

关于建立对象模型的探讨

李味辛 (石油管道局信息中心)

管理信息系统是一个复杂的大系统。用面向对象的方法来建立管理信息系统的模型就象用地球仪模仿地球一样,是一种最贴切最直观地描写现实世界的方法。与迄今为止的其他方法论相比,面向对象的方法论最接近人类长期以来认识世界的习惯,最容易学习和掌握。

一、对象建模的三种模型

一般来说,一个典型的软件肯定包含了这三种性质:它使用数据结构(对象模型),顺序执行操作(动态模型),并且完成值的变化(功能模型)。面向对象的方法论用对象模型、动态模型、功能模型三种模型来描述现实世界。

1.对象模型是通过描述系统中的对象、对象间的关系,标识类中的对象的属性和操作表组织系统的静态结构,是三种模型中最重要的模型,它为动态模型和功能模型提供了实质性框架。

2.动态模型描述与时间操作和次序有关的系统属性,它关心的是控制,从事件和状态的角度描述系统的数据结构。

3.功能模型从值和功能的角描述系统的计算结构,它不考虑系统干什么或如何干。功能模型描述了对象模型中的对象唤醒和动态模型行为唤醒的功能。

这三种模型描述的是系统的不同方面,但各种模型之间又互相联系,对象模型描述了动态模型和功能模型中操作的数据结构,对象模型中的事件及功能模型中的功能。由于用途和目的编译系统重在功能模型。而数据库的目的不是数据的变化,而是将数据有条理地存放管理起来,因此甚至可以完全不做功能模型。

模型设计是识别对象,对象的归类,详化对象的属性和操作定义,处理对象与对象之间的关联,这样一个从简单概括到详细复杂的反复过程。如果沿用数据模型设计分为需要分析、概念设计、逻辑设计、物理设计阶段的说法,那么,三种模型适应于设计的全过程。

以往在谈到 MIS 系统设计时,常把分析现实世界搞得很抽象,实际上,现有报表系统是长期企业管理经验的积累,是新系统的基础,是识别对象的主要根据。在原有系统的基础上,新系统要考虑的无非以下两点:①由于市场变更对企业及其管理的新要求;②由于使用计算机这一高新技术而带来的管理手段的革命而引起的变革。

二、面向对象建模的有关概念

1.对象和对象标识(Object and OID)

现实世界在计算机世界的映射称为对象。

对象具有一个可以唯一地识别自身的标识,称之为对象标识,可记为 OID。特别要注意的是,对象标识和该对象被看成完全是同一个东西。OID 是由系统自动生成的,使用者不能直接指定或修改。

对象是属性和操作的封装(encapsulation)。

对象::={DS,MS,};

DS={D1,D2,...,DN}描述对象的 N 个属性的数据结构;

MS={M1,M2,...MN}对象的行为即操作的集合。

操作在程序中是一系列过程(或者叫子程序)的集合,操作由消息来激活。每个操作都有唯一的消息格式。对象的消息,也称为对外接口,消息由对象名称、操作名称(程序中就是过程的名字)和施行某操作的参数三部分组成。面向对象的系统的运行,依靠各对象间的信息发送和反馈来完成。

由于封装特性,面向对象的方法论把程序的层次结构和数据结构统一起来,设计为单一的对象类的层次结构,形成了一个区别于传统的方法论的主要特点:

2.类和例(class and instance)

具有共同特性的对象,被抽象为类。其中的对象称为这个类的例。类本身也是对象。针对例的操作在类中定义,叫做例操作,简称操作。比如有一个类「职工」的例

操作“建立”,当调用它时,就可以生成一个新的对象,比如:“张三,38...”在这里,“建立”是例操作。针对类的操作,叫做类操作。类操作在后面将要提到的元类中定义。

在数据库中除了属性以外,还有一些专门为编程需要设置的变量。记述例的性质的变量称为例变量。例变量在类中定义。例变量类似于关系数据库中的字段名。记述类的性质的变量称为类变量。类变量在元类中定义。

3.超类和子类(SuperClass and SubClass)

超类和子类可用来表现一般和个别,全部和部分的关系。某个类的例变量、类变量、例操作、类操作在其子类中都可以被继承和重新定义。

4.定义域 (Domain)、主类 (OwnerClass)、从属类 (OwnedClass)

定义域即属性的取值范围。例如:“姓名”的取值范围是“8 字节字符串”,年龄的取值范围是“3 位整数”。而如果要表现类「工资」的属性「职工编号」的取值范围必须是类「职工」中已登记的“例”的集合这一事实,则可以定义类「工资」的属性「职工编号」的“定义域”为类「职工」。实际上,在面向对象的模型中“8 字节字符串”、“3 位整数”都用对象类来表示,在概念上也是统一的。

另外,在上例中,如果类「工资」不作为一个独立的类存在,那么把它叫做主类「职工」的从属类。一个对象加上它的全部从属对象,构成一个有主从关系的复杂对象。从属类随着主类的生成而生成,消灭而消灭。如果把关系数据模型的表看作是一个二维空间,那么,在对象模型中,通过类的主从关系,可以构成具有复杂构造的多维空间的数据模型。通过定义域还可以建立类似于关系换型中通过外来键建立的参照关系。

5.元类(MetaObjectClass)

元类是一种更抽象的类,反之,一切类都是元类的例。它类似于关系模型的数据字典。因为元类的例就是应用数据模型的类,所以,元类的操作,即针对元类的例的操作,也就是针对整个应用数据模型的所有类的操作。它不仅包括对各属性的值修改、编辑功能,还包括对整个应用系统的,从生成到消灭的整个生命周期的管理。

三、元类的模型

元类的模型实际上是数据模型的核心问题。1992 年日本规格委员会通过了称为 JDMF(JSA Data

Modling Facility)的数据模型标准。它继承了关系数据模型 SQL 的优点,是一规范化的,能为多种目的提供数据共享的面向对象的数据模型。本文参考该标准,根据国内的企业管理信息系统设计实践,以及系统设计方法论的调研,提出如下元类,同大家一起探讨。

元类 META0 有 7 个子类 META1—META7,它们之间的关联如图 1 中的箭头所示。META2、META3 是 META1 的从属类。META1—3 记录了应用模型中的类。META6 记录超类—子类关系,META7 记录主从关系。META1、2、6、7 建立了整个系统的类与类之间的关联。META4 保存关于用户的信息,META5 保存用户对类的所有权信息。

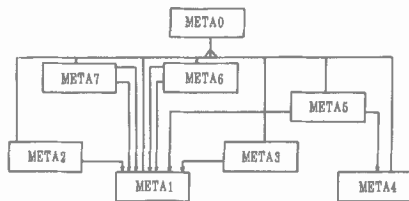


图 1

META1:「对象类」类(Class「ObjectClass」),该类的属性有:汉字类名、类名代码、更新周期、类特性等。

META2:「属性」类(Class「Attribute」)记录并管理了整个系统的属性。该类的属性有:汉字属性名、属性代码、定义域、数据类型、字段长度、小数点位数、唯一性、非空值、其它约束条件、所属类等。其中的定义域与数据类型、字段长度、小数点位数从概念上讲有所重复,这里是为了照顾现有的数据库平台。

META3:「操作」类(Class「Method」)该类以应用系统的操作为例。该类的属性有:汉字操作名、操作代码、备注、所属类等。

META4:「用户」类(Class「User」)管理了整个系统的用户的情况,其属性有:用户代码、用户名、负责人、联系人、电话、上级对口部门等。

META5:「用户—对象类关系」类(class「User—ObjectClass Relation」)该类记录了每个用户所能使用的类的信息,类似于用户权限的记录,其属性有:用户代码、类名代码、建立、使用等。

从面向对象的观点来看,应用系统是一组状态变化着的对象,因而,用户权限也是针对对象类而言,一旦声明拥有对象类,就同时拥有它的数据和操作,从而避免了

以往的过程设计时的重复。

META6:「超类-子类关系」类(class「Super-Sub Relation」)。该类的属性有:超类代码、子类代码。

META7:「主类-从类关系」类(class「Owner and Owned Relation」)

该类的属性有:主类代码、从属类代码。凡在 META7 中登录的具有主从关系的类作为一具统一体生成和消灭。

META0 的操作有:

- (1)META1-7 的例生成、修改和编辑;
- (2)为某个用户打印可调用的类(包括属性定义词典和操作定义词典)的信息;
- (3)各种查询;
- (4)检查唯一性、定义域等约束条件;
- (5)统一修改类名代码、用户代码等。

上述元类适用于需求分析、概念设计、逻辑设计阶段。正如前面提到过的,在这个阶段,还没有加进那些“属性”以外的,仅用于编程的例变量,也没有生成、管理、消灭应用系统的操作。META0 仅仅从概念上搭起了整个应用系统的骨架,并成为用户与系统分析人员的桥梁。

笔者编写了“MIS 应用数据模型逻辑设计辅助程序”,可完成上述功能。若在此基础上,根据实际的软硬件平台,加上必要例变量,并进一步加以完善,即可成为能够生成和管理系统的元类。

四、关联的处理及设计的统一性

值得特别提出的是系统设计本身的规范化。由于对客观世界的认识和表现方法的不同,即使是同一事物,其设计结果也会因人而异。这不仅大大降低了软件生产的效率,更重要的是不利于信息共享和软件对环境的适应性。因此,模型设计也要象机械设计那样,必须注意设计本身的规范化和一致性。以下提出几种关联的设计,以供讨论。

1.表示每个国家都有一个首都这一事件的 1 对 1 的关联。设计为:定义类「城市」和类「国家」,并把类「城市」作为类「国家」的属性“首都”的“定义域”。

2.要表示一个人可以在一个以上的公司里工作,而每个公司都有许多职工的多对多的关联。建议设立一个类,用来记录那两个类之间的关系。如图 2 所示,可以定

义三个类:类「人」、类「公司」和类「雇佣关系」,并把类「人」作为类「雇佣关系」的属性“职工”的“定义域”,类「公司」作为类「雇佣关系」的属性“公司”的“定义域”。

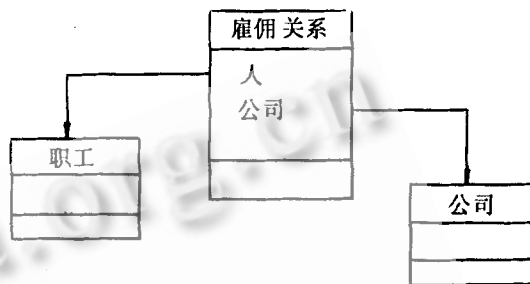


图 2 多对多的关联的处理方案

3.假设要做一个微机仓库的模型,仓库中有成套设备,有零部件,而且成套设备是由零、部件组成的。是否采用如图 3 所示模型:

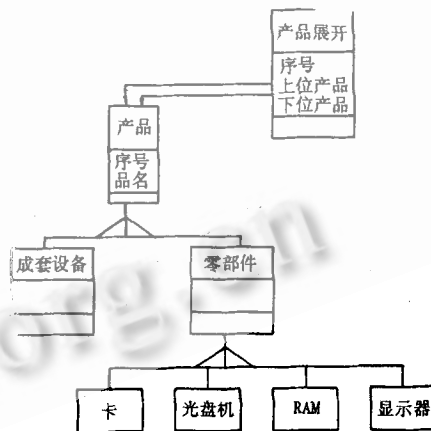


图 3

(1)用超类子类关系来处理产品的分类,超类「产品」的属性“序号”、“品名”等都可被它的所有子类继承。例如成套设备 AST486、66 配置 1,由 1024M 的 RAM、双速 CD-ROM、... 等组成。AST486、66、RAM、CD-ROM、...等的有关信息分别记录在相应的类中。

(2)设立类「产品展开」来记录产品的组成。类「产品的展开」的属性“上位产品”、“下位产品”的定义域是类「产品」,亦即“上位产品”、“下位产品”中记录的是各类产品 OID。

以上建模方法应用了面向对象的概念,但在关系型的数据库语言中也同样可以实现。