

面向对象空间数据组织方法与应用研究

白晓东,黄为民

(南京师范大学数学与计算机科学学院,南京 210097)

[摘要] 总结了当前空间数据组织的一般方法和各种应用,并基于面向对象的方式,在空间对象类的组织和管理、空间对象高效的索引方式以及分离空间数据与空间数据应用等方面提出了自己新的观点。

[关键词] 面向对象空间数据库,对象关系数据库,网格索引,SVG

[中图分类号] TP311.13; [文献标识码] A; [文章编号] 1001-4616(2001)04-0037-06

0 引言

地理信息系统(GIS)技术的研究与发展从其诞生以来一直是信息科学中最活跃的组成部分之一。它在空间信息的采集、建模、数据组织、可视化表示和分析等领域取得了大量的研究成果。其中,空间信息的数据组织是GIS领域研究的核心问题,这是因为空间数据的特点是类型繁多、异常复杂的,一种空间数据的组织方式决定了它应用的方便性和检索效率。

本文针对当前GIS研究领域中间数据组织的几个问题,从原理和实现两个角度,提出了一种全新的空间信息的组织模型,给出了这种模型快速定位空间信息的索引算法。这种模型适合GIS数据的分布式应用和Web应用,笔者还结合当前最新的W3C的Web矢量化图形标准——SVG,提出了一种WebGIS实现方法。

1 空间数据信息组织的特征分析和现状

矢量化的空间数据是研究地理信息系统的基础,矢量信息模型一般分为两种形式,即基于域的模型和基于对象的模型。笔者认为矢量化空间数据的本质特征是对象的集合,而基于域的表达最终可以转化为对象的方法调用,因此,空间对象,包括点、线、面对象及其它的相互关系是组织空间数据模型的核心。当前流行的GIS系统平台正是基于这样一种思想构建的。著名的GIS桌面平台MapInfo的数据组织反映了当前空间数据组织的一般形式,即:

(1)空间数据分层存储;

(2)空间对象的描述分为属性要素和空间要素两种形式,空间要素描述空间对象的空间位置及其关系,而属性要素描述空间对象的意义及应用要求;

(3)空间要素(几何数据)使用特定的矢量图形格式存储于图形文件中,而属性要素可以存放于数据库中,也可以存放在文件中,二者通过索引文件相互对应。

从空间数据存储介质的角度,随着计算机技术的发展,经历了几个不同的阶段:文件存储、

收稿日期 2001-07-21

作者简介:白晓东,1969—,南京师范大学数学与计算机科学学院助教,主要从事智能化图形图像与地理信息系统等方面的教学与研究。

万方数据

文件——数据库混合存储、完全数据库存储。当前,文件——数据库混合存储空间数据仍然是 GIS 应用的主流。正如前面所说的那样:属性要素存储在关系型数据库中,而几何要素以某种特定的矢量图形格式存储在图形文件中。存储介质的差异主要受数据库技术发展的制约,由于传统的关系型数据库缺乏表达复杂数据结构的能力,无法构建空间对象的数据类型,矢量数据很难存储在关系型数据库中。GIS 系统的开发厂商采用了折中的办法,把空间数据的几何属性放在图形文件中,通过索引文件把空间对象的属性要素与之联系起来。文件——数据库混合存储方式由于采用了异构的数据源,给空间数据的应用带来了诸多不便。面向对象技术和对象——关系数据库技术的发展,空间数据的数据库存储成为可能。一种可行的办法是利用当前数据库系统支持的可变长二进制类型,在应用系统中定义支持空间对象的结构,然后把此结构的数据当作一种可变长二进制数据类型存储在数据库中。例如,可以定义“点”结构如下例所示:

```
struct point{
    float x;
    float y;
}map;
```

如果有一个 map 变量的值是(12.4,1.3),把此数据转化为数值的 ASCII 码形式(用 16 进制表示)31 32 2E 34 2C 31 2E 33,用 0X 作此 16 进制串的开头就可以把它插入到数据库表中(假定表中有一个 point 字段与之对应:INSERT INTO TABLE point_table (ID, point) VALUE ('10006',0X31322E342C312E33)。可以写二进制域存储过程将其转换为应用程序可以理解的数据形式来提高变长二进制域的存取速度。由于某个空间对象的数字化点数不同(每个点对应一个 x, y 值),而不同个数字化点但同种空间对象很难在同一关系表中存储,因此,用可变长的二进制类型表示空间对象的位置信息有一定的优势。

Oracle 与 MapInfo 合作推出的网络 GIS 系统——Spatialware 中,使用了 Oracle 的对象——关系数据库的 DBMS 扩展,可以定义与地理信息系统有关的数据类型,例如,可以在关系表中增加这样的域:

```
Alter table 'temp' adds column location point
```

还可以支持与地理信息系统有关的查询,如下:

```
select r.name
from temp j, temp r
where j.name = 'Joe' and distance(j.location, r.location) < 1
```

本例查询与名字为“Joe”的人居住距离在 1 公里范围内所有人的姓名。

Spatialware 用到了 Oracle 对象——关系数据库的基本操作方法,即支持用户自定义类型和用户自定义函数。但是,新一代数据库标准 SQL-3 还没有被大多数系统所采纳,当前流行的数据库系统,如 SQL Server 7.0 等还不支持这种操作方法。同时,即使 SQL-3 被广泛采纳,作为地理信息系统这样一个特殊的数据库应用领域,通用的数据库系统也不可能做到完全的支持,必须和 GIS 公司合作,正如 Oracle 与 MapInfo 合作推出 Spatialware 一样,而这种合作仍然需要针对特定的系统。对象——关系数据库中所谓复杂空间数据类型都可以分解成关系型数据库中支持的简单数据类型。同时,针对 GIS 空间数据查询的要求,对象——关系数据库采用 R-树等空间索引方式,提高了空间数据检索的速度。

1.1 空间数据模型的基本组织方法

空间数据结构可以用一种复杂数据类型来表达.例如可以视一条公路为一个空间对象,许多涉及地理数据的操作都是以一个空间对象为单位,一个空间对象的空间位置信息可以由许多离散的空间点构成:公路 1 ($(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$).

对象—关系数据库模型中用嵌套关系描述这样一种结构,即可以允许非原子的域存在.这样的模型不符合 1NF 关系.并且,它允许直接表达层次结构,也就是利用结构类型、集合类型、引用类型等表示复杂关系.同时,扩展了的 SQL 支持这样的表达.存在嵌套关系的数据库系统可以看成是一种三维的关系表集合,即对作为关系数据库基础的二维表进行扩充.这样,第三维表示蕴涵复杂类型数据的简单数据集,使整个数据表仍然满足原子性要求. GIS 应用中往往涉及空间位置信息的局部应用,例如“查找与某条街道最近的饭店”,这样的操作是针对某个空间离散点进行的,离散点不是对象—关系数据库中的原子对象,而是对象的某个属性,很难在对象—关系数据库模型中找到对这种查询的支持.因此,笔者采用另一种关系数据库扩展方式—面向对象数据库模型来组织空间数据.面向对象数据库系统以过程化的面向对象语言操作数据,可以大大提高程序的执行速度.同时,它可以满足空间数据的各种应用要求.

1.2 空间对象的存储与访问

空间对象的存储应满足对象的持久化要求.首先,对象的数据部分必须要针对每个对象单独存储.本模型中,用关系型数据库存储空间对象的数据部分,每个空间对象有一个唯一标识它的 Oid,而关系表的每个元组对应一个空间对象的属性数据.关系表设计满足类层次的要求,每个关系表对应一个空间对象类,类与类的继承关系(类层次)由关系的聚集表示.在逻辑上,实现类的方法的代码应该作为数据库模式的一部分存储在数据库中.在本模型中,将代码存储在数据库外的文件里,表现为类的动态连接库(.dll)的形式,这样做是为了避免必须将如编译器这样的系统软件集成到数据库中,省去了许多麻烦.模型系统包含一个 Manger 类,它管理持久化空间对象,定义了三张系统表格:类表、类层次表和类引用表,它们用来由 Manger 类直接管理用户定义的空间对象类:

```
private:
    ClassTable class_table;           // 类表
    RefTable Ref_table;               // 类引用表
    LayerTable Layer_table;           // 类分层表

public:
    Manger();                          // 读各表内容进行初始化
    ~Manger();                         // 存储各表内容
    int create(char * ,char * );       // 建库
    int store(OID);                    // 空间对象存储
    int restore(OID);                  // 空间对象恢复
    int deleteObj(OID);                // 空间对象删除
    int select(OID, int);               // 空间对象选择
    .....
```

Manger 类、用户自定义的空间对象类,包括它们的属性集和方法集的关系和存储介质如图 1 所示:

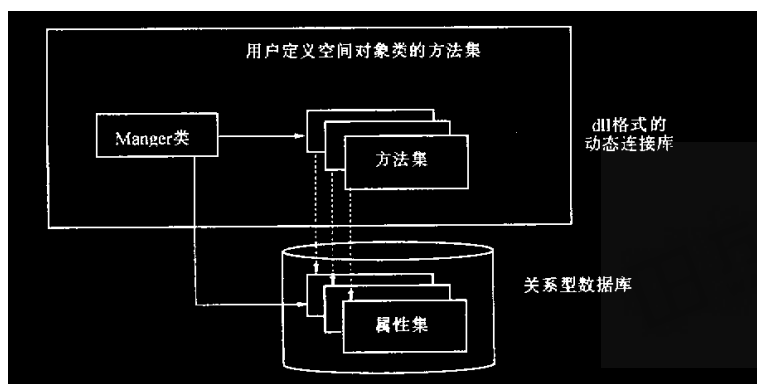


图 1 面向对象空间对象的逻辑组织形式

1.3 高效的空间对象索引方法

本模型系统采用了一种高效的网格索引文件用于检索空间数据。它的基本思想是把应用空间等分成一定大小的网格,网格(称为 Grids)分割空间对象,换言之,每个网格(称为一个 Grid)中会含有一定数量的空间对象的局部信息(空间位置)。空间数据的特点是海量数据,因此,高效的检索对空间数据的有效应用是至关重要的。而网格索引文件可以很好的解决这一问题。顺序检索或者普通的检索方式优化后检索代价仍然是 $O(n)$ 或者是 $O(n/m)$,而用网格索引的检索代价是 $O(n/m^2)$,只需要经过很少的计算就可以进行空间定位。下面用空间数据应用中经常用到的“选择”操作来说明网格索引的运作原理。

用户单击鼠标在图形区域发出一个选择请求,捕获这个坐标值。设图形区域为 $(0,0) \times (6000,6000)$, $n=60$,这时 $x=890.5$, $y=1000.3$,那么,此 (x,y) 应落在 $[890.5/(6000/60)]=8$ 行, $[1000.3/(6000/60)]=10$ 列,即第 680 个 Grid 中,查 Grids 索引文件可知 680Grid 中有 $Oid=1$ 、 $Oid=9$ 、 $Oid=10$ 三个图形对象,它们分别是点对象($Oid=1$)、点对象($Oid=9$)和线对象($Oid=10$),它们对应的空间坐标如下:

$(Oid=1, 890.5, 1001.3)$ 和 $(Oid=9, 820.3, 1020)$ 和

$(Oid=10, 90.3, 200.1) \mid 10, 180.5, 350 \mid 10, 780.6, 1200.8 \mid 10, 2000, 5000)$

假定空间对象的误差允许范围是 2,先计算点对象与选择点的距离:

$$D_1 = \text{sq}((890.5 - 890.5) * (890.5 - 890.5) + (1000.3 - 1001.3) * (1000.3 - 1001.3))$$

$$D_2 = \text{sq}((820.3 - 890.5) * (820.3 - 890.5) + (1000.3 - 1020) * (1000.3 - 1020))$$

显然对象 1 更符合要求。

再计算线对象与选择点的距离。在代入方程前,先查找符合要求的线段,选择点 (x,y) 应满足 $(x_1 < x < x_2)$ 或 $(x_1 > x > x_2)$ 和 $(y_1 < y < y_2)$ 或 $(y_1 > y > y_2)$,对线对象($Oid=10$)的三条线段进行比较运算,发现它们都不符合要求,因此,线对象 10 不是请求的线段。 D_1 的计算结果是 1,小于误差允许范围,所以,点对象 1 是请求点。

在此过程中,请求点的定位一次完成,并且只经过两次计算和三次比较运算就完成了要求的全部操作。网格空间索引以文件的形式存储在系统中,应用时可以以结构数组的方式加载到内存,由于它只包含空间对象的标识信息,因此,它对计算机资源的占用是相对较小的。

1.4 模型的逻辑结构

用图示给出整个系统的逻辑结构,见图 2。

系统中并没有把空间数据分解为点表、线表和面表分别存储在数据库中,这是因为模型采用如上述的可变长二进制域存储空间几何数据,这样可以实现空间对象与数据库表的元组之间的一一对应关系。同时,这样的存储模式可以根据空间数据的逻辑关系编排数据而不受空间数据特性的制约,例如,可以把公路以及与公路相关的数据(例如房屋等)存储在同一数据库表中,这样更有利于数据检索的方便,而传统的点、线、面表的空间数据分割方法很难实现这样的存储。

2 分离数据和“表现”应用模式

涉及图形的应用系统,数据往往是和绘制图形的“画布”绑定在一起的。这种绑定的“画布”可以是控件(control),例如,MapInfo的MapX也可以是某个应用系统平台,例如,Arc/Info的

ArcView。这种绑定关系大大限制了空间数据的应用范围,使得一种格式的空间数据只能在一

种特定的平台上使用,即使以控件的形式存在,支持空间数据与其他系统的集成,也由于控件“画布”的存在,限制了空间数据的应用范围。例如,控件可以内嵌在浏览器中,支持空间数据的Web应用。2000.8.2,W3C推出的新的“可升级矢量图形”标准—SVG,可以在浏览器中快速直接地显示图形,而不需要在网页中内嵌图形控件。

本模型系统采用完全面向对象的设计和思想方法,同时考虑到前台应用的多样性,从系统的组织角度实现数据与数据的“表现形式”的分离。

- 1、据采用完全数据库存储,通过统一的调用接口管理和操作数据;
- 2、程序以动态链接库的形式存在,可以方便地加载到任何一种开发平台中,来实现空间数据的前台应用;
- 3、空间数据的组织以空间对象为基础,一个元组对应一个空间对象,符合空间对象的操作习惯,并且保证空间对象操作的方便性,实现了系统的通用性。

模型系统给出了直接基于浏览器的SVG图形格式的一种应用,由于SVG数据格式是基于XML的,它继承了XML中对对象的支持,因此可以和本模型的空间对象很好的结合起来。本模型系统绝不仅仅支持空间数据浏览器表现这样一种应用,其实,用户可以构建任何应用平台,由于系统的完全面向对象的特点,使系统组织的空间数据可以和任何前台应用相结合。

3 小结

空间数据的组织和管理是地理信息系统领域研究的核心问题之一。信息组织的好坏和空间数据检索的效率是衡量一种空间数据应用系统成败的关键。文章中所研究的空间数据模型、空间对象类的组织形式和基于网格的空间数据索引方式以及模型的前台应用等已经在笔者开

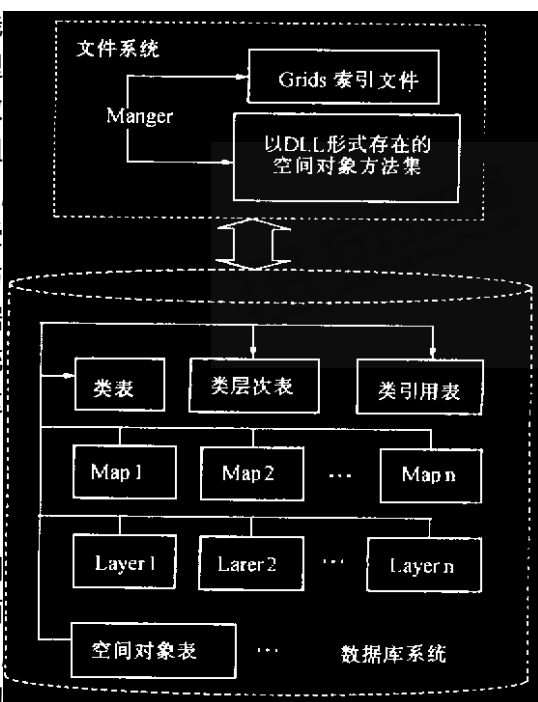


图2 系统的逻辑结构

发的空间信息系统—WebCity 中得到了部分实现,但是,WebCity 还只是一个没有被应用的软件模型,首先,WebCity 还没有自己的空间数据的获取手段,开发中,没有设计图像矢量化模块和与其他系统进行数据转换的模块,其次,空间对象类集合的设计还不理想,需要在实践中改进.

[参考文献]

- [1] 边馥苓.地理信息系统原理与方法 [M].北京:测绘出版社,1996.
- [2] 罗英伟.WebGIS 的构件设计[J].中国图形图像学报,1999,4:79—84.
- [3] 曹晶.空间数据库的聚类方法[J].计算机科学,2000,7.
- [4] [美]Robert W.Stewart Graphics Programming Design[M].北京:清华大学出版社,1995.
- [5] 王家耀.地图学与地理信息系统的现状和趋势[J].测绘通报,1997,06.
- [6] 吴信才.地理信息系统的基本技术与发展动态[J].地球科学,1998,04.

The Organization and Application of the Object-Oriented Spatial Data

Bai Xiaodong ,Huang Weimin

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

Abstract :The article summarizes the methods and all kinds of applications of the spatial data organization. Base of object-oriented way ,the article proposes new views in the following fields : The organization and management of the spatial object , the effective index way in the spatial object and the separation of spatial data and the application to spatial data.

Key words the object-oriented spatial database ;object-relation database ;grid index ;SVG

[责任编辑 陆炳新]