

语义数据模型(1)

Joan Peckham Fred Maryanski

摘要 语义数据模型是适应更强表达能力的概念数据模型的需要而产生的。当代的数据模型不能直接支持联系、数据抽象、继承性、约束、非结构对象,及应用的动态特性等。尽管普遍认为数据模型需要更丰富的语义,但还没有一种方法赢得公认。本文描述了语义数据模型的基本性质,给出了自七十年代中期以来提出的有代表性的模型。除了解释单个模型的特征之外,还提供了一个比较它们的指南。最后讨论了概念数据建模领域的未来方向。

引 论

虽然关系模型已向数据库专业人员提供了独立于物理实现细节的建模方法,但许多设计人员仍认为,它并没有为那些难以直接列成表的问题提供足够丰富的概念模型。过去十年中,已出现了许多数据模型,其目的是向建模者提供更强的表达能力并将更丰富的语义融入数据库中。这些数据模型可大致归类于“语义”数据模型,因为其共同特征是试图提供比关系模型更多的语义内容。

数据库所表示的那一部分“现实世界”通常被称为“企业模式”,数据库模型的结构或模式是“企业模式”中元素及元素间联系的一

注意到cut的插入可能会改变程序的终止性,使得本来并不终止的程序在插入cut后可以终止。

6、结束语

PROLOG的语义通常是根据一段逻辑的模型论给出的。但是,这常常并不足以推导PROLOG程序的计算特性和确认程序变换的强正确性,因为它不能支持这类程序计算性态的推导。如果我们希望研究含有“非纯逻辑”成份(例如cut)的语言时,问题

个表示,用于这些元素建模的构件称为“对象”或“实体”。企业模式往往不是静止的,所以数据库模型就要有一些结构用于操纵数据库模式对象的建模操作,这些结构可以是原子操作或者更复杂的事务,而且可以认为是含有其间已定义了联系的对象。

现实世界的任何一部分的建模都有一固有的问题,即人类对企业模式的理解不同于计算机,必须用一定的方式将这些结构组织起来,以有效地存贮和实现。这导致了考虑用户的概念模型、机器的物理模型及二者间映射关系这三个数据库建模层次的出现,这些层次在ANSI/SPARC关于数据库体系结构标准化建议等中描述如下:

变得更加严重。

在本文中,我们给出了PROLOG在含有和不含cut情况下的指称语义和操作语义;证明了它们的等价性;并且,通过验证PROLOG程序的两种优化变换说明了这种语义的作用。我们相信,虽然模型论语义对于理解PROLOG程序是非常有用的,但也需要用我们这样的指称描述来推导出操纵和转换PROLOG程序的机制。

(附录1、附录2和参考文献略)

【译自J. LOGIC PROGRAMMING
1988:5:61—91 仲源校】

(1) 外部层次：不考虑实现和存贮问题，只是用户对企业模式的逻辑视图。

(2) 概念层次：是信息模型，提供了从逻辑到物理（或内部）层次的映射，描述了实体和联系的语义，包括联系和一致性约束描述。

(3) 内部层次：是涉及数据访问路径和存贮的物理数据库的抽象模型。

运用这一划分，关系模型及其扩充模型可被视为概念模型。虽然关系模型中的“表”缺乏表达全部逻辑模型对象间联系的能力，但它的确为到物理模型的映射提供了方便手段。实体-联系模型可以被视为概念模型和外部模型，因为它不仅说明对应于用户企业模式逻辑视图的对象和联系的一种方法，而且是把这些数据结构映射到物理结构的一种方法。

本文讨论的概念语义模型利用了实体、联系和约束来描述企业模式的静态、动态和时态特征。期望能够不管物理模型，使企业模式的表示很类似于用户的理解。通常，带有语义的标准联系（包括相应的操作和约束）是预定义的，并提供了访问和存贮策略，用户只需选择并结合这些联系，就能对现实世界企业模式建模。为此，语义概念模型提供了不同层次的建模方法。

早期的数据库研究集中于数据库物理结构，很少考虑用户对数据的理解，而最关心的是提供一致的、高效的数据库存贮和检索所需的物理结构和信息结构。层次和网状模型在记录级上向用户提供了导航数据库的方法，也提供了导出更抽象结构的操作，关系模型增加了一个数据结构层次，不必执行基本记录级的操作。前者是操作型的方法，后者是结构型的方法。这些方法的能力与数据库的记录结构紧密相关。

七十年代中期，研究者们试图通过提供能支持用户数据视图的建模结构来简化数据库的设计和使用。Chen 1976, Schmid and Swenson 1975, Smith and Smith 1977这

三篇论文提出了两个重要的数据建模的思想，标志着语义数据模型的出现。第一个思想是数据独立性。数据库研究者们受程序设计语言发展的影响，觉得用户应该从数据库物理结构细节中解放出来。这样，用户就可按照人们对问题的理解来建模数据。第二个思想是在数据建模的过程中获得更多的语义。现有的数据模型可以定义一些语义，如关系理论的函数依赖为数据模型建立了一些低级语义。已有人试图将这些语义扩充到关系间的依赖或联系。

Schmid和Swenson[1975]的论文是上述的一个例子，其中描述了“联系”、“特征”和“联合”及“相关”的语义，Smith和Smith[1977]的论文则把一些心理学家和人工智能研究者所定义的抽象概念（诸如概括、聚集等）用到了数据库建模中。实际上，几乎所有的现行语义数据模型都提供对这些抽象概念的支持。

概括忽略相似对象的不同之处，形成较高级类型，在较高级的类型中强调相似性。例如，图书馆数据库中的PUBLICATION实体，它体现了图书馆收藏的所有出版物的特征。在图1中，PUBLICATION与其它更具体的实体间的联系是用类型层次来表示的，其中，上面是概括类型，下面是具体类型，如PUBLICATION可以认为是数据库中JOURNAL-PAPER、BOOK和CONFERENCE-PAPER

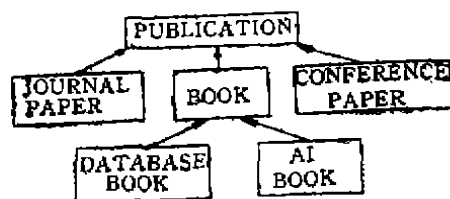


图1 概括



图2 聚集

的概括。

聚集是一种在高级类型中用到低级类型之间联系的方法。关系数据模型运用这一概念将实体类型或关系聚集成较高层次的实体，如在我们的数据库例子中，TITLE和AUTHOR类型聚集为PUBLICATION类型，如图2所示。

除概括和聚集外，许多语义模型还支持分类和联合的联系。分类是一种把一组对象视为一高层次对象类的抽象概念，它表示了一种“实例”联系。例如，在我们的数据库例子中，BEST-SELLING-BOOK对象类由所有销量超过10,000册的BOOK对象构成，对象IT是BEST-SELLING-BOOK对象类的一个实例。分类提供了一特定对象类型的说明结构，而特化则是从一现存类型导出另一类型的定义。

联合是一种将成员对象之间的联系视为一较高层次集合对象的抽象概念。例如，集合DATABASE-BOOKS是一些BOOK对象的联合，AI-BOOKS也一样。集合成员资格的标准，一般是基于一些谓词的可满足性，例如对于DATABASE-BOOKS集合：TOPIC=DATABASE，但集合定义的结构可以是纯粹外部的。若将GOOD-BOOKS作为例子，可知决定其成员资格的是终端用户，而不是模式设计者。

虽然聚集和联合都是从已定义的类型定义新类型，但它们是本质不同的抽象概念。聚集提供了一种说明新对象类型的属性的方法，而联合是一种定义类型的机制，其值是某一类型对象的集合。

随着语义数据模型的增多，其好处也越来越明显了。

(1) 表示的经济性 语义数据模型中，用户能完整地象以往那些模型一样方便地从数据库中提取各式各样的信息，所有这样的模型都提供了比完整性更强的经济性，即用户不仅可以准确提取相同的信息，而且可以随意提取信息的部分内容。

在关系模型中，用户必须留心关系间联系的隐式定义中包含的一些属性，并对这些属性要用投影、连接等来实现综合操作，以提取这些联系中的信息。这样，联系的语义是包含在用户程序中的。在语义模型中，操作明确地定义在联系上，于是语义存在于数据模型本身中。

考虑图书馆数据库中处理作者和书的一个例子，图3表示了关系模式，图4是对应的语义模式。

WRITER	NAME	INCOME	STIPEND
BOOK	AUTHOR	TITLE	SALES PUBLISHER
PUB-HOUSE	NAME	GROSS	

图3 图书馆关系模式

一个有趣的查询是确定全部由大出版社出版有成功书籍的穷作者。“穷”、“成功的”及“大”这些量刻画出现在查询语句中，这一查询语句的关系表示为：

```
SELECT NAME FROM AUTHOR
WHERE WRITER. INCOME < 20,000
AND WRITER. NAME = BOOK.
AUTHOR
AND BOOK. PUBLISHER = PUBLISHER.
NAME
AND PUB-HOUSE. GROSS > 1,000,000
同样的查询可用语义模型及Tsichritzis and
ovsky (1982) 的记法表示如下：
SELECT NAME FROM AUTHOR
WHERE WRITER. INCOME < 20,000
AND BOOK. SALES > 50,000 / WROTE
AND PUB-HOUSE. GROSS > 1,000,000 /
PUBLISHED
```

由于联系的语义体现在对关系的查询中明确要求的连接过程内，语义模型的查询比关系模型更紧凑。

(2) 完整性维护 传统的模型迫使用户要么记住数据库对象间的联系，要么通过

物理层联系的导航来维持对象内部的一致性；语义模型提供了完整性约束定义的机制，同时允许用户在不管低层记录结构的层次而查看数据。虽然所有的语义模型都具有这一功能，但程度不同（见1节中关于插入\删除\修改约束的讨论）。

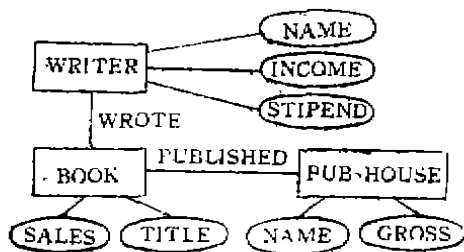


图4 图书馆语义模式

(3) 建模灵活性 多数传统*)的数据模型只提供一种表示数据的方法，而语义模型通过运用抽象概念，允许用户在多个层次上建模和处理数据，这就增强了对“现实世界”建模的能力。如实体-联系(E-R)模型允许用户把两实体间的联系理解为一个联系或一个实体。

(4) 建模效率 设计者在构造数据库模式时，不必在低层次进行，多数语义模型包含内部基本操作和约束。如一引用联系，即数据库中一类对象可通过其属性引用另一类对象，在我们的数据库例子中，BOOK对象可通过其属性AUTHOR引用WRITER对象。引用联系有特定的操作及与之相关的约束。例如，一插入操作及其约束，这一约束指出，如果一引用对象引用了数据库中不存在的对象，它就不能插入，这就把设计者从每定义一个引用联系就要执行一次操作和约束中解放出来了。这并不是一个新思想，在含抽象数据类型的语言理论中也是这样的。事实上，可以说多数语义模型都提供了这些良好的程序设计技术。

自从提出了第一个语义建模结构后，十

年来已建立了许多这样的模型。本文着眼于在语义数据库的逻辑设计和说明中使用的抽象概念模型，分析并比较了这些模型中具有代表性的一些模型。分析表明，这些结构的出现能满足一般语义建模的目标。

一些系统，诸如ACM/PCM提供了对整个数据库系统生命周期的支持，包括说明、设计和维护。这些系统超过了本文的范围，CRIS学术会议着眼于这种信息系统的发展。该会的报告提供了各种计划的深入描述，包括ACM/PCM, ISAC, NIAM和D2S2等。其它这种系统在Braegger等(1985)和Bryce and Hull(1986)中也有论述。

本文强调的是语义数据模型，例如应用于ACM/PCM概念模型中的SHM+，它所描述的实体、操作和约束是我们所感兴趣的，而ACM/PCM的其它部分则不在此范围(见2.8节关于SHM+的讨论)。

1. 比较的基础

近期语义模型的迅速发展使人们认为，在这一庞大的、看来彼此独立的领域里似乎不存在比较的基础。试图提供一些比较基础的作者们通常把语义模型分成若干类，例如，Brodie(1984)把模型分类为经典模型的扩充、数学模型、不可约数学模型、静态语义层次模型和动态语义层次模型；Tsichritzis and Lochovsky(1982)将模型分类为：传统模型、实体-联系模型、二元模型、语义网络模型和信息逻辑数据模型；Hull和King(1987)用了一种略为不同的方法：把由基本语义建模部件组成的一个模型与每一模型进行比较。Hull和King的论文具有特别的教学意义，它用教学数据模型来讨论数据库的语义建模，其中强调的是围绕语义模型发展的数据库系统的实现，而本文则更着重于概念的建模问题。

Kerschberg等(1976)按下列参数：结构的、概念的和语义的将概念数据模型分

*)“传统”意指层次、网状和关系数据模型。

类。他首先应用集合论及图论参数，建立了数学基础来评价模型的结构特征，用表示实体和建立模型的数学结构的方法来阐明模型的概念性质，采用语言学的方法来决定抽象语义概念的层次。

本文阐述衡量每一模型的语义建模能力的全部概念。介绍方式着重于支持联系、联系表示的抽象概念、说明语义的方式和动态建模方法（如果有），其结果可用来比较和对比这些模型，并识别哪种方法能最完美地实现所述语义建模系统的目标。

每一语义模型都有对象（或实体）、联系（函数的或关系的）、动态性质和处理完整性约束的方法。联系可由它们所能表示的抽象概念及其表示方法来刻画，动态性质可通过简单说明操作及事务建模的插入、删除约束引伸而得，约束可由用户收集并由模型的联系语义表示和（或）自动隐含。信息表示的层次和机制均用以描述和比较模型。

综上所述，我们认为下面几点是语义数据模型的基本特征。

(1) 非结构对象的表示 在Tsichritzis and Lochovsky (1982) 中，非结构数据类型被定义为低层次的和初始的类型，它们不能由更低层次的类型聚集而得。串、整数和实数都是低层次的标准类型，基本计算机系统的硬件直接支持这些类型。根据特定应用而开发的某些模型[Christodoulakis等1986; Su 1983; Woelk等1986]提供了比大多数机器或编译程序要详尽的原语，例如支持统计、文本、声音和图象数据的类型。

(2) 联系表示 联系由建模者借助其表示来分析的。从概念上说，联系结构在模型中可以属性、实体、单独的元素或者函数的形式出现。一个由属性体现的联系是这样的，即其中某一对象的属性联接到或指向另一对象，或由另一对象派生出来，例如，如果BOOK类型中的AUTHOR属性被定义为WRITER类型，而WRITER是一实体类型，那么BOOK和WRITER间的联系就是通过

BOOK实体的属性来表示的。

如果联系的两个或多个对象从概念上说描述了一个不同的模型对象，那么它就作为实体来表示。例如，我们可选择REVIEWER和BOOK间的REVIEW联系，将其表示为模型中的实体，它除具有属性RATING和DATE（审校者的）外，还具有属性NAME和TITLE（书的）。

联系可被认为是与实体不同的独立对象，这并不意味着在物理数据库中用不同的结构表示联系和实体，但至少概念层次上，实体被认为是独立于联系的。例如，在图4中WROTE和PUBLISHED联系与WRITER和PUB-HOUSE实体的出现有所不同，在这种情况下，系统的用户会把联系视为两种类型间的简单连结而不是前述的一个独立的实体或属性。

如果允许在数据定义语言中通过函数定义来说明对象的联系，就可以形成函数表示。例如，语句

```
DECLARE AUTHOR(BOOK) =
  >> WRITER
```

允许设计者定义BOOK和WRITER类型间的一个联系，其中给定的BOOK对象的AUTHOR是BOOK对象的一个函数，且是WRITER对象。在所述的系统中，我们将发现各种表示方法的例子。

(3) 现有的标准抽象概念 正如在引论中所讨论的，语义数据库中最常用的抽象概念是：分类、概括、聚集和联合。

(4) 网状或层次联系 几乎所有的语义数据模型都为模式的概念化提供了图示结构，在大多数这类模型中，此图示表示了模型的基本建模抽象概念。最常见的例子是表示对象类型衍生过程的概括/特化图（或称IS-A图），图的本质（层次的与网状的，有圈的与无圈的）在数据模型的刻画中起了重要作用。模型基本抽象概念的图形表示也可不采用IS-A图，例如，图2中给出的情况可扩展为图5的聚集层次，这里PUBLICATION

是TITLE和AUTHOR的聚集，而AUTHOR定义为NAME、INCOME和STIPEND的聚集。其它模型，如实体-联系模型〔Chen 1976〕只支持网状联系，如图6所示。以这种方式，不存在联系间层次结构表示的规定（虽然可隐含）。在第3节，我们考察的各个模型仅限于把层次（相对于网状）作为组织结构。

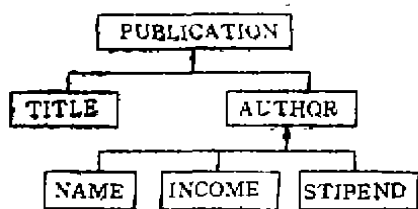


图5 聚集层次

(5) 派生/继承 语义模型处理数据库模式内的重复信息有两种方法：语义连结和派生。单个对象类型的重复可以通过定义两个带有语义连结的独立类型来处理，这样，就降低了冗余度（见2.5节）。类型间的重复用派生来处理。派生这种方法使某一对象的属性来自对其它对象的计算或继承，或者，可以用类的属性来完整地保存某一类对象的派生属性。例如，类型BEST-SELLER的类属性AVERAGE-SALES可按所有BEST-SELLER中实体的SALES数的平均数来计算。

概括层次中的继承是一种将更一般对象的属性传递给更特殊对象的方法。这可看作是一种常见的派生，其中计算是恒等函数，即若PUBLICATION被定义为BOOK的上级类型，且设属性TITLE是BOOK从PUBLICATION继承而来，那么这也可以认为是一种派生，这里，BOOK.TITLE=PUBLICATION.TITLE。

多重继承是一种允许一个概括/特化层次结构中的对象从多个上层对象继承性质的机制，这对于某些应用是很方便的，但可能难以控制。当某一对象从两个上层对象继承同样的性质时，麻烦就出现了。考虑图7的概括/特化和聚集层次，LIBRARY-FIG-

URE类型表示了那些既是作者又是审校者的人，LIBRARY-FIGURE是WRITER和REVIEWER的特化，于是从两者继承性质。REVIEWER和WRITER都有STIPEND属性，这样，由于REVIEWERER.STIPEND指审校者审校一本书所获酬金，而WRITER.STIPEND指写一本书所付的总酬金，所以就出现了“继承”冲突。语义模型中可以禁止多重继承的使用，或为可能出现的冲突处理提供内在的机制。

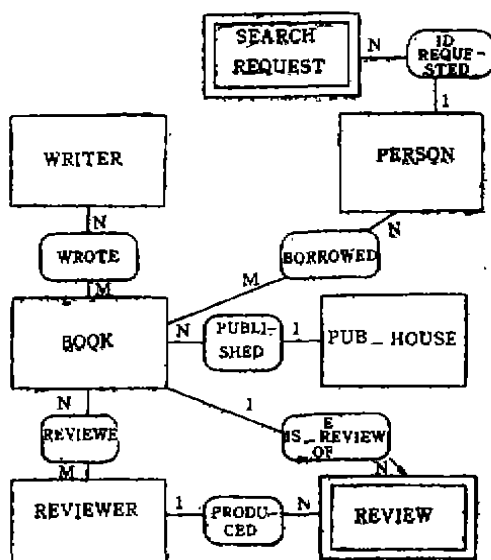


图6 联系网络

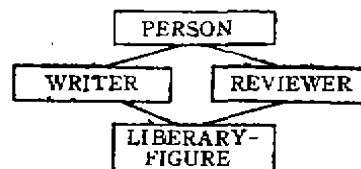


图7 多重继承

(6) 插入/删除/修改约束 插入和删除约束用于维持语义数据库的完整性是其最重要特征之一，约束的说明是对这个模型语义的物理解释和操作解释。如果对象通过联系连结起来，那么一个对象的插入、删除和修改将会影响到所有与其相连结的其它对象的存在状态。

对于数据库设计者和端点用户来说，模

型的联系清晰地反映数据库联系的语义是至关重要的,即,所有的用户都应该对数据库操纵的结果有清晰的了解。显然,意外的副作用是不适宜的,例如,IS-A是一常见的概括联系,如果BOOK IS-A PUBLICATION,那么当PUBLICATION对象从数据库中删除以后,相应的BOOK对象也应删除。应该将IS-A联系的语义明白地告诉模型的设计者和用户。另外,有些模型允许设计者说明联系的插入/删除/修改语义。

(7) 联系语义的表达能力 有的模型给语义表达式赋予主值、空值或倒排联系、派生性、继承性(部分/全部或在那些表示非概括的联系上)或缺省值,其它模型则完全地定义了一个或多个这样的特征。因而,与此相对应,这些模型的一个重要区别就是它们提供给设计者更多的灵活性。

(8) 动态建模 动态建模是指对数据库事务中语义性质的描述。动态建模的反面是静态建模,它是对数据对象和联系性质的描述。在多数情况下,语义数据模型的研究都强调其静态方面,但最小程度的动态建模,如插入及删除约束的说明、前后条件的处理、错误恢复及事务定义,将出现在大多数建模系统中。

有些模型在帮助用户定义高级操作和事务时能扮演积极角色。某些方法中只在其修改对象的层次上有用,而在另一些方法中则根据面向对象的模型来定义建模原语,其中操作被封装在它们所修改的对象中。参看Dittrich(1986)对面向对象的一般数据模型的描述和Dayal and Dittrich(1986)对一些特殊模型的详细描述。然而,这些模型的重点仍是静态对象而非动态事务。除了面向对象的观点,一些模型还支持一个概括层次方法,它是数据驱动的或是比数据驱动更具有时态性的信息流方法。

通过将基本的建模抽象概念应用到事务上,一些模型使事务的描述前进了一步。通过运用特化和聚集,这些模型能支持更复杂

的事务。有关这些技术的具体应用在第三节讨论,而有关支持综合静态、动态方面的信息建模系统则已超出本文范围,NIAM [Verheijen and Van Bekkum 1982]和ISAC [Olle et al. 1982]中有所论述。

1.1 图书馆例子

这里构造了一个图书馆例子来表示本文所讨论的各种模型,以便给读者提供一个比较模型的非形式化途径。这个例子说明了各模型的某些有趣特征。该例是讲授性的,因此并不打算表示一个图书馆数据库各个方面的完整描述。数据库中的元素如果是通过已表示过的特征重复而被模型化,那么它就不包含在内,因为它们不会增加读者对模型的理解。

本例选择的企业模式是一个图书馆。为了不使读者偏向一种建模方法,也不假设读者以前具有数据模型的经验,此图书馆的信息以文本形式给出。假设下面给出的语句是通过会见一个图书管理人员而获得的,并且插入了会见者的一些数据库术语。对第2节介绍的每个语义数据模型,我们根据下列语句建立了一个图书馆的概念模型:

(1) PUBLICATION实体有TITLE和AUTHOR两个属性;

(2) BOOKS、JOURNAL-PAPERS和CONFERENCE-PAPERS是PUBLICATIONS的几种特殊种类;

(3) BOOK实体有SALES、TOPIC、AUTHOR、PRICE、PUBLISHER、INTEREST-INDEX和ACQUISITION-PRIORITY属性;

(4) BEST-SELLING-BOOK的SALES超过10,000;

(5) WRITER实体有INCOME、NAME和STIPEND属性;

(6) BOOK中的AUTHOR属性为WRITER类型;

(7) PUB-HOUSE实体有NAME和GROSS属性;

共享库概述

徐泽同

涂健

(中国科学院数学所) (铁道科学院电子所)

摘要

本文简略介绍以多台可能不同类型的数字计算机作为顾客,一台数据库计算机作为服务员,经通讯设备连接而成的共享库的体系结构和性能,也分析共享库对实现分布式数据库管理系统产生的影响。

一、引言

数据库技术的研究,已30余年的历史了。30年来从文件系统发展为层次、网状、关系型数据库系统。而今在集中式关系数据库管理系统方面已有DB2、DATABASE III、INGRES、ORACLE等著名产品;在分布式关系数据库管理系统方面也成功在望。

这是成功的一面;但另一面,由于上述这些研究,都是以通用数字计算机为基础,在此基础上实现的系统,当用户增多时其响应能力下降,因而近十余年来又激起了人们对数据处理专用的数据库计算机的研制,而今实现数据库机已不成问题。这种机器虽长于数据处理,但又失去了通用数字计算机的长

(8) PERSON实体有一个NAME属性;
(9) REVIEWER实体有一个STIPE-ND属性;

(10) LITERARY-FIGURE既是WR-ITER又是REVIEWER;

(11) BORROWED实体有PUBLICA-TION和DUE-DATE属性;

(12) LIBRARY实体有ACQUISITION和ORDERED-PUB-LIST属性;

(13) DATABASE-BOOKS集合包含所有TOPIC=DATABASE的BOOK且具有TOTAL-COST属性;

(14) AI-BOOKS集合包含所有TOPIC=AI的BOOK;

(15) GOOD-BOOKS集合中的BOOK是由端点用户指定的;

(16) DB-AI-GROUP-BOOKS集合由既是DATABASE-BOOK又是AI-BOOK的

BOOK组成;

(17) 集合RESEARCH-GROUP-CO-LLECTIONS中所有的BOOK都是GOOD-BOOK;

(18) WROTE是BOOK和AUTHOR间的联系;

(19) PUBLISHED是BOOK和PUB-HOUSE间的联系;

(20) REVIEW有属性RATING和DA-TE,是BOOK和REVIEWER间的联系;

(21) REVIEWER是一个PERSON;

(22) HOLD是BOOK和LIBRARY间的联系。

(未完待续)

[纪岳、何未艾、王红、赖维生、陆强、钟颖译自、姚卿达校自ACM Computing Surveys, 1988, vol. 20, №3]