

# 工程数据库管理系统 设计、实现的几个关键问题\*

\* 顾宁 \*\* 林宗楷 \*\* 郭玉钗

\*(复旦大学计算机科学系 上海 200433)

\*\* (中国科学院计算技术研究所CAD开放实验室 北京 100080)

**摘要** 本文针对工程领域的特点,提出基于关系/网状的语义超图模型描述,以表达工程中不同语义类的多对多复杂对象关系;将该模型描述和表达的管理机制和基本数据在内存中统一管理和运行,使工程数据的存取效率大为提高;增加了大量通用和特殊功能,形成了功能更强、使用方便的C语言界面。在上述基础上自行研制并有自主知识产权的工程数据库管理系统EDBMS(engineering database management system)已在建筑CAD领域中得到实际应用。

**关键词** 工程数据库,数据模型,内存管理,C界面,数据库应用。

工程领域所处理的对象常常是具有复杂结构和复杂语义的复杂对象,对象间的关系可表现为不同语义类的多对多关系。提供良好的描述机制,以方便用户建模对于工程数据库管理系统EDBMS(engineering database management system)是非常重要的。传统的着重描述记录间多对多联系的网状数据模型<sup>[1]</sup>难以满足要求,而“纯”面向对象方法在描述复杂对象时也受到类层次结构限制。<sup>[2]</sup>

在CAD应用中,用户希望EDBMS有高的存取效率。将数据尽可能多的放入内存,尽可能地减少I/O次数是提高存取效率的一个关键。由于工程数据中含有大量的图形、图象等多媒体数据,将这些数据全部同时放入内存,在目前尚不具备条件。<sup>[3]</sup>但将EDBMS的模型所表达的全部管理机制和基本数据放入内存,并进行统一管理和运行是现实和可能的。

数据库语言界面是数据模型及其操作的具体表达。EDBMS通常作为CAD集成系统的支撑工具,其工程用户需要EDBMS提供功能更强的宿主语言界面,以便和各CAD系统紧密相连。如PLUS系统<sup>[4]</sup>提供用于管理几何数据的C语言界面,CAD \* I<sup>[5]</sup>提供用于支持机械CAD应用的FORTRAN语言界面。

\* 本文研究得到国家“八五”攻关项目基金资助。作者顾宁,1955年生,博士,讲师,主要研究领域为工程数据库,多媒体数据库。林宗楷,1934年生,研究员,博士生导师,主要研究领域为工程数据库,多媒体数据库,中文数据库,协同设计。郭玉钗,女,1937年生,研究员,博士生导师,主要研究领域为电子CAD,工程数据库,多媒体技术。

本文通讯联系人:顾宁,上海200433,复旦大学计算机科学系

本文1996-04-05收到修改稿

### 1 语义超图模型描述

在定义 1.1~1.6 中,均令  $S$  是唯一属性名的有限集.

**定义 1.1.** 若  $A_1, \dots, A_n$  是  $S$  中的不同属性名,且属性值是不可分的数据单位,则  $R_1 = (A_1, \dots, A_n)$  称为  $S$  上的基本关系,  $R_1$  称为基本关系名.  $A_i (1 \leq i \leq n)$  是基本属性名. 记  $value(A_i) (1 \leq i \leq n)$  是基本属性值.  $d(A_i) = \{value(A_i)\} (1 \leq i \leq n)$  是基本属性域.

**定义 1.2.** 嵌套关系递归定义如下:若  $B_1, \dots, B_n$  是不同的关系名,则  $R_2 = (B_1, \dots, B_n)$  称为  $S$  上的嵌套关系,  $R_2$  称为嵌套关系名.  $B_i (1 \leq i \leq n)$  是嵌套属性名. 记  $value(B_i) (1 \leq i \leq n)$  是嵌套属性值.  $d(B_i) = \{value(B_i)\} (1 \leq i \leq n)$  是嵌套属性域.

**定义 1.3.** 若  $C_1, \dots, C_n$  是  $S$  中的不同属性,且属性值是数组名、工程长记录名或变长属性名,则  $R_3 = (C_1, \dots, C_n)$  称为  $S$  上的特殊关系,  $R_3$  称为特殊关系名.  $C_i (1 \leq i \leq n)$  是特殊属性名. 记  $value(C_i) (1 \leq i \leq n)$  是特殊属性值.  $d(C_i) = \{value(C_i)\} (1 \leq i \leq n)$  是特殊属性域.

**定义 1.4.** 若  $A_i, B_i, C_i \in S (1 \leq i \leq n)$ , 则  $R = (A_1, \dots, A_n, B_1, \dots, B_n, C_1, \dots, C_n)$  称为  $S$  上的关系,  $R$  称为关系名.

**定义 1.5.** 令  $R$  是  $S$  上的关系,关系域  $D(R)$  定义如下:

$D(R) = d(A_1) \times \dots \times d(A_n) \times 2^{d(B_1)} \times \dots \times 2^{d(B_n)} \times d(C_1) \times \dots \times d(C_n)$ , 其中  $2^{B_i}$  是  $d(B_i) (1 \leq i \leq n)$  的幂集.

**定义 1.6.** 对于  $S$  上的关系  $R$ , 令  $T_i \subset D(R) (1 \leq i \leq n)$ , 对于任一  $O(I, V) \in T_i$ , 其中  $I$  是对象标识,  $V = [n_i, v_i] (1 \leq i \leq n) (n_i \in S, v_i \in d(A_i) \cup d(B_i) \cup d(C_i))$ , 称  $O(I, V)$  为  $D(R)$  上的对象或元组.

**定义 1.7.** 数据库模式  $D$  是一个语义超图五元组  $D = (X, S, E, F, M)$ . 其中  $X = \{x_1, x_2, \dots, x_n\}$  是  $n$  个顶点(对象)的有穷非空集合.  $S = \{s_1, s_2, \dots, s_n\}$  是  $n$  种语义的有穷非空集合.  $E = \{e_1, e_2, \dots, e_m\}$  为语义超边集, 语义超边  $e_i (e_i \in E \wedge e_i \neq \emptyset)$  为  $X$  的某一子集, 且  $\bigcup_{i=1}^m e_i = X$ .  $F: 2^X \times 2^S \times 2^M \rightarrow 2^X$  为语义映射关系.  $M = \{m_1, m_2, \dots, m_n\}$  是  $n$  个方法的有穷集合.

**定义 1.8.** 语义子超图  $D_P = (X_P, S_P, P, F_P, M_P)$ , 其中  $P \subset E, X_P = \bigcup e_i (e_i \in P), S_P \subset S, M_P \subset M, F_P: 2^{X_P} \times 2^{S_P} \times 2^{M_P} \rightarrow 2^{X_P}$ .

**定义 1.9.** 语义子超图中结点间的指向边称为语义边. 语义子图  $D_G = (X_G, S_G, G, F_G, M_G)$ , 其中  $X_G = \{x_1, x_2, \dots, x_n\}$  是  $n$  个顶点的有穷非空集合,  $G$  是语义边的有穷集合,  $S_G = \{s_1, s_2, \dots, s_n\}$  的  $n$  种语义的有穷非空集合,  $F_G: X_G \times S_G \times M_G \rightarrow X_G$  为语义映射关系,  $M = \{m_1, m_2, \dots, m_n\}$  是  $n$  个方法的有穷集合.

**定义 1.10.** 对  $\forall x, \forall y (x, y \in X), \exists s (s \in S), \exists m (m \in M)$ , 若  $x$  对  $y$  有语义联接, 且对  $x$  或  $y$  有相应的方法操作(方法集合可以为空), 记为  $x \xrightarrow{s_i} y (i = 1, 2, \dots, n)$ .

**定义 1.11.** 若  $x \xrightarrow{s_i} x$ , 称  $s_i$  是自反语义. 若  $x \xrightarrow{s_i} y$  且  $y \xrightarrow{s_i} x$ , 称  $s_i$  是对称语义. 若  $x \xrightarrow{s_i} y$  且  $y \xrightarrow{s_j} z$ , 称  $s_i$  是可传递语义.

**定义 1.12.** 含有嵌套属性的对象不能有自反、对称和可传递语义联系.

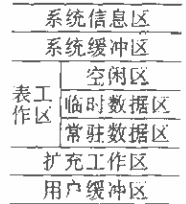
定义 1.1~1.5 突破了 1NF 的限制, 允许在关系中定义嵌套和特殊属性. 定义 1.6 将对

象和元组统一定义. 扩充实体联系模型(EER)<sup>[6]</sup>通过实体、属性以及实体间联系来表达复杂对象, 其描述受到面向对象方法的类层次结构限制. 定义 1.7~1.12 所定义的超图和语义子图处于 2 个表达层次. 由用户所定义的语义联系可突破类层次限制(满足定义 1.12), 定义 1.10 允许对象的方法操作. 超图层次主要用于高层表达和交互界面演示操作, 如超图的交、并、差、否定等操作. 而语义子图层次则更多地面向工程用户, 用户可通过语义子图的操作语句实现对象间联系的插入、更新、查寻和方法实施等操作.

## 2 存储管理和调度

图 1 给出了管理系统的内存分配. 其中系统信息区存放表工作区控制块、对象和语义联系控制块、索引和版本控制块以及各特殊属性控制块在内存的首地址; 系统缓冲区存放当前表字典和当前元组值; 表工作区存放全部表字典和表内容; 扩充工作区存放除表工作区控制块以外的系统控制块及相应内容.

我们把表、记录的内外存调入调出称为表切换和记录切换. EDBMS 的内存调度策略如下:



(1) 当内存足够大时, 将除了  $L$  属性(该属性值是图形、图象等文件)的全部数据库内容读入内存.

(2) 当内存空间不够时, 依次不读入(或淘汰)下列内容:

· 扩充工作区. 其中的各控制块和相应内容次序为:(1)版本机制(2)索引机制(3)工程长记录(4)变长记录(5)嵌套属性(6)对象及联系机制.

· 表工作区.

· 系统信息区. 不读入(或淘汰)内容的时机与上述(6)一致.

(3) 当对内存中数据库内容做插入和建立对象间联系等增加内容的操作时, 若内存空间不够时, 做第(2)步骤(淘汰).

由于表工作区存放的是系统中全部表的数据字典和表的实际内容, 所以对数据库的多种操作通常只需对表工作区进行操作. 对表工作区的具体管理策略如下:

(1) 允许用户定义存放表的区域——表工作区的大小, 并允许用户定义每一个表在表工作区中的记录个数, 即用户可以将某个表的一部分放入表工作区. 若用户放弃定义, 系统便自动将所用到的表尽可能地全部读入工作区.

(2) 系统根据用户定义的情况, 将某些用户定义的表的全部(或一部分)放入表工作区中的常驻区, 其余放入表工作区中的临时区, 表工作区中放不下的继续留在外存. 系统对内、外存中所有的表统一管理和控制.

(3)  $N$  个记录一次处理具有 2 种含义. ①表中  $N$  个记录在内、外存的数据交换一次 I/O 便可完成, 即从外存一次读入  $N$  个记录到内存, 或从内存一次写  $N$  个记录到外存. 这样, 使记录切换变为包含  $N$  个记录的记录块的切换. ②用户对表工作区的  $N$  个记录可以同时存取, 即可同时插入  $N$  个记录到表工作区, 也可同时从表工作区中取  $N$  个记录. 所以, 对每个记录块的各种检查和处理只要进行 1 次, 这就比单表记录方法的  $N$  次处理所花费的时间要少.

(4) 若表中只有  $N$  个记录在内存, 其余在外存, 则随机和顺序操作便会产生不同的影

图1 内存分配

响,必须采用不同的策略.

• 对于顺序操作:当读取的记录不在内存,系统自动从外存的该记录起一次读入  $N$  个记录到内存(原内存中的记录若被修改则要回写),再从内存读取该记录.

• 对于随机操作:若读取的记录不在内存,则自动到外存去读该记录,不做  $N$  个记录的一次读入.

(5)对常驻区的内容不淘汰,直到用户释放或事务结束时,才将常驻区的内容释放.当用户申请常驻区而空闲的空间不够时,便到临时区去查找,若该表在临时区,便直接作为常驻区内容.若用户申请的表不在临时区中,便要将临时区中的内容淘汰,淘汰策略是以表为单位,后进先出.

(6)若数据仅在外存,当修改数据(插入、删除和更新)时立即修改外存数据暂存区;若数据在内、外存中同时存在,当数据发生修改时,仅修改内存数据,在下列时机,通过 1 次 I/O 回写到外存数据暂存区:①表切换;②表中的  $N$  个记录的记录块放满,还要插入记录;③常驻区的数据块释放;④系统发生错误,将上一次修改的数据回写;⑤事务结束.当提交事务时,用外存暂存区的内容替代外存数据区中的内容.

常规的存储管理策略<sup>[7]</sup>将数据库分为大小相等的页,或页的集合——段,并设计了各种置换算法,如 LRU, RAND 等,其目的主要为了减少页置换的 I/O 操作.本文提出的存储管理方法具有如下特点:(1)在内存允许时,所有管理机制(包括语义联系、版本、索引等)和基本数据全部放在内存.(2)允许用户采用第 1 节提出的方法直接对表中元组(对象)进行内存中的语义超图建模和操作,使得对已建立语义联系的对象查询非常方便、快捷.(3)以可供用户自行定义的表大小为基本单位,进行统一管理和 I/O 操作.这样,若需要,工程设计人员便很清楚内存中表的数量和运行情况,并可对特定表进行干预,以便加速整个系统的设计过程.

### 3 C 语言界面

EDBMS 的 C 语言界面功能如表 1 所示,在表 1 中,打“√”的部分表示有此功能.例如:在  $V$  属性操作中,提供的操作功能有:用数组元素(一维或二维)对表中记录的变长属性插入和更新;并可获取表中单个和  $N$  个记录的  $V$  属性值,放入数组中.

我们对 EDBMS/2 的 C 语言界面<sup>[8,9]</sup>进行了大量地增加、修改和扩充,形成了 EDBMS 的 C 语言界面.在功能安排上,除了库、表、记录、索引和统计等通用功能外,主要体现在:(1)建模.机械 CAD 领域要求属性值是复杂结构,第 1 节的定义 1.3 可适应这种需要,定义 1.7~1.12 是适应建筑 CAD 要求的不同语义类的多对多关系而提出的.当然,这同时也适应机械、电子 CAD 的相应建模要求.语义联系的有关操作语句是针对语义子图层次的,超图层次的语句不在表 1 中.(2)存取效率.由于提供内存管理操作语句,再与语义建模语句配合,使得对具有语义联系的对象操作可在内存进行,使存取效率得到提高.(3)特殊功能.对数组、结构数组、 $V$  属性和  $L$  属性的插入、提取、更新以及对字符属性的特殊处理等功能可满足工程领域的需求.以数组为例,大量建筑 CAD 应用软件是用 FORTRAN 编制的,其待处理的数据以数组形式准备,处理后的数据也希望放在数组中.因此,工程设计人员希望 EDBMS 具有数组接口.

表 1 EDBMS C 界面的功能

操作对象	操作功能							其它
	创建	删除	打开	关闭	插入	提取	更新	
数据库	✓	✓	✓	✓				获取长库名
库 LINK	✓	✓						向前查找
								向后查找
表	✓	✓						条件拷贝内容
								条件拷贝结构
								获取结构
								表联接, 表 Link
								获取表注释
记录					✓	✓	✓	条件提取
								条件更新
								码名互查, 预
								删, 条件恢复
记录语义 联系	✓	✓						实际删
								向前查找
结构数组					✓	✓	✓	向后查找
								N 个记录
索引	✓	✓						满足条件
								条件查询
数组					✓	✓	✓	N 个查询
V 属性					✓	✓	✓	表达式计算
L 属性					✓	✓	✓	N 个记录
								N 个记录
字符属性								满足条件
统计								转换
内存管理	过滤, 拼接, 满足条件, 转换							记数, 求和, 平均
	申请, 释放工作区, 随机和顺序操作, 提取和插入 N 个记录							

#### 4 EDBMS 在建筑 CAD 中的应用

EDBMS 作为国家“八五”攻关项目“民用建筑集成化微机 CAD 系统 BICAD”的一部分已形成产品推向市场. 该系统如图 2 所示. 我们对图 2 中的结构荷载、楼板计算和平面框架的数据处理时间进行了统计, 所涉及的表有总信息表、楼层表、截面表、墙表、梁表等 67 个表. 系统运行环境为: PC-486/33, 16M 内存, 500M 硬盘; DOS 6.0, WATCOM C 32 位编译器. 对各类数据通过 18 个工程实例分别在 EDBMS/2 和 EDBMS 时运行并进行比较, 得到工程应用程序的平均提高效率为 1.2 倍.

#### 5 结束语

在前期工作的基础上, 本文讨论了基于关系/网状的语义超图模型描述、存储管理和 C 语言界面. 由于提供了语义超图的建模手段, 工程用户可以不必修改已经建立好的关系模

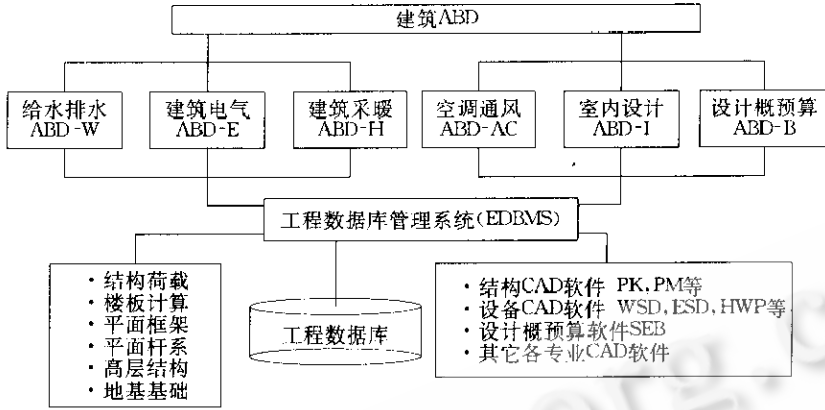


图2 BICAD系统

式,进一步建立不同对象间的语义联系,形成更高层次的新模式;将上述模式和基本数据全部放入内存进行管理,使查询和浏览更为方便和快捷;用户通过C界面可把EDBMS和CAD系统集成在一起.版本管理和长事务管理在本系统中已实现并实用化,详细内容另文介绍.

**致谢** 感谢中国建筑科学研究院魏文郎研究员、黄如福和张汉义高级工程师所提出的用户需求和设计草案;感谢同课题组的林守勋研究员、余遐助理研究员、余涛硕士和林志丹硕士等全体同仁所做的大量工作.

### 参考文献

- 1 Stig Ulfby *et al.* TORNADO; a DBMS for CAD/CAM systems. North-Publishing Company, IFIP. 1982. 335~346.
- 2 Khoshafian S. Object-oriented databases. John Wiley & Sons Inc. , 1993.
- 3 Garcia-Molina Hector, Salem K. Main memory database systems: an overview. IEEE Trans. on Knowledge and Data Eng. , 1992,4(6):509~516.
- 4 Garling D *et al.* Non-standard database management for CIM. In: Proc. of International Conference on CAD & CG'89, 1989.
- 5 Raflik M, Patzold M. CAD \* I database: an approach to an engineering database. Springer-Verlag. 1990.
- 6 Hughes J G. Object-oriented databases. Prentice Hall Int. (UK)Ltd. , 1991.
- 7 周龙骧. 数据库管理系统实现技术. 武汉:中国地质大学出版社,1990.
- 8 Lin Zongkai *et al.* The database technologies used for electronic CAD system. In: Proc. of International Conference on CAD & CG'89, Beijing, 1989. 372~376.
- 9 Zheng Weidong *et al.* A data manager for engineering application. J. of Comput. Sci. & Technol. , 1993,8(4):307~316.

## SEVERAL KEY RESEARCH RESULTS IN DESIGN AND IMPLEMENTATION OF ENGINEERING DATABASE MANAGEMENT SYSTEM

GU Ning

*(Department of Computer Science Fudan University Shanghai 200433)*

LIN Zongkai GUO Yuchai

*(CAD Laboratory Institute of Computing Technology The Chinese Academy of Sciences Beijing 100080)*

**Abstract** According to the features of engineering domains, this paper presents a semantic hypergraph model based on relation and network, in order to express many to many relations of different semantic classes. The management mechanism expressed by the model and the basic data of databases are managed and run in main memory, thus the access efficiency of engineering data is increased. C language interface which includes general and spacial functionalities is proposed for meeting requirements of engineering applications. The EDBMS(engineering database management system) with above research results is implemented and has been actually used for engineering domains.

**Key words** Engineering database, data model, memory management, C language interface, database application.