

# 一个面向对象数据库系统的研究与实践\*

徐庆云

王能斌

(华东石油地质局研究中心, 南京 210031)

(东南大学, 南京 210093)

杨传钧

毛兆余

黄玮

(华东计算所, 上海 200233)

(机电部 11 所, 北京 100011)

(北京信息工程学院, 北京 100101)

**摘要** 本文介绍了一个适用于 CIMS、CAD、AI 和 GIS 等领域的面向对象数据库系统 WHYMX。WHYMX 给出了聚合规范化和概括规范化的理论描述, 提出了复杂对象的并发控制策略、存贮管理策略和对象管理方法。它支持嵌套事务处理和大对象的存贮管理。它在对象描述层下面增加了基于元组和面向值的操作, 从而把面向对象模型与关系模型有机地结合在一起。

**关键词** 面向对象数据库, CIMS, 总体结构, 数据模型, 对象管理, 事务管理, 存贮管理。

80 年代以来, CAD、CIMS、AI、OA 和 GIS 等领域发展迅速, 这些领域迫切希望新的数据库技术给予支持。为了适应新的应用需要, 发展数据库技术已势在必行。

近年来, 数据库技术的研究主要有 3 个方向。第 1, 是构造一种统一的数据模型, 对不同的应用领域均采用这种模型来处理不同类型的数据。这样的数据库系统十分庞杂, 性能很差, 并且不易扩展与维护, 各应用领域已取得的各种数据不易直接移植到新的数据库管理系统(DBMS)之中。第 2, 是在每一领域都设计专用的数据模型和 DBMS, 如专家数据库、演绎数据库、知识库和工程数据库等。每一系统只处理某一特定领域的数据库, 具有较好的性能。但由于各应用领域的数据库可能有一些公共操作, 在上述系统中, 可能存在着相似甚至相同的功能模块, 导致人、财、物的很大浪费, 总体效率差, 而且各数据库间难以通讯, 难以共享数据。第 3, 是研制一种便于扩展的数据模型, 使数据库系统具有良好的扩展性, 可以根据不同领域的需要, 增加新的功能模块, 而无需修改 DBMS<sup>[1,2]</sup>。这种方法比前两种方法更灵活, 更有吸引力, 面向对象的数据库技术正体现了这种思想。WHYMX 就是我们针对 CIMS 需要而研制的一个面向对象的数据库管理系统。

## 1 WHYMX 的总体结构

WHYMX 如果要支持 CIMS 和 CAD 等应用, 其总体结构必须要适应工程设计环境中

\* 本文 1992-06-04 收到, 1993-03-09 定稿

作者徐庆云, 1958 年生, 工程师, 主要研究领域为数据库, 图形学。王能斌, 1929 年生, 教授, 博士生导师, 主要研究领域为数据库及信息系统。杨传钧, 1965 年生, 工程师, 主要研究领域为数据库, 人工智能。毛兆余, 1963 年生, 工程师, 主要研究领域为数据库, 自动控制。黄玮, 女, 1966 年生, 讲师, 主要研究领域为数据库。

本文通讯联系人: 徐庆云, 深圳 518031, 深坊大厦 B 座 1809-1810 室, 傲能科技(深圳)有限公司

由多人分工协作完成同一任务这一特点.为此,我们把 WHYMX 逻辑上划分为公共数据库、项目数据库和专用数据库.公共数据库存放着所有设计人员可以共享的数据.项目数据库存放着本项目的的设计以及与设计有关的数据.专用数据库存放着设计人员尚未交付的设计对象. WHYMX 以联邦式 DBMS 为依托,其公共数据库和项目数据库存放在全局结点上,各专用数据库存放在每一

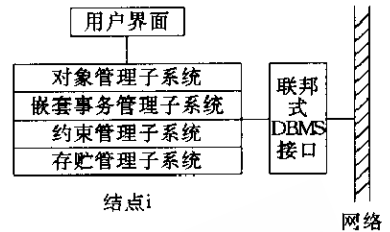


图1 WHYMX总体结构

设计人员的局部结点上. WHYMX 的总体结构如图 1 所示. 其中对象管理子系统主要分析与处理来自用户的消息, 包括类与实例的查询与更新、版本控制和授权等; 嵌套事务管理子系统对嵌套事务提供并发控制和恢复机制以保证事务处理的原子性和一致性; 约束管理子系统在设计对象修改之后采用增量方式或批量方式进行约束条件检查; 存贮管理子系统负责采用有效的存贮结构来组织、存贮结构相异的对象; 联邦式 DBMS 负责沟通局部结点之间、局部结点与全局结点之间的联系, 每一结点上的数据库可通过联邦接口与其它结点上的数据库进行通讯和协商, 它们具有较高的自治性并可以实现有选择、有条件的共享.

## 2 数据模型

在面向对象数据模型中, 现实世界任一实体都被模型化为一对象. 为减少信息冗余和管理方便, 将所有具有共性的对象构成的集合抽象为一个类, 类中的每一对象称为类的一个实例. 类定义包含属性、操作、约束条件和直接超类, 统称为类的特性. 属性是类定义的结构部分, 操作是类定义的行为部分. 类的所有实例有着相同的属性和操作并满足同样约束条件.

一个问题域中的类之间具有代数格关系. 类与类之间或具有聚合联系(即低层类的高层类的组成部分)或具有概括联系(即高层类是低层类的抽象). 在概括联系中, 高层类是低层类的超类, 低层类是高层类的子类. 一个类中的对象可以继承其所有超类部分或所有特性.

对象间通讯是通过对象名来传递消息进行的. 消息中只需指明要求对象执行的操作功能而无需说明操作过程. 当对象接收到消息后, 便去执行该对象所属类定义的操作, 当然所执行的操作程序也可根据需要再将某些消息发送给其它对象. 这种独特的数据组方式使得面向对象模型有以下缺点: (1) 对象管理复杂, 存取对象效率低; (2) 不能提供丰富的查询语言; (3) 实现基于集合的操作较难; (4) 难以处理复杂对象之间的连接操作; (5) 并发控制与恢复管理效率差; (6) 无法在任意域上建立索引. 而传统的关系数据模型以关系代数为基础, 对关系内或关系间的任何操作其结果仍是关系. 这种基于元组或面向值的操作给关系模型带来了许多优点, 能够消除上述面向对象模型的不足.

因此, 为了更好地描述和处理客观事物, 需要将面向对象模型与关系模型结合起来, 这两条途径: (1) 在关系模型基础上增加抽象数据类型、过程类型以及类格模式的继承性等. Postgres 就采纳了此方案<sup>[1]</sup>. 这样既完全保持了关系模型的重要特点, 可用于传统的以记录为基础的数据管理, 同时又能处理多媒体数据, 具有面向对象模型的许多优点; (2) 在面向对象模型基础上, 在对象描述层下面增加基于元组记录和面向值的操作, 这样就能避免前面列举的 6 条缺点, 又保留了面向对象模型的所有优点, WHYMX 采纳了这一方案.

WHYMX 数据模型把应用领域中所描述的对象类划分为下列几种: (1) 基本类, 包括实

型、布尔型和字符串型等；(2)复杂类；(3)集合对象类，包括集合、栈、队列、树和关系类等；(4)描述类集合自身特性的类，包括元类、版本类、历史数据类和索引类等；(5)非结构化对象类，包括函数、文本、声音和图象等。

### 3 对象管理

WHYMX 的对象管理主要包括类创建、类模式修改、对象的创建、查询及修改、版本控制、历史数据处理、用户授权和多媒体信息管理等<sup>[3-5]</sup>。

#### 3.1 类的创建

类的创建是由用户对象发送一个创建新类的消息给系统元类而实现的。消息的参数应包含下列内容：(1)新类的标识符(即类名)；(2)一组相关的属性，包括属性名及属性定义域(即属性所属的类名)；(3)一组操作程序，描述类内对象或对象中某些属性行为；(4)类内对象间或对象中某些属性必须满足的语义完整性约束条件集合；(5)新类可继承的超类集合。

一旦一新类被创建，系统将为该类自动加上一个动态标志属性项的集合，用于描述该类的有关动态控制信息，包括版本开关、类历史数据说明及通讯控制信息等。

在 WHYMX 模型中，子类可以对超类中的某些属性或操作重新定义。类定义中，约束条件集合定义为合取范式的条件表达式。

#### 3.2 类模式修改

类模式修改包括对类定义中属性集、操作程序集、约束条件集、超类集中元素的增删改以及整个类的删除，这是通过发送类修改的消息给系统元类进行的。

类中属性的增加，必须将类的所有对象增加此属性域并设其初始值为 NULL。类中属性的删除，必须将该类的所有对象的对应属性域删除，类中属性的修改需视具体情况再对其实例作相应的改变。所有属性的更新都将影响到其所有的子类。

类内约束条件的更新需使涉及到的所有对象满足新的约束条件。

类定义中的超类集的更新及类本身的删除可能会影响该类及相关类的前承/后承关系。为此需要做以下操作：(1)检查类格结构不出现环路或断路；(2)类格中属性、操作和约束条件的继承性的检查与更新以及相应对象的改变。

#### 3.3 对象的创建

在某一类内插入对象是通过向该类发送创建新对象的消息来完成的。对象内容与对象所在类的属性需满足类型匹配，并且符合类内所定义的约束条件。

#### 3.4 对象的查询

对象查询是通过向该对象发送消息完成的。在 WHYMX 中，类被看作是元类的对象，操作、函数和过程被看作是非结构化类的对象，因而有 4 种对象的查询：(1)通过类名查询类结构；(2)通过对象名查询对象或某些属性值，查询消息中的对象名可以是对象的别名，因此别名相同的对象可以一次查出；(3)通过类名查询若干类内的满足某些条件的对象或对象的属性值；(4)对类内某些操作项的查询。操作查询的实质就是执行程序，操作程序的参数来自类集中的若干对象或属性对象。这类查询也可用于执行非结构化类中的某些过程或函数对象，如打印文本、显示图像、播放声音及用户的交互对话的过程。

#### 3.5 视图类和对象体的处理

用户常常只对一些类中的部分属性或操作感兴趣,或对一些有关类中的部分对象感兴趣。为此,WHYMX 分别定义了视图类和对象体作为用户的视图。视图类的定义采用对象操作中的类结构查询语句建立视图类与内模式中若干类的映射联系。对象体的定义采用对象操作中的通过类名查询对象的语句。

### 3.6 版本管理

#### 3.6.1 对象的版本管理

消息形式: VERSION OBJECT object-name = new-object-name FROM class-name

其语义是将类 Class-name 中的对象 object-name 复制一份,并定义新对象的标识主名为 new-object-name。这样对 object-name 或 new-object-name 的一次修改都不会丢失对旧对象的保存。

#### 3.6.2 类的版本管理

消息形式: VERSION CLASS class-name; new-class-name

其含义是:(1)建一个名为 new-class-name 的空类;(2)将 new-class-name 类中超类域置为 class-name。这样 class-name 类将成为 new-class-name 类的直接超类。根据继承规则, class-name 类和 new-class-name 类将具有完全相同的类结构,但 new-class-name 类下面没有任何对象。因此,如果要保存类 class-name 下的对象版本,只能修改类 new-class-name 的结构,而保存类 class-name 作为旧的类版本。

### 3.7 历史数据的处理

在 WHYMX 模型中,具有 3 种表示时间和时间间隔的抽象类型。

## 4 存贮管理

WHYMX 的可扩展性之一是存贮结构及其算法的扩展,这就要求具有标准化界面的开放性体系结构,也就是找出一类易实现且有较好性能的算法,然后设计一个能支持这类算法的界面。WHYMX 存贮系统提供了文件对象、块、缓冲管理等标准化界面<sup>[6]</sup>,包括有创建或删除存贮对象、群聚对象和文件对象的原语等。

### 4.1 存贮对象

存贮对象是存贮管理的基本数据单元,它是一个连续的字节序列,其大小是可变化的。存贮对象可以是小对象(几百个字节)或大对象(上千个字节以上)。小对象被存放在单页上,而大对象可占有多页。存贮对象标识符 OID 可表示为〈页号,槽号〉。小对象的 OID 是一个指向磁盘上该存贮对象的指针。大对象的 OID 是一个指向其“大对象头”的指针,该大对象头与其它大对象头和小对象数据一起被存放在一页上,且该大对象头含有指向该大对象表示所涉及那些页的指针。当小对象经增、改使之增长到一页放不下时,存贮系统自动地把它转变成一个大对象,并把它的大对象头保留在它原先所在的位置。图 2 显示了大对象的结构。

从概念上看,一个大对象是一个连续的字节序列,而从物理上看,它在磁盘上被表示为一个关于它内部字节位置的 B 树索引和一组叶块(数据块)。一个大对象内的全部叶结点仅含有对象的数据,为提高访问效率,在所有叶结点中设置一链域,该域存放其右兄弟叶结点的首页号。这样在顺序扫描大对象内的若干相邻叶结点时,可以减少 I/O 次数。

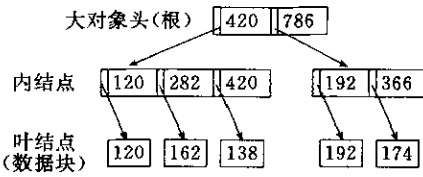


图2 大对象示意图

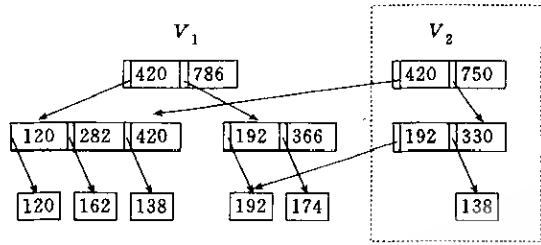


图3 一个大对象的两个版本

### 4.2 存贮对象的版本

对一个版本化对象修改时,将大对象头(如果是大对象)或整个对象(如果是小对象)复制到磁盘的同一页或邻页上,对此修改产生一个新版本.

图3说明了大对象的版本管理方法.版本 $V_2$ 是删除了版本 $V_1$ 最后36个字节而派生出来的, $V_2$ 共享 $V_1$ 中所有未被改变的结点,它只拥有每个被修改过的结点的副本.一个大对象的新版本含有一份从根结点到某个新叶结点这一路径上的结点拷贝.由于这样的路径长度通常较短,并且相对于实际的数据页(即叶结点)来说,内结点页数又较少,所以用这种方法对大对象进行版本控制的开销比较小.

### 4.3 群集对象

群集对象把一个复杂对象层次的所有对象聚集在一块,它是一组存贮对象的集合.群集对象的表示有些类似于大对象.它的对象标识符CID是指向它的根页(即群集对象头)的指针.群集对象用一棵B+树作索引,该索引逻辑上可以作为群集复杂对象的所有对象数据页的一种手段,大大提高了访问群集对象的效率.

创建一个群集对象时,首先生成一个对象头,然后插入它的构成对象.群集的全部对象物理上应尽可能相邻.删除群集对象中的一对象时,只需把它从所在页上消掉即可.若要删除整个群集对象,首先要删去群集对象索引中所列出的全部对象,然后删去其索引.

### 4.4 文件对象

文件对象把相关的对象存放在一起,它是存贮对象和群集对象的集合.文件对象的对象标识符FID是一个指向其根页(即文件对象头)的指针.存贮对象、群集对象和文件对象的识别是依据它们各自的对象头中的标识位而区分的.文件对象也采用B+树作索引.

创建一个文件对象时,首先为其分配一对象头,尔后,如果要在该文件对象中创建存贮对象或群集对象,则调用相应的创建原语.从文件对象中删除一小对象,只需把它从所驻留的页上消掉即可.若要删除大对象或群集对象,必须递归地删除其所有页的数据.若要删除整个文件对象,则必须首先删除该文件对象中的全部对象,然后删除它的索引.

## 5 事务管理

WHYMX的事务模型为嵌套结构<sup>[7,8]</sup>.嵌套事务由若干项目事务由一组协作事务组成,项目事务间是相互独立的.每一项目事务由一组协作事务组成,每一协作事务是委托者一承包者事务的一个层次,每一委托者一承包者事务由数据库操作和承包者事务组成或仅由数据库操作(即单级事务)组成.

嵌套事务允许一事务把它的任务分解成几个子任务,每一子事务又可产生新的子事务.

由于子事务可以并发执行,并且以此作为恢复单位,所以嵌套事务可望获得较高的并发度和对故障的较高的恢复度。

### 5.1 并发控制

在嵌套事务的每一级上,必须保证原子性和偏序,实施一致性约束。

项目事务在项目数据库上采用封锁法进行并发控制.项目事务对一对象施加封锁的目的,并不是要直接访问它,而是要让其子事务去访问该对象,并限制其它项目事务对它的访问.项目事务最终是由许多并发执行的单级事务组成的,在单级事务这一级上采用经过扩展的封锁法进行并发控制.鉴于面向对象数据模型的特点,WHYMX 支持分层封锁、类格封锁和复杂对象封锁,我们在文献[5,9,10]中,提出了类格与复杂对象的并发控制策略与协议。

每一委托者/承包者事务在专用数据库中拥有自己的专用空间、委托者空间和承包者空间(见图 4)。

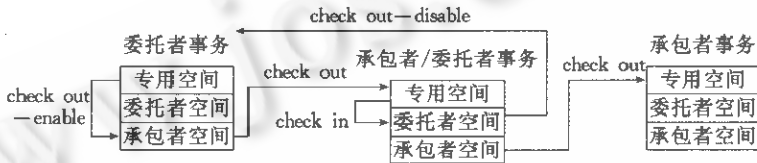


图4 转包工作过程

为保证数据一致性,规定检出检入协议如下。(1)在第 1 阶段,委托者事务将转包工作所需的数据从专用空间移到它的承包者空间(checkout—enable),承包者事务再从委托者事务的承包者空间检出数据送入自己的专用空间(check—in);(2)在第 2 阶段,承包者事务将处理过的结果数据从自己的专用空间检入自己的委托者空间(check—in),然后再由委托者事务取回到它的专用空间(checkout—disable)。一旦承包者事务 check—in 之后,就不能再次从委托者事务的承包者空间(check out)数据。

为了保证承包者事务对于委托者事务的原子性,委托者—承包者事务层次应符合偏序关系.我们提出了一种虚拟时间戳算法,该算法为每一委托者/承包者事务指定了时间戳,从而使委托者/承包者事务在执行中保持偏序.这一算法保证了每一对象被委托者/承包者事务按正确的次序处理。

### 5.2 恢复管理

在 CIMS、CAD 环境中,由于设计人员需要与机器交互以及设计人员之间的相互协作,使得嵌套事务要延续很长时间.在这期间,可能出现事务故障、系统故障、介质故障和通讯故障.由于经历时间长,事务应能人为地中止,留待以后继续执行.因此,不可能把整个嵌套事务作为恢复的最小单位.WHYMX 允许嵌套事务分若干次保存一些中间结果,挂起某一子事务甚至整个嵌套事务,当需要时返回到嵌套事务的某一步.只有这样,才能允许嵌套事务分几段时间来完成,也才能在出故障时,不至于丢失中间结果,使设计工作前功尽弃。

WHYMX 采用日志技术和检查点方法来解决嵌套事务的恢复问题.一方面由系统每隔一定时间设置一检查点,另一方面在查询语言中提供这样的语句,由设计人员视需要在事务中间设立检查点.当因各种故障或资源不足而使事务中止时,允许系统返回到前面任一检查点,以避免嵌套事务重新从头开始执行,从而大大减少了恢复工作量。

## 6 结束语

WHYMX 是用 C 语言编写,在基于 UNIX BSD 4.2 开发环境的 3 台 SUN 工作站上实现的. WHYMX 具有以下重要特征:(1)用面向对象的方法研制,结构较好;(2)实现了复杂对象的共享、持续存贮和并发控制;(3)实现了部分基于值或集合的操作,查询操作语言丰富;(4)提供菜单类、文本类和图形编辑器等用户接口;(5)修改和扩展系统方便;(6)与关系模型联系较深,使用简便;(7)支持 CAD、CIMS 领域的嵌套事务管理.当然,该系统仍有一些不尽完善之处,例如,用谱系对象来管理类的版本不太适宜,版本合并、嵌套事务的恢复管理尚不尽人意,模式动态更新使系统效率降低等,这些有待进一步研究解决.

## 参考文献

- 1 Stonebrake M *et al.* The postures model. Proc. of the 13th Int. Conf. on VLDB, 1987.
- 2 Darrell Woelk. Multimedia information management in an object-oriented database system. VLDB, 1987.
- 3 杨传钧,王能斌,徐庆云等. OODB 模型. 全国第九届数据库学术会议论文集,1990.
- 4 黄玮,王能斌,毛兆余等. WHYMX 系统查询机制初探. 全国第九届数据库学术会议论文集,1990.
- 5 徐庆云,王能斌,陈钢. 复杂对象及其并发控制. 软件学报,1993,4(5):46-50.
- 6 毛兆余,王能斌,黄玮等. WHYMX 存贮系统中的对象和文件管理. 全国第九届数据库学术会议论文集,1990.
- 7 徐庆云,王能斌. WHYMX 的嵌套事务管理. 计算机应用,1991,9(5):45-49.
- 8 徐庆云,王能斌,杨传钧等. WHYMX 单级事务的设计与实现. 计算机研究与发展,1992,29(3):1-6.
- 9 Xu Qingyun *et al.* Concurrency control of an object-oriented database system suited to CAD/CIMS. 2nd International Conference on CAD&CA, 1991.
- 10 徐庆云,王能斌,杨传钧等. 面向对象数据库系统的并发控制. 全国第九届数据库学术会议论文集,1990.

## RESEARCH AND PRACTICE OF AN OBJECT-ORIENTED DATABASE SYSTEM

Xu Qingyun

Wang Nengbin

(East China Bureau of Petroleum Geology, Nanjing 210031) (Southeast University, Nanjing 210096)

Yang Chuanjun

(East China Institute of Computing Technology, Shanghai 200233)

Mao Zhaoyu

Huang Wei

(Beijing Laser Institute, Beijing 100011) (Beijing Institute of Information Engineering, Beijing 100101)

**Abstract** This paper introduces WHYMX object-oriented data base system suitable for a certain of domains such as CIMS, CAD, AI and GIS. WHYMX gives the theoretical description of aggregation and generalization standardization, presents the concurrence control strategy, the storage management strategy and the object management mechanism of complex objects. WHYMX supports nested transaction and management of large storage object. WHYMX adds the operations of tuple-based and value-oriented in the object description. Therefore, it has many advantages of the object-oriented data model and the relation data model.

**Key words** Object-oriented database, CIMS, architecture, data model, object management, transaction management, storage management.