计算机科学1993 Vol.20 No.5

对象标识的语义及构成

李天柱 （河北大学 河北保定071002）

摘 要

Based on the three database schema in ANSI/X3/SPARC, the full object identity, including objective object identity, conceptual object identity, internal object identity, is introduced. The objective object identity is a special primary key, the internal object identity corresponds to the object identity oid in usual object-oriented systems. The object equality based on the full object identity is also discussed. The fall object identity can be used not only to identify the objects in user interface, but also to manage the objects in computer systems, makes object identity further perfect. And it is also available in combining object-oriented model with value-oriented model.

1. 引言

在面向对象模型中，对象标识是一个重要概念，文〔3，9〕均将其做为O-O模型的必备特征。在NF^2关系模型中，也有人引入了对象标识(元组标识和关系标识)。现在人们之所以重视对象标识，是因为在描述嵌套结构的复杂事物、处理数据共享、描述历史数据和版本特征等方面，对象标识具有重要作用。文〔10〕强调了由系统管理、独立于内容、独立于结构、独立于地址的强标识；文〔15〕指出了对对象标识的错误理解，提出了怎样使强标识与唯一访问（unique reference）协同存在的问题；文〔7〕指出，在扩展的关系模型中，只有不存在不变的key的情况下，才引入对象标识；文〔9〕指出，标识的概念包括比较运算、逻辑标识、对象标识、对象生成。究竟对象标识的语义是什么，怎么构成，作用是什么，是值得深入研究的重要问题。本文主要就面向复杂对象数据库中的对象标识进行进一步讨论。

2. 不同级别模式下的对象及对象标识

数据库中的数据模型是对客观事物的描

李天柱 副教授. 主要研究方向为数据库与知识库. 收到日期：92-09-19.

〔4〕 Tripathi, A.P., An Overview of the Nexus Distributed Operating System Design, IEEE Trans. on SE, Vol.15, No.6 1989

〔5〕 Agha, G., An Overview of Actor Languages, SIGPLAN Notices, Vol.21, No.10 1986

〔6〕 Kim, W. and et al., Integrating an Object-Oriented Programming System with a Database System, in «Research Foundation in Object-Oriented and Semantic Database Systems», ed. by A. F. Córdenas and D. McLeod, Prentice-Hall, 1990

〔7〕 Bronnenberg, Win J.H.J. and et al., DOOM: A Decentralized Object-Oriented Machine, IEEE Micro, Vol.7 No.5, 1987

〔8〕 d'Arielli, L., A Message Management System for Personal Computers, IBM System Journal, Vol. 28, No.3, 1989

述模型。面向对象的数据模型（O-O模型）是近年来发展迅速、倍受青睐的数据模型。按照著名的ANSI/X3/SPARC，描述客观事物的数据模式分为概念模式（全局概念模式和子概念模式）及内模式。概念模式既要能正确描述客观事物，又要便于转换成内模式，以在计算机中实现。相应地，对于不同的模式级别，应有相应的对象及对象标识的概念。

2.1 客观对象及其标识

客观世界中的对象是客观事物，不妨叫做**客观对象**。客观对象不依赖于人们怎么描述它而客观存在，相互区别。至少，不同客观对象的空间位置不同。

人们区分客观对象是靠其属性或空间位置，这往往较繁锁，故人们对客观对象赋于名字或编号，以简单地标识和区分客观对象。这些名字或编号往往有一定的适用范围，如人名、职工号等。当建数据库时，最好是能在所涉及的范围内，人工地给每一客观对象赋于一个所有用户都承认的、不重复的、不变的名字或编号，如零件号，身份证号等，我们称它为**客观对象标识**，记为o-id。客观对象标识也是客观对象的一个属性。

客观对象是不断变化的。一个零件会损坏，一个人的工作和职务会变化，图纸会被修改。这样，一个客观对象在历史上会有不同的状态。为了区分一个客观对象的不同历史状态，人们给它们赋于历史标记，时间戳或版本号，我们称之为客观对象的历史标识。这些历史标识是客观对象标识的一部分，也是客观对象的属性。

一般通常所说的客观对象是指客观对象的现时状态，省略其历史标识。

2.2 概念对象及其标识

概念模式下的对象是对客观对象以某种形式的描述，不妨称其为**概念对象**。O-O模型中的型（type，用类实现[9]）是一种概念模式，类中的对象应该是概念对象。概念对象应包含所描述的客观对象的客观对象标识o-id作为其一个特殊属性。

但对客观对象的描述方式可以是多种多样的：所取的属性集可以不同，可以具有不同的嵌套结构等。因此，不同的概念对象可以是描述同一客观对象的，即客观对象与描述它的概念对象是1对多的。为了便于区分概念对象，可以给每一概念对象赋于一个标识代号，不妨叫做**概念对象标识**，记之为c-id。o-id和c-id之间也是1对多的。

2.3 内对象及其标识

内模式下的对象是概念对象在计算机中的象，称之为**内对象**。内对象可重复拷贝，因此概念对象与内对象间是1对多的。内对象的重复拷贝的存储地址不同，但对应于同一概念对象，描述同一客观对象。为区分内对象，可引入**内对象标识**，记为i-id。可见，内对象标识是与存储地址有关的，每个i-id直接或间接地对应一个存储地址。c-id与i-id之间是1对多的。

基本的元对象，如数字、字符串等，结构简单，是构成其它对象的最基本成分；其本身是抽象的，没有确定的语义，可用其值本身相互区分，不必引入新的对象标识。需要引入对象标识的是构造型对象。通常的O-O模型就是这么做的。

此后，对象一词泛指客观对象、概念对象、内对象。

2.4 三种标识间的关系及其特性

由上面讨论知，客观对象$\xrightarrow{1\,n}$概念对象$\xrightarrow{1\,n}$内对象；相应地有o-id$\xrightarrow{1\,n}$c-id$\xrightarrow{1\,n}$i-id。计算机内存储的是内对象，它应同时包含上述三种标识。内对象可表达为〈id，value〉，其中id＝〈o-id，c-id，i-id〉叫做**全标识**。对二个内对象，o-id同$\rightleftarrows$c-id同$\rightleftarrows$i-id同。

三种标识都具有唯一性。

显然，客观对象标识独立于值、独立于结构、独立于地址，具有文[10]中所述强标识的特性。它标志着概念对象、内对象描述那个客观对象。

概念对象标识是独立于值的，修改对象

属性值时，它保持不变；但它不独立于结构，当概念模式结构改变时，其下的概念对象就变成了另外的概念对象。概念对象标识也是独立于地址的。

内对象标识是独立于值的，也可以是独立于结构的（同一存储空间可存放不同结构的内容）；但它不独立于地址，每一个i-id直接或间接地对应于一个存储地址，当i-id不同时，其所对应的地址也不同。

3. 对象相等

这儿的对象相等指对象个体区分和判别，不考虑文[13]中所述的属性组意义上的任意相等（arbitrary-equal）。

设o_1和o_2是二个内对象。由于前述的o-id，c-id，i-id之间依次的1对多关系，这三种标识间的合法组合有下面四种：

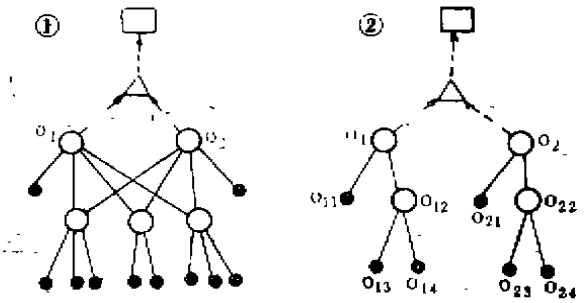
(1) $o_1.o\text{-}id=o_2.o\text{-}id$，$o_1.c\text{-}id=o_2.c\text{-}id$，$o_1.i\text{-}id=o_2.i\text{-}id$；

(2) $o_1.o\text{-}id=o_2.o\text{-}id$，$o_1.c\text{-}id=o_2.c\text{-}id$，$o_1.i\text{-}id\neq o_2.i\text{-}id$；

(3) $o_1.o\text{-}id=o_2.o\text{-}id$，$o_1.c\text{-}id\neq o_2.c\text{-}id$，$o_1.i\text{-}id\neq o_2.i\text{-}id$；

(4) $o_1.o\text{-}id\neq o_2.o\text{-}id$，$o_1.c\text{-}id\neq o_2.c\text{-}id$，$o_1.i\text{-}id\neq o_2.i\text{-}id$。

情形（1）叫做o_1与o_2**同一**。o_1和o_2是同一内对象，占用同一存储空间，对应同一概念对象，描述同一客观对象。



①o_1，o_2浅概念相等

②o_1，o_2深概念相等

$o_{12}.o\text{-}id=o_{22}.o\text{-}id$

$o_{12}.c\text{-}id=o_{22}.c\text{-}id$

$o_{11}=o_{21}$　$o_{13}=o_{23}$　$o_{14}=o_{24}$

图1　概念相等

（注：□—客观对象；△—概念对象；○—内对象；●—原子对象；…→—描述）

情形(2) 叫做**概念相等**。o_1和o_2是存储于不同存储空间的不同内对象，但对应同一概念对象，描述同一客观对象。若o_1和o_2顶层的非构造性属性值相同，构造性成分对象同一（符合情形(1)），则o_1和o_2叫做**浅概念相等**；若o_1和o_2的各层的非构造性属性值相同，构造性成分对象符合情形(2)，则o_1和o_2叫做**深概念相等**。如图1所示。

情形（3)叫做**客观相等**。o_1和o_2存储于不同空间，以不同的概念模式，描述同一客观对象，没有深、浅相等之别。这种情况较典型地对应于文[13]中的引用相等（referential-equal），情形（1）和（2)是引用相等的特例。

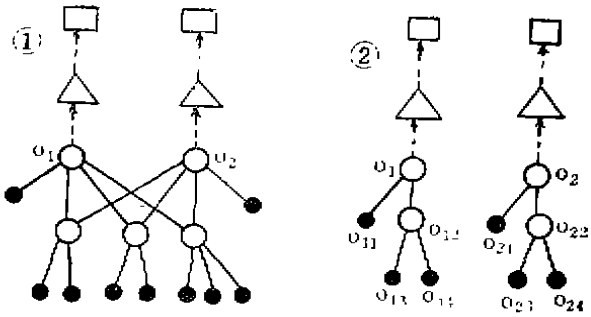
情形（4)，一般情况下，o_1和o_2是不同的内对象，对应于不同的概念对象，描述不同的客观对象。当o_1和o_2的概念模式相同，且$o_1.o\text{-}id\neq o_2.o\text{-}id$时，若它们顶层的非构造性属性值全相同，构造性成分对象同一，则o_1和o_2叫做**浅纯值相等**；若o_1和o_2的各层上的非构造性属性值（不含o-id）全相同，则o_1和o_2叫做**深纯值相等**（图2)。例如，在概念模式{[姓名，班级，{课程}]}下，同名同班的二个学生内对象浅纯值相等（同班学生所学课程相同，在内模式上共享课程集合对象)。深纯值相等的例子容易得到，略。若考虑到o-id也是一个属性，则无纯值相等。

文[10]中不含客观相等，其中的深、浅相等包括这儿的概念相等和纯值相等二种情况；这儿的$o_1.o\text{-}id=o_2.o\text{-}id$对应于文[13]中的引用相等。

4. 全标识与通常对象标识oid的比较

4.1 全标识的变形

根据前面分析，o-id和c-id由用户生成并赋于对象。概念对象是在某种方式（概念模式）下对客观对象的描述，没有必要对同一客观对象在同一概念模式下做重复描述，



①o_1，o_2浅纯值相等

②o_1，o_2深纯值相等

$o_{11}=o_{21}$　$o_{13}=o_{23}$　$o_{14}=o_{24}$

图2　纯值相等（图注同图1）

故c-id可由〈o-id，type〉代之，不必单设c-id。那末，全标识就变为id＝〈o-id，type，i-id〉。

i-id是内对象标识，与地址有关。存储地址有多种表达方式：物理地址、相对地址、间址、虚址等。可将虚址空间设想为无限大，对每一个i-id赋于一个虚址，从不重复[5]，这就相当于文〔10〕中的强标识代号(surrogate)。这时，此代号与物理地址间应有一对照表。于是，i-id分为二级：称代号一级为i-id-s，称对应的物理地址一级为i-id-p。

内对象的构成也与系统实现方式有关。在子类格中，可将描述同一客观对象的所有属性值聚集在一起，赋于一个i-id（首地址），此时的i-id标识的内对象对应于该子类格中最小子类（文〔8〕中的精确类，不一定显式地在所定义的子类格中出现）中的概念对象。其它子类中的概念对象由此最小子类概念对象在该子类对应的型（type）上投影而得到。这种方式需要变长记录文件支持。这是较典型的O-O系统实现策略。

另一种实现方式是在内模式中，内对象按概念模式聚集，每个子类中的每个概念对象对应的内对象有自己的i-id。如果考虑到i-id可分二级，则描述同一客观对象的概念对象对应的内对象可具有相同的i-id-s，不同的i-id-p，公共的i-id-s起逻辑上的聚集作用。这种方式允许将数据按子类分割存储。当然，上边的i-id-s也可象i-id-p一样总是不相同的，这时，i-id不显示属性聚集特征。这种将对象属性集按子类分割的实现策略具有关系模型实现的特征。

图3示出了上述二种内模式实现策略。

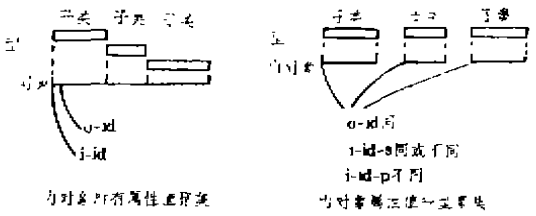


图3　内对象属性值集的不同聚集策略

4.2　关于通常对象标识oid的讨论

我们以下面的讨论说明，通常的oid实质上是某种i-id。缺乏o-id有时使得使用不便。

关于oid的实现，文〔10〕强调了以surrogate实现的强标识，有的称其为逻辑标识。现实的系统，以不同的对物理地址的独立性引入了oid，如Gemstone，Orien等就引入了逻辑标识（Surrogate）及与物理地址的对照表[8]；Smalltalk以指针实现oid，O_2系统以物理记录标识作为oid[5]。O_2系统以变长记录（所有属性聚集）存放内对象，OBMS/IDKE以按子类分割方式（逻辑聚集）存放内对象[15]。这些都在上面关于i-id的讨论的范围之内。

缺乏o-id，oid在使用中有时显得不便。一是唯一访问问题[15]：人们习惯于用对象属性值来区分对象，而O-O模型声称oid是标识和访问对象的唯一手段。用户要在一个子类中插入一对象，系统首先必须以oid判别在其超类中有无此对象；用户在一个子类中删除一对象，系统必须自动在其子类中删除所有具有同一oid的对象（物理删除或修改访问计数器reference count），所有这些都包括依oid对对象的唯一访问，都要求用户以某种方式指定oid。但oid是由系统按千篇一律的方式生成，对用户来说，由指定oid

来标明所要访问的对象是不方便的。在Smalltalk，C++中，引入了对象名（类似于变量名）供用户标识对象使用，以此解决用户唯一访问对象和系统唯一访问对象的不匹配问题。数据库中有大量对象存在，对每一对象由用户即席指定一个名字是一个负担。引入o-id，用户可指定概念对象标识〈o-id，type〉，由系统确定i-id，以此实现对对象的唯一访问是方便的。我们强调指出，对象名像变量名一样对应一个地址。二是对象拷贝问题：拷贝对象X而得到对象Y(deep或shallow)，按文〔10〕，X和Y具有不同的oid，但无法判别X和Y是否描述同一客观对象。对象的拷贝占用不同的存储空间，具有不同的oid，说明oid与地址有关。事实上，X和Y为概念相等。三是不同用户对同一客观对象的不同描述问题：二个用户独立地建立了二个数据库，一个描述运动员，一个描述学生，其中所有对象的oid全不同，难以依oid判定是否有一个运动员对象和一个学生对象描述的是同一个人。又如，概念模式{〔姓名，{〔课名{成绩}〕}}和{〔姓名，{〔成绩，{课名}〕}}是对学生学习成绩的不同方式的描述，其下的所有对象的oid全不同，无法依oid判定在上述不同描述方式下，二个对象是否是对同一学生学习情况的描述。这就是文〔13〕引入引用相等要解决的问题，实质上，判引用相等的根据是某种形式的o-id。

由上述三个问题可见，oid不同，则对象的存储地址不同，oid相当于i-id，而不同于o-id。

文〔12〕将现实世界的情况与计算机内的情况作了比较。现实世界中的客观对象的空间位置不同，为区别客观对象，人们赋于它们名字；在计算机内，对象的存储地址不同，并赋于对象以名字或其它标识，以区别不同对象。这种观点实质上认为通常所说的对象标识oid是对象的存储地址的一种代号，即oid相当于i-id。

5. 面向对象数据模型与面向值的数据模型

面向对象模型处理的基本单位是概念对象，语义上，概念对象是对客观对象以某种方式的描述，即使其纯属性值集(不含o-id)全相同，也是不同的概念对象，具有不同的c-id。对象标识在对象辨别中起重要作用。

面向值的模型处理的基本概念对象是元组，元组被看作属性值域的笛卡尔积中的元素。概念对象纯粹以属性值集合来区分，割断了概念对象与客观对象的联系。当属性值集相同时，即使描述的是不同的客观对象，也被看作同一概念对象，只用一个元组表之。关系模型中的key起重要作用，但它仅是元组的代表，作用域只是一个关系。这些导致了众所周知的关系模型的若干弱点[19]。

事实上，在面向对象模型中，属性值集也是描述对象的基本要素。本文引入的o-id既是客观标识(概念标识的一部分)，又是对象的属性。在概念模式中，它是一个特殊的、不变的key，在描述具有o-id的同一客观对象的所有概念对象中，o-id都是key，超出了一个关系的范围。o-id的引入，建立了面向对象模型和面向值的模型之间的联系。同时，内对象的属性值集按概念模式分别聚集（对应于概念对象）的实现策略具有关系模型的特征，也是对面向对象与面向值的模型相结合的支持，为引入带有标识的选择、投影、连接等典型的关系运算提供了方便[14,17]。

6. 后记

文〔6〕中指出，下一代数据库的特征应包括对象标识；由系统生成的对象标识和关系模型中的key。本文基于ANSI/X3/SPARC引入的全标识（〈o-id，c-id，i-id〉，其变形为〈o-id，type，i-id〉），较完整地说明了对象标识的语义及构成，包括了文〔6〕中指出的对象标识的特征。粗略地讲，这里的o-id和i-id分别相当于文〔9〕中的逻辑标识和对象标识，但o-id比文〔9〕中的逻辑标识语义上更明确，且明确了o-id与i-id可以是1对多的。全标识丰富的语义和对象相等的

协议工程 模型 计算机网络 ⑤

计算机科学1993Vol.20No.5

21—26

协议工程行为模型的研究

TP393

李腊元 （武汉水运工程学院 武汉430063）

摘 要

This paper has made the study on behavioral model for protocol engineering. A behavioral model which is hierarchical is described. In terms of this model, main concepts and properties of protocol engineering can formally be defined, and semantical relations of different FDT's are also discussed.

近年，随着计算机网络和分布式系统的不断发展，各类新型通信技术和分布式应用已开始出现，其中主要包括高速光纤网、多介质通信、宽带综合业务数字网(B-ISDN)、智能网，以及综合语音、数据和图象服务等。它们已对计算机通信协议的设计和实现带来了很大影响。为了适应这种形势的发展，一门新兴的高科技学科—协议工程已应运而生。协议工程本质上是计算机硬件工程和软件工程的理论方法在通信协议设计和实现中的具体应用，但由于协议具有实时、互操作和同步性等特点，因而其复杂度又要比一般传统软件大得多。协议工程的主要研究目标是使协议软件的生产怎样更好地实现规范化、工程化和自动化。它所包含的基本内容是：协议及服务的形式描述，协议验证和证实，形式描述的自动生成，协议转换，性能分析，半自动实现以及一致性测试[3]。迄今为止，协议工程在开发形式描述技术(FDT)、研制适应于某种FDT的编译器，以及探讨其综合开发环境或产生其自动工具等方面已取得不少可喜的成果，但从整体来看，协议工程的研究仍处在不断发展和完善之中，许多问题尚待进一步研究和探讨。为此，本文将研讨适应于协议工程的行为模型，旨在为协议工程有关概念及性质的形式描述，以及探讨FDT之间的语义关系提供一种抽象的形式模型。

一、一种行为模型

为了形式地描述通信系统或协议的各种行为特性，我们可描述一种抽象层次模型，其结构如图1所示：

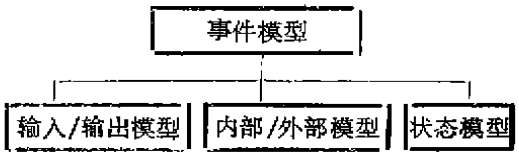


图 1 抽象层次模型

该模型分为两层：高层包括事件模型，它主要依据抽象事件来刻画系统的行为，并可用来描述和定义各种设计原理和相关的行为特性。低层主要包括输入/输出模型、内部/外部模型、状态模型，它们为表达事件提供了更为详细的形式途径，可进一步描述某些辅助设计原理和性质。此外，图1所示的模型还为各种FDT的综合分析（比较）提供了一种统一的基础。

1.1 事件模型 该模型是其它行为模型的基础，它所包含的基本概念是时间和事件。在下面的讨论中，我们用Time表示所

收到日期：92—12—05。李腊元 教授，IEEE INFOCOM、ICCC、ISDSRC等国际学术委员，国际论文评委，从事计算机网络和协议工程学的教学和科研工作。

概念，弥补了通常面向对象模型中的对象标识oid和关系模型中key的不足，为面向对象模型和面向值的模型的结合创造了条件。

（参考文献共19篇略）