

## 语义数据模型(2)

Joan Peckham

Fred Maryanski

### 2 有代表性的语义模型

以下将讨论一些语义模型,用以表示各种主要的概念建模方法,其中涉及了各模型的主要特征,并用第1节讨论的比较参数进行了分析。

#### 2.1 实体-联系模型

实体-联系(E-R)模型[Chen 1976]是一个早期的语义数据模型,它统一了传统模型的各种特征,有利于处理各种语义信息。顾名思义,两个基本建模结构是实体和联系。从概念上看,被建模的企业模式可视为由图形表示的一组实体类型和联系类型(与网状模型类似);从表达层次上看,实体和联系实例的信息结构与关系很相像。

前面图6表示了一部分图书馆数据库的E-R图,而图8描述了实体属性和联系类型的属性。从很抽象的观点来看,如图8,联

系存在于类型和其属性之间。虽不像在实体-联系模型中所讨论的,但类型及其属性间的联系对数据建模者提出了与聚集相同的抽象概念。

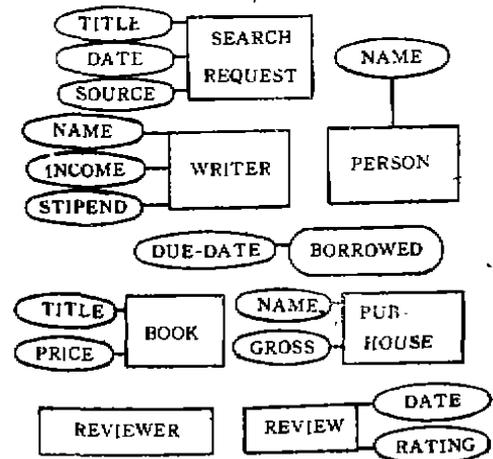


图8 E-R模型中的类型和属性

2) 如果  $\{B_{11}, B_{12}, \dots, B_{1n}\} \subset \{B_{21}, B_{22}, \dots, B_{2m}\}$ , 则  $ES_1$  的能力弱于  $ES_2$ ;

3) 如果  $\{B_{11}, B_{12}, \dots, B_{1n}\} \cap \{B_{21}, B_{22}, \dots, B_{2m}\} = \emptyset$ , 则  $ES_1$  与  $ES_2$  不可比。

本文从一般化角度讨论了问题的定义与求解,并讨论了评价专家能力的方法。应该看到,本文对所涉及的问题类型作了一定的限制,所给出的定义、方法是否具有普遍性,还需实践的检验。最后,感谢陆汝铃教授在我研制UNION过程中所给予的精心指导和帮助。

#### 参考文献

- [1] 陆汝铃,“分布式专家系统”,新一代计算机文集, P114—124, 1988.
- [2] Arch W. Naylor, “On decomposition theory: generalized dependence”. IEEE TRANSACTIONS ON SYSTEMS,

MAN, AND CYBERNETICS, Vol. SMC-11, No.10, 1981.

- [3] David M. Himmelblau, “Decomposition of Large-Scale Problems”. North-Holland Publishing Company, Amsterdam, 1973.

- [4] Michael N. Huhns, “Distributed Artificial Intelligence”. Morgan Kaufmann Publishers, Inc., Los Altos, California.

- [5] 庄庆雨,“分布式专家联合系统UNION中的任务分解与分配”, 89年全国人工智能及其应用学术会议文集, P189—195.

- [6] 管纪文,刘大有等,“知识工程原理”, 机械工业出版社.

E-R模型对约束的多重性提供了很强的支持。图6中的字符1、 $\infty$ 和 $N$ 表示联系的维数,这种联系受到其所在弧的约束, $\infty$ 和 $N$ 表示“多”,因此这种模型显然支持一对一、一对多和多对多联系,插入/删除约束是采用存在依赖定义的。图6中,REVIEW实体的存在依赖于BOOK实体的存在,因而如果BOOK实体被删除,则与之相关的所有REVIEW实体也将从数据库删除。存在依赖表示如下:首先把依赖的实体(REVIEW)画在一个双边长方形中,然后将标识“E”插入联系方块中(IS-REVIEW-OF),并包含一指向依赖实体(REVIEW)的箭头。依赖实体类型在实体-联系术语上是作为一个“弱实体”。

实体的标识能够依赖于实体参与的联系本质,实体可以通过一组属性的值来唯一标识,即WRITER的NAME。然而有些实体的标识会依赖于与其它实体的联系,例如在图6和图8中,SEARCH-REQUEST实体由TITLE、DATE和SOURCE构成,它不具有唯一的标识,但一定要由PERSON实体出发,通过REQUESTED联系来访问。再者,在图6中,标识依赖采用图形表示为:双边长方形围着SEARCHREQUEST实体,箭头在联系弧上,而标识在联系框中。

虽然已提出了包含概括的扩展模型(Teory等1986),但E-R模型直接支持的唯一抽象概念是聚集。Chen(1985)、March(1988)和Spaccapietra(1987)也给出了E-R模型的一系列扩充和应用。E-R模型的主要贡献在于,它是早期试图吸收网状和关系模型的最佳特征来提供多种抽象层次的模型之一。有关这类模型描述参看Pirrotte(1977)和Tschritzis and Lochovsky (1982)。

## 2.2 TAXIS

TAXIS [Borgida等1984; Mylopoulos等1980; Nixon等1987; O'Brien 1983]是一种交互式数据库系统设计的语言,其重点放在分类、概括/特化和抽象层次。该数据模型吸取了程序设计语言和数据库理论的

思想以支持以下能力:

(1) 数据封装:数据库对象的定义包含了其上的操作;

(2) 语义数据建模:特化从数据库定义的静态方面扩展到动态部分,这包括事务和例外处理操作的建模,后面将给出例子;

(3) 嵌套的数据库类型和操作:TAXIS中的数据库类型、“类”,及其操作和“事务”能嵌套到更高级的语言中(如PASCAL)。

其结果是一个高度结构化的模型,为数据库的静态及动态部分提供了完整的建模结构,概括/特化是支持TAXIS建模方法的基本抽象概念。如图1所示,这个概念抽象的形式是一个“IS-A”层次,在该图中,BOOK IS-A PUBLICATION,因而继承了BOOK的所有性质,另外,BOOK还拥有其本身的性质。TAXIS中的BOOK类定义如下:

```
dataclass BOOK with
  attributes
    AUTHOR: set of WRITER;
    SALES: 0..99999999;
    TITLE: string;
    PUBLISHER: PUB-HOUSE;
    BORROWER: set of PERSON;
    REVIEWERS: set of REVIEWER;
    REVIEWS: set of REVIEW;
end BOOK;
```

TAXIS允许如第1节中所提到的多重继承,若不能正确处理就会引起麻烦。但是TAXIS通过曾简单描述过的例外处理机制解决了继承冲突问题。

TAXIS的一个特别之处是用特化来建模数据库的动态部分,类的概念也应用到了事务定义上。考虑PUBLICATION类中的OBTAIN事务:

```
transaction OBTAIN with
  parameters
    p: PUBLICATION;
    l: LIBRARY;
  prerequisites
    Not-in-library? , (p not-in-l.acquis-
```

```
itions);
Not-yet-ordered? : (p not-in l.ordered-
ed-pub-list);
```

```
actions
a1: add p to the l.ordered-pub-list;
end OBTAIN;
```

在TAXIS的事务中，前件是前提条件，当它为“TRUE”时才执行动作。本例中，前件检查对象p表示的一出版物是否已在图书馆中或已订购。本事务的结果为：该出版物已由采购部门登记在订购单上以便进行下一步的加工，因为购买一种书要涉及专门的操作，如检查书价和把订书单交给出版商。执行买一种新书的事务能通过把OBTAIN事务特化后而得；OBTAIN的特化版本运行的条件是，作为参数p送来的对象是BOOK类型的，如下面所示。

```
specialize OBTAIN(p:BOOK)
add
prerequisite
Cost-reasonable? (p.price<=60);
action
a4:order-book(p, p.publisher);
end OBTAIN;
```

在建模例外处理时也沿用了特化技术。对每个事务定义了一个一般的例外处理器，它规定了处理一切例外的一般操作，特化的例外能与一般的例外处理器结合，定义处理特殊例外的操作。例如，一般例外处理器可定义为OBTAIN事务，而特化例外处理器对每个例外类型执行专门的动作，图9为OBTAIN事务展示了一个例外的层次。修改事务以处理例外及例外处理器的例子如下所示：

```
transaction OBTAIN with
parameters
p:PUBLICATION;
l:LIBRARY;
prerequisites
Not-in-library? :(p not-in l.acquisit-
ions);
Not-yet-ordered? :(p not-in l. order-
ed-pub-list);
```

```
actions
a1:add p to the l.ordered-pub-list,
for exception e in OBTAIN-EXCEP-
TION with pub<-p,lib<-l
use EX-HANDLER(e);
end OBTAIN
Script class EX_HANDLER(e:book-out-of-
print)

transitions
send message;
actions
inform user that book is out of print
end;
```

BOOK-OUT-OF-PRINT这个例外操作出现的条件是OBTAIN事务被一个不成功的候选值作为参数调用，在TAXIS中用“脚本”来实现例外处理器，这些脚本被概括为具有显式通讯和同步机制的进程。这里，脚本的通讯方面是有趣的，因为例外频繁地引起给用户显示信息。

所以，TAXIS是这样一个系统，它强调分类和概括层次并扩展到数据库的动态部分上。该系统的一个突出特点是，利用数据库抽象建模例外处理器和事务。

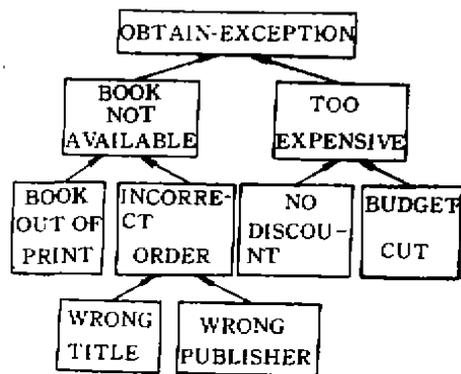


图9 例外层次

### 2.3 SDM

这个综述讨论了很多模型，给建模者提供了典型的一两个基本的抽象概念。回忆一下带有实体和联系的E-R模型及TAXIS，它能提供类、概括/特化的层次。SDM[Hammer and McLeod 1981]采用了一种不同的方

法, 它将各种建模结构表示为单独一个抽象概念——“类”, 意在使数据库设计者的概念能清晰地表达数据库的“含意”, 而多数别的语义模型为设计提供了一组原语, 由此可构造更复杂的概念对象。SDM 试图提供一组完整的建模设施。按照语义建模的观点, 建模的目标之一就是允许存在灵活的和多视图的数据, 同时要小心控制在这种需求下引起的重复建模劳动。聚集、概括、分类、联合和派生都是为了实现这一目标的。

在 SDM 中, 分类和联合比聚集和概括更重要, SDM 中的数据库是组成类或类型的实体 (实例) 的集合, 设计者定义了类, 在这个框架内定义了成员、类属性、类间联系和派生。这与实体-联系模型不同, 它着重定义类本身, 而不是通过 E-R 图及 IS-A 层次所表示类间联接。

图书馆数据库的另一个例子说明了 SDM 方法。假设下面给出了 BOOK 和 DATABASE-BOOK 的定义, 对 BOOK 的描述只包含反映单本书性质的成员属性, 但 DATABASE-BOOK 除其成员属性外还包含了类间联系的描述和类属性。类间联系的描述表明, DATABASE-BOOK 是 BOOK 的子类, 其中 DATABASE 是 BOOK, Topic 的一个元素。类属性把类的性质作为一个整体进行描述。

#### BOOK

description: all books within the library  
member attributes:

Title

value class: STRING

Author

value class: PERSON

Publisher

value class: PUB-HOUSE

Price

description: price of book at library  
rate

value class: DOLLARS

Topic

value class: TOPICS

Interest-Index

value class: INTEGERS

Acquisition-priority

value class: INTEGERS

derivation: order by decreasing Interest-Index within BOOK.

identifiers

Title

DATABASE-BOOK

description: all database books within the  
library interclass connection,

sub-class of BOOK where BOOK.

Topic = 'database'

member attributes:

Subtopic

value class: DB-SUB-TOPICS

class attributes:

Total-cost

value class: DOLLARS

derivation: sum of Price over mem-  
bers of this class

正如上面 Total-cost 和 Acquisition-priority 所解释的, 由属性所假设的值能够通过运用一组复杂的派生原语从数据库模式的任何其它信息, 如统计、布尔、排序计算和递归模式追踪等计算出来。

类间联系的两种类型是子类型和组, 它们可在类定义中说明。子类型和组联系以前分别处理概括和联合这两个抽象概念, 如在多数含概括层次的系统中, 要用到属性的继承, 由于允许多重继承, 所以当两个以上的上级类型具有同名而不同值或派生的属性时, 要定义规则来处理继承冲突。当沿概括层次下移时, 属性也被特化了。在上述的定义中, DATABASE-BOOK 是 BOOK 的一个子类型, 子类型的定义中, 用户可明显给出隶属关系如下:

GOOD-BOOKS is a subclass of BOOK  
to be specified by the user.

其它类中的隶属关系也能用以定义类, 如:

DB-AI-GROUP-BOOKS is defined as  
a subclass of BOOK

where class membership is defined as books that are in the classes

DATABASE-BOOKS and AI-BOOKS.

组联系针对所说明的用户或约束用于定义包含一组对象的类，例如，RESEARCH-GROUP-COLLECTIONS的定义如下：

```
RESEARCH-GROUP-COLLECTIONS
interclass connection:grouping of BOOK
as specified
membership attributes:
  Good-book-list
  description:the good books that are
    in the collection (if any)
  value class: BOOK
  derivation:subvalue of Contents where
    is in GOOD-BOOKS
    multivalued
```

interclass connection 语句说明了：第二个研究小组的元素是由用户给出的，在这种情况下，集合包含由研究小组成员选定的书，GOOD-BOOK-LIST 是一个多值、派生的成员属性，它包含了那些所选出的归于好书一类的书。content 是为每个组类自动建立的一个多值的成员属性，属性的值是类的成员的集合，类在一个组中，而组形成了该成员的内容。本例中，RESEARCH-GROUP-COLLECTIONS组中的每个成员即是每个研究小组中书的集合，其Contents 属性值与研究小组中由用户选出的书的集合相同。GOOD-BOOK-LIST 属性的 derivation statement 确定了哪些 Contents 属性集成员同时也在 GOOD-BOOK 类中。

SDM 将类抽象作为其主要的概念建模手段。尽管采用 SDM 方法能有效地用类表示实体，但联系还是用类间联系（作为类定义的一部分）来表示。SDM 为类抽象提供了各种约束、继承、派生等，其突出特点之一是着重于类的说明而不采用层次或网状来表示各类间的联系。

虽然更多的建模结构能提供更出色的功

能，但有人会指出各有侧重，不过 SDM 仍可认为是提供了能表达用户数据视图的一组特别丰富的结构。SDM 为其它语义模型充当了一个原型系统，许多后来的系统也确实选用了 SDM 的一个子集，为特定的企业模式或方法提供了一组有用的建模工具。

## 2.4 函数式数据模型

函数式数据模型 [Shipman 1981] 是伴随着数据定义语言 DAPLEX 而构造出来的，其目的是提供一个模型及能自然而简易地表示应用的定义/操作语言。许多模型用各种结构来使建模更灵活（这些结构在第3节的表1中进行了总结），但这个系统的设计者发现，把结构限制为实体和函数能为数据定义和操纵提供一种直接、简明的语言。

在函数式数据模型中，能够用函数来定义属性的聚集，形成实体。BOOK 可定义如下：

```
DECLARE BOOK( ) ==>> Entity
DECLARE Title(BOOK) ==>> String
DECLARE Publisher(BOOK) ==>> PUB-
HOUSE
DECLARE Author(BOOK) ==>> PERS-
ON
```

因为 AUTHOR(BOOK) 和 BOOK( ) 是多值的，==>> 用于表明多值函数。联系也用函数方式表示，AUTHOR(BOOK) 的说明描述了 BOOK 和 PERSON 之间的联系。一个多元函数如

```
DECLARE REVIEW RATING(BOOK,
REVIEWER) ==>> RATING
```

能表示多实体间的聚集联系。作为比较的基础，E-R 模型把上面的函数表示成带属性的联系，REVIEW-RATING 应是带 RATING 属性 BOOK 和 REVIEWER 之间的联系。

函数式模型并不为概括和分类提供明确的工具，虽然用户可以定义表达这些抽象概念的函数。在实体类型上的空函数将返回实体类型的元素集合，于是就提供了一种联合，如图书馆中书的集合表示成 BOOK( )。

虽然函数式数据模型并未能像其它模型一样提供各种建模技术,但很明显,其优点之一在于能清晰、简洁地表示实体间的联系。**Buneman**和**Nikhil**(1984)讨论了函数式数据语言**FQL**,其中的一个小函数集为函数式数据库的操作提供了查询集合。

## 2.5 RM/T

**RM/T** (**Tasmanian**模型) [**Codd** 1979] 是**Codd**关系模型 [**Codd** 1970]的一种扩充,试图通过引入联系和完整性规则来进一步获得概念模型中更多的含义。图3所示的关系模型,提供了一个表格式的概念模型,其中所有表间的联系是以表中的数据值为基础动态地形成的。

引入了关系模型后,研究者们便把精力投于模型的描述形式,以保证高度的一致性及防止修改异常 [**Codd** 1970; **Fagin** 1977, 1979]。但保持高阶范式的结果通常是大大缩小了用户原来定义的关系,因而与用户的概念模型差别很大。

例如,如果用户要定义一个包含完整的作者描述的**BOOK**关系,定义范式的规则将迫使这种关系分解为**BOOK**和**WRITER**两个关系,以保证一定程度的数据库一致性。这两个分离关系的定义允许一个字段(如作者地址)的改变只出现在某一处,而不必修改此作者所写的所有书的元组。然而,用户还要考虑到这两个关系之间仍存在联系或联接,这种关系间的联系在关系模式中并不明显存在。例如,当图3自认没有任何支持文件时,在**BOOK**和**WRITER**间就不出现明显的联接。**BOOK.AUTHOR**和**WRITER.NAME**间的语义联系仅存于用户的心里或应用程序的代码中。关系的联接操作是用来联接那些字段中有值匹配的元组的,但在关系模型中,一个联接也可能出现在**BOOK.TITLE**和**WRITER.NAME**之间,因为二者均为文本字段。其结果从技术上是成立的,但不是所需要的那种联系。

虽然关系模型提供了一个定义良好且合

理而简明的概念模型,但其语义单薄,把建模重担留给了端点用户或应用程序员。**RM/T**描述了在保持其基本特征的同时提高关系模型语义表达能力的方法,按此精神,**RM/T**模型定义了实体和联系类型及它们之间相应的存在约束。实体类型用**E-关系**定义,每种类型存在一种,而**P-关系**则定义了实体类型的性质。**E-关系**有一个项,是实体类型中每个实例的唯一标识符,**P-关系**直接与**E-关系**相关,并且对每个性质都保存一个值。图10描述了有关**BOOK**实体的**E-关系**及一组**P-关系**。

**RM/T**用“联合实体类型”表示多对多联系,用“指称实体类型”表示多对一联系。**BOOK**和**PERSON**之间的联系**BORROWED**是属于多对多联系,它作为一种在**RM/T**中定义的联合实体类型(将在后面谈到)出现。**BORROWED**关系包含有关的**BOOK**和**PERSON**元组,再加上一条指明图书馆中书的总册数的属性。

```
CREATE E-RELATION HOLD
ASSOCIATING (BOOK VIA BOOK-
ID, LIBRARY VIA LIBRARY-ID);
CREATE P-RELATION PHOLD FOR
E-RELATION HOLD
PROPERTIES (SUR-BOOK SURR-
OGATE FOR BOOK, SUR-LIB SURRO-
GATE FOR LIBRARY);
CREATE P-RELATION PHOLD-COP
FOR E-RELATION HOLD
PROPERTIES (COPIES DOMAIN
(COPIES));
```

**BOOK**和**WRITER**之间的联系可以这样来表示:通过给**E-RELATION BOOK**的定义增加如下所示的“**DESIGNATING**”短语来定义从实体类型**WRITER**到实体类型**BOOK**的指称访问,这样就认为**BOOK**是一种指称实体类型。

```
CREATE E-RELATION BOOK
DESIGNATING (AUTHOR VIA
```

WRITER-ID);

E-RELATION		
BOOK	BOOK-ID	

---

P-RELATION		
AUTH	BOOK-ID	AUTHOR
TITL	BOOK-ID	TITLE
PUBL	BOOK-ID	PUBLISHER

图10 E-关系和P-关系

RM/T 为各种实体类型提供了许多内部完整性规则，下面的规则 [Date 1983]应用于联合实体类型BORROWED, Codd[1979]和Date[1983]都包含了全部 RM/T 完整性规则。

一给定的BORROWED实例可存在于数据库中，仅当对于这一实例，BOOK和PERSON的每一E-属性或是空值，或确定了一个存在的适当类型的实体。

实体类型的E-属性是内部的、系统范围的、唯一元组确定的属性，在面向对象的系统中，一个E-属性大致对应于对象标识。上面的完整性规则的一般形式对于模式中所有的联合实体类型都成立。为了避免空值，也可重新定义规则。任选项适用于这里定义的BORROWED类型。

RM/T通过其SUBTYPE子句明显地支持了类型层次。在RM/T类型层次中，子类型通过指定的每一属性值来区别，如图11。联系DATABASE-BOOK IS-A BOOK用DATABASE-BOOK的 E-关系定义表示如下：

```
CREATE E-RELATION DATABASE
-BOOK SUBTYPE OF BOOK PER
CATEGORY TOPIC;
```

短语“PER CATEGORY TOPIC”表明，TOPIC属性的值决定了DATABASE-BOOK子类型中的隶属关系。

多重继承可出现在 RM/T类型层次中。RM/T做了一点假定，由于没有一种明确的

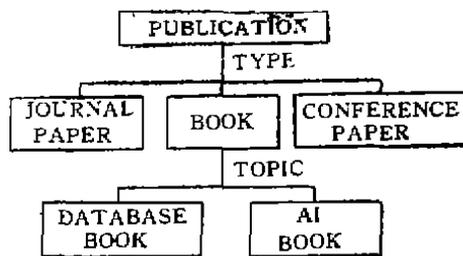


图11 RM/T类型层次

完整性规则会使多重继承出现，所以不会产生命名冲突。

RM/T 模型的目标与其它语义模型稍有不同，其动机是处理因插入/删除与关系间依赖有关的元组引起的数据库的不一致性，因此，这种模型与关系模型一样是面向信息结构的。大多数其它语义数据模型提供了类似的建模抽象，但起点不是基础的信息结构，而是在更高的概念级上。RM/T本身是一种增强的关系模型，而不是一种新的概念建模方法。但是，它仍可看作是一种语义模型，因为这些语义的定义不仅为这些联系赋予了更多的涵义，而且提供了必要的数据结构以运用数据建模抽象。

### 2.6 SAM\* 语义联系模型

SAM\*[Su 1983]是一个原为科学统计数据库而设计的语义模型，后来发展为明显地支持计算机集成制造的应用。由于SAM\*是为一类特殊应用而设计的，它通过支持非传统的对象类型和联系来方便这些数据库的设计。用SAM\*构造的模型，其最一般结构是一种原子和非原子概念(或对象)的网络，非原子结构通过递归嵌套来为对象类型定义提供一个好的结构和语义一致的方法，在最低的层次上，概念表示为一组抽象数据类型。从一般的字符串和数字扩展到这些类型是由于以下原因：

- (1) 提供那些与用户所看到的实际类型相对应的对象类型来作为应用的最原始单位；
- (2) 提供定义良好的、与问题领域中所出现的动作相对应的信息单位上的操作。

由于不必花时间在原来类型的基础上建

立高层次的结构，以上两点节省了设计人员的劳动。

作为计算机集成制造环境的需求分析结果[Su 1986]，在SAM\*中建立了以下类型及其操作：

- (1) 系(有序的和无序的)；
- (2) 向量和矩阵；
- (3) 时间和时间序列；
- (4) 正文；
- (5) G-关系(扩展关系)。

SAM\*的G-关系是一种扩展的关系，其属性可以是任何有效的SAM\*类型，包括关系。G-关系的子类型通过将类型层次组织成一个语义网络来表示，映射隐含地把G-关系类型嵌套在别的G-关系定义中。例如，图12描绘了PUBLICATION的G-关系含有CONFERENCE-PAPER、JOURNAL-PAPER和BOOK关系作为PUBLICATION的特例这一事实。

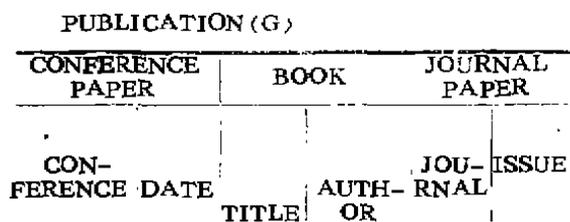


图12 G-关系

SAM\*为设计人员提供了七种内部联系(SAM\*的术语为联合)，它们可以组织成一个网络，以建模一特定企业的语义。下面给出了这七种联系[Su 1986]，其中有些已在前面的模型中出现过，这里只是一带而过。

- (1) 隶属：“是……的成员”；
- (2) 聚集：已在引论中定义；
- (3) 交互作用：提供了维数与供参考的完整性约束来建模E-R中的任意联系；
- (4) 概括：扩展概括具有定义在概括层次的兄弟类型之间的互斥、集合相等、子集、集合相交等约束。图13说明了互斥约束，它表示一PUBLICATION必

而是BOOK、JOURNAL-PAPER或CONFERENCE-PAPER之一；集合相交约束表示一BOOK可以是AI-BOOK和DATABASE-BOOK。在此图中，一个G结点表示概括，一个A结点表示聚集；

- (5) 合成：是……的一部分；
- (6) 叉积：类型分组，为了用下面定义的总和联系表示统计分析，每个类型的实例是其它组成类型实例的叉积；
- (7) 总和：支持统计聚集和非聚集，结合叉积与总和，设计人员可以使用n种不同的实体类型的属性来建立统计实体。图14表示了一组允许收集出版的财务方面的总和统计。

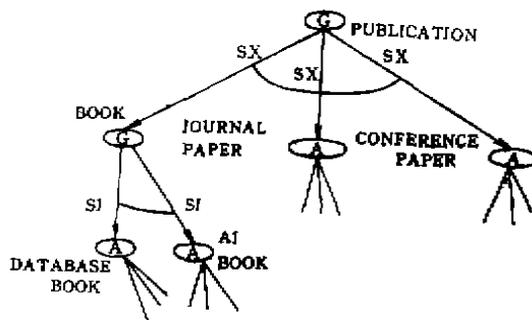


图13 SAM\*概括层次

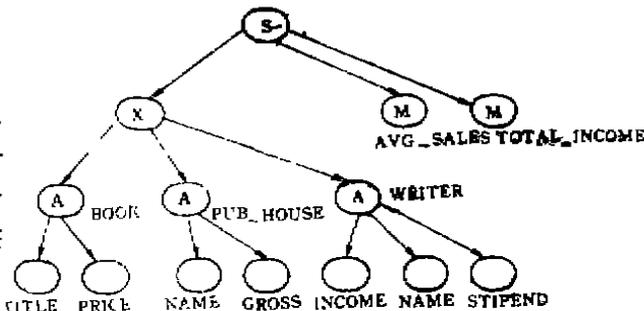


图14 叉积与总和

SAM\*是面向统计数据库和CAD/CAM应用的，它包含了非传统的数据。SAM\*中有大量不同的原始数据结构，而且，G-关系为表示各种不同的信息提供了支持。SAM\*的建模结构是在评价了CAD/CAM环境的概念建模需要后选择的。在SAM\*模型基础上，开发者

已实现了一个数据库管理系统原型 IDMAS<sup>4</sup> 以评价模拟技术对计算机集成制造应用的适用性 [Krishnamurthy 等 1987]，一种加强的模型与系统正处于设计阶段。

### 2.7 事件模型

事件模型 [King and Mcleod 1984, 1986] 是一种用函数支持概括，用属性表示聚集的语义数据模型，其特点是一种动态建模方法。

在事件模型中，子类型联系用于将静态模式组织为层次。子类型中的成员关系用属性求值的谓词来定义。例如，若 DATABASE-BOOK 是 BOOK 的子类型，则要在 BOOK 中说一个谓词，定义 BOOK 的成员，如：

```
Subtype; DATABASE-BOOK
    all BOOK
    where topic="database".
```

下面列出的属性约束是建立在模型中的 [Farmer 等 1985]：

- (1) **Primary**：唯一标识对象的属性；
- (2) **Ordered**：可作为一种排序关键码的属性；
- (3) **Single valued**：以一原子元素作为其值的属性；
- (4) **Multivalued**：以一集合作为其值的属性。

因此，属性的语义取决于模式设计者强加在属性类型定义中施加的约束，这就给设计人员提供了更多的灵活性。BOOK 类型的定义表明了事件模型的静态建模设施：

```
TYPE; BOOK
    primary attributes; ISBN-string(single valued, nonnull)
    dependent attributes;
        AUTHOR-WRITER(multivalued, nonnull, ordered)
        SALES-integer(single valued, ordered)
        TITLE-string(single valued, nonnull, ordered)
        PUBLISHER-PUB-HOUSE(single valued, nonnull, ordered)
```

```
BORROWER-PERSON(multivalued)
REVIEWERS-REVIEWER(multivalued)
REVIEWS-REVIEW(multivalued)
TOPIC-string(multivalued, ordered)
```

交互式动态建模设计方法学包括一系列描述应用动态结构（事件）的设计步骤。在事件设计的第一个步骤中，逐步求精的过程用于定义应用环境的功能，后者表示为过程和功能链的层次。过程事件对应于应用环境中的处理单位，功能链用以表示事件的层次。若一个单独的人或过程负责两个事件，或者一个事件逻辑地嵌入另一个中，则在两个事件之间定义一功能链。图15提供了一个购书事件的例子，这种方法学和许多EDP系统分析技术相似。这种带有概念数据建模方法的系统分析方法将事件模型与其它语义模型及传统的EDP方法区别开来。

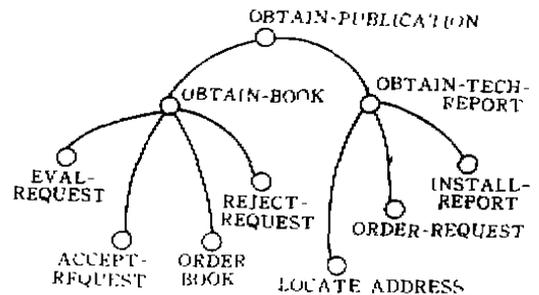


图15 功能链说明

接下来，有向通讯链是用来表示数据库的信息流的，如图16所示。所得的图示与状态图或Petri网相似，不过它们表示得更具体。图16给出了得到出版物的信息流，最后一步是消除二义性和设计求精。

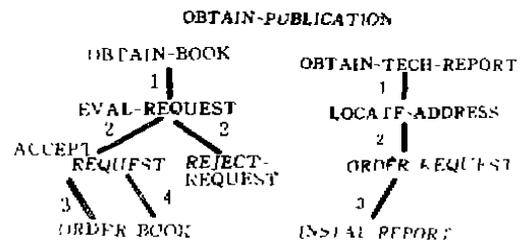


图16 信息流

正如作者指出，这种模型不适宜于信息流不确定或不规范的应用。然而，当模型的性态可适当预测时，这种模型就的确为数据库的动态方面提供了一种高级的建模方法学。这种模型的设计思想不仅指出了机制，确定了设计人员提出信息的顺序，而且还可扩展到静态建模。如果它有助于设计者说明一种应用的语义，则这种思想是很值得包含于模型定义之中的。

使用事件模型的工作包括 Sedaco [Fae-mer等1984,1985]，即一种语义数据库构造工具，它提供了用于数据库实现的原语，及 Sembase [King 1984]，即一种语义数据库管理系统，它采用了基于图形的用户接口。

### 2.8 SHM+

如同前面的模型，SHM+[Brodie 1984; Brodie and Ridjanovic 1984]是为了解决应用的静态和动态建模问题。数据对象及有关事务的说明通过运用一种抽象数据类型思想和相关的建模方法学来实现。与关系表和实体-联系图不同，对象和行为模式用于从设计者那里获得对象和操作模式，这些模式用来建模高层和低层的数据、联系和操作对象，因而为表示应用语义提供了一个统一的结构。

SHM+的基本建模结构是原始对象和操作，分层形成更复杂对象和操作的组台规则及施加于所有原语、组合规则和层次的约束。

为了建模应用的静态结构，使用了对象模式将对象和结构联系的总特征集中起来。一个利用聚集的例子如图2所示，对概括和联合也有类似的模式。

SHM+通过联系而不是概括来支持继承性。如果我们假定图2中的 AUTHOR 涉及其它对象类型，则该类型的其余属性就可以被聚集对象 PUBLICATION 所继承。除此之外，这种模型还允许用户指定应该实现的继承范围（如，AUTHOR 的哪一些属性应作为 PUBLICATION 的一部分）。当概括层次出

现在设计中时，SHM+规定了严格的继承性。

“行为模式”是SHM+事务总特征的明确图形表示，一个行为模式中的结点代表事务中的对象，而弧上标有对对象的操作，图的结构定义了事务的控制流。图17和图18分别说明了一串计算和选择决定，这些控制抽象概念的图形表示等同于用于表示聚集和概括结构抽象概念的图式。这样，SHM+就为静态和动态对象提供了一种统一的建模方法学，图19表明了OBTAIN-BOOK事务的SHM+定义。

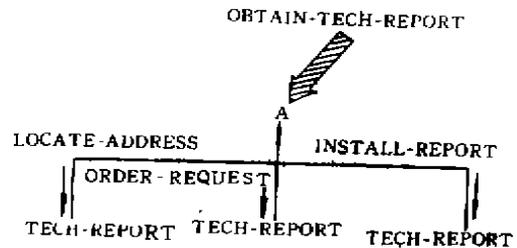


图17 顺序模式

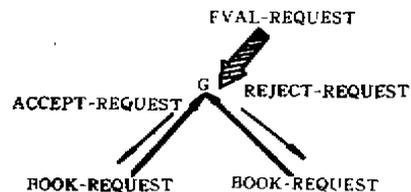


图18 选择模式

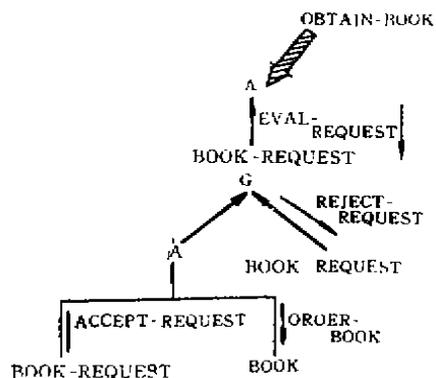


图19 行为模式

# NF<sup>2</sup>关系数据模型及其在工程数据建模中的应用

聂培尧 (山东财政学院计算机系)

**摘要** 目前关于支持层次结构的复杂工程对象的研究,大多是扩充原有的关系DBMS,但这种方法在概念上和执行等方面都存在许多缺点。本文介绍一种嵌套关系数据库系统中的NF<sup>2</sup> (Non First Normal Form) 数据模型。该模型可按一种规范的方法处理标准关系(1NF)、列表和层次结构,最后讨论NF<sup>2</sup>数据模型在工程数据建模中的应用。

在CAD/CAM环境中,常常需要嵌套层次较深的结构和多种不同的元组类型来表示复杂工程对象<sup>[1,2]</sup>。由于数据聚类和避免不必要联结之类的语义(概念)与执行上的原因,这种复杂对象<sup>[3]</sup>不能简单地作为普通的标准关系(1NF)存储。这就要求DBMS能提供一种直接支持层次结构的功能。

文献[3,4]论述了一种扩充关系DBMS来更有效地支持层次结构的方法。该方法的实现是加入了一些新定义的属性类型和系统产生的标识码(以表示层次联系和加快联结速度)并且提供了适当的新DML语言(如对复杂对象的插入、检索、更新、删除等操作)的专用接口。这种方法的优点是对原有DBMS的修改部分较少,但缺点是对于底层的DBMS来说复杂对象仅是一种特殊的对象,并且不能按照通常的标准关系进行处理。也就是说,这种方法不能像标准关系那样支持复杂对象的投影、选择以及聚类等操作。为了真正实现标准关系与复杂对象的统一表示,应当开发一种能体现这两种数据类

型的数据模型。

一种较好的方法是扩充数据模型,使之能在数据模型中表示标准关系与复杂对象,又不放弃高层关系查询语言。这种扩充方法的主要思想是允许在一关系中关系本身也可以作为属性值出现。这就意味着应放弃关系必须满足1NF的要求。我们称这种关系为非第一范式(NF<sup>2</sup>)关系<sup>[4,5,6]</sup>亦即带关系值属性的关系<sup>[7]</sup>。

## 一、工程DBMS中的数据模型

数据模型是构成DBMS的基础,根据目前的工程DBMS,我们可把它的数据模型分为三大类。

### 1. 传统的数据模型

现在很多用于工程设计、制造等应用领域的DBMS仍是使用60年代末,70年代初开发的关系、网状、层次之类的传统数据模型<sup>[8]</sup>。基于网状模型的一个DBMS的例子是由phil-

SHM+模型最重要的方法就是它对动态数据建模和有关动态及静态模式的一致建模方法所作的贡献,图2和图17中结构的相似性说明了SHM+中数据和事务建模方法的共同性。由于数据库对象的语义总是对数据库行为指定了约束,所以在数据库建模中定

义事务(动态对象定义了数据库行为)是合理的。

(未完待续)

[纪岳、何未艾、王红、赖维生、陆强、钟颖译自、姚舞达校自ACM Computing Surveys, 1988, vol.20 No3]