

专家数据库系统引论

欧阳为民 (安徽大学计算中心)

摘 要

本文描述了一类称为专家数据库系统的新型计算机系统的特性。专家数据库系统涉及到专家系统技术与数据库管理系统技术的结合。专家数据库系统将用于开发那些需要对共享信息进行知识制导处理的应用。专家数据库系统可望在本世纪末成为最重要的应用开发工具之一。本文重点讨论该种系统的结构与设计原则。

一般说来,专家数据库系统(EDS)涉及到数据库管理系统(DBMS)与专家系统(ES)两种技术的结合。这里,ES用于对存贮在DBMS中的或从DBMS中检索出的数据实施智能化处理。这方面的应用,如CAD/CAM、办公自动化、军事指挥与控制等,与日俱增。ES与DBMS的集成系统是现今的一个新兴研究领域。本文重点讨论EDS的体系结构与设计原则。

1. 体系结构

本节评述DBMS与ES的主要特性,介绍一种EDS的体系结构,接着描述在设计EDS时的技术难点。

1.1 数据库管理系统

DBMS是一种用于开发那些要求存取共享信息应用的软件工具。当多个用户(应用程序)为了更新或检索而要对同一信息集合进行存取时,DBMS是适用的工具。在这种环境下为保护共享信息需要一些实用程序,如一致性维护、恢复机制、并发控制和安全控制。

DBMS通过高级事务说明语言来提供上述功能用户用该语言生成事务,而DBMS承担如下责任:管理物理数据存取、重叠操作的交错、系统失败的恢复和存取权限的管理。为使用户摆脱物理数据表示的细节,DBMS提供了一种特殊的知识表示语言,即数据模型。

保护共享信息是DBMS存在的主要理由,它还可以在控制数据的冗余与分布方面和使应用程序更易开发与维护方面提供便利。无论共享信息量是多或

是少,也不论数据驻留在内存或外存,DBMS的各项功能成分都是需要的。不过,当今所设计的DBMS大多是面向数据驻留在外存的。这种情形很可能在将来随着内存的扩充和性能要求的提高而改变。

在专用(非共享)信息的应用中,DBMS仍然是一个有用的工具,只是它的许多功能成分将成为多余。使用DBMS有利于应用开发迅速进行,但如此开发出的应用软件其性能不高,我们必须在两者之间加以权衡。

总之,DBMS适合于那些需要存取共享信息的应用,而不适合于那些利用专用信息且性能要求又很高的应用。

1.2 专家系统

专家知识是指人类专家用以解决领域问题的原理与过程,而专家系统就是围绕这种专家知识的直接表达和利用而建造的应用系统。一般来说,专家系统是以诸如逻辑规则、产生式规则或框架这类离散形式来表达专家知识的,并力图在知识片断与知识表达中的规则或框架之间保持一种一一对应关系。这样的知识表达方法可以使复杂的病态结构的应用易于开发、调试与发展。而且,业已证明交互信息(如解释)可以做得对用户更有意义。

当然,为获得上述优越性能是付出了代价的。知识表达要保持简单的离散形式,就要忽略面向性能的计算细节。而这些细节则不得不通过强有力但却低效率的控制策略加以补偿。为确定应用知识片断(规则或框架)的次序,控制策略必须进行搜索,这种通用搜索机制的性能是限制该技术应用的主要

因素。

利用多重知识表达方法表达专家知识是专家系统研究的一个发展趋势。单纯的逻辑规则,语义非常简洁,并可有效表达无副作用的知识。产生式规则适于处理副作用。框架适于根据规则在问题求解的不同阶段所起的作用来进行规则的聚集。

1.3 专家数据库系统

顾名思义,EDS是开发那些需要DBMS和ES的应用的软件工具。这种工具的应用是很多的,实际上,可以说当今大多数DBMS应用都将从这种工具中获益。稍严格一点说,EDS是一种用于开发那些需要对共享信息进行知识制导处理的应用的软件工具。

图1表示的是EDS的一个假想结构。在这个结构中,用户与信息之间有三种系统成分:1)进行知识制导处理的ES;2)进行共享信息管理的DBMS;3)进行处理特殊格式数据的实用程序(这种实用程序的数量与类型取决于应用具有的特性)。各成分之间的连线反映的是控制流,而不是信息流。每一成分均可对其低级成分的数据与命令进行直接存取。用户将通过ES接口存取EDS。实用程序是用于进行特别的数据处理,例如图象处理,有限元分析等。由于用户感兴趣的不是数据的结构,而是通过特定功能的执行所呈现的表现,因此,任一知识表达语言(或数据模型)均未涉及数据结构。但是处理特殊数据的能力是日益增长的ES与DBMS应用的要求,作为通用的软件工具,EDS势必应有这种能力。

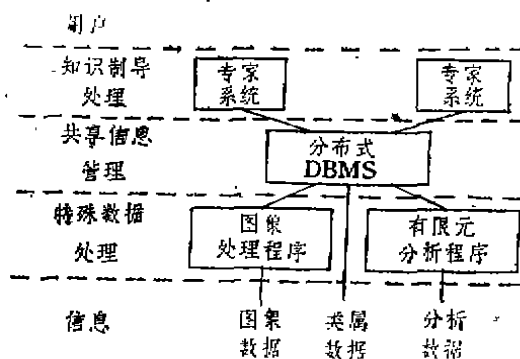


图1 EDS体系结构

如图1所示,把特殊处理实用程序置于DBMS的控制之下是非常有益的。这样,特殊处理实用程序所执行的功能可以被综合进DBMS的查询语言中,所有传向或产生于特殊处理实用程序的数据可以由DBMS进行分类。

一般地,DBMS将分布于多于处理器与存储设备上。多于处理器既可以通过计算机网络进行松散耦合,也可以利用共享存储进行紧密耦合。于是,每个ES均可与自己的DBMS结点进行交互。DBMS结点可以把信息存储在内存中以便ES快速存取。数据的聚集与复制可以大大提高DBMS的并行处理机会。

事实上,图1所示的结构是基本的,其它可选结构将与此大同小异,无本质不同。DBMS必须是一个独立的模块以便保护共享信息。特殊处理实用程序也必须是一个独立的模块以便限制对其内部功能的调用。同样,为压缩应用中的问题求解活动,每个ES必须是一个独立模块。

这种体系结构存在两个问题。第一,EDS的各构件应怎样分散到ES和DBMS上。数据模型、查询语言等其它原来由DBMS提供的构件是否要从DBMS中取消一部分,而代之以ES提供?反之,原由ES提供的构件是否也这样做?第二,EDS的全部构件应怎样展示给应用系统的开发人员?应用开发的接口应显示ES与DBMS之间的界限吗?应用开发人员需要意识到ES与DBMS之间的信息移动吗?

性能是研制EDS的最主要难题。EDS是两种搜索机的必然组合:其一为求解应用问题而搜索知识片断(如规则,框架),并为此生成对共享信息的查询;其二则为回答查询而搜索共享信息。设计不当就可能造成处理时间的组合爆炸。

系统的性能应是ES与DBMS之间进行系统构件分散的准则。一个构件应分散到能最有效执行它的哪个系统成分上。两个主要性能指标是外存DBMS句型搜索和ES的搜索周期。这里我们有一个基本论断,即DBMS对一个存取请求的总目标了解得越多,它就越能更好地对该请求进行优化。DBMS优化的最坏情况是当DBMS接收到大量独立的信息请求时,它对这些请求的最后目的又一无所知。此时DBMS别无选择,只能分别作出响应,从而丧失了对性能起决定作用的全局优化的机会。因此,把传递构件放到DBMS可导致较有效的外存存取。为提高系统的总效率,DBMS是否可以接替ES的某些工作?以及如何接替?下一节将着重讨论这个问题。

2. 搜索:推理与查询求值

搜索在ES中支持推理,而在DBMS中则支持查询求值,但是,这两类系统对搜索的支持是颇不相同的。本节将比较ES搜索与DBMS搜索,以得出它们在EDS中如何结合的结论。注意,下文是以一阶

谓词逻辑的演绎推理为ES推理技术的代表,而以关系型DBMS的查询求值作为数据库查询求值的代表进行对照比较的。

2.1 演绎推理

在演绎推理中,知识被表达为一个公式集A,查询是个关于自由变量V的公式Q(V),查询的结果是自由变量例示值S的集合, S满足, Q(S)可由A导出。

考察图2中关于船的位置的例子。它有两个公式。第一是说Kennedy位于Point-U,或者位于Point-V,第二是说Kennedy不在Point-V。查询是问Kennedy的位置。由于公式集A可导出Position(Kennedy, Point-U),而不可导出Position(Kennedy, Point-V),查询结果自然是Point-U。

Axioms:

1. Position (Kennedy, point-U)
or position (kennedy, point-V)
2. not (Position (kennedy, point-V))

Query:

? position (kennedy, P).

Inference:

find P's such that Position (kennedy, P) is provable from the axioms.

Results:

P=point-U

图2 演绎推理

从逻辑上讲,“Q(S)可由A导出”与“A蕴含Q(S)”是等价的。因此,推理可定义为要求证明A蕴含Q(S)对谓词符号的所有可能解释均为真的搜索。一个n元谓词的解释就是一个已知集上的n元关系。注意,仅仅证明“A蕴含Q(S)”对某一特别解释下为真是不够的,必须对所有解释均为真。正是这一点使得推理的计算复杂性较高。

2.2 查询求值

在查询求值中,知识被表达为定义集D和一个解释I。I给出针对元谓词集给出的一种解释。D定义的是由元谓词导出的谓词符号。一个查询就是一个带有自由变量V的公式Q(V)。查询的结果是自由变量例示值的集合S, S满足Q(S)/D在解释I下求值为真。

Definition:

Position (x,y) is

range r1, report, type (r1) = pos.

r2, report, r1=r2, id*(r1)=id*(r2)

r3, report, type(r3)=neg

(gr1) (vr2) (gr3) (ship (r1)

=x&position (r1)

=y &position (r2)=

position(r3)&ship(r2)=ship(r3))

Interpretation:

report

id#	type	ship	Position
b-2	pos	kennedy	Point-U
b-2	pos	kennedy	Point-V
f-4	neg	kennedy	Point-V

Query:

? position (kennedy, P)

Evaluation:

Find P such that position (kennedy, P) evaluates to be true under the interpretation.

Result:

P=point-U

图3 查询求值

图3中的例子所表达的知识与图2的相同。谓词符号Position是由元谓词report定义的。report有肯定和否定两类,肯定型report是说船可能在某一已知位置,而且,在ID*与ship均相同的肯定型report中船在且仅在某一确定位置。否定型report是说船不在某已知位置。否定型report中ID*仅出现一次。

对用户而言,图3中所表达的知识内容是不同于图2中的,尽管两种表达都用谓词逻辑的语法和语义,可表达的形式却非常不同。

查询求值所涉及的搜索被定义为寻找在解释I下Q(V)/D的真值。注意,查询求值只要在一种解释下进行。

2.3 比较

现在我们来比较查询求值中的搜索与演绎推理中的搜索。

查询求值中的搜索要比演绎推理中的搜索简单

得多。前者只涉及一个解释,而后者则涉及所有的可能解释。当然,查询求值中的公式 $Q(S)/D$ 要比演绎推理中的公式 $A \supset Q(S)$ 复杂,但在大多数情况,与解释的数量相比公式的复杂性对性能的影响甚微。查询求值一般都比演绎推理快得多。

用于演绎推理的知识表达方法要比用于查询求值的知识表达方法更为有力、简洁。

演绎推理方法要比查询求值方法更为灵活。特别是当要增加新知识时,前者更显优越;在推理方法中增加新知识对已表达的知识没有影响;而在查询求值方法中,以新元组形式增加知识到关系 I 中就可能使原定义 D 无效。例如在图3的例子中增加一否定型report,其ID*与已存储了的某个否定型report的相同,那么原定义 D 就失效了。在DBMS中为维护定义 D 的有效性而对解释 I 施加的约束就是所谓的“完整性约束”。

实际上,演绎推理的灵活性对很多应用来说是一种不必要的奢侈。在大多数应用中总存在一部分固定不变的知识。对这部分知识的最佳利用是把它作为完整性约束放在搜索过程中,这样可以改善系统的性能。在演绎推理中,完整性约束与其它知识不加区分,搜索过程无法从其中获益。在许多应用中,完整性约束用于检测/修改输入数据中的错误。在这种场合,约束知识必须与其它知识区分开来。

查询求值方法与演绎推理方法一样也有解释能力,后者的解释涉及到证明树的回溯,而前者则涉及查询求值树的回溯。前者在解释时所利用的东西与后者所利用的东西相比是低层次的细节。为使用户理解所解释的内容,而又不必陷入技术细节,我们可以引入中间定义,其术语必须为用户所理解,

否则任何解释都毫无意义。

我们假定演绎推理的特性是各类型ES推理的典型特性。基于此,下述结论具有普遍性。

如果一阶谓词逻辑用于DBMS的谓词定义,DBMS是不会象ES那样能表达很多知识,除非采用更强功能的定义(如可递归的谓词定义)。

在ES与DBMS表达相同知识的场合,两种系统均支持相似的查询与解释功能。不过,DBMS回答查询的效率要比ES的高。

ES具有一个比较灵活的知识表达形式,但在大多数应用中这种灵活性未必是必需的。

ES的搜索机制功能强大,可计算代价高昂。该搜索机制对所有知识都是同一的。DBMS的搜索机制相对来说不那么有力,但效率却高得多。为弥补功能不足,可通过定义来扩充DBMS的搜索机制以处理附加类型的知识。

在EDS中,各构件如何分布于ES与DBMS上是设计EDS的关键。一般说来,其设计准则是,ES应该仅仅用在那些确实需要ES搜索机制的功能的场合;否则,应该用较简单的搜索机制。特别是对于共享信息的搜索,DBMS应该最大限度地担负起责任,力图减少ES搜索周期中所涉及的知识量,尽可能扩大DBMS优化的范围。

参考文献

- (1) L. Kerschberg, (Ed), Expert Database System, The Benjamin/Cummings Publishing Company, 1986.
- (2) M. J. Coombs, (Ed), Development in Expert System.

国际学术会议消息

会议名称: 第18届国际自动机、语言和程序设计讨论会ICALP 91

时 间: 1991年7月8—12日 **地 点:** 西班牙马德里市

论文提交截止日期: 1990年11月16日

接 受 人: Prof. Mario Rodríguez Artañeja

Departamento de Informática y Automática, Facultad de Matemáticas, Universidad Complutense, Av. Complutense s/n, 28040 Madrid, Spain

论文要求: 7份全文或详细摘要,最好不超过12页

内容: 理论计算机科学所有方面,包括(但不限于):可计算性,自动机,形式语言,算法分析,计算复杂性,数据类型与数据结构,数据库与知识库理论,程序设计语言的语义,程序规格说明,程序变换和程序验证,逻辑程序设计与函数式程序设计的基础,逻辑设计与布局理论,并行计算与分布式计算,并发理论,符号计算与代数计算,重写系统,计算几何,密码学,机器人理论。

(江 海辑)