

信息资源字典环境下 OODB 的模式修改

周 姪 楼荣生 罗文化

G202

(复旦大学计算机科学系 上海 200433)

摘 要 The schema modification of OODB is important subject in designing and implementing software systems. IRDS (Information Resource Dictionary System) is an important schema of information management, it emphasizes meta data's management which is not restricted by data model. This paper present a approach with IRDS to represent object-oriented data model and to modify schema of OODB.

关键词 OODB, Information Resource Dictionary (IRD), Schema modification.

信息资源字典系统 (Information Resource Dictionary System, IRDS) 是信息资源管理的一个重要模式, 在国际上受到极大重视, 各发达国家纷纷出台各自的标准, 并由国际标准化组织 (ISO) 制定了 ANSI IRDS 框架标准。信息资源字典强调元数据管理不受数据模型限制, 其构造思想运用于关系数据库的表示, 可使数据库结构修改与数据修改一样方便地进行, IRDS 环境下的应用程序在视图上操作, IRD 内部数据库中数据的修改与应用程序无关, 实现了数据结构对应用程序的独立性, 保证了数据的正确性、安全性、完整性, 便于数据库的维护。本文对 OODB 引入信息资源字典的构造思想, 探讨面向对象数据库表示的基本方法及在 IRDS 环境下 OODB 的模式修改。应用 IRDS 的构造思想讨论 OODB 的模式修改是一种新思路, 使 OODB 的模式修改易于进行。

1 信息资源字典系统 (IRDS)

信息资源字典 IRD 是数据字典 (DD) 的扩充, 不但可表示和存储用户数据库中的数据, 而且还可表示和存储信息资源范围内有关软硬件系统的资料, IRDS 管理了整个软硬件系统的全部资源, IRD 为四级结构: (1) 基本级; (2) 模式级 (字典定义级 DDL); (3) 字典级 (DL); (4) 应用级。相邻两级构成了一个级对 (基本级和模式级构成顶级对, 称之为 IRD 定义级对; 模式级与字典级构成中级对, 称之为 IRD

级对; 字典级与应用级构成低级对, 称之为应用级对)。上级信息描述了下级信息的类型, 即一个级对的上级是对象的类型, 下级是类型的实例。

(1) 基本级。建立用于定义字典的基本概念, 如数据基本类型, 属性类型的特征等, 其内容在系统中完全固定, 基本级的实例存储在模式级中, 按 1986 年发表的美国家标准, 基本级由九个文件组成, 分别是数据类型、非空限制、唯一性限制、实现者定义界限、DDL 文件、DDL 属性、DDL 组合属性、DDL 组合属性内的属性、引用限制等文件形式, 详见文 [1], 这些文件定义了存储在 IRD 模式级数据的类型。

(2) 模式级。定义了 IRD 中实例的类型, 即定义了 IRD 的结构及有关控制机制。模式级与基本级构成顶级对 (即 IRD 定义级对), 全部文件名都登录在基本级的“DDL 文件”中, 文件的属性登录在基本级的“DDL 属性”文件中, 文件之间的相互关系登录在“引用限制”文件中。模式级是基本级所定义数据类型的实例。同时, 又定义了字典级的数据结构类型, 其数据在字典级中实例化, 也就是模式级与字典级构成中级对 (即 IRD 级对)。IRD 标准为 IRD 级对提供了 IRD 的功能设施, 包括在 IRD 中插入、修改或删除对象、类、属性等, 对于面向对象的数据库, 模式级由以下几个文件组成:

a. DDL-OBJECT (元对象名, 录入日期, 元对象号, 元对象类型)。元对象类型: = 元类/类/属性/方法。注: 这里的元对象有别于 OODB 中的对象。该文



件建立元对象名与元对象号之间的对应关系,对于在其他文件中出现的元对象,可用元对象号代替元对象名。

b. CLASS-TYPE(类号,类型),类型:=系统定义/数据/外部引用,该文件对每一被管理的类,确定其类型,登录编号。

c. ATTR-DOMAIN(属性类型号,属性域),属性域:=字符串/整数/实数/文本/日期/时间/类,该文件对每一个属性,定义其属性域,登录属性编号。

d. METHOD-REFER-ATTR(方法类型号,属性类型号),描述某一方法类型引用的一个属性类型。

e. CLASS-CONTAIN-ATTR(类号,属性类型号),描述某一个类包含的属性类型。

f. CLASS-CONTAIN-METHOD(类号,方法类型号),描述某一个类包含的方法类型。

g. SUBCLASS(类号,子类号),描述某个类的子类。

h. SUPERCLASS(类号,超类号),描述某个类的超类。

用以上文件的数据可以方便地表示 OODB 中的类级次结构,使 OODB 的模式修改也易于进行。

(3)字典级(DL),是存储 IRD 的数据的数据级,登录了组成用户数据库的全部对象,其对象类型规定在模式级中,下面给出字典级的文件,文件表示格式与模式级文件的表示格式相同。

a. CLASS(类实例名,实例号,类号,录入日期),记录类及其实例的对应关系,给每个类的每个实例编号即实例号。

b. METHOD(方法号,方法类型号,实例号),记录方法及其引用方法的实例之间的对应关系,为每个实例引用的每个方法给定一个编号即方法号。

c. ATTR(属性号,属性类型号,应用,实例号,方法号),指出每个属性应用的实例号或方法号,并给每个属性编号即属性号。

d. COMPOSITION-ATTR(复合属性号,复合属性类型号,复合类级次号,应用实例号),复合属性即类复合级次中的属性,其定义域是一个类,该文件记录复合属性与类复合级次号(即类号)及应用实例号的对应关系,并给定复合属性的编号。

e. ATTR-VALUE(属性号,属性值),指出属性

号和属性值的对应关系。

文件 a,b,c,d 规定了 OODB 中全部的类、属性、方法,文件 e 中的“应用”是指该属性的应用对象类型;文件 e 是属性与值的对应表,不必是一一对应。一个属性可对应多个值,即可表示多值属性,字典级各文件的数据实例组成应用级,是现实世界中的数据,IRDS 不为该级提供任何支持设施,对该级的操作由 DBMS 提供。

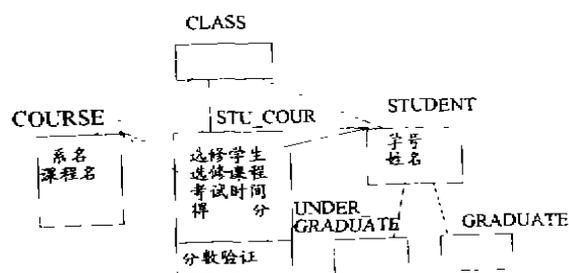
2 IRD 表示面向对象模型

2.1 面向对象数据模型

面向对象模型中最基本的数据结构是对象类型,对象包括实例变量(Instance variable)和方法(Method),实例变量用以描述对象的状态,即对象属性值的集合,而方法是定义在对象实例变量上的操作,实例变量和方法被封装(Encapsulation)在一起,每个对象都有一个唯一的标识符,对象标识由系统自动生成,在整个系统内保持唯一不变,具有相同的实例变量(即属性)和方法的所有对象组合在一起构成一个类(Class),此时对象称为该类的实例(Instance),类的属性定义域可以是基本类型,如整型,实型,字符串等,也可是具有自己的属性和方法的一个类,构成类的嵌套结构,即类复合层次(Class-composition hierarchy)。

一个系统中所有的类按超类子类关系组成一个有根的有向无环图,叫类层次(Class hierarchy),相似的类概括而成超类,一个类细化为其子类,一个类从其类层次中直接或间接超类继承(Inherit)所有的属性和方法,同样,一个类的所有属性和方法被其子类继承,一个类的所有实例也是其所有超类的逻辑实例。

下图是学生成绩管理数据库的面向对象模型。



2.2 OODB 时间模型的表示

使用信息资源字典系统(IRDS)中的模式级可以很方便地表示 OODB 的类级次和类复合级次结构。根据图1,填写模式级的文件,文件填写的内容即构成字典级。

1. DDL-OBJECT

元对象名	录入日期	元对象号	元对象类型
CLASS	95.06.10	1	元类
STUDENT	95.06.10	2	类
UNDER-GRADUATE	95.06.10	3	类
GRADUATE	95.06.10	4	类
COURSE	95.06.10	5	类
STUT-COUR	95.06.10	6	类
学号	95.06.10	7	属性
姓名	95.06.10	8	属性
系名	95.06.10	9	属性
课程名	95.06.10	10	属性
选修学生	95.06.10	11	属性
选修课程	95.06.10	12	属性
考试时间	95.06.10	13	属性
得分	95.06.10	14	属性
分数验证	95.06.10	15	方法

2. CLASS-TYPE

类号 类型

- 1 系统定义
- 2 数据
- 3 数据
- 4 数据
- 5 数据
- 6 数据

3. ATTR-DOMAIN

属性类型号 属性域

- 7 整数
- 8 字符串
- 9 字符串
- 10 字符串
- 11 类
- 12 类
- 13 日期型
- 14 整数

4. METHOD-REFER-ATTR

方法类型号 引用属性类型号

- 14 13

5. CLASS-CONTAIN-ATTR

类号 属性类型号

- 2 7,8
- 5 9,10
- 6 11,12,13,14

6. CLASS-CONTAIN-METHOD

类号 方法类型号

- 6 14

7. SUBCLASS

类号 子类号

- 1 2,5,6
- 2 3,4

8. SUPERCLASS

类号 子类号

- 2 1
- 3 2
- 4 2
- 5 1
- 6 1

2.3 实例的表示

IRDS 中,每个级对的上级定义了下级数据的类型,而下级是上级类型定义的实例化,正如 IRD 级对中模式级的文件内容构成字典级一样,在应用级对中填写字典级的文件,文件内容即构成应用级。既然使用 IRDS 中模式级的文件方便地表示了面向对象数据库中类级次结构,那么类的实例也可在字典级文件中方便地表示出来,以构成应用级。对上述模式级文件内容,利用用户数据填写字典级各文件以组成应用级内容。(以下字典级文件的部分内容)

1. CLASS

类实例名	实例号	类号	当人日期
01	1	2	95.06.10
02	2	2	95.06.10
03	3	2	95.06.10
04	4	5	95.06.10
05	5	5	95.06.10
06	6	6	95.06.10
07	7	6	95.06.10

文件中的“类实例名”由系统给出,这里假定为

01,02,...,将通过实例属性值具体化。

2. METHOD

方法号	方法类型号	实例号
1	15	6
2	15	7

3. ATTR

属性号	属性类型号	应用	应用实例号	应用方法号
1	7	类	1	
2	7	类	2	
3	8	类	1	
4	8	类	2	
5	9	类	4	
6	10	类	4	
7	13	类	6	
8	14	类	6	
9	14	方法		1

4. COMPOSITION-ATTR

复合属性号	复合属性类型号	复合类级次号	应用实例号
1	11	2	6
2	12	5	6

5. ATTR-VALUE

属性号	属性值
1	0001
2	0002
3	李伟
4	田英
5	计算机
6	数学
7	95.10.20
8	98

利用关系代数运算,自然联接上述字典级各文件,再作适当选择和投影,可生成各种用户视图。从上述应用级的数据中,可以很方便地作查询。如:计算机系学生李伟(学号0001),选修了数学,得分为98分,考试日期是95年10月20日,等等。

3 模式修改

3.1 分类

面向对象的数据库系统能够动态定义与修改数据库的结构,其类型结构的可扩展性和可修改性是OODB的重要性质。在我们前面定义的IRDS环境下,使OODB的扩展和修改更容易进行。在实际应

用领域中,用户需要的数据模式修改,反映在类型结构上,有以下两种:

1. 类定义的改变,包括属性,方法,类名的改变。

①改变属性。类中增加新属性;从类中减少属性;改变属性名;改变属性域;改变属性缺省值。

②改变方法。对类增加新的方法;减少方法;改变方法名;改变方法代码。

③改变类名。

2. 类级次结构的改变,包括增加或减少一个类及在两个类间改变继承关系即超类/子类关系。①增加或减少一个类。②在两个类间改变超类/子类关系,使类S成为类C的超类;从类C的超类中去掉类S;创建新类C作为n个类的超类:(*)将类C划分为n个新类;(*)n个类合并为一个类。

3.2 软变化和硬变化

由于类层次上的继承,一个类的改变会影响其它类及其实例,也影响该类的实例。例如,如果删除一个类的属性,则所有的类实例中该属性值必须被逻辑删除,该属性也必须从继承了该属性的直接或间接子类及其实例中删除。一般地,一个类的变化逻辑上要求修改数据库,即修改类实例和其子类的实例。在IRDS中,也就是需对IRD中字典级文件进行修改。但是,对许多逻辑上要求修改数据库的模式变化,可以避免实际对数据库的修改。如,从类中删除一个属性,不必立即删除该类属性值,象这样的模式变化,我们可称之为软变化。而有些模式变化要避免数据库修改是不可能的,也是不现实的。如,一个类被分成两个或多个类,类实例为了反映它们在新类中的成员关系必须重新组织,象这样的变化称为硬变化。因此,我们可以对软变化和硬变化作如下定义:

软变化是不要求已存在实例改变它们所属的类,避免修改数据库的模式变化,在前述的模式修改分类中,除(*)外,其它都是软变化。硬变化是迫使已存在实例改变它们所属的类,不可能避免修改数据库的模式变化。前述模式修改分类中,如(*)所示,为硬变化。软变化中的模式修改只需在模式级文件中进行,而硬变化的模式修改还需修改字典级文件。

3.3 模式修改

下面,说明在我们定义的IRDS环境下的各类模式修改。

1. 软变化。利用关系代数运算,对模式级的文件作修改即可作软变化。以下面几种模式修改为例来

说明 IRDS 环境下的模式修改原理。

①类 C 中增加一个新属性 V

参数格式:(类名 c, 属性类型 V, 属性域)

首先将类名 c 转换成相应的类号:

类号 $c = \text{Num}(\text{类名 } c)$

$= \Pi$ 元对象号 s 元对象名 = 类名 c
(DDL-OBJECT)

如果新属性类型 V 不在 DDL-OBJECT 文件中, 则需对 DDL-OBJECT, ATTR-DOMAIN, CLASS-CONTAIN-ATTR 三个文件增加元组。

DDL-OBJECT \leftarrow (属性类型名 V, 录入日期, 新元对象号, 属性)

其中新元对象号 = 最大元对象号 + 1 = 新元对象类型号 V

ATTR-DOMAIN \leftarrow (新属性类型号 V, 新属性域)

CLASS-CONTAIN-ATTR \leftarrow (类型 C, 新属性类型号 V)

如果新属性类型 V 已在 DDL-OBJECT 中, 则将属性类型名 V 转换成属性类型号:

属性类型号 $V = \text{Num2}(\text{属性类型名 } V) = \Pi$ 元对象号 s 元对象名 = 属性类型号 (DDL-OBJECT)

当查询文件 CLASS-CONTAIN-ATTR, 若属性 V 已是类 C 上局部定义的属性, 则新属性 V 被拒绝。若 V 为类 C 从其超类上继承的属性 (查询 CLASS-SUPERCLASS ∞ CLASS-CONTAIN-ATTR 即得类 C 的继承属性) 或 V 根本不是 C 属性, 则 CLASS-CONTAIN-ATTR \leftarrow (类号 c, 新属性类型号 V)

②从类 C 中删除一个属性 V

参数格式:(类名 c, 属性 V)

从类 C 中删除一个属性, 该属性只能是该类上定义的属性, 继承属性不能被删除。

若属性类型 V 仅在类 C 中定义, 则须从 DDL-OBJECT, ATTR-DOMAIN, CLASS-CONTAIN-ATTR 三个文件中删除。首先确定要修改的类号和属性类型号。

类号 $c = \text{Num1}(\text{类名 } c)$, 属性类型号 $V = \text{Num2}(\text{属性类型名 } V)$

DDL-OBJECT- $\{x \in \text{DDL-OBJECT}/x.$ 元对象号 = 属性类型号 V}

ATTR-DOMAIN- $\{x \in \text{ATTR-DOMAIN}/x.$ 属性类型号 = 属性类型号 V}

CLASS-CONTAIN-ATTR- $\{x \in \text{CLASS-CONTAIN-ATTR}/x.$ 类号 = 类号 c, x. 属性类型号 = 属性类型号 V}

③增加一个新类 C

如果增加的新类 C, 未指定其超类, 则元类为其超类, 修改 DDL-OBJECT, CLASS-TYPE, SUBCLASS 及 SUPERCLASS 四个文件。

DDL-OBJECT \leftarrow (类名 c, 录入日期, 新元对象号, 类)

其中新元对象号 = 最大元对象号 + 1 = 类号 c;

CLASS-TYPE \leftarrow (类号 c, 数据);

SUBCLASS \leftarrow (1, 类号 c);

SUPERCLASS \leftarrow (类号 c, 1)

如果类 C 被指定了多个超类, 参数格式为:(类 C, 超类 C_1, C_2, \dots, C_n), 同样也是修改上述四个文件, 只是 SUBCLASS 和 SUPERCLASS 分别需要增加 n 个元组。

2. 硬变化, 例如: OODB 中将一个类划分为 n 个子类就是一种硬变化。如果类 C 水平划分成子类 C_1, C_2, \dots, C_n , 其划分条件是针对每个类实例给出的, 如果类 C 的一个实例满足子类 C_i 的划分条件, 则该实例的类就变成 C_i , 否则仍为该实例的类。如果类 C 实例的类变成 C_i , 则实例的对象标识也须改变。因此, 在做硬变化的模式修改时, 除了需做字典定义级文件的修改外, 还需修改相应字典级的文件, 如文件 CLASS, METHOD, ATTR 等。

结束语 本文在信息资源字典环境下讨论了 OODB 的表示和模式修改的问题。IRDS 本身是用关系模型表示的。在 IRDS 中, 面向对象数据库的结构信息, 包括类级次和类复合级次都以关系数据形式存放在模式级文件内, 而模式级文件与字典级文件一样, 可独立地进行处理, 从而使得 OODB 的模式修改易于进行。

参考文献

- [1] 楼荣生, 信息资源字典系统环境下数据库的表示, 计算机信息与工程, 1991年第1期
- [2] Kim W., Introduction to object-oriented database, The MIT Press, 1990
- [3] 冯玲、冯玉才, 面向对象数据库的模式进化与版本管理, 第十二届全国数据库学术会议论文集, 1994
- [4] ISO/IEC 10027, Information Technology - Information Resource Dictionary System (IRDS) framework
- [5] 黄金明, 信息资源字典系统 (IRDS) 的解剖与实现, 复旦大学硕士论文, 1989