

# 地理信息系统的数据库模型与系统结构

黄波

(中国科学院遥感应用研究所 北京 100101)

**摘要** 数据模型是描述数据内容及其相互关系的工具,它是数据库设计的核心问题之一。空间数据相对于一般事务性数据具有数据量大和更加复杂的特点,因此,地理信息系统(GIS)中空间数据模型的研究显得尤为重要。商业化的GIS一般采用关系数据模型,而关系数据模型缺乏表达空间实体及其相互关系的能力,因此形成了图形库与属性库相分离的GIS系统结构。这种结构不便于空间实体的整体管理。为解决这一问题,人们探索了扩展的关系模型和面向对象的数据模型。本文介绍了这三种GIS数据模型及其与之对应的三种系统结构,分析了这些模型与系统结构的优越性和不足之处,并揭示了数据模型与系统结构之间的内在关系。

**关键词** 地理信息系统,数据模型,系统结构

## 1 引言

目前,最为普遍的地理信息处理方法是混合的方法(hybrid approach),即图形数据由文件管理系统管理,而属性数据则由关系数据库管理系统(relational DBMS)如DBASE、Oracle和Ingres等管理。基于这种混合数据处理方式的GIS系统结构,称为双重结构(dual architecture),如图1所示。其典型代表有ARC/INFO(ESRI),MGE(Intergraph),SI-CAD(Siemens)和ARGIS 4GE(Unisys Corporation)。

以ARC/INFO为例,它由ARC和INFO两部分组成。ARC主要用于图形数据的存贮、分析和处理,而INFO则是一个关系型数据库管理系统,用于属性数据的管理,图形与属性通过用户定义的关键字相联接。类似于ARC/INFO这种系统结构的GIS的最大优点是:属性数据的管理可以充分利用

商业化数据库管理系统的功能且图形的处理与检索效率较高。但是,GIS中的空间实体本身不仅具有图形部分,也具有属性部分。图形与属性的硬性分离势必带来许多问题:

(1) 不利于空间查询。因为查询常常涉及到空间实体的图形部分和属性部分。依赖于这样的结构,只能对图形库和属性库分别进行查询,然后将两者的结果组合在一起。这样势必降低了查询效率。

(2) 数据的整体一致性(Integrity)易遭到破坏。例如,一实体在图形库中的部分

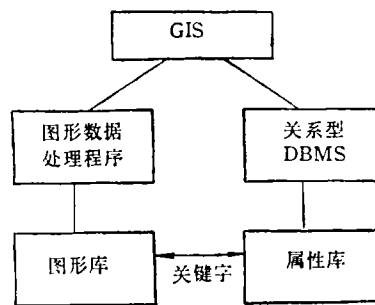


图1 GIS的双重结构  
Fig. 1 Dual architecture of GIS

收稿日期: 1994年6月27日;收到修改稿日期: 1994年9月5日

可能已被删除,而对应的属性库部分却可能没有被删除。

(3) 不利于分布式数据管理。因为图形数据独立于属性数据之外,自成一体,无法利用标准的数据库管理系统提供的分布式数据管理功能,而且在分布式数据库中,图形与属性的协调更加困难。

之所以存在以上问题,主要是因为具有双重结构的 GIS 所基于的关系模型缺乏表达图形实体及其相互关系的能力。从这里我们可以发现 GIS 的系统结构与 GIS 的数据模型之间存在着一定的关系。下面,以 GIS 的数据模型为主线,具体分析 GIS 的每一种数据模型及其与之对应的 GIS 系统结构。

## 2 关系模型

与层次和网络模型相比,关系模型是目前商业化的 GIS 最常用的数据模型。关系可以定义为一组域 (field) 的笛卡尔积的子集。给定一组域  $D_1, D_2, \dots, D_n$ , 其笛卡尔积为:

$$D_1 \times D_2 \times D_3 \cdots \times D_n = \{(d_1, d_2, \dots, d_n) | d_i \in D_i, i = 1, 2, \dots, n\}$$

其中每一个元素  $(d_1, d_2, d_3, \dots, d_n)$  叫做一个  $n$  元组。关系  $R(D_1, D_2, \dots, D_n)$  是元组的集合,且

$$R(D_1, D_2, \dots, D_n) \subseteq D_1 \times D_2 \times \cdots \times D_n$$

关系的具体实现是一个二维表结构,表结构中的行对应于元组,列对应于域,列的名字称为属性。 $n$  元关系必有  $n$  个属性。满足一定条件(如第一范式 1NF)的规范化关系的集合,就构成了关系模型。

关系模型的优点是结构灵活,而且具有严密的数学基础(关系代数),可以满足所有用关系运算和数学运算规则形成的询问要求。另外,关系数据库还能搜索、组合和比较不同类型的数据。但是,用关系模型描述空间数据,还存在许多不足之处<sup>[1,2,3]</sup>:

(1) 关系数据库管理系统不能定义几何数据类型如点、线、面和空间操作算子如相交、包含、距离、面积等,因而无法完成地理意义上的空间分析和查询。

(2) 图形数据的表示涉及参考坐标系、坐标、属性、拓扑关系等,因而用一系列关系表格来表示就显得非常复杂。

(3) 表格形式缺乏直接的语义表达能力。关系模型仅提供两种结构表达实体特征以及实体与实体之间的关系,一种是表内的,一种是通过公共值跨表联接。这种关系的联接是隐含的,并且这种表格间的联接十分费时,效率低。

(4) 缺乏构造复杂对象的能力。造成这一缺陷的主要原因是关系模型要求关系至少要满足第一范式的要求,而且属性的取值应是原子的,即关系表中不能包含另一个关系。这样就不能用对象嵌套的办法去构造复杂对象,而只能借助于关系的规范化分解来实现。过多的不自然分解必然导致模拟和操纵的困难和复杂化。

关系模型在空间数据处理中的不足之处,促使人们去探索新的模型。这里有两种方法:一是扩展现有的关系模型,使其能容纳图形数据,并进行空间运算和查询,即扩展的关系模型;二是用新的数据模型如面向对象的数据模型取代关系模型。以下分别介绍扩

展的关系模型和面向对象的数据模型。

### 3 扩展的关系模型

通过以上对关系模型的分析,我们知道,关系模型不适合于图形数据处理,主要原因是缺乏定义几何数据类型和空间操作算子的能力。因此,为适应 GIS 的要求,在关系模型、的基础上,需要增加几何数据类型和空间操作算子。其中几何数据类型包括点、线、面、矩形等;空间操作算子包括距离、面积、相交、包含等。这里以作者研制的软件系统 Topo DBMS<sup>1)</sup> 为例作一说明。

Topo DBMS 建立在商业化关系数据库管理系统 Postgres 基础之上,可以进行拓扑空间查询。与一般的数据库管理系统不同,Postgres 是一个开放式的系统。这意味着新的数据类型、函数、算子和索引结构可以添加到系统中。除了具有通常的数据类型如逻辑型、数字型、字符型、日期型等,Postgres 还提供四种基本的几何数据类型,Point, Lseg, Path 和 Box。其中 Point 代表点类型, Lseg 代表弧段类型, Path 代表线和多边形类型,而 Box 则代表矩形类型。使用这些数据类型,加上 C 语言函数库和 Postgres 本身提供的算子如求和、平均值、最大、最小值等,作者定义了判断六种拓扑关系算子,即:

面/面关系如相离 DisjointAA(a1, a2)、相触 TouchAA(a1, a2)、包含 ContainAA(a1, a2) 等;

面/线关系如穿过 Bisect(l, a)、渗入 Permeate(l, a)、包含 InLA(l, a)、相触 TouchLA(l, a) 等;

面/点关系如包含 ContainAP(a, p)、相离 Disjoint(a, p) 等;

线/线关系如相交 Intersect(t1, t2)、相触 Touchll(t1, t2) 等;

线/点关系如相离 DisjointLP(t, p) 等;

点/点关系如相离 DisjointPP(p1, p2) 等。

其中 A 代表 Area (面), L 代表 Line (线), P 代表 Point (点)。

利用以上介绍的几何数据类型和空间操作算子,可以完成基本的拓扑查询。例如,查找“权属主是 Smith, 面积大于 2000 平方米且位于武昌区的地块”可表示如下:

```
Select *( : Path)
From Parcel
Where owner = 'Smith'
and area > 2000
and ContainAA (*, '武昌区')
```

由于已定义多边形数据类型和算子 ContainAA, 所以很容易实现这个查询。

如图 2 所示,使用扩展的关系模型,其根本目的是用同一个数据库管理系统同时管理图形和属性数据,从而建立一整体的 GIS 系统结构 (Intergrated Architecture)。具有这种结构的 GIS 有: System9 (Computer Version), Geoview (University of Edinburgh)、

1) 黄波. GIS 与专家系统的结合及其在温州市控制性规划管理中的应用: 数据模型部分. 见: ITC 硕士论文, 1993。

TIGRIS (Intergraph) 和 Geo++ (TNO Institute)等。

用同一个数据库管理系统管理图形和属性,除了可以实现图形实体整体意义上的查询之外,还具有以下优点<sup>[1,2,3,4]</sup>。

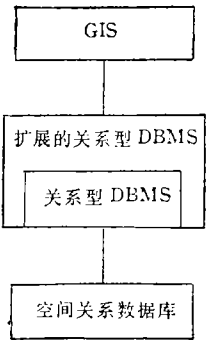


图 2 基于扩展的关系模型的 GIS 整体结构

Fig. 2 Integrated GIS architecture based on extended relational data model

(1) 可以对数据进行集中控制,包括数据的完整性、安全性和一致性控制等。

(2) 可以充分利用数据库管理系统提供的功能。

(3) 数据库的开发应用程序独立于数据本身。

(4) 便于分布式数据管理。

(5) 提高了系统的开放性,便于用户使用和二次开发。

尽管存在以上优点,但是是否用统一的数据库管理系统管理图形和属性还有争议,而且扩展的 SQL 标准以及查询的优化问题还有待解决。

同时,我们也应看到,扩展的关系模型虽然克服了关系模型的部分局限性,但仍然摆脱不了关系模式的束缚,如属性取值为不可再分的简单数据,不能处理复杂对象等。因此,GIS 学者们仍然在探索着新的数据模型。面向对象模型的出现则为这种探索开辟了新的道路。

模型的出现则为这种探索开辟了新的道路。

## 4 面向对象的数据模型

‘面向对象’是当今 GIS 文献所广泛引用的术语。它的基本思想是将现实世界看成一组彼此相关,并能互相通信的对象,用更直观和自然的方法建立这些对象的模型,从而尽可能达到直接求解问题的目的。换句话说,这种思想提高了数据的抽象化程度,方便了用户的使用。

由 E. F. Codd 于 1970 年提出的关系模型定义明确,而迄今为止,面向对象的概念还没有确切的定义。不过,专家们普遍认为它有以下几个重要元素即:对象 (object)、类 (class)、方法 (method) 和消息 (message)。

对象与常规程序中的变量相似,而类则类似于数据类型,类的实例 (Instance) 即是对象。方法是施加于对象上的操作,对象通过发送信息来激活另一对象以完成某些操作。通常,一个图形实体的图形部分、属性部分及其操作方法、限制规则可以封装在一起成为一个对象。

面向对象的数据模型是以面向对象概念为基础,支持复杂对象表示和操作的高层次抽象模型。它涉及四个抽象概念:分类 (classification)、概括 (generalization)、聚集 (aggregation) 和联合 (association),以及继承 (inheritance) 和传播 (propagation) 两个语义工具 (详细介绍见[4])。面向对象模型的核心是抽象对象及其操作。它涉及的对象的形式化描述为:

$$\text{Object} = \langle \text{ID}, \text{S}, \text{M}, \text{IO} \rangle$$

其中:

ID 为对象的标识;

S 为对象的内部状态,它可以是一组属性值,也可以为另外一组对象的逆归;

M 为对象所能承受的操作的集合,即方法集:  $M = \{\text{method-1}, \text{method-2}, \dots, \text{method-n}\}$ ;

IO 是对象的对外接口,即对象可接受的消息集。它向外界提供了对象的可用方法。

$IO = \{\text{message-1}, \text{message-2}, \dots, \text{message-n}\}$ ,

一个对象的 IO 和 M 之间应有一定的对应和联系。

对若干对象进行分类抽象,便得到一组包含所有同类型对象的类。其形式化描述为:

$$\text{Class} = \langle \text{CID}, \text{INH}, \text{CS}, \text{CM}, \text{CIO} \rangle$$

其中:

CID 为类的标识;

INH 为继承描述,给出被继承的类的标识;

CS 是除继承之外的结果描述;

CM 是除继承内容之外的方法定义;

CIO 是除继承内容之外的接口描述。

显然有:

$S \in \text{CS}$  和  $M \in \text{CM}$ , 当 object 是该类的实例时。

对类进行概括抽象,从具有共性的类中提取出更加综合的特性,便构成了更高层次的超类 (superclass), 而前者则叫子类。超类的实例为较高级的对象 (或复杂对象); 子类的实例即为子对象。子类可以继承超类的属性结构和操作方法。

对类进行聚集抽象,即把多个具有不同性质的类组合成一个高层次的复合类。复合类的实例为复合对象。复合对象的组成部分为组件对象 (component object)。复合对象的属性值来源于两方面: 一部分属性值由该复杂对象本身所属的类定义, 与构成它的组件对象无关; 另一部分属性值则通过传播工具从组件对象中派生而来。

可见,面向对象的数据模型就是通过聚集、概括等方法对空间实体和联系进行模型化。

根据面向对象的数据模型的思想,将图形与属性封装在一起,并由一个面向对象的数据管理系统进行管理,就形成了如图 3 所示的 GIS 系统结构。其典型代表有 Smallworld GIS (Smallworld Systems Ltd.)、ARC/INFO7.0 (ESRI) 和正在开发的 GEOSTAR (武测)等。具有这种结构的 GIS 除了具有图 2 所示结构的优点外,还体现了面向对象数据模型的优点。这里以 Smallworld 为例,说明如下:

- (1) 所有的地物以对象形式被封装,而不是以复杂的关系形式存贮。
- (2) 以对象为基础,消除了分层 (layer, coverage) 的概念。这样当空间查询涉及两

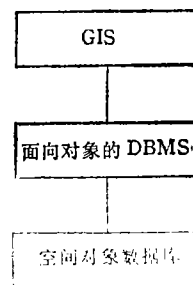


图 3 基于面向对象模型的 GIS 系统结构

Fig. 3 GIS architecture based on object-oriented data model

类对象时,毋须进行叠加(overlay)的操作。原先的 ARC/INFO (7.0 之前版本)将“学校”(点状要素)和行政区划(面状要素)分成两层,若要找出位于某一行政区内的学校,需将这两层叠加起来,而 Smallworld 则不需要这样的操作而能直接得出结果。

(3) 可以处理复杂的地物类型。例如 Smallworld 可以将某湖中所有小湖定义为一个复杂数据类型 Region,再利用面向对象数据抽象工具一传播,则很容易完成对整个湖的查询和统计工作。

(4) 根据面向对象 Late-binding (后编译)的思想,用户可以在现有抽象数据类型和空间操作箱的基础上,定义自己需要的数据类型和空间操作方法,因而增强了系统的开放性和可扩充性。

(5) 基于 icon (肖像)的面向对象的用户界面,便于用户的操作和使用。

当然,类似于 Smallworld 的 GIS 只是使用了面向对象的部分思想,还没有体现它的全部思想。

## 5 结 论

通过以上的分析,作者得到了如下的一些体会:(1)尽管 GIS 数据模型与 GIS 的系统结构不存在直接的关系,但空间数据模型在一定程度上决定了 GIS 的系统结构。良好的 GIS 系统结构,可以方便 GIS 用户对数据的有效管理,但如果没有合适的数据模型做支撑,则无法建立高效的 GIS 系统结构。(2)尽管标准的关系模型存在许多不足之处,但是目前大部分实用数据库和 GIS 仍采用这种模型,而且在关系模型基础上进行扩展,也是一条实用有效的途径。(3)具有整体系统结构的面向对象 GIS 代表着 GIS 的一种发展方向。但由于目前尚还未有纯粹的面向对象数据库管理系统和面向对象 GIS 存在,所以面向对象数据模型的优越性还有待进一步检验。

最后,需要说明的是,在国外,面向对象的 GIS 已远非象当初面向对象刚出现时那样声张旗鼓。因此,我们需要以谨慎的态度对待面向对象的 GIS 的开发和研究。

### 参 考 文 献

- [1] Peter Batty. Why use a single Database Management System for GIS?, International GIS Sourcebook, 1991.
- [2] T. Vijlbrief etc., The GEO++ System: An Extensible GIS. EGIS 92, 1992.
- [3] P. van Oosterom etc., A Geographic Extension to the Relational Data Model. Geo'89 Symposium, The Hague, 1989.
- [4] C. Webster. The Object-Oriented Paradigm in GIS. International Archive of Photogrammetry and Remote Sensing, 1990b, Vol. 28.

## GIS Data Model and Its Architecture

Huang Bo

*(Institute of Remote Sensing Applications, CAS)*

**Abstract** Data model is a representation of data and its relation. It is considered very important in database design. In contrast to alphanumerical data, spatial data is characterised by its large magnitude and complexity. Therefore, it is of significance to do research on spatial data model. Most commercial GISes are based on relational data model. However, its drawbacks on representing spatial entities and their interrelations have led to the separation of graphic database and attribute database. In order to overcome this problem, extended relational data model and object-oriented data model are explored. This paper introduces these three data models and their corresponding GIS architecture, analyses their pros and cons, and presents the relation between spatial data model and GIS architecture.

**Key words** GIS, Data model, Architecture