

面向目标空间数据库——实例研究

P. 米尔恩等

1. 引言 地理信息系统 (GIS) 所关心的是可以想象得到的最广泛的信息。其研究对象是现实世界, 而不是一些商业上 (例如商业数据库) 的抽象模型 (虽然诸如此类的数据库可以成为GIS的组成要素)。GIS的应用范围非常广泛, 包括保护、开发和其它主要工业、国防、紧急服务、环境管理、设施管理、土地管理和规划。

涉及土地信息管理的组织机构与从硬拷贝而来的几何数据库紧密相关, 同时, 先进的数据库更新与转换技术正在使巨型地理数据库成为现实。日常生活中, 许多GIS的应用都涉及几个横跨不同组织机构的数据库。应用的广泛性和目标的复杂性超出了数据库技术的能力, 使得其必须进一步深化。因此, GIS正在推动数据库技术的发展。

为GIS用户服务的工业团体正在迅速地演化, 目前, 已经突破了原有的专门化的应用领域, 例如地理数据处理、环境中开发的数据的采集技术。由于涉及图形数据类型的操作需要, 这些工业团体已经抛弃了传统的数据库的发展方向。目前, 面向目标的数据库技术正在适应工程需要 (例如GIS)。可以预料, GIS将在适合其需要的一般目的数据库平台上得到发展。

本文讨论了基于具有商用可能的、面向目标的数据库管理系统的地理数据库 (OODBMS) 发展中的一些问题。澳大利亚空间信息中心 (CSIRO) 信息技术室开发了这种数据库。这种数据库要求能够支持地理目标和它们属性的交互输入、检索和显示。图 1 (略) 给出了一个例子。

用作设计的概念数据模型是以空间数据转换标准草案 (SDTS) 中的数据模型为基础的。§ 2 中作出了描述。§ 3 中叙述GIS设计。§ 4 中讨论涉及 OODBMS 的一些问题。§ 5 中阐述 OODBMS 和数据库设计。§ 6 中讨论一些中间运行结果。

2. 空间数据转换标准 空间数据转换标准的建立 (美国地质调查局1991) 使现今GIS协会感兴趣的数据概念模型达成了一致意见。用实体和类型的术语来描述这种模型, 它的反面是: 目标和类型, (参§ 4), 实体可以考虑为一个限定类型的空间现象。可是, 如图 2 中格网概念模型所描述的数据聚集体并不是对应于一种具体的真实世界中的实体。图 2 显示了这个模型的典型部分, 这个模型涉及到SDTS称之为的空间目标, 用位置、形状、大小和拓扑关系来描述。其它属性 (包括实体) 通过关系模型结合在一起。

空间实体的概念模型必须用图形形式表示成地理系统的内部模型 (例如数据模型)。如果面向目标的数据库被用于这个系统, 那么空间实体即用图形形式表示成复杂目标。例如。一个线目标的数据组成可以是它的起始结点、终止结点、确定此线的中间点的集合、以及一些可打印属性。

SDTS概念模型的非空间成分是一个关系模型, 这个模型专门负责建立空间目标和实体结构和过程。

- (d) 为系统提供答复有关时间变化探测一类查询的能力; 以及
- (e) 提供允许系统的归纳学习过程独立进行操作的过程和控制结构。

宛铭译自《IJGIS》1987, 第1卷, 第2期

有关组成之间的相互关系。一般有两种类型的关系：原始的和派生的。原始关系是指本质关系（即，它们的存在独立于其它关系），其元组通过指定键项（即系统产生的识别器）与空间目标相联系。派生关系对应于特征和有联系的（association）关系。特征关系是指由原始关系中的元组所表示的实体的一组多值属性。相联系的关系表示实体关系（参见§3）和它们的属性。

原始关系中的元组可能仅仅与一个空间目标相对应。因此，这种中心元组图示成相联系的空间目标的一个唯一的概念实体。空间目标可以对应于多个原始元组，以使多个概念实体有相同的空间描述（如一个山体和一个三角测站具有相同的地理点位）。SDTS不提供涉及键、规范化、属性和域的定義的关系模型的语义成分。

SDTS中主要的数据模型组成间的相互关系如图3所示。空间目标被划分成两类——矢量和栅格。数据质量是定义数据源、精度、一致性和完善性的一个静态属性集。

SDTS模型没有提到GIS空间数据产生三维造型的一些要求，另外，SDTS没有提到非常规数据类型，如声音。此模型的非空间部分存在的其它一些严重的不恰当性，在此文中没有讲述。此外，SDTS模型可以为空间DBMS模型和实用性服务，并建立了一个基准点。这个空间目标模型是§3所给的系统设计的一个组成部分。

3. 一个GIS实例的概念设计

设计分两个层次描述。在下一节中，我们将简明地描述系统的构造风格，在这种系统构造风格中，数据库模块必须可操作，本节将给出数据库的设计思想。

3.1 系统结构 构成系统的主要成分如图4所示：1图形用户界面（GUI）；2地图模块；(3)图形显示模块；4地理数据库模块

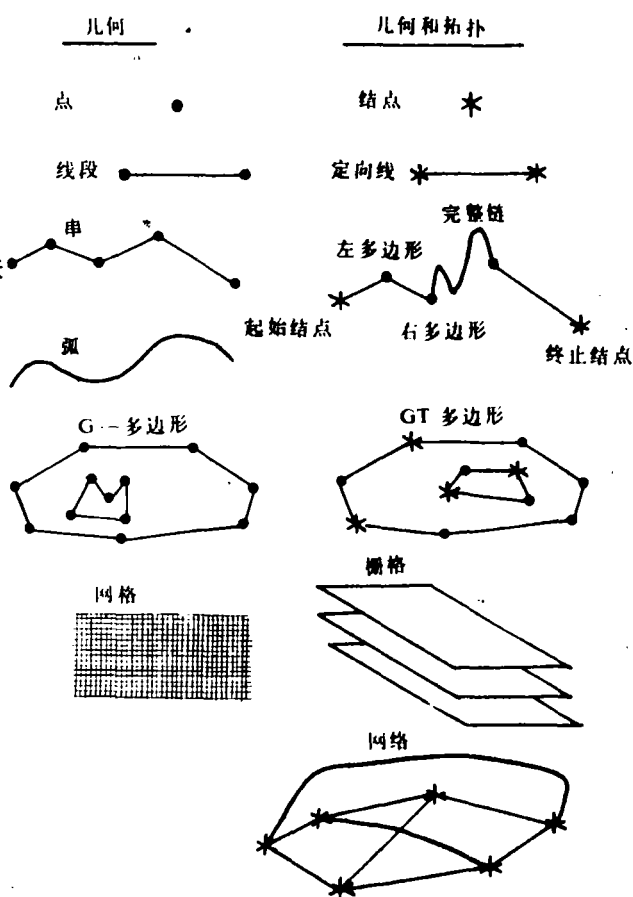


图2 地理目标SDTS概念模型的代表子集

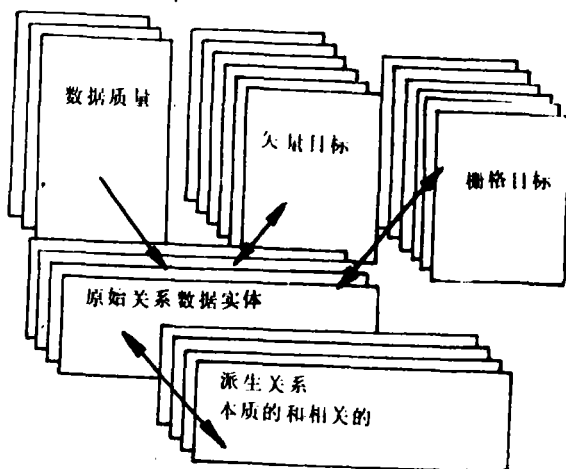


图3 SDTS数据层次集

地理数据库模块的人机界面由图形窗口支持，包括复杂地理目标（例如河流、道路和植被）的建立、更新和检索。

当地图在GUI下显示时，地图模块涉及到形成地图的较大组成部分的说明、存储、检索和显示。显示模块显示矢量或栅格表示的目标，并且支持图形数据输入。GUI或地图显示交互界面都是基于一般的X-Window系统及相关工具，如MOTIF。

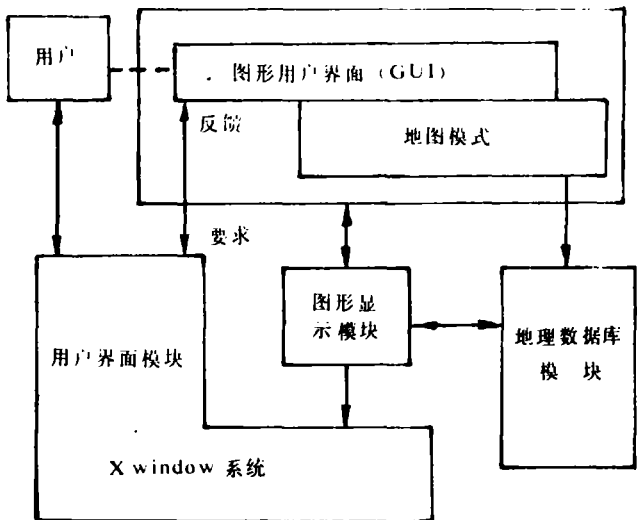


图4 系统模块

3.2 一般概念 一个面向目标数据库设计的一般组成是目标、分类、关系、继承关系和集合。利用Hull和 King (1987)创建的符号，图5表示了一个数据库设计草案。图5模拟了图2中某些空间目标。三角符号代表基本物理或抽象的目标类型，圆圈代表已获得或继承的目标类型。位于继承性箭头起点的目标类型是更一般的，尾部的那些是一般类型的扩展和具体化。例如，一个点是几何目标类型的具体说明。扁平的椭圆代表原始的可打印的目标。这种目标伴随主要目标在功能属性关系中产生。目标间的所有函数关系通过双向箭头的多值函数实线表示。一般目标关系的反演关系也用符号表示。表示网络链的目标关系仅描述开始结点和终止结点，但一个结点可以是很多链的开始或者结束。车轮符号表示一组类型的数据构造者，因此可以表示同一类型的一组目标。通常情况下，这些构造都具有其它结构，如有序结构。

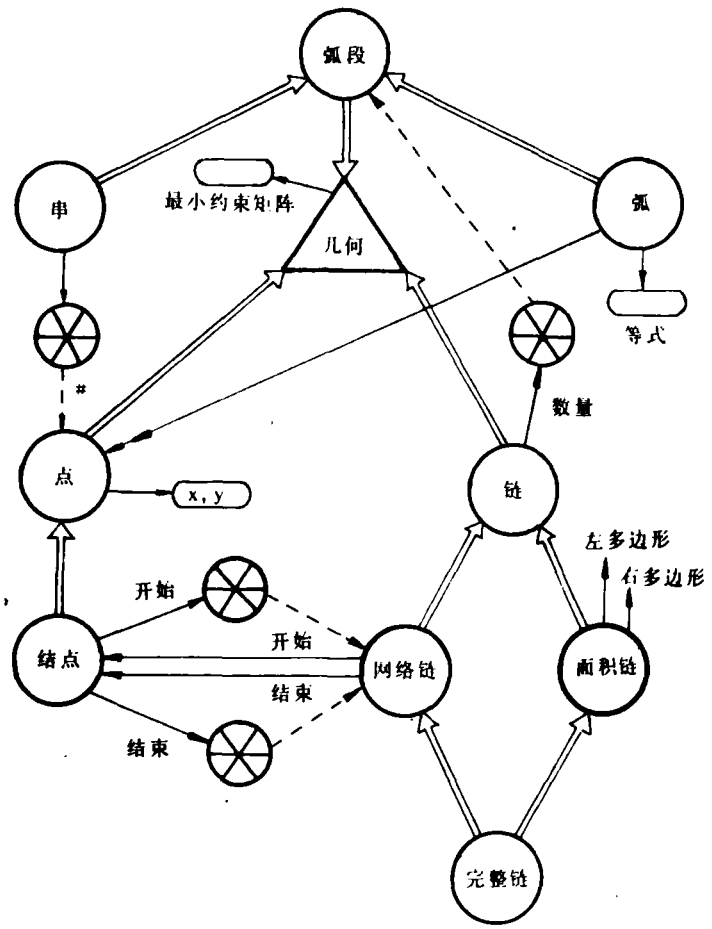


图5 空间数据库设计

3.3 几何目标 某些目标类型如图 5 所示在继承的等级中被描述, 最一般的类型被标以“几何”, 并且有最小边界长方形 (mbr) 的样本属性。结点、栅格、链、环、多边形和组合都是一般类型的具体化。组合是与这些目标相适应的亚类型, 这些目标由不具有任何特殊的拓扑关系的各种空间目标构成。例如, 一个不连续 (组合) 区域是由两个或更多的多边形组成, 这些多边形不具共同的边界。

3.4 拓扑目标关系 几乎图 5 所示的所有的非原始的目标关系都是拓扑关系。这些完成了如图 2 所示的特定目标类型的定义。例如, 一个网络链联系于它的起始结点和终止结点; 作为一个基础链, 对于它的组成弧段, 它也有一个有序的关系。相反地, 一个结点联系于所有以它为起点或终点的线性目标。

3.5 地理目标 地理目标继承了如前所述的几何目标的基本空间性质 (图 6)。许多地理目标继承于这些目标 (如组合) 的集合。地理目标有许多非空间属性, 最常见的一项为名称。一般地理目标被专门分解成各种继承原有等级的亚类型, 例如道路、河流、江心洲和人口中心。每一个都有它自身的结构属性, 并与SDTS模型的原始的和派生的关系相对应。

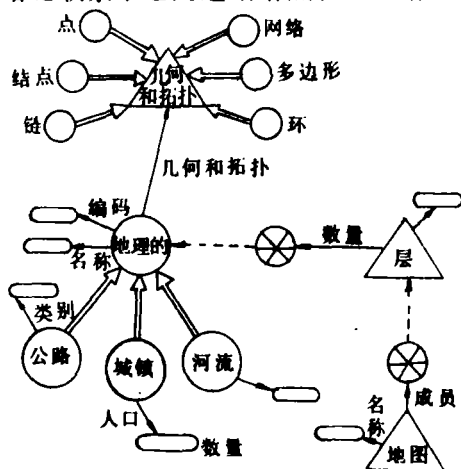


图 6 地理数据库设计

3.6 地理集合 本文设想的主要应用将是产生、显示和存储地图。对于被一幅图覆盖的地理窗口, 我们期望把含有目标的许多组 (层) 收集在一起。有时这些层被看作为地图的特征。因此, 最高层次上, 地图目标是组的集合, 同时, 在最低层次上, 每一组又是同类地理目标的一个集合 (如道路)。相比较而言, SDTS组合是几何目标的集合, 而地图则是地理目标的集合。与单个目标相结合的大量的集合可以是一权衡结果。栅格是特例, 其设计决策基本上就是存取整个采集象元的长方形窗口。

3.7 其它目标 在一个完整的数据库设计过程中, 还有一些其它的目标类型。例如, 用于表示地图上每一特殊目标的图标和控制信息。也存在一些具体用途的地理目标, 但绝大部分重要的论点都表示在熟悉的地理类型中。

另一个设计思想涉及到冗余的关系。这些在数据库设计中常常出现。并且总是完整性的数据库的额外负担。因为这种数据库中的许多目标类型是复杂的, 存在大量的冗余是不可避免的。例如, 所有的河流系统在支流和较大的小溪之间都有一种互相联系的树叉结构。这种相互联系也可从组合目标内集合在一起的链的基本拓扑关系推出 (图 5)。

3.8 过程 GIS的特征是以数字模型 (近来更多的是逻辑模型) 服务于决策支持。被模拟的自然现象十分复杂, 因此, 决策支持过程必须包括计算结果并以地图表示的可视产品。基于这样的理由, 数据库的功能必须支持子集的查询、产生和维护地图, 以及数据库更新。这些需要可以表示成下述的一般数据库过程。(1)指定作为数据库目标子集的地图目标; (2)显示地图目标; (3)存储和修改地图目标; (4)通过地图显示界面选择目标子集; (5)转换和拷贝其它模块的子集; (6)通过地图子集浏览数据库; (7)建立和修改一些类型的目标。

由于决策支持通常是实时的应用, 这种集合的过程的主要含意是执行数据库查询。

4. 面向目标的DBMS(OODBMS) OODBMS是目标范例的集合, 用程序语言 (例如 Smalltalk) 和 DBMS 编制。有用的核心概念定义被 Atkinson 等于1989年和X3项目DBSSG

/OODBTG于1991年给出,并且1991年Vossen出版了最新的文献目录。在数据库中,一个目标与一个SDTS实体相关,虽然通常情况下后者可以对应于目标的组合(正如前述一样)。目标范例在下面的条款中被描述。

①封装是目标外部行为语义和内部执行之间的显著分离。这个黑箱根据操作行为被定义。一种操作根据一种方法进行。

②消息,对于某个目标而言,是一种操作的一个运算域(其可以有几个运算域)。如果一个目标具有可区分的运算域,这种运算域是一种操作的接受者,那么这种操作可以被称为一个消息,其它的运算域可以是参数。

③目标的状态,通过内部数据结构执行,这种结构(作为消息状态变化的结果)可以改变目标状态。某些消息仅仅存取操作。一个消息变化状态的结果随后将改变至少一些消息的结果。

④识别符提供一种方法,独立地用符号表示一个目标的行为和状态。

⑤类型(TYPE)定义一组相似目标,这些相似目标是类型的实例。这是为数据库定义语义整体约束的方式。

⑥类别(Class)是一个类型的具体化。它确定了实例的秩序、消息和性质。一个类别的扩展是实例的结合。

⑦组合目标从逻辑上讲是由其它目标构成,通过操作组合成的复合目标。

⑧联接是与具有秩序的一个操作相联系的行为。

⑨多态性是指联接秩序中的一个选择。因此操作中可以加载其它典型的同类别。

⑩继承性是指新的特征可以从已存在的特征中获取。类型继承性图有单向性联系,这种联系使得子类别(子类型)可以从它们的上一级类别(高级类型)中分化得到。

⑪可扩张性是指确定新的类型和类别(例如子类型)的能力。

虽然在关于OODBMS的意见达到统一之前还有大量的研究需要进行,但面向目标的数据模型可以成为数据库的范例。某些数据库特征(如更新和查询)在很大程度上独立于数据模型之外。OODBMS现行的实现集中于实现数据库对于不同目标适应性,它是在面向目标的编程环境中,采用一种高效的方式产生的。在数据库中的目标标识符和程序语言中的目标基准的变换被建立,从而使得高效的数据库存储的实现成为可能(参见§4.1)。

OODBMS可以有效地处理所谓“阻抗错配”(impedance mismatch)问题,这与关系系统(RDBMS)相比是一个很重要的优点。关系模型的本质是指正常化的关系,包括属性值的 n -元组(行),以及符合原始类型的每一个值。这些属性的某一部分构成了主关系或其它关系(外部关系)的键项(元组标识符,从而也是目标的标识符)。键项是复杂目标和关系的表现方式。在数据库和用于计算的程序语言执行环境之间的数据传输过程发生阻抗错配。如果后者是基于面向目标的程序设计环境,那么在一个OODBMS中,数据模型就与复杂目标的表示十分协调。另一方面,在RDBMS的数据传输中必将产生重要的数据转换和数据构造/分解。

关系模型和数学逻辑有紧密的联系,因此二者有统一基础,以使得陈述查询语言(如SQL)的发展成为可能。另一方面,OODBMS把面向目标和完成计算的程序设计语言(如C++)和数据库结合到了一起,是一重要的优势,虽然数据库语言理论还没有很好地发展。在关心这方面研究的研究协会中曾经有过重要的讨论。

OODBMS中缺失的主要元素之一就是查询模式和关联查询最优化。通过操作人员访问

可能有非常复杂结构的目标所包含的封装数据，这一过程存在着严重的最优化问题。在查询语言方面不一致，相关查询语言（查询的输出是相关的）的封闭性质，在现有的使用类别等级和方法的面向目标的语言中不存在。后一个问题，Scholl等于1991年已阐述过，他们描述了关于目标查询代数的概念。

4.1 主存数据库 这里，主存数据库主要指传递存贮的目标和模拟把一个复杂目标变换成其它数据模式的程序过程。OODBMS具有这种能力，它是我们决定用这种技术作为实现§3.8的基础的一个因素。在其它方法中的许多困难是与程序设计环境和数据库环境间的阻抗错配联系在一起的。

图7围绕OODBMS和RDBMS中的数据存储的使用进行了因子对比。讨论基于这样的假设，即存取仅仅是可读写的或者（看作为工程数据库的一个例子）仅仅存在于专门缓存中数据的一种费时的更新交互操作。所有DBMS以客户/服务器结构的形式进行属性计算，以支持基于同一数据库的多台工作站。因此，程序化的目标（可从数据库中检索）存在于客户数据库中（图7）。由于“阻抗错配”，当使用RDBMS时存在数据冗余。（我们使用术语“超高速缓存”（cache）表示所有缓存，包括应用和系统中的）在OODBMS的客户不存在数据的冗余。目标间的标志（关系）仅仅可以通过数据库服务器交互作用决定。另一方面，当这种关系作为相关“或”被处理时，这必须通过关系数据库服务器决定，占用网络和处理费用。因此，从发展看，关系系统不支持目标的超高速缓存。

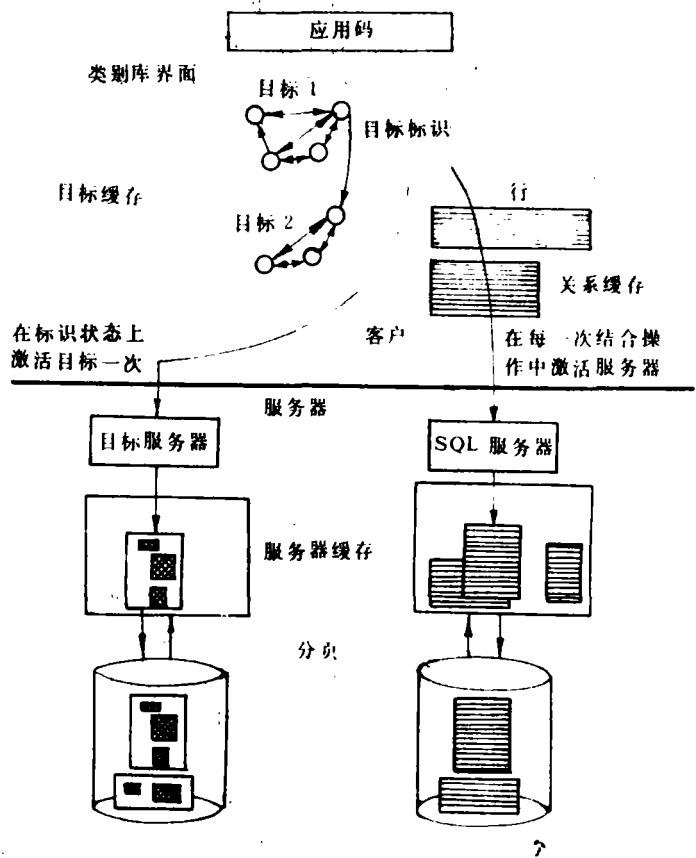


图7 OODBMS和RDBMS上的目标缓存

GIS数据库是典型的支持复杂目标的工程数据库。现行的与RDBMS相关联的专有GIS数据库系统的二方面使用反映了这样一种形势，即我们已被引导去为地理数据库进行OODBMS的试验。在DBMS设备的选择中还需要考虑许多特征（例如，客户/服务器体系和用户设备，语言接口，交互处理，并行控制，锁，查询，图示设计和图解演示，数据字典，浏览和用户界面工具），这里我们仅讨论对于复杂目标和查询程序的支持问题。

5.1 面向目标DBMS的 ONTOS ONTOS (Ontos 1991) 是一个多用户的、分布式(客户/服务器)的，面向目标的数据库系统，这个系统具有C++类别库交互界面。有关ONTOS系统的文件资料表明ONTOS提供了如上所述的广泛的设备支持，这些设备服务于数据库的

存取、处理、保护和管理。但此项目未提供关于这些设备的评价。ONTOS允许C++程序变量表示的目标的生命期比用程序表示的长，并且，它也允许C++程序查询持久性的目标（其它程序产生的目标）进入用§4.1介绍的方式为主存数据库服务的程序变量。目标具有持久和不可改变的标志。

ONTOS也提供了不同类别的C++，它在数据库和面向目标的两方面都扩展了C++的能力。这些包括对聚集、迭代、异常处理和人机界面结构的支持。数据库的主要工具和用途与应用发展如图8所示。

5.1.1 面向目标语言 C++

ONTOS由C++实现，C++是目前唯

一可以操作ONTOS数据库的语言。C++是从C发展而来，它在三个方面扩展了C语言：

①改进了目前C的结构，尤其在打印方面；②为建立及应用封装数据提供支持；③为面向目标的设计及编程提供支持。C++与C语言两者是兼容的。

在C++中，用户定义的目标数据类型被归并为一个类别，这个类别是指定数据元素的集合（数据可以是不同的类型），以及一系列的操作（这些操作是为操作这些数据设计的）。因此，具有封装和类别交互功能。它还具有继承性。据此，子类别可以从根类别继承实施，动态接口是由C++支持的面向目标程序的另一关键方面。

5.2 地理目标类型分层 在ONTOS，数据库的实现是由大量的被定义为“类型”、“属性”、“程序”构成。一种ONTOS类型是反映C++类别的数据库。数据库中的每一个例子都是某一个类型的，同时，也有与之相对应的C++类别。ONTOS中的类型是较类别更为有力的概念。也就是说，这些类型是可解释的，并且允许判读应用的组合。任何类型都是可扩展的，所选择的例子是为集合处理而设计的。属性和程序是与C++类别及数据成员及成员功能相应的目标类型。在ONTOS中，它们也可以在解释方法中得到应用。

图6中的类型是从图5中的目标类型获取的，表达了由数据库实现的地理类型分层，亚类型的地理目标是特别为区别地面目标及其它空间目标而设计的。例如，那些从地图上提取作为应用的一个方面的区域应用目标。图6中表示了从地理类型中继承的对应于文化、植被、水系的特殊目标。

5.3 目标标识 (references) 图5及图6中的目标关系必须由ONTOS中的标识来实施。在ONTOS中，数据库目标标识应由单一、独特的识别空间中获得的UID来实现。当一目标在数据库中被激活时，需对应分配存储区，并将在存储区中的目标标识返还。在首次标识时，所有激活目标包含的标识也被译成基于存储的标识。因此，在应用中，存储标识可以自由地应用，如同真实的UID。在本项目中正在进行关于类别库的工作。类别库将支持作为一级类别的关系。

5.4 地理目标类别库 数据库的交互实施很象C++类别库。所有利用地理分类的程序

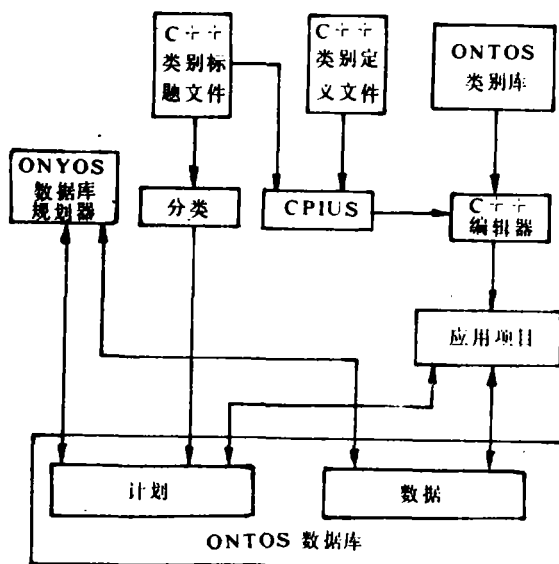


图8 ONTOS 工具和用途

(也就是建立地图目标程序)必须包含于相关的标题文件中。与此相类似,每一类型的程序(成员函数)也必须能够和可执行程序相连接。这一点可以通过建立图8所示的C++库来获得支持。这种实用分类产生了图形源目标(如类型、属性、程序),C++的编译功能为生成需要用ONTOS库编译应用程序的多功能信息提供了便利方法。

5.5 空间存取方法 空间存取方法由空间索引类别来实现。这种方法应用了ONTOS的字典类型来构成一个索引化的成员集。索引键是一种约束矩阵。所有的几何及地形目标都有这种约束矩阵性质。空间索引类别程序为检索和更新实现了一种简化的SIRO-DBMS的存取方法。

6. 性质测试 图1中的数据库中等高线层面可用于三种不同的数据库实施比较。这种比较着重于对比每一系统的组合及复合目标的处理能力,并着重于展示从OODBMS中的目标自动生成的优点。所对比的三个系统是ORACLE, SIRO-DBMS和ONTOS。

每一等高线内几种属性(也就是高度、最小约束矩阵、图形属性)并用一系列三维空间点来定义位置。

ORACLE数据库设计由两个表构成: 1.第一个表具有主键、CONTOUR-ID和相应于上面提到的属性列; 2.第二个表具有复合键CONTOUR-ID和POINT-NO。每一排都包含基点坐标。POINT-NO定义了沿等高线分布的点构成的串状的顺序,两种关系都在CONTOUR-ID上指示,没有使用聚类方法。

SIRO-DBMS完成于ORACLE之上。SIRO-DBMS数据库设计的几何目标使用了直接关联形式。在这种数据库中,含有一种相关,对应于一系列点,这些点为包含一系列数据的一条等高线。当查询时,SIRO-DBMS将ORACLE字节值解释成一系列点坐标。

C++ONTOS数据库设计以等高线分类为基础。等高线分类数据成员是一个链目标标识。这些链目标由定义等高线的点坐标串构成。

为了预测可测试的服务器缓存和客户目标缓存的效用,对上述三种数据库,开发了如下一种简单的测试作业。每一数据库中含有2568条等高线,共有266029个坐标点。测试作业包含一个可检索数据库中每一等高线的途径。该途径在测试作业结束前共执行二次。

在ORACLE和SIRO-DBMS中,途径的核心是含有一个SQL选择语句的PRO*C命令。在ORACLE的应用中,选择语句含有一个联合词(Join term)。ORACLE应用利用了一个400维的数组缓冲器,即,在一次“取”命令中,400条等高线的点被返回执行环境,见表1(ORACLE执行时间在没有使用到缓冲器时,达到1500秒)。对ONTOS而言,途径是通过对于集目标成员,每一标识链目标的标识链目标是在第一途径中激活的。为激活目标给出一个标识仅仅只需做一个地址变换。

运行于Sun Sparc Station 2(具有40Mbytes主存)进行测试。测试是在工作站上做的,且没有其他数据库使用者。得出了同一测试(每一个含有二种检索途径)连续二次执行的时间。结果证实了我们的预期,也即,在ONTOS中,经第一次检索,大量的等高线及链目标将在客户/应用存储中被激活,并可利用相同途径迅速存取。相反,通过使用基于SQL的过程,则不会有这样的优点。

二次结果的不同可归因于ONTOS中的服务器超高速缓存。也许会认为在SIRO-DBMS和ORACLE中应该有相同的结果存在。可能是在二次执行之间的非激活状态的数据库导致了ORACLE服务器刷新其缓冲区。

1989年,Abel提供了利用空间索引对空间目标随机检索的测试结果。对ONTOS,也开

发了相同的索引方法，希望这一目标检索方式在此同样具有那些优点。

在主存储器中，必须具有§4中描述的数据库能力。这一点在为完成§3中的需求而决定是否使用OODBMS时起了决定性的作用。我们在该项目中工作的重点主要集中于建立20MB已描述过的复杂目标的数据库，并完成特定的检索和图形显示功能。这一应用还要求交互创立某些类型的空间目标。数据库的应用着重于单用户方式。在二次释放的ONTOS中，采用了如下的全面测试显示模块及数据库模块方法。

复合目标数=5372

描述目标的总点数=84996

显示中使用的总点数=46170。

ONTOS 版本1.5 (XXM图形模式) (Sun SPARC Station 1)

从数据库中读及显示 (以前无任何目标的读取) :

从数据库中读并以地图形式显示的时间=4分53秒

速度18.3目标/秒; 290点/秒。

读及显示 (目标在主存缓存器中) :

以地图形式显示的时间=53秒

速度: 101目标/秒; 1603点/秒。

ONTOS 版本2.0 (Xs图形模式) (Sun SPARC Station 2)

从数据库中读和显示 (以前无任何目标的读取) :

从数据库中读并以地图形式显示的时间=31.7秒

速度: 170目标/秒; 2681点/秒。

读和显示 (目标在主存缓存器中) :

以地图形式显示的时间=8秒

速度: 767目标/秒; 12142点/秒。

2.0版本对1.5版本的数据库性能改进主要是由于在数据库组织中应用了成组存储管理模式。该方法对类目标文件进行集中 (Cluster) 管理。因此，减少了某种存取方式的磁盘存取次数。应用该形式的存储管理，在目标文件中，加快了指针的标识。2.0版本下为测试而用到的Xs图形模式，是用来表示图4中的显示模块的面向目标的分类库。XMM是实施中的C的前驱。Xs可以以超过15000点/秒的速度将矢量地理目标文件传输到屏幕上，差不多可达到30000点/秒的速度。因此，从主存储器数据库中进行的操作得到改进，其速度分别提高到原来的5.5和11倍。

7.讨论 我们赞同这样的观点，即，面向目标的设计是在概念水平上最自然的设计方法。图5中使用的扩展了的目标——关系概念方便地给出了面向目标的数据库设计。另外，模型子集在由面向目标语言如C++构成的应用程序中得到支持。假如选择了非面向目标的程序语言，将在概念模型和数据库模型之间产生语义差。利用SQL关系数据库同样会产生语义差及阻抗错配。另一些方法如SIRO-DBMS (Abel, 1981) 为C语言编程者减少了这种

问题。但 SIRO-DBMS 在设计中需要与 RDBMS 一起使用。

面向目标方法提供了一种从概念到实施的统一模型。不用面向目标技术而只用面向目标规则，便可得到这种联合的好处，但目前的工具使这种方法更实用、更有效。C++ 由于其在打印（typing）方面的优势，意味着大量的 C 调试可以免去。尽管如此，一般是把 ONTOS 数据库标识打印出来的。

从长远来看，之所以采用 OODBMS，最有力的证据是其在面向目标设计和实施方面软件工程上的优势。在与此项目有关的其它工作中，我们正致力于几何及拓扑类别库的开发。该软件将为未来的空间数据库和其它应用的开发者提供可重复使用的、大量的模块。

正如本文中已经讨论过的，面向目标软件的开发较其它方法需要较少的时间和努力。在数据库交互中，不需要编程去解释目标。（即，无阻抗匹配）。而且可以重复使用以前开发好的模块。例如，ONTOS 提供集合类、合并的存取方法。这对产生开发空间目标分类及空间存取方法特别方便。这些类型可以简单地从 ONTOS 中得到继承，或者，通过合并予以使用。因此，应用是对 OODBMS 软件的扩展。

要获得参加数据库测试的数据，花费是很大的（人工数字化、解译花费）。这使得我们决定获取一些矢量数据，用它们来测试一个 OODBMS 数据库优越的代表能力及实现潜力，而没有必要再使用其它形式来存取这些数据。为此，我们提倡带有关系表的 USGS 数字线图（DLG）格式，这些关系表表明另外的拓扑关系。为装入 ONTOS 数据库，开发了一种输入应用程序，由于该程序是从线性输出/输入格式中建立复杂目标文件，它通过缓冲区数据执行大量传送，建立数据库目标文件，并解释目标标识。

利用以上数据，也可以表明面向目标方法中的一个最显著的优点。可以利用鼠标选择界面，在一幅地图中定位一条河流弧段，并且可以检索相关的河网水系。一个具有上百个相交连的目标的复杂河流系统，只需几秒的反应时间。这种类型的锁合计算在一些有关的系统中也可实现，但通常速度很慢。

上述例子解释了 OODBMS 中查询语句的局限性。而此在 ORACLE 中用少量高级的相关命令就可表现。当 ONTOS 中具有目标—SQL 时，它达不到关系联合的完整功能。因此，一些查询是通过过程目标途径来实施的。但是，正如前述，通过先期定义关系的辅助定向是非常有效的。

任何团体，当决定采用面向目标技术时，都不应低估服务于这里所描述的学习曲线。C++ 和 ONTOS 无论在学习概念方面，还是在利用该工具形成生产力方面对一般的软件工程师将会提出有力的挑战。迄今为止，我们发觉 ONTOS 产品对这项任务有效的，它只不过展示了出自于高技术公司的新产品的惯常现象。在版本 1.5 中，最主要的困难是不合适的文件编制和释放间的不兼容。这种情况在版本 2 中已得到显著改善。

许多 ONTOS 系统中的关键操作特点，我们都没有测试和评价。我们着重研究了支持地理数据库的定义、建立、更新和检索的技术能力，该数据库由相当多数目的具有大量关系的复杂目标类型构成。ONTOS 已被证明是一个可进行多项应用的可靠产品。这些并不能证明它是一个完善的 DBMS。

章燕燕译自《IJGIS》1993，第 7 卷，第 1 期