

OBMS

IDKE

面向对象
白

⑧

计算机科学1992Vol.19 No.2

数据库

OBMS/IDKE的动态模式管理*)

44-49

王宗斌 王 珊 武志文

TP314.13

(中国人民大学数据与知识工程研究所, 北京100872)

摘

要

本文从各数据模型下的模式管理问题出发,讨论了面向对象数据库系统OBMS/IDKE (Object-Base Management System/Institute of Data and Knowledge Engineering) 的数据模式管理所面临和待解决的复杂语义问题,并阐述了该系统模式管理的外部界面和内部实现。

数据模型是数据库的逻辑结构,有其严格定义的概念集合。数据模型分别用数据结构、数据操作和完整性约束精确地描述数据库系统的静态特性、动态特性和数据完整性约束条件。

最早的层次和网状数据模型为了解决应用文件系统管理数据所产生的诸如数据缺乏独立性和安全性、数据冗余过大等问题,作为它们的核心管理之一的模式管理,其中心任务在于保证对数据库中的数据进行独立的、完整的和有效的逻辑描述。为了简化问题、突出重点,其一般策略是:数据模式一次成型,在对应的数据库生命周期内不再改变。

但是,随着基于网状和层次数据模型的数据库系统的广泛应用,其模式管理所用的数据模式一次成型、不能改动的策略对数据库设计显得过分苛刻。它一方面要求设计师花很长时间来设计数据库以尽早地发现错误和不当,以降低设计成本;另一方面一旦在数据库运行中发现需要修改模式错误,就意味着停用旧数据库、重装新数据库,对用

户来说这个环境不太友好。

七十年代初出现了关系数据库。关系模型非常简明,它用一张二维表来描述实体及实体之间的联系。其数据模式管理的任务除了对数据库中的数据进行描述外,还允许对所定义的表进行修改。这样就简化了数据库的设计工作,为设计师提供了一个更加友好的数据库使用环境,在一定程度上也支持原型开发。然而,关系模型的简明性也使得它描述的语义范围不可能很宽,无法满足日益增多的数据库应用领域(多介质的OA及CAD/CAM, CASE等)提出的新要求。顺应于形势,在传统数据模型的基础上又发展出了面向对象的数据模型。按照[1]中的观点,一个面向对象的数据模型必须满足二个标准:首先它必须是一个数据库管理系统,即具备持久性、辅存管理、并发性、恢复和即席查询的功能;其次它还必须是一个面向对象的系统,即具备下列八个特性:复杂对象、对象标识、封装性、类型或类、继承性、重载和滞后联编、可扩充性及计算完备性。显然面向对象的数据模型可以分别在二个基础上

*)国家自然科学基金资助项目

00 44:11

建立,一是在现有的传统数据库上通过扩充OO(Object-Oriented,面向对象)特征得到(代表性系统有在INGRES关系数据库上扩充得到的POSTGRES系统),另一个则是在OO环境下建立数据库特性(代表性系统有以Smalltalk为基础的Gemstone等)。从这两个标准可以看出,OO数据模型的语义较之传统的数据模型要丰富得多,如包含在继承性中的ISA语义、由复杂对象决定的ISP语义等都是一般关系模型难以表达的。在OO模型下的数据模式管理通常要比传统模型下的模式管理任务复杂得多,至少表现在:要描述的语义更加复杂且模式动态变化。

一、OBMS/IDKE的OO数据模型及数据模式管理

面向对象的数据系统OBMS/IDKE是研究用的实验模型,是在分析了国际上目前影响较大的OODBS以后,在C++的基础上建立起来的。OBMS/IDKE的设计力图满足前述二个标准,即5个DBMS特征和8个OO特征。具体说来,OBMS/IDKE的数据模型除了传统数据模型具备的基本特征外,还具备如下的OO特征:对象标识(包括对象外部标识),复杂对象,对象的封装性,类、类型(C++提供的基本类型、结构及扩充的set类型)和快照(Snapshot;用来存放查询结果),多继承,及利用C++实现的重载、滞后联编,可扩充性和计算的完备性。对整个模型的详细说明请参见[17,18]。

在OBMS/IDKE模型中,数据模式管理的任务十分复杂。它要描述并管理如下的语义关联:

1. ISA关联(继承关联)

ISA是OO系统类之间的一种最基本的关联。它表明了二个类之间这样一种关系:其中一个类(设为A)是另一个类(设为B)的子类(或类B是类A的超类)。记作类A ISA类B。类A继承了类B的所有公用属性和动作。

在OBMS/IDKE中,每个对象库都有一

个公共的超类(直接的或间接的)SUPER-CLASS,它是这个对象库中类之间构成的层次关系的根(或顶点)。

2. ISP关联(合成关联)

ISP表示了二个类之间这样一种关系:其中一个类的实例对象是另一个类的实例对象的一部分。它是OO模型支持复杂对象的必然结果。所谓复杂对象就是允许一个对象的属性取另一个对象为值或者说允许一个类在定义中把一个属性的值域定义成另一个类(在我们的模型中把前者称为主体类,后者称为成员类)。这种ISP关联有时又称为聚集(Aggregation)。

设M为主体类,A为成员类,则M的对象Om有四种把A的对象Oa取为自己属性值的方式:

- a)排它方式:Oa只作为Om的一部分;
- b)共享方式:Oa是Om的一部分,也可以是其它对象的一部分;
- c)依赖方式:Oa依赖于Om而存在;
- d)独立方式:Oa独立于Om而存在。

前二种方式(a,b)和后二种方式(c,d)进行两两组合构成四种有意义的合作语义,即排它且依赖的,排它且独立的,共享且依赖的,共享且独立的。

3. R关联(引用关联)

R关联指的是一个类使用另一个非超类的共用属性或动作的情况。因为被引用类的某些变化会影响引用类对信息的正确使用,所以这种语义关联必须为数据模式管理程序所控制。

4. N关联(数量关联)

N关联用来表示非超/子类关系的二个类的对象之间的数量关系。像学生和课程二个类的对象之间可以定义这种数量关联。这种关联的好处在于一方面保留了传统数据库中的习惯表示,另一方面保证了二个类的对象之间在不具备ISA和ISP关联但具备某种数量关联的情况下的数据的完整性。

二、OBMS/IDKE动态模式管理的外部界面和内部实现

1. 内部实现

(1) **模式定义** 模式定义的任务主要是用数据字典和二个系统文件来记录对象库中所有类的结构、类之间的语义关联及函数体的源代码和目标代码。数据字典是模式管理的核心。

在OBMS/IDKE中,数据字典在构造上是一个特殊的类,记作DD。这样设计的益处主要在于系统所管理的内容(不论是用户的还是系统本身的)都以类或对象的统一形式出现,从而使得一些处理也可以统一进行。

系统中的每个对象库对应一个数据字典或DD类的一个实例对象集合。类DD的第一个实例对象是关于DD本身的描述,其它的每一个对应,或对应于一个用户的类或结构(相当于无动作的类),或对应于一个系统的类或结构(如用户权限表等)。

用户每定义一个新的类就相当于在系统中增加一个类为DD的对象。这个对象的属性值包括了对这个新类的结构描述(类名,属性个数,属性名,动作个数,动作名,…)及这个类与其它类之间的语义关联。

(2) **模式更新** 模式更新指的是在对象库的运行过程中增加新的类或对现行类的属性或动作进行修改的活动。模式管理程序必须保证更新操作后对象库中的数据完整性。这通常通过一组模式不变式和更新规则来实现。

(i) 模式不变式(Schema Invariants)

在OBMS/IDKE系统中,我们构造了一组模式不变式,对于类定义的任何变化,对应的对象库的类层次结构都必须保证这一组性质的成立。

I1: 类层次不变式

对象库的类层次是一个带根(一个特殊的结点SUPER-CLASS,其它所有类都是它的直接或间接子类)的有向无环图。

I2: 名区别不变式

系统中每一个类的名字必须是可区分的,在整个系统中不允许出现同名的类。每个类中属性名和动作名(不论是自己定义的还是继承来的)也必须是可区分的。在一个类定义中,除了按规定说明为overload(重载)外,不允许出现同名的属性和动作。

I3: 全继承不变式

一个类必须继承其超类的所有属性和动作,除非(a)全继承导致了上述I2不成立或(b)多继承时多个超类的属性发生冲突,这时对多个超类按规定顺序进行处理。

(ii) **更新规则** 模式不变式对保证模式更新操作的正确性起指导作用。具体的是用几组规则来对模式进行维护。

第一组 名冲突解决规则

规则1.1 如果定义一个类时,类名与系统中已定义的类名发生冲突,则按出错处理,要求重新命名。

规则1.2 如果一个类的属性或动作的命名与该类中其他属性或动作发生名冲突,则按重名错处理,要求重新命名。

规则1.3 如果一个类的属性或动作与从超类继承来的属性或动作发生名冲突,则用该类定义的属性或动作阻止超类的同名属性或动作的作用并向向下传递继承。

规则1.4 如果一个类有多个超类,这些超类间的属性或动作有相同的名字,则按超类的顺序来继承,继承在顺序链前面的超类的属性或动作。

第二组 类层次修改规则

规则2.1 (节点增加规则) 定义一个新的类,加入到类层次中时,规定其只能成为类层次的叶子,即要求这个类的超类必须或只有OBJECT或用户在定义该类之前已经定义。若该类还含有合成关联(即它是一个复杂类),那么其成员类必须是已经定义了的,或者与该类一起定义,这时先生成成员类的定义。

规则2.2 (节点删除规则) 删除一个已

定义的类，要求只能从类层次的叶子上进行。若一个类要从类层次中删除，必须满足下列条件：

a: 该类无子类，若存在子类则首先要删除其子类。

b: 该类无关联，若存在与其它之间的关联则先删除关联。

类一旦删除后，所有属于该类的对象被自动清除，空间随之释放。

规则2.3 (增加一条边) 如果一个类A要成为另一个类B的超类，则要求该关联的加入不能在类层次中出现环。A作为B的最后一个超类（即从超类链的尾部增加）。

规则2.4 (删除一条边) 如果A是B的唯一超类，A从B的超类集中删除后，B将立即成为A的每个超类的子类，B的超类次序同A一样。

规则2.5 (增加关联规则) 如果已定义的类之间要建立关联，则要求这两个类的相应属性没有数据，增加关联的属性是自定义属性。否则操作不允许进行。

规则2.6 (删除关联规则) 如果对于定义有关联的类，要删除这个关联，可以直接进行。

第三组 类修改规则

规则3.1 (增加属性规则) 如果对已经定义的类增加新属性，则要求该属性与该类的其它属性不发生名冲突。否则按规则1.2处理。若该属性与其超类的属性发生名冲突，则按规则1.3处理。

规则3.2 (增加动作规则) 如果对已定义的类增加新的动作，则要求该动作不能与该类的其它动作发生名冲突。否则按规则1.2处理。若该动作与其超类的动作发生名冲突，则按规则1.3处理。

规则3.3 (性质修改规则) 对于类的属性和动作，允许将PRIVATE (私有)型的改为PUBLIC (公有)型的。反之不允许。对一个超类的继承若为PRIVATE型的，可以改为PUBLIC型。反之不允许。

第四组 性质传播规则

规则4.1 当一个类中的属性或动作的性质发生变化后，这种变化将传播到该类的所有继承其属性或动作的子类。除非这些性质在子类中已经重新定义了。

规则4.2 对于名字的修改和新增加的属性或动作，仅传播到那些不发生冲突的子类，如果发生了冲突则按名冲突规则解决。

2. 外部界面

在OBMS/IDKE中，我们把模式的更新操作分成两大类：一类是修改类层次图中的节点及边，包括增、删结点和边；另一类是修改类层次图中的节点的内容，包括增加属性、操作及一些性质。

用户可以在OBMS/IDKE的系统环境下，用专用的ODPL语言中的DDL语句来对模式进行更新。

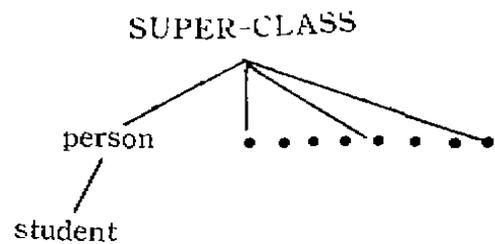
例1 在类层次中插入一个在根OBJECT下的新类person。

```
OBMS/IDKE> $ CLASS person
```

例2 在类层次中插入一个新类student，且student ISA person。

```
OBMS/IDKE> $ CLASS student:PUBLIC person
```

这条语句执行后，从系统内部的类层次上看有如下结构：



例3 我们还可以在根SUPER-CLASS下插入另一个新类course

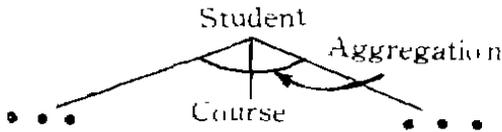
```
OBMS/IDKE> $ CLASS course
```

例4 在例2的student的定义中，我们除了可以说明student和类person之间具有ISA关联外，还可以说明course和student之间具有ISP关联。

```
OBMS/IDKE> $ CLASS student:PUBLIC person
```

```
{
  ...
  $ course crs-selected{ };
  ...
}
```

即course和student在内部有如下的合成关联 (Aggregation) 图:



例5 左类student可以增加一个新的属性: 不及格的课程

```
OBMS/IDKE> $ADDATTR student
```

```
{
  $ course crs-fail{ };
}
```

每当用户进行一次模式更新, 系统内部自动调用相应的规则进行检查以保证三大不变式的成立。

三、结束语

OBMS/IDKE是为了更加全面、深入地研究面向对象数据库而建立的一个实验模型。在这个实验模型的动态模式管理方面参照了国际上目前一些较成功的面向对象数据库管理系统的相应内容, 如Gemstone, ORION, POSTGRES等。与这些系统相比, 我们的实验模型有自己的独到之处, 尤其在语义关联及更新操作给这些关联产生的影响方面作了更为深入的讨论和考虑。OBMS/IDKE系统支持复杂对象、对象标识、对象的封装性、多继承、类型及类和快照、重载及滞后联编、可扩充性、计算的完备性; 支持继承语义 (ISA) 关联、合成语义 (ISP或Aggregation) 关联、数量语义关联和引用语义关联; 支持对类层次中的结点、边和结点的内容进行动态更新。

整个研究工作尚未结束, 关于更新方面的一些特殊技术和策略, 其使用的效果还需作进一步的实验。我们期待有关问题的讨论能在以后与大家见面。

参考书目

- [1] Atkinson, M. etc., The Object-Oriented Database System Manifesto, The Proceedings of 1st Conference on Deductive and Object-Oriented Databases, pp.40-57 (中文译载《计算机科学》1990.3.王宗斌, 王珊译)
- [2] Banerjee, J. etc., Data Model Issues for Object-Oriented Applications, ACM Transactions on Office Information System Vol.5, No.1, 1987.1 pp3-26
- [3] Bancilhon, F. etc., The Design and Implementation of O₂, an Object-Oriented Database System, Proceedings of OODBSII Workshop, 1988.9
- [4] Carey J. etc., Object and File Management in the EXODUS Extensible Database, VLDB 86 pp 91-100
- [5] Enrich, H.D. etc., Abstract Object Types for Databases, Technique Report
- [6] Enrich, H.D. etc. Objects, Object Types and Object Identification, Tech. Report
- [7] Enrich, H.D. etc., Algebraic Implementation of Objects over Objects, Tech. Report
- [8] Fishman, D.H. etc., Overview of Iris DBMS, Object-Oriented Concepts, Databases, and Applications pp219-250
- [9] Jackson M.S., Tutorial on Object-Oriented Databases, Information and Software Technology, Vol.33, No.1 1991.1-2 pp4-12
- [10] Maier, D., Development of an Object-Oriented DBMS, ACM OOPSLA/86
- [11] ONTOS Object Database Programmer's Guide, 1990.2
- [12] Rowe, L.A. etc., The Postgres

请

计

请

OBMS 面向对象
IDKE 数据库

存储

⑨

计算机科学1992Vol.19No.2

OBMS/IDKE中对象存储/存取
子系统的若干关键问题*)

49-55

王珊 任永杰[✓] 唐元昌

TP311.13

(中国人民大学数据与知识工程研究所, 北京100872)

摘 要

The storage/access subsystem is one of the kernel module in OBMS/IDKE. It should support the object-oriented data model PUC, store and manage massive, shared and permanent object effectively and efficiently. In this paper, we discuss some important issues and main ideas in designing and implementing the storage management, buffer management and access path management of OBMS/IDKE.

一个数据库系统称为是面向对象的数据库系统首先应满足以下两个条件:

- (1) 支持一个面向对象数据模型内核。
- (2) 支持传统数据库所有数据库成分。

当然为了与面向对象数据模型内核的语义一致, 必须作适当的语义扩充和修改。

从文[1]图1中可以看到OBMS/IDKE是一个面向对象数据库系统。其体系结构包括了传统的DBMS几乎所有的模块, 只是在OO环境下有了新的内容, 有些则是传统的DBMS没有的。OBMS/IDKE核心可以大致分为三部分:

- | | |
|--|---|
| <p>Data Model, VLDB 87, pp83-96</p> <p>[13] edited by Stonebracker, M. etc., The POSTGRES PAPERS, 25 June 1987</p> <p>[14] Weiser S.P. etc., OZ+ An Object Oriented Database System, Object-Oriented Concepts, Databases, and Applications, pp309-337</p> <p>[15] Kim, W., Features of the ORION Object-Oriented Database System, Object-Oriented Concepts, Databases and Applications, pp251-382</p> <p>[16] Bretl, R., The GemStone Data</p> | <p>Management System, Object-Oriented Concepts, Database, and Applications, pp283-308</p> <p>[17] 王珊、任永杰、唐元昌, OBMS/IDKE对象存储/存取系统的若干关键问题, 本期《计算机科学》</p> <p>[18] 唐元昌、王珊, OBMS/IDKE: 模型及其无缝的面向对象的编程语言, 本期《计算机科学》</p> <p>[19] 萨师焯, 王珊, 数据库系统概论(第二版), 高教出版社</p> <p>[20] X3/SPARC/DBSSG/OOBTG FINAL REPORT 12-SEPT-1991</p> |
|--|---|

*)国家自然科学基金资助项目