

一种基于图形数据 OODBMS 的设计方法⁽²⁰⁾

马晓琴

(青海铝业有限公司, 西宁 810100)

摘要:介绍了一种利用面向对象方法设计和实现的基于图形数据的面向对象数据管理(OODBMS)。讨论了 OODBMS 的结构和功能, 定义了基于图形的数据模型和数据模式以及对象管理技术。

关键词:图形; 对象; 类; 数据模型; 数据模式

中图分类号:TP315

文献标识码:B

文章编号:1008-858X(1999)04-0076-05

近年来, 面向对象的方法以其独特的优点引起了计算机科学界的广泛重视。用这种方法描述的数据模型, 不仅可以描述数据的结构特征, 还能描述它的行为特征。因此, 面向对象的方法特别适合数据结构具有动态性和不完整性的数据类型, 而在工程环境中, 最常见的就是此类数据。对于这样的问题, 利用以往的事务数据管理方法来管理上述数据, 由于他们在数据结构上的差异, 可能会导致难以描述或根本就无法描述的结果。因此, 研究新的数据管理方法, 研究和开发新一代数据库管理系统, 是非常必要的。

OODBMS 是用 Borlan C++ 和面向对象方法实现的。本文不仅介绍 OODBMS 的实现技术, 还介绍基于图形数据的数据管理方法。OODBMS 包括界面部分: 面向终端用户的交互式查询界面、面向应用程序员的数据库宿主语言器; 数据模式定义子系统部分: 数据模式定义、修改、一致性约束; 对象管理子系统部分: 对象管理器、类型管理器和方法管理器; 存取子系统部分: 数据库的存储组织和有效的管理。数据管理包括如何从图形数据抽象数据模型的问题。

1 数据模型的描述

在 OODBMS 中, 能够完成任何图形的存储、适用、管理问题。所谓图形, 不外乎是由点、线段、矩形、圆、弧、任意曲线经过有限次诸如放大、缩小、移动、裁剪、添色等操作得到的, 其结构如图 1 所示:

图 1 既体现了图形分类的直接性, 也体现了图形之间的继承性, 图形如此分类则定义了它的范围, “规则图形”继承“图形”的属性和操作; “圆”继承“规则图形”的属性的操作; …… , 则定义其继承性。各子图的属性只是属于这一类图形最基本的数据, 如位置、画法、颜色等属于与超图共有的属性可通过继承得到。一般认为图形基本数据表示其结构特征, 面对其操作则表示了图形的行为特征。如果定义了图形结构和对它的操作, 就定义了图形的数据模型和此

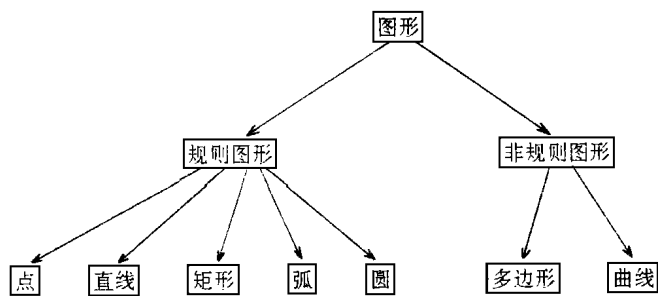


图 1 图形的结构
Fig.1 Structure of graphics

DMBS 能力。图形结构和其操作的归纳表示式如下所示：

〈图形〉：：=〈规则图形〉⋈〈非规则图形〉

〈规则图形〉：：=〈点〉⋈〈直线〉⋈〈矩形〉⋈〈圆〉⋈〈弧〉

〈非规则图形〉：：=〈曲线〉⋈〈多边形〉

〈操作〉：：=〈放大〉⋈〈缩小〉⋈〈移动〉⋈〈旋转〉⋈〈裁剪〉⋈〈添色〉

2 数据模式的建立

定义了图形的数据结构以及 OODBMS 对图形的操作功能后,数据模型即确定。下面的工作是设计 ODDL。ODDL 包括定义图形的数据结构和检验方法,以及数据模式的动态修改。ODDL 的定义有多种方法可以引用。一种方法是设计全新的对象数据描述语言;另一种方法是如果系统提供了对象数据的定义和描述,则利用之,然后补充对象数据描述语言的其他部分。本文采用了第二种方式,数据结构定义借助了 Borland C++ 数据。如图 2 所示。

```
class 类名 {
    结构特征:
        私有[...]
        共有[...]

    行为特征:
        私有[...]
        共有[...]
}

end class
```

图 2 Borland C++ 中 class 的结构
Fig.2 Structure of a class in borland C++

由图 2 看出,Borland C++中只能完成对数据结构和操作的描述,却不能完整地描述数据模型,特别是无法描述对数据的检验。所以,ODDL 必须补充对数据结构属性和操作的封装、一致性约束功能;补充定义对数据结构的动态修改功能。封装包括两部分:属性和操作。对于图形数据,应能从数据模型得出,其属性和操作方法大致有一定的范围。因此,OODBMS 设计时定义了一些属性和操作,初始封装时只需要将属性名字添上,有些属性可通过人机交互时输入。如果此属性是私有、共有或从超类上继承的,要在属性表中的特征一栏添清楚。操作的封装只需指出操作名和特征即可。另外,利用封装表可随时检验数据模式的完整性。如某些操作不存在,ODDL 提供定义新操作的接口,使封装达到动态封装的目标。封装功能是通过建立封装表实现的。如图 3 所示。

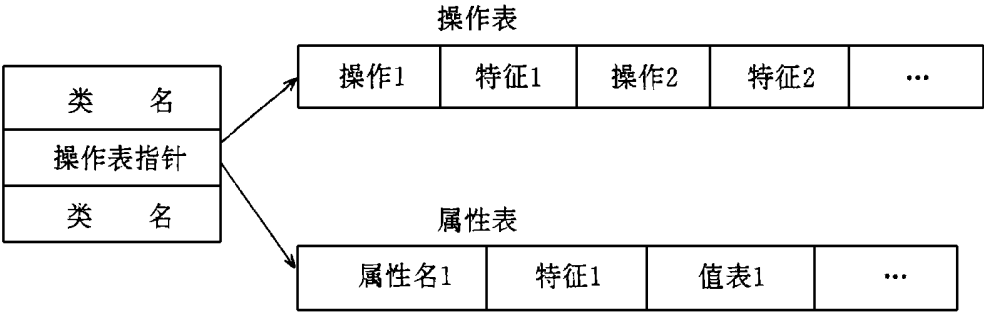


图 3 ODDL 的封装表
Fig. 3 Packaging table of ODDL

ODDL 中关于数据模式的一致性约束管理,包括一致性约束和对不一致性的处理。其中一致性约束实现比较复杂。在工程应用中,图形数据相互间关系比事务数据复杂得多,必须满足种类繁多的一致性约束条件。这些一致性约束条件,需要建立约束条件表来管理。此表的结构与封装表大致相同。此外,还需建立灵活的约束检查功能和约束条件删除功能。这是因为在图形设计过程中,有些中间结果可能会违背某些约束条件,但随设计的深入,这些约束条件又会逐步得到满足。这种复杂的一致约束检查,不可能让 OODBMS 自动完成,所以必须有上述接口。用户可以在他认为适当的时候做一致性检查。ODDL 中对出现的暂时不一致情况,只给出警告信息,允许继续操作数据。

ODDL 除定义数据模式外,还包括定义对数据模式动态变换。数据模式的动态变换包括如下功能:

- (1) 派生新子类:新子类继承它的超类的属性操作,同时拥有自身定义的属性和操作。
- (2) 删除子类:删除当前子类及隶属于它的所有下级子类。如果消除根类,则这个数据模式消失。
- (3) 修改子类:当前子类中的属性及操作修改后,除私有部分外,它所有下级子类如果继承了这些属性和操作,则自动修改。

3 对象管理

OODBMS 的对象管理,给用户提供了一个方便灵活的界面。用户仅用简单的命令,即可完成诸如此类的定义、对象的插入、删除和修改等工作。这些功能包括类型管理器、对象管理器和方法管理器。其结构如图 4 所示。

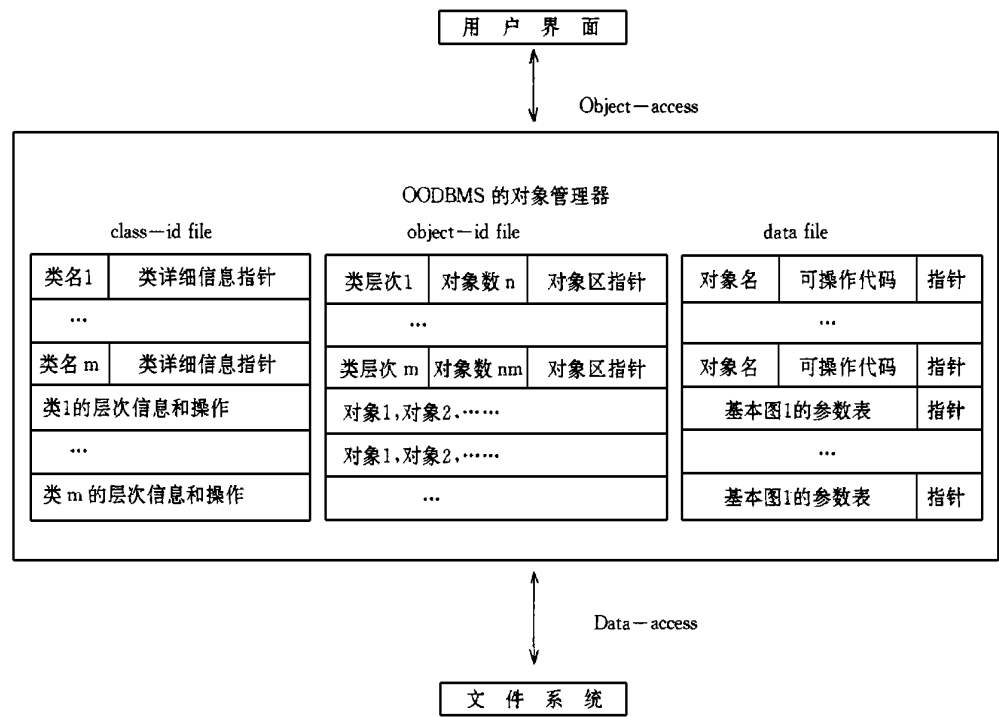


图 4 OODBMS 的对象管理功能结构

Fig. 4 Structure of object managing function of OODBMS

Class-id file 和 Object-id file 属于系统级文件,由系统建立和管理。设立这两个文件的目的是为了对类和对象进行有效的管理。当用户定义一个新类(子模式)时,Class-id file 文件打开,并且 Class-count 加 1,存入用户对该类详细描述信息,如属性、操作和继承,并做相应检查。当用户输入一幅图(一个对象)时,系统先将 Object-id file 文件打开,将此类的对象计数器 Object-count 加 1,添加对象信息,并重新检查一致性检验等工作,然后才能在 data file 中写入数据,关于类的修改、删除等可依次类推。

4 结论

本文从设计一个新的面向对象数据库管理系统的角度出发,讨论了对面向对象数据库技术中必须涉及的一些问题,并用 C++ 语言予以实现。虽然本文所介绍的 OODBMS 只是一个实验系统,但所阐述的一些方法是面向对象数据库技术的共同问题。

参 考 文 献

[1]李昭原.数据库技术新进展[M].北京:清华大学出版社.

A Design Method for an OODBMS Based on Graphics and Data

MA Xiaoqin

(*Qinghai Aluminum Limited, Xining 810100*)

Abstract

This paper illustrates an OODBMS system based on graphics and data, which is designed and realized by using the object-oriented method. Discussions were made for the structure and functions of the OODBMS system. And definitions were made for a graphics-based data model and data modes, as well as the object managing technology.

Key words: Graphics, Object, Class, Data models, Data Modes.