计算机科学 1994 Vol. 21 №. 5

# 知识库系统和语言的演变

朱扬勇　施伯乐

（复旦大学计算机科学系　上海 200433）

**摘　要**　In this paper, the development of knowledge-base systems and languages during the past decade is discussed. Three kinds of knowledge-base systems which have been implemented since 1982 are intruduced. They are "Prolog+SQL systems", "DATALOG extended systems" and "Procedural language+DATALOGe systems". A brief survey of these systems and the future work of knowledge-base systems and languages are given also.

**关键词**　Knowledge-base system, Database system, Logic programming.

## 1. 引言

Prolog 作为一个良好的 AI 语言已经得到广泛接受，市场上已经有多种 Prolog 产品（包括 Prolog 机），但是，Prolog 不能有效地管理持久数据且"每次一个元组"的求值方式不适用于数据密集型应用（如：CAx、CIMS、专家系统、……）。另一方面，数据库管理系统（DBMS）能够有效安全地管理大容量数据，但不能处理递归，而递归是上述基于知识系统的基本能力。因此，研制同时具有 Prolog 和 DBMS 能力的新型数据系统就成为必要。这类系统称为知识库系统、或者演绎数据库系统、或者专家库系统。虽然研究侧重略有不同，但目的一致。

知识库研究始于八十年代初，日本第五代机计划中将知识库作为其核心。经过十几年的研究和实践，知识库研究取得了许多成果，提出了众多的查询优化算法，并且研制了许多实验系统。关于"知识库出论文，面向对象数据库出系统"的论断在 1990 年前后引起了研究者的重视[14,15]，1990 年的 SIGMOD 会议还对这一论断设了专题讨论。因而，近几年研制的系统比较强调实用性。本文根据知识库研究的发展，将知识库语言的演变分成"Prolog+SQL"、"Prolog 扩充和 DATALOG 扩充"和当前的"过程语言+DATALOG 扩充"这样三个阶段，并对每个阶段的典型系统进行了讨论。

## 2. Prolog+SQL

AI 与 DB 技术相结合的最简单办法就是 Prolog 与 DBMS 的松藕合，即在 Prolog 与 DBMS 之间建立一个接口。这类系统主要出现在 1988 年以前，1989 年的拉古那海滩会议认为这类系统前景不好[5]，并且 1993 的拉古那海滩会议再次否定了这类研究[6]。下面我们介绍几个典型的系统。Prolog 作前端、DBMS 作后端是这类系统的特征。

**JCV 系统**[1]　是第一个这样的系统，由纽约大学商学院于 1984 年完成。该系统是在 Prolog 与 SQL 之间建立了一种中间语言 DBCL，DBCL 是 Prolog 的一个无变量的子集。程序执行时，一个要查询数据库的 Prolog 语句，首先翻译成 DBCL 语句，然后转成 SQL 语句由 DBMS 执行返回结果。DBCL 中间语言有一个优化器，而 SQL 的优化由 DBMS 完成。JCV 有两个假设：①查询结果是小的，从而可以装入 Prolog 的内部数据库；②在任何的 Prolog 语句中，数据库谓词都是集中在一起的。

**PROSQL 系统**[2]　是 IBM 研究所于 1986 年完成的。在 PROSQL 中，Prolog 加入了一个一目谓词 SQL(X)，其参数为任何定义操纵数据库的 SQL 语句。所有的数据库查询结果都装入 Prolog 空间，因而对数据库的访问就好象使用 assert 和 consult 谓词一样。

**EDUCE 系统**[3]　是欧洲计算机工业研究中心于 1986 年完成的。EDUCE 以两种方式（松藕合与紧集成）实现 Prolog 与 DBMS(INGRES)的通讯。在 EDUCE 中，Prolog 加入了一些新的访问数据库的谓词，这些谓词以 QUEL 命令为参数。对于松藕合方式，用 Prolog 进程与 INGRES 进程通讯（传递数据）来实现这些谓词；而紧藕合是直接用 INGRES 的访问方法来实现的。

**BERMUDA 系统**[4]　是威斯康辛大学于 1987 年完成的。在 BERMUDA 中，规则存于 Prolog，事实

朱扬勇　博士生，研究兴趣为数据库、知识库系统，施伯乐　教授，博士生导师，长期从事数据库、知识库研究。

存于外存的数据库中。访问外存数据库对用户是透明的。BERMUDA 的实现也是 Prolog 与 IDM(智能数据库机)数据库系统的松耦合。

## 3. DATALOG 扩充与 Prolog 扩充

由于松耦合系统的性能很差，集成系统成为研究热点。首先是扩充 DBMS，如：POSTGRES[7]；其次是基于 DATALOG 的系统，如：NAIL![8]，LDL[9]，KBASE[10]；另外，扩充 Prolog 使其具有高效管理大容量数据的能力的研究也在进行。如：NU-Prolog[11]。

1985 年提出的 DATALOG[12]语言为"每次一个集合"地处理递归奠定了基础。从表达能力上看，DATALOG 为不带函数的 Prolog，但 DATALOG 是"每次一个集合"地求值并且是纯说明性的，而 Prolog 是"每次一个元组"地求值且是过程性的(而这正是松耦合系统低性能的主要原因之一)。由于 DATALOG 的表达能力较弱，所以一般将其扩充到支持函数、否定以及集合。以 DATALOG 为基础实现的系统，由于 I/O 及数据库更新难以处理，而无法实用。这一现象是导致"知识库出论文，面向对象数据库出系统"的论断的主要原因。

**POSTGRES 系统**[7] 是加州大学伯克莱分校于 1986 年提出并一直在研究的系统。POSTGRES 是 INGRES 的后继，其目标是通过简单自然地扩充 INGRES 来开发一个知识库系统。POSTGRES 的语言是 POSTQUEL，它是 QUEL 的扩充，最主要一点是增加 retrieve * 语句来实现递归查询。这样，任何 DATALOG 规则都可以用 POSTQUEL 语言表达。

**NAIL! 系统**[8] 是斯坦福大学研制的，该系统始于 1985 年并一直在研究。NAIL! 是 DATALOG 的扩充，支持函数和否定，是说明性语言。这是一个实验系统，实现了大量的查询优化技术(如：MAGIC-SET，COUNTING 和左右线性变换等等)。系统将 NAIL! 原程序翻译成中间语言 ICODE，ICODE 是扩充了的关系代数语言，有一个优化器对其进行优化。解释器把 ICODE 语句翻译成 SQL 语句进行求值。

**LDL 系统**[9] 是美国 MCC 集团提出的。该系统始于 1985 年，一直到现在仍在研制当中。LDL 是一个雄心勃勃计划，试图从零开始开发一个知识库系统。LDL 也是 DATALOG 的扩充，它支持函数、否定、集合以及 I/O 和 DB 更新操作，并且仍然具有说明性语义。目前实现的 LDL 版本采用了大量的查询优化技术，但还没有开发出一个 DBMS 来支持它，有关数据库的操作也还只是内存的数据操作。

**KBASE 系统**[10] 是复旦大学研制的，1990 年完成了第一个版本，在 MICROVAXII 机器上用 INGRES 数据库和 C 语言加以实现。1991 年用 ORACLE 数据库和 C 语言实现了一个微机版本。KBASE 语言和 NAIL! 一致，也是说明性语言，实现技术也基本相同，此外，KBASE 还提出并实现了模式规范化和查询模式等优化技术。

**NU-Prolog 系统**[11] 是墨尔本大学研制的。它是 Prolog 的一个扩充，具有 DB 更新、查询、事务等谓词。在处理上，加入了自底向上的求值方法，并采用了超级连接技术(superjoin)。它的研究重点不在查询优化上。

## 4. DATALOGe 嵌入过程语言

进入九十年代后，由于知识库的查询优化技术已经达到一定程度，DATALOGe(指 DATALOG 的某种扩充)作为知识库查询语言已经得到认可，实用性就成为近两年知识库系统实现的目标。前十年的研究表明：扩充说明性的语言 DATALOG 使其成为通用的知识库程序设计语言是困难的，因此将 DATALOGe 嵌入过程语言就成为必然。下面介绍的系统都是在 1991-1993 年期间研制的，并且仍在研制当中。它们都能够独立地(不再需要其它语言或系统)用于开发应用系统。

**Glue-Nail 系统**[16,17] 是斯坦福大学研制的，以 NAIL! 作为查询语言。Glue 是一个过程语言，支持 I/O 和 DB 更新操作，能处理集合、聚合运算，具有 REPEAT/UNTIL 循环语句和 Glue 过程模块。在 Glue-Nail 语言中，NAIL! 语句被编译为 Glue 代码，然后程序由一个解释器执行。

**CORAL 系统**[18,19] 威斯康新大学研制的。其语言由说明性的查询语言、命令子语言和 C++ 扩充语言组成。它的查询语言是 DATALOG 的扩充，支持集合、聚合否定，带变量的 EDB；其命令子语言支持数据库的查询、更新，具有顺序、循环等结构，能够在 CORAL 提示符下交互执行；其 C++ 扩充语言是在 C++ 中增加了新数据类型和结构，能够直接访问数据库。

**COL 系统**[21] 是 INRIA 研究所研制的。COL 是 DATALOG 的扩充，是说明性的，它支持集合构造器和数据函数。COL 的最终目标是开发面向对象的知识库程序设计语言。当前有两个初步实现的系统：一个是用 CAML 语言开发的；另一个是用 PROLOG 开发的。相应地用"COL+CAML"和"COL+PROLOG"来开发一个应用系统，做法和"SQL+C"相同(将 COL 语句用{……}分隔开来)。

**KBASE-P 系统**[20] 是复旦大学正在开发的一

个系统，KBASE-P是一个开发环境，支持KBASE-P语言的编程。KBASE-P以KBASE作为查询语言，以FD-PROLOG(我们开发的一个PROLOG扩充)为过程性的宿主语言执行I/O和DB更新操作(用扩充的内部谓词)。系统自动识别KBASE谓词和PROLOG谓词，并分别用KBASE和FD-PROLOG求值。因而KBASE查询语言嵌入过程语言PROLOG对用户来说是透明的，并且由于KBASE和PROLOG具有相同的语句形式和语法，所以这种透明性是自然的，从而用户在使用KBASE-P语言时没有两个语言的感觉。为了减小KBASE"每次一个集合"求值与PROLOG"每次一个元组"求值的不匹配，我们还开发了一内存事实管理器。在KBASE-P语言中，所有对数据库的操作都是以"每次一个集合"的方式进行的，因而，KBASE-P是一个比较实用的知识库程序设计语言。

## 5. 讨论

我们以Ullman的一段话开始本节的讨论："如果任何数据库查询语言都嵌入具有图灵机能力的宿主语言中，那么原则上讲，任何数据库系统均可以做其它系统所能做的事情。一个数据库系统和一个知识库系统之间只是程度上的差别而不是能力上的差别"[27]。

我们知道文件系统与数据库系统都能够处理持久数据，一个根本的差别就是：数据库系统能够高效地处理查询，而文件系统则不能。其原因是数据库系统实现了大量的查询优化技术，使得查询处理具有较高的效率(用户可以接受的执行时间)。显然，当前的商用DBMS系统的查询优化技术不能有效地处理带有递归定义的知识库查询。因此要研究递归查询处理的优化技术，同时还要给出递归查询的表达方式(即查询语言)，这就是知识库研究的二个重要方面。

查询语言是容易设计的，在POSTGRES系统中，仅在QUEL中增加了表示递归查询的retrieve * 命令，就使得POSTQUEL具有DATALOG的表达能力。之所以选择逻辑语言(注意：SQL也是基于一阶逻辑的)，是因为：①PROLOG成为公认的AI语言；②逻辑语言相对更高级(比C，COBOL更具有说明性)且表示递归更方便；③逻辑语言数学基础好。因此，人们提出了DATALOG作为知识库查询语言的基础。对DATALOG的优化研究，已近十年，取得了众多的研究成果[12,13]。

随着理论研究的发展，系统实现也相应地变化。早期的PROLOG与DBMS的松藕合系统(PROLOG+SQL)除了SQL本身的优化外没有对递归查询提供任何优化(如：JCV，BERMUDA，PROSQL，EDUCE等)。因此，这类系统的低效性以及后来被专家们否定都是必然的；稍后的基于DATALOG的系统致力于查询优化的研究和实现，主要是探讨了知识库查询语言的合适的版本(如：LDL，NAIL!等)；进入九十年代，由于递归查询优化技术已经达到一定程度，所以近几年的研究者致力于实用系统的开发(如：CORAL，Glue-Nail，KBASE-P等)，主要工作是将扩充的DATALOG语言嵌入到一个通用的过程语言中。

但是，知识库系统要想像当今的数据库系统一样走向市场，还需要进一步的努力。问题当然在查询处理的效率上。试想一下，如果完成一个数据库查询(非递归)要用T个单位时间，则完成一个递归查询可能要做成百上千个非递归查询(由问题的递归深度决定)，即要花费成百上千个T时间。可见，知识库查询在当前的硬件环境下是固有低效的。我们在文[28]中讨论了，即使采用当今最好的递归查询优化技术，也很难保证查询效率令人满意。我们认为：硬件技术的革命(大容量内存、并行机、可高速访问的外存)与良好的并行算法将是提高知识库查询效率的关键。

## 6. 未来的发展

"知识库出论文，面向对象数据库出系统"的现象使人们注意到：OODBMS能更快地进入市场。从而，一些系统在逐步加入面向对象的特征，如：LDL$^{++}$，CORAL$^{++}$等。从长远目标来看，面向对象的知识库系统(或者是具有推理能力的面向对象数据库管理系统、或其它名称)最终将成为真正的下一代数据(或知识)管理系统。这类系统的主要特征是面向对象、演绎推理[22,23,24,25,26]。虽然知识库系统与面向对象数据库系统的结合还存在理论上的问题，但二者结合的系统实验确在进行当中。

基于DATALOG查询优化的研究已达到相当程度，很难再有实质性进展，所以当前的支持否定、函数、集合的扩充的DATALOG语言将逐步成为知识库查询语言的标准。在系统实现方面，当前"DATALOGe嵌入过程语言"结构的系统已经可以实用(尽管查询效率仍不尽人意)。进一步将考虑的是规则作为持久数据存放以及持久规则的共享、一致性等，即如何有效地管理持久规则。在KBASE-P语言的设计中，程序的独立模块可以存入系统的用户自定义库(UdLIB库)中，UdLIB库是可共享的。

最后，我们以Ullman的一段话作为本文的结束："面向对象的系统(指OODBMS)将先成为可用的，面向值的系统(指KBS)最终也会成为可用的，并且将证明是更吸引人的"[27]。(参考文献共28篇略)