的限制, 需结合一定的查询工具, 如 JDO 等进行, 这也是在以后的工作中需要认真研究的一个新方向.

[参考文献] (References)

[1] 钟志农, 景宁. 地理标记语言 GML [J]. 微型机与应用, 2003 22(6): 12-15
ZHONG Zhong, JING Ning. Geographic markup language GML [J]. Microcomputer and Its Application, 2003 22(6): 12-15 (in Chinese)

[2] 兰小机, 闫国年. 基于 GML 的空间数据建模研究 [J]. 工程勘察, 2004(6): 54-56
LAN Xiaojie, YAN Guonian. Spatial data modeling based on GML [J]. Geotechnical Investigation and Surveying, 2004(6): 54-56 (in Chinese)

[3] 廖述梅, 陶皖. (对象)关系数据库中 xml 文档的存储技术 [J]. 计算机与现代化, 2003(3): 41-50
LIAO Shumei, TAO Wan. Storage technology of xml in (oriented) relational database [J]. Computer and Modernization, 2003(3): 41-50 (in Chinese)

[4] RICHARD HO, LI BAI, DAV ID ELLMAN. A new data model for XML databases [J]. Int J Intell Sys Acc F in Mgmt, 2002 11(9): 149-157.

[5] 吴洪森. 面向对象数据库的发展与研究 [J]. 计算机工程与应用, 1998(7): 3-5
WU Hongsen. Development and research on object-oriented database [J]. Computer Engineering and Application, 1998(7): 3-5 (in Chinese)

[6] 张大斌, 朱绍文. 面向对象数据库设计的研究 [J]. 计算机工程, 1998 24(11): 10-12
ZHANG Dabin, ZHU Shaowen. Research on design of object-oriented database [J]. Computer Engineering, 1998 24(11): 10-12 (in Chinese)

[责任编辑: 刘 健]