

复杂对象数据库研究：历史和现状

石桥林 (吉林大学计算机系)

摘

要

本文简要回顾了复杂对象数据库的发展历史，比较全面地介绍了国际上当前正在开展的研究工作，尤其是对复杂对象模型的形式化理论研究工作做了重点介绍。本文最后所列的文献可作为读者进一步了解这方面工作的指南。

本文将复杂对象数据库的研究分为两个阶段，1986年以前为早期，1987年以后为近期。这样划分的主要依据是复杂对象数据库研究的发展水平，同时也是为了便于叙述。对每一阶段的工作，都将从复杂对象模型的形式化理论研究及复杂对象数据库系统的建造这两个主要方面进行介绍。

1. 早期研究工作(1986年以前)

数据库技术中，关系模型的产生和发展大大促进了计算机信息管理在很多领域中的应用，但是，随着应用的深入和范围的扩大，传统关系模型的局限性也日益突出，主要是以下三个方面：

(1) 基于第一范式(1NF)的关系模型，即平关系模型(flat relation model)的数据抽象能力不足，在表达复杂对象数据时，语义不直观，语义信息有丢失，而且结构不灵活。

系统可以“通告”铁路某部分何时成为瓶颈，并可以建议调度机构减少该地区的车辆。一个简单的方法是定义一些规则，统计在一段时间有多少列车通过一特定区域或车站等。还可以定义其它一些规则，标记何时一个区域的统计数比其余区域高得多。

这些例子说明了运用规则管理系统作为应用程序的智能数据操纵工具的技术。所讨论的应用程序可以作为专家系统外壳。这样，有些规则可以对DBMS规定，而另一些

(2) 在诸如工程数据管理、CAD/CA M、办公室信息系统(OIS)等许多领域中，需要增加数据库的语义内容，以使更多的信息能以可用的形式包容在数据库中，而不是分散在访问和操作数据库的各种程序中。但是在平关系模型数据库中要实现这一点是不可能的。

(3) 面向对象的程序设计也对数据库提出了面向对象管理数据的要求，以消除程序设计语言(包括非面向对象的程序设计语言，如 Pascal、C、等)与数据库管理语言及数据库数据结构之间的不匹配。

复杂对象模型正是为克服平关系模型的这些缺陷而产生的，最初是彼此独立地出现在一些应用领域中，在术语上并不一致，例如语义数据库模型(SDM)^[66]、非第一范式关系(\neg 1NF)^[40, 63]、格式模型(format mod-

规则可由外壳实施。在这两个系统之间研制最好最小的接口是一个值得进一步研究的有趣课题。

参考文献(略)

〔徐宝文译自《proceedings from the second International Conference on Expert Database Systems》, The Benjamin/Cummings publishing Company, 1989 声钟仲源校〕

el)^[67]、工程数据模型(EDM)^[16-78],等等。早期的这些工作主要集中在工程数据管理、CAD/CAM、以及OIS等领域中。

由于复杂对象模型与传统的平关系模型相比具有很大的优越性,在这些早期工作的推动下,关于复杂对象模型的形式化理论研究也迅速得到了广泛的重视。1984年,MCC建造了一个面向对象的数据库原型IRIS,在此原型系统上进行了复杂对象模型的形式化理论及实现技术的初步研究^[15]。Carlo Zaniolo在1985年的VLDB大会上提出了一种扩展关系代数(ERA),用于复杂对象数据的操作;这一工作标志着复杂对象模型的形式化理论研究的开端;但是该文所处理的复杂对象并不具有一般性,只是一种特殊情形——嵌套关系,不过比标准的嵌套关系要复杂一些,例如属性的值可以通过多种方式求得。

早期的形式化理论研究主要是围绕嵌套关系进行的。在1985年的ACM SIGACT-SIGMOD数据库系统原理大会上,Z.M.Ozsoyoglu及Li-Yan Yuan介绍了他们关于嵌套关系的研究;他们给出了嵌套范式(NNF)的形式定义,在此基础上提出了一种将给定泛关系模式U关于多值依赖集M进行嵌套范式分解(NNF分解)的算法。S. Abiteboul提出了一种嵌套关系代数;P.S.Fisher(Vanderbilt大学)也一直在从事这方面的研究工作^[46]。

针对更一般的复杂对象的形式化理论研究也有所开展。1986年,F. Bancilhon及S. Khoshafian在ACM SIGACT-SIGMOD数据库系统原理大会上提出了他们构造的一种复杂对象演算系统;他们首先定义了“归约对象”的概念,然后基于子对象关系将对象集抽象为一个偏序格,给出了归约对象集中对象之间的 \cap 、 \cup 两种运算;在此基础上,他们定义了wff及规则,以及它们各自的形式化语义,从而形成了一个复杂对象演算系统,并且允许规则中存在递归。

S. Abiteboul及R. Hull给出了“型”及“型的范式”的形式定义;在“型”的定义中引进了“记号(token)”概念,从而初步建立了型的记号模型;下面所列是他们对“型”的定义:

定义: 型是一棵有根树(V, E),其中 $V=V_+ \cup V_* \cup V_\times \cup V_0$ 是所有+节点、*节点、 \times 节点、以及原子节点(它们分别对应于元组构造子、集合构造子、联合构造子、及原子型)的不相交并集,并且:

- (1) 一个节点是原子节点当且仅当它是叶节点;
- (2) 给一个节点赋予一个属性名,当且仅当它是原子节点;
- (3) 同一节点的不同子节点被赋予不同的记号;
- (4) *节点只能有一个子节点。

他们还基于型的定义及其记号模型给出了“型的域”的定义,从而定义了“型化的对象”,在此基础上建立了型化对象的一种代数系统,引入了重写操作,给出了相应的形式语义;他们最后讨论了型之间的信息容量比较及型的重构,并且得出结论:重构变换的有限序列可表达为一个简单重写操作。

由于复杂对象都具有层次式结构,因此建立型和对象的记号模型对于方便复杂对象的管理和共享、提高复杂对象的操作效率具有重要意义。记号模型的思想起源于使用标志符标识对象的做法^[68]。对象的记号模型与型的记号模型类似^[7,24,68]。

上述研究工作构成了复杂对象模型的形式化理论研究框架,为复杂对象数据库的设计及更进一步的理论研究提供了指导。

这期间也发表了一些复杂对象数据库建造方面的成果,例如S. Abiteboul及R. Hull的语义数据库模型IFO、惠普公司的IRIS、俄勒冈研究生中心的TEDM及GemStone,等等。这些实验性系统的开发促进了复杂对象数据库理论研究的深入及其在世界范围的普及。

另一重要的早期工作是关于“复合对象”的研究。使复杂对象数据库支持复合对象就是使它具有将一组通过IS-PART-OF关系相互关联的一组对象定义为一个逻辑实体，并对之施行某些操作的能力，这种能力对语义完整性、有效存/取、以及并发控制等具有重要意义。这方面的早期工作见[14]及[62]。

值得指出的是，至1986年前后，由于专家系统的发展，知识的有效管理成为日益严重的难题，因而从复杂对象数据库领域中又派生了一个子域：面向对象的知识系统^[65]，也称为“面向对象的知识库管理系统”^[69]，这两个术语是完全相同的。但是，这方面的研究迄今仍处于起步阶段。

2. 近期研究工作(1987年以后)

约从1986年开始，大量的研究工作都侧重于实验性复杂对象数据库和复杂对象操作语言的设计，下面列出一些比较典型的例子。

COL 这是一个基于逻辑的复杂对象操作语言，由INRIA的S. Abiteboul及S. Grumbach研制。他们针对由元组和集合构造子构造的型化对象给出了该语言的形式定义及语义，并讨论了它在过程数据、语义数据模型等方面的应用。

LPS(Logic Programming with Sets) 该语言以多类逻辑中的二类逻辑为基础，能处理的复杂对象只能是原子对象及原子对象的集合；该语言可看作是一阶逻辑和Horn子句逻辑的扩充^[69]。

LDLI 该语言由MCC的C. Beeri等人研制，类似于LPS，但功能更强一些。

FAD 这也是由MCC开发设计的一种复杂对象操作语言^[13]。

IRIS 是由惠普公司实验室实施的一项研究计划的成果，一个实验性复杂对象数据库系统，其中包括了IRIS OODBMS及IRIS SQL(一种以SQL为基础进行扩充而得的复杂对象查询语言)。

VODAK 是由GMD-IPSI研究所实施的VODAK计划的核心数据模型，该模型能支持复杂对象、抽象数据类型、继承及消息传递，是一个比较全面的复杂对象模型；该模型中也采用了符号模型的概念。

ALGRES 是欧洲共同体ESPRIT计划支持的

一项研究工程，包括ALGRES复杂对象模型及复杂对象操作语言ALGRES-PREFIX。

TEDM 是俄勒冈研究生中心早期研制的一个复杂对象数据库，并得到不断改进和完善。

O₂ 是Altair协作工程中开发的一个复杂对象数据库系统，支持由元组构造子及集合构造子构成的复杂对象，支持型、方法和继承，并且提供了一种数据库程序设计语言和环境CO₂(其主要功能是复杂对象的操作)。

值得特别指出的是，上述几个复杂对象数据模型中，TEDM、O₂及VODAK是支持方法和继承的，因而更接近于面向对象的程序设计中的“对象”概念，但是迄今的多数复杂对象模型都不考虑这两点，而只处理纯数据的复杂对象，但是，从发展趋势看，考虑方法和继承是必然的。这方面目前已开展的工作还有：

C. Lecluse等人在O₂模型中，根据所建立的符号模型对方法作了形式化定义，初步讨论了方法的模型论语义，然后在此基础上给出了方法间的一种偏序关系，反映了复杂对象模型的“方法继承”特征。

关于继承，近年来在复杂对象研究中出现了一些关于“多重继承”的研究成果^[25, 42, 87]，多重继承在有的文献中也称为“多重映射(polymorphism)”，一些成果见于[32, 73, 86]中。

在[73]中，C. Lecluse等人对多重继承进行了深入的讨论，给出了继承和多重继承的集合包含语义(set inclusion semantics)。

值得特别一提的是，在面向对象的程序设计研究领域中，人们对继承和多重继承已做了很多深入的研究，其成果无疑是可供复杂对象数据库领域中的相应研究利用和借鉴的。

一系列实验性复杂对象数据库及复杂对象操作语言的研制促进了对一些基本实现技术的研究，主要包括：

- (1) 复杂对象的存储管理^[80, 105]
- (2) 对象查询的优化及执行策略；
- (3) 操作语言中各种算子的有效执行策略^[66]。

主要指三方面的实现策略:(i)复杂对象的散列(hashing)及散列值附带(tagging);(ii)唯一性和不重复性的推断;(iii)有重复时复杂对象算子的执行。

(4)版本管理、事务管理、授权模式动态进化等的研究。MCC在复杂对象数据库系统ORION的开发研究中,对这些问题都作了比较深入的探讨,使复杂对象数据库向实用化方向有了新的进步。

关于复杂对象查询的优化,目前已有的工作见[19, 45, 49, 103];此外,IRIS复杂对象数据库中有一个基于规则的查询优化器,它根据启发式规则首先产生一个执行查询的计划,然后利用规则和一个基于A*算法的穷尽搜索算法进行寻优。

这方面工作的一个共同特点是,针对具体的复杂对象代数系统(或演算系统)总结出查询表达式转换的知识(以产生式规则表示),然后利用这些知识对查询表达式进行优化。这些知识还可能涉及到应用级语义以及关于查询的实现技术的知识。

关于复合对象的研究,由于对复杂对象数据库管理具有重要意义,得到了进一步的开展。W. Kim及J. Banerjee等人在ORION中实现了一种复合对象机制,给出了复合对象的形式定义及语义,并对基于复合对象的模式进化、版本管理、对象实体的存储、并发控制中的粒度封锁等问题作了深入的讨论。

在实验性复杂对象数据库的研制中,一项有代表性的研究工作是西德信息研究所的Heuer自1988年起在IFO的基础上着手研制一个具有四个描述级的所谓“关系型对象管理系统”EXTREM,其中的四个描述级自上而下依次是:受限的IFO模型、扩充的实体-关系模型、一种综合了NF²关系及RM/T关系的所谓EXTREM关系模型、以及传统的平关系模型;同时还提供了这四个级依次之间的映射。该系统在复杂对象的操作上,实现了一个与带复杂投影及选择的NF²代数类似的对象代数。

A. Heuer的这一研究工作代表了一个重要的研究方向:利用传统的平关系模型实现

嵌套关系模型,进而实现受到一定限制的复杂对象模型。这方面已开展的工作还有[41, 44, 56, 75, 100]。这些工作与前面介绍的关于嵌套关系的形式化理论研究是紧密相关的;正因为如此,近年来关于非1NF关系代数和非1NF关系演算的研究受到了广泛的重视^[35, 37, 67, 81]。这些工作的共同特点是,充分利用传统1NF关系数据库理论中的成果,进行改造和推广,例如将传统1NF关系代数中的基本运算、交、并、联接、笛卡尔积等进行扩展,以使其适合嵌套关系的操作。

对复杂对象数据库的研制,这些工作的意义是明显的,但究竟能有多大,这取决于实际应用中各种复杂对象模型映射到嵌套关系模型的可能性和可行性的多少,以及由此实现的复杂对象数据库所能达到的潜在效率的高低,还有待实践的验证。

在复杂对象模型的形式化理论研究方面,对象和型的符号模型得到了进一步的发展^[25, 30, 74, 85]。在[25]中,给出了对象符号模型的形式化描述,而关于型的符号模型,目前尚未见有形式化定义;但由于型是对象的抽象,因此可以断言这二者的符号模型之间有相通之处。在复杂对象模型中引入符号模型的意义在于:

(1)使对象的管理、操作与对象的实际值分离,从而可利用对象标识符进行。这对提高管理与操作的效率有很重要的意义。

(2)数据共享可通过对象标识符间接实现,从而避免了对数据实体的重复拷贝和传输。这一点对分布式复杂对象数据库尤其重要。

(3)通过符号模型可以将对象或型分层“截断”^[65],从而将涉及复杂对象的高阶推理分解为不同的层次,以适合于一阶推理。

在复杂对象数据库TEDM的进一步研究中,Jianhua Zhu等研究者引入了“抽象对象”,用于复杂对象查询中的匹配;抽象对象表达的是与相应对象相关的询问和操作指令,与复杂对象一样存储在永久性数据库中,从而解决了传统数据查询语言不能支持

通用查询重用模式、难于定义可在多个事务中使用的查询过程、以及难于将查询进行复合等等问题。此外，他们还定义了一种型演算，用于在TEDM中定义型和对型进行操作。

在型方面，人们研究了各种型作为一个系统（型体系）所具有的性质。〔74〕中给出了型的一个形式体系，内容包括型结构、型结构之间的一种偏序关系（反映了复杂对象模型中的“属性继承”特征）、型及其体系；还给出了型的一种模型论语义，以及子型关系与型结构间的偏序关系的联系。这些成果构成了复杂对象数据库 O_2 的理论基础。

GMD-IPSI研究所的H. Duchêne等人也提出了型的一个类似的形式体系，而且也采用了符号模型。这两个体系之间的差别仅在于术语上的不一致：〔74〕中区分型与型结构，并将对偶 $(t, m) \in \text{TypeId} \times \text{MethodId}$ 称为型，其中TypeId及MethodId分别是型结构标志符集和方法标志符集；而Duchêne等不作此区分，他们将 (t, m) 称为方法。由于这一区别，在这两个形式体系中，对型与方法的处理便有所不同，但实质却是一样的。

在1988年的第二届国际数据库理论大会上，宾夕法尼亚大学的A. Ohori公布了他所定义的一个简单的型化复杂对象语言，该语言的一个重要特点是允许表达式中出现广义联接和广义投影，同时还形式地定义了型体系 (DBType, \leq) ，其中 \leq 是型集合DBType上的一个偏序关系；Ohori依次构造了单个型的语义、型体系的语义、广义投影的语义、以及表达式的语义，从而得出了该语言的完整指称语义。

关于对象代数的研究，1988年的第二届面向对象数据库系统国际工作会议上，Western Ontario大学的S.L. Osborn报告了他的研究工作：按照Wegner对复杂对象的定义建立了一个复杂对象模型，并定义相应的对象代数，其主要特点是，除通常的集合操

作之外还提供了几个特殊的对象操作，包括Combine（组合）、Partition（划分）、Choose（挑选）、及Name（命名），因而具有较强的复杂对象操作能力。Osborn还给出了相应代数表达式的转换规则。

在对象演算的研究中，〔25〕从型的重构出发得出了一个对象演算系统，其主要工作是引进了虚型和相容重构两个概念，并说明虚型的相容重构表达式即为对象演算中的规则。

3. 结束语

本文对复杂对象数据库的发展历史和近况作了一个梗概性的介绍，从中可以看到，关于复杂对象数据库的研究大致可分为三个方面：

(1) 复杂对象模型的形式化理论研究，包括复杂对象的定义、型的定义、型的形式体系、方法的形式体系、继承、对象代数、对象演算、型的重构等主要内容。

(2) 复杂对象数据库的技术基础研究，目前已开展的工作主要是：嵌套关系模型的理论研究和实现、受限对象模型与嵌套关系模型之间的映射、嵌套关系模型与平关系模型之间的映射、面向对象的物理存储模式、复合对象，等等。

(3) 实验性复杂对象数据库的开发设计，包括复杂对象操作语言的设计（已设计的语言中，既有针对嵌套关系等特殊复杂对象的，也有针对一般复杂对象的）、复杂对象查询的优化、并发控制、版本管理、模式动态进化，等内容。

表1中列出了国外的一些主要研究机构及其相应的研究工作。从表中我们可以注意到，无论从数量上看，还是从工作的深入程度看，公司或其研究机构占了很重要的位置，这也从一个侧面反映了复杂对象数据库所具有的潜在应用价值是很大的。

对象这一概念，以及面向对象的结构，是知识系统、数据库、以及程序设计语言这三个领域中的一个最有前途的统一模式，这可从以下三个事实中看出：

(1) 面向对象的程序语言已经得到了很大的发展；

(2) 复杂对象数据库方面已开展了大量的研究，正在向实用化发展；

(3) 知识的框架表示法和语义网络表示法在AI领域中应用广泛，而这两者在结构上都可以抽象

表1 国外一些主要研究机构及其开展的工作

机 构 名 称	研 究 内 容	主要文献
MCC (微电子和计算机公司)	LDL1 FAD 复杂对象操作语言 ORION 复杂对象数据库	[13,22,98] [19,20,27,48]
INRIA (法国国家计算机与控制研究所)	COL 复杂对象操作语言 IFO 复杂对象模型(与Brown大学合作)	[6] [8]
Altair协作工程(由法国INI、LRI及INRIA资助)	O ₂ 复杂对象数据库	[12,21,73,74]
ESPRIT工程 (欧洲共同体ESPRIT工程)	COMANDOS CIS 复杂对象模型及操作语言 ALGRES 复杂对象数据库,包括ALGRES-PREFIX语言,(由意大利的Modena大学等单位承担研究任务)	[18,26,29] [30,31]
Oregon Graduate Center (俄勒冈研究生中心)	TEDM 复杂对象数据库(得到Intel, DEC, Xerox, Apple 等公司资助) Gem Stone 复杂对象数据库	[77,104,106] [80,89]
Hewlett-Packard (惠普公司实验室)	IRIS 复杂对象数据库	[15,16,28,43,71,76,93]
ECRC (欧洲计算机工业研究中心)	KB2 复杂对象数据库	[101]
GMD-IPSI (西德GMD-IPSI研究院)	VODAK工程 包括复杂对象数据库的研究	[36]
Univ. di Pisa (比萨大学, 意大利)	GALILEO 复杂对象操作语言	[1,4,8]
CCA (美国计算机公司)	PROBE工程 包括复杂对象模型PDM Hi PAC工程 FUGUE 复杂对象模型(与Brown大学合作)	[33,39,40,78,84] [34] [58,59]
UMA (麻省大学, Amherst)	ADABTPL DB系统开发环境,支持复杂对象 Mneme工程 开发复杂对象数据库与程序设计语言间的接口Mneme	[96,97] [82]
Texas Instrument Inc. (德克萨斯仪器公司)	ZEIGEIST 复杂对象数据库	[94]
Univ. of Wisconsin—Madison (威士康辛大学)	EXODUS工程	[23,47]
Univ. of Texas—Austin (德克萨斯大学, Austin)	GENESIS工程 开发一个可扩充的DBMS, 具有处理非INF关系的能力	[9,10]
Rensselaer Polytechnic (Rensselaer工学院)	ROSE 复杂对象数据库	[50,58]
Institute für Informatik (信息学院, 西德)	EXTREM 复杂对象模型	[51,52,53]

图 象 理 解 及 其 推 理 机 制

丁德恒 吕维雪 (浙江大学科仪系)

摘 要

在研究实用图象理解系统过程中,人们逐步认识到推理机制的重要性。本文从推理方式的角度讨论图象理解中存在的问题及其产生原因。重点分析多重信息融合方法和假设推理过程。

图象理解过程一般可分成两大步骤:

(1) 分割 (Segmentation) 从输入图象中抽取图象特征 (边缘或区域), 形成图象的结构描述。

(2) 匹配 (Matching) 将图象特征与待识别对象的模型进行匹配, 形成对图象所表示的景物的描述。

当我们要构造一个图象理解系统 (Image Understanding System), 对复杂景物能进行稳定分析时, 仅仅孤立地研究分割或匹配的个别算法是不够的, 还必须将分割与匹配这两个处理过程有机地结合起来考虑^[1]。也就是说, 必须研究贯穿于这两个过程中的推理方式和控制机制。由于在一个复杂的场景中, 各种物体交错排列, 照明环境难以描述, 阴影部分或高光强部分受相互反射的影响极大, 所以各算法所基于的模型 (如边缘检测的Step-Edge model, Shape from shading的漫反射模型) 只适用于图象的局部。对于理解复杂景物, 必须使用局部不同的图象、照明、物体模型。为此, 不仅要在匹配过程中计算图象特征与识别对象的

相似度, 还必须生成对图象的哪一局部应采用何种分析模型的控制信息和解析规则, 以指导整个分割过程。

一、图象理解中存在的问题

图象理解中产生的主要问题有:

(1) 分割问题

(1a) “有意义”的图象特征有时被漏检;

(1b) “无意义”的信息有时被误检测为图象特征。

(2) 匹配问题

(2a) 与某一图象特征相匹配 (往往是部分匹配) 的识别对象模型有多个;

(2b) 由于遮挡和阴影, 使识别对象的某些局部被消去, 故只能进行不完全匹配;

(2c) 因图象特征的模糊性, 需要有评价匹配优劣的标准。

一般来说, 分割过程中存在的上述两个问题是不可避免, 因为:

为复杂对象。

这就引发了人们对面向对象的程序设计与复杂对象数据库综合、逻辑程序设计与复杂对象数据库综合的研究兴趣。这两方面的研究对开发强功能的程序设计语言、强功能的知识系统具有重要意义。

事实上, O₂中就已强调了前一方面, IRIS及ALGRES中则强调了后一方面; 尽管如此, 这两方面的工作, 尤其是后一方面的工作, 仍只是刚刚起步。

(参考文献共106篇略)