关系数据库和面向对象数据库应用比较

郭 江 廖越虹 (北京航空航天大学软件工程研究所)

摘要:本文分析了在自动制造环境下使用关系数据库及面向对象数据库在技术上的优势和局限性。

面向对象的数据库目前应用还不广泛,但它已在研究人员和软件开发者中引起了极大的兴趣和关注。将关系数据库和面向对象数据库这两种数据库模型进行比较后,我们能发现它们都基本解决了以下几个主要技术问题:

.程序设计语言的支持

.查询语言的支持

.知识表示

.性能

.数据分布

商品化的可用系统是我们要集中考虑的问题。虽然这些问题最终都是相关的,但我们在此将对它们分别进行讨论。

一、程序设计语言的支持

典型的数据库应用都使用程序设计语言来实现大部分功能,并用永久存储器(如在 DBMS)来存放它要使用和将生成的永久性数据。例如,一个 CAD 系统可以用 C 程序设计语言编写,而用关系数据库来保存数据。当 CAD 软件需要从 DBMS 那里得到数据时,比如用户要系统显示一幅图,这时软件就通过查询语言接口访问 DBMS。这些对 DBMS 的查询常被嵌入应用语言中,这里即为 C 语言。数据检索到后,应用程序可以不调用 DBMS 而运行,直到用户再次运行到需查询数据库的时候。

在这一过程的执行中有两点很重要。第一,对于应用开发人员来说,由于要和 DBMS 有接口,从而使软件设计增加了一定的复杂性。虽然它增加了开发和维护这样一个系统的费用,但它对终端用户来说并没有造成什么影响;第二,应用程序同数据库的连接要花费一定时间,这样,系统运行速度就会减慢,这一点直接影响到了终端用户,在诸如 CAD 的工作站应用中尤为突出,研究表明,即使系统的性能稍有下降,也会明显的影响用户的生产率。

使用面向对象数据库技术进行软件开发,不但能改善一个应用系统的设计,还能改善它的性能。一些可用的面向对象数据库(或仅是对象数据库)通过函数调用提供了一个程序设计语言的接口。在这种情况下,如果用诸如 C 的语言来进行应用开发,那末底层的 DBMS 不管它是关系的还是面向对象的都很简单,只是面向对象的数据库速度可能更快一些。

但是,有的对象数据库在特定程序设计语言和数据库本身之间提供了一个无缝连接,这样就可以基本上消除那些通常用来与 DBMS 接口的应用代码。这样就可以简化应用系统的开发和应用,在某些情况下还能提高速度。另外,软件的复杂性减小了,具有良好性能的高级版本也就可以更快开发出来。有的对象数据库提供的和面向对象的程序设计语言,如 C++、Smalltalk、Lisp 等之间的无缝连接相当完美。

了解程序设计语言(可能是面向对象的,也可能是非面向对象的)和 DBMS(可能建立在面向对象的、关系的、层次的、网状的模型之上)之间的区别是非常重要的,应用中往往把程序设计语言和 DBMS 结合起来使用,我们应分别对这些部件进行分析。许多应用使用了非面向对象的程序设计语言(如 C, COBOL)和关系数据库,而用面向对象的程序设计语言(如 C++)和关系数据库来生成应用也是可以的。这种应用目前尚无法广泛地加以使用,但随着面向对象语言的普及,会逐渐广泛起来的。

二、查询语言的支持

关系数据库技术最强有力的一点是其分解数据关系的能力,这样集合论中的操作就能在数据上进行操作了。并、交、选择、联接等操作是由易使用的查询语言来

支持的(如 SQL),并且对终端用户和软件开发人员都是可用的。当应用需对大量数据进行分类检索时,使用诸如 SQL 的查询语言可节省几百小时的开发时间,而用常规的编程方法就要费时得多,制造资源规划(MRPII)系统就是制造环境的一个实例。这些系统常常生成几百份报告,使用诸如多级排序、控制断点、排列统计等报表生成特征。一些老的 MRPII 系统开发时没有高级查询语言,包含几千行不必要的代码,而终端用户要修改这些代码是非常困难的。

不幸的是,对象数据库常常是不支持查询语言和报表工具的。只有一个对象数据库提供了 SQL 接口。有些对象数据库提供了有限的查询工具,但是这些对需要用非编程途径生成复杂的格式化报表的报表密集型应用而言,几乎没什么用。

许多对象数据库界的人员认为 SQL 是专门为关系数据库模型设计的,把它用于对象数据库只会限制它的表达能力。这一结论可能是对的,但在对象数据库界没有给出一种至少和 SQL 功能一样强的面向对象查询语言之前,ANSI 标准的 SQL 就仍是查询语言的选择。这并不是什么好的选择,因为它从本质上妨碍了对象数据库在如 MRPII 的报表密集型应用中的使用,而它本来应在这一领域起更大的作用。目前仍可以认为关系数据库技术在报表密集型制造应用中占主导地位。

三、知识表示

制造数据库理想上应支持几种知识表示。为了方便,将在后面分别讨论结构的、过程的和说明的对象表示。应当指出的是,一个知识表示方案应提供所有这三种类型的统一表示。

1.结构知识

结构表示是描述一个项所必需的数据属性的分组,例如,钻机的属性有机器的制造者、编号、通行的装配及可能的装配范围。这里需要几种通用的数据类型,编号可以用简单的数据类型来表示,如整型或数组;而机器制造者就需要用更复杂的用户定义的类型来表示。为描述机器的结构就需要在概念上把机器分为简单的部件,而它们又是由别的,或更简单的部件组成,换句话说,有的项要表示为一个其它项的组合体,它们中的几个或全部都有自己的复杂结构。

理论上讲,关系和面向对象数据库都能表示这样的结构。虽然在允许用户定义数据类型方面关系数据库的限制更多,但建立一系列关系表也能达到同样的效果,表中的外码象指向用户定义类型的指针一样。对象数据库在用户定义及扩充数据类型方面功能很强,用户定义的数据类型有时能和简单数据类型同样处理。使用对象数据库来编写应用程序通常复杂度小一些,尤其是在选用了合适的面向对象程序设计语言之后更简单些。这种情况又引起了关于结构表示的几个问题,它们可能会在制造应用系统中产生变体记录结构、合成结构及版本。另一与结构表示有关的问题在任意型位图中讨论。

2.变体记录结构

信息表示在制造系统中总是产生变体记录结构,通常从一般的或抽象的记录结构中导出特殊的记录结构。例如,过程规划和调度系统使用的制造操作表示,在一个生产过程使用几百个操作的制造中也是如此。这种制造的过程规划包括切削、磨、抛光、热处理等操作。理论上讲,这些操作都有一些共同的属性,如操作码和操作描述;还有一些特殊的属性。例如,切削操作有描述刀具路径和切削深度的传统属性,而热处理操作则有关于温度和热处理持续时间的属性。特定类型的操作,如切削、研磨或热处理可以看作是一个操作的抽象概念的派生。

在商品化的关系数据库中,没有从抽象概念到特殊类型的表示派生的内部机制(虽然扩充的实体关系模型技术的使用提供了这种能力)。对使用关系技术的 MRPII 软件包中的表结构进行研究可以证明这一点。我们只能表示诸如操作、机器、部件的抽象概念,而所有的操作、机器、部件必须用由抽象概念提供的有限的属性集来表示。如果一个这样的系统用户需要更精细的表示,应用软件就必须修改了。对象数据库的最大优点之一是它提供了一种继承机制来表示从抽象的高级概念派生的变体结构,这种机制称作继承,是对象数据库的中心观点之一。

3.合成结构

把某些结构看作独立于程序设计实现的逻辑上可区分的实体通常是很有用的。例如,一个高层组装是一个逻辑上可区分的实体,虽然软件的实现可能把这个组装表示为一个复杂的数据结构,它可以包括很多资料,别的组装及部件。当设计者想编辑这个组装的几个方面,而

同时让别的设计者编辑这个组装用到的子部分的设计时，就会产生一些问题。这样最好有一个数据库级的功能，它能让设计者声明一个由别的实体组成的高级实体，一个分级功能，可以让它们在合成实体的逻辑概念基础上跟踪系统的改变。例如，在设计一个子组装时的改动可能暗示着高级的设计需要更新。虽然这一级功能能在逐个情况的基础上编程，但这种情况很普遍，数据库级的支持更为合乎需要。

把合成结构作为逻辑上可区分的实体的概念源于面向对象程序设计，虽然从理论上来说可以把这一功能加到关系数据库中，但在对象体系中这一观点的实现更为自然。一些面向对象的数据库就是专门为支持这一概念而设计的。

4.版本

虽然版本不是一个结构表示本身的问题，但对版本的需求常和对合成结构的需求发生在同一上下文中。那就是，数据库跟踪合成结构版本的功能是很必要的。当一个实体处于版本控制之下时，对这个实体的修改就产生一个新的版本，通过一个版本派生图就可以跟踪所有的改动。例如，在一个工程修改控制系统中，一个工程修改可能在交付工作未完成时就是有效的，而在工作交付后，设计版本仍有效，用它来完成开放式工作一般较为合适。再重复一下，这种逻辑可以在一种情况的基础上实现，但是在制造应用中，这种情况的频繁发生使得数据库级的支持变得非常重要。因为在需要合成对象结构的时候，版本支持通常也是必要的，所以数据库应同时支持这两个概念。

5.任意型和位图

结构知识表示的另一个能产生混淆的领域称作任意型。任意型是一个通用数据类型，一些关系数据库已把它加到自己的仓库中，使得在应用中出现的任何数据类型都能存储。特别是任意型能存储位图或二进制代码图象，在对象数据库中，它们也可以用用户定义的数据类型来表示。

关系和对象数据库的厂商都有一种错误概念，即任意型和位图在计算机辅助设计(CAD)应用中有某些关系。实际上，CAD 系统总是用过程的方法来存储和生成图形的。作为一个例子，我们来看一下线框 CAD 系统，它使用少量的原始过程来画点和点之间的线。把一个 CAD 位图的产生形成一个文件，通常称作图形元文件，它包括各种过程的原函数的代码，和可用作调用原函数的过程参数。为显示一幅图，CAD 系统检索图形元文件，并在元文件中代码的基础上调用适当过程的原函数。最终显示在屏幕上的形式可以是一个位图，但这实际上至多和 CAD 系统的操作相切。任意型和位图虽然可能在某些应用(如图象处理)中有用，但和 CAD 系统没有特别的关系。

6.过程知识

过程知识描述了诸如组装一台机器、装配一个部件、或处理一个部件这样的工作是怎样通过一系列操作来进行的。一个过程上的描述有些可以不太确切，如对机器的操作人员的注意事项;有些必须是确切的，如在一个数控机床程序中的算法。随着工厂自动化程度的提高，对各种工厂过程的确切算法描述也随之要求提高，因为在自动化程度低的工厂中机器缺乏智能，也不了解工人们怎样工作。这样，制造数据库就更加依赖于存储的过程知识了。

实际上这种情况某种程度上被简化了，就是多数过程知识是以特定的程序设计语言的形式存在的，例如 APT(自动的程序设计工具)语言是用于数控机床程序设计的，AML 语言是用于机器人程序设计的。这些语言不太可能被更为通用的程序设计语言所代替，因为它们是面向当前的任务的，并且用这种语言编程所需的熟练程度，尤其是机器人程序设计，一般超出了典型的高级语言程序员的熟练程度。这减轻了制造数据库的压力，因为在这种情况下数据库可以真的只存储源代码，或可执行的代码形式，它们只需要象字符串或二进制这样简单的数据类型，而这是任何数据库都很容易提供的。因此，表示的过程知识包括数控机床和机器人程序，而对象的和关系的数据库看来没有什么更大的优势。

7.说明性知识

说明性知识描述的领域有:逻辑规则，数据间的关系和对数据的约束。力(如引力和磨擦力)和性质(如空间延展性和时间性)能使真实世界的实体具有特性，而这些特性是很难用计算机软件模拟的。正如生产系统的目标越来越宏大，模拟它的特性所需要的已不再是个议题所能解决的了，涉及到的说明性知识就日益成为研究的重点。

在 CAD 系统的发展中可以看出这方面的进展。由

于原始厂商有时会生产模糊的或物理上无法设计的几何图形，所以就发明了固体布线包。而固体布线包有时也会产生物理上非法的设计图，而无法投入生产；最终以特征为基础的 CAD 系统就向前发展了。通过对 CAD 领域的研究可以发现，处理逻辑规则和约束的能力是今后 CAD 系统发展的一个基本特征。

目前，关系和对象数据库在处理说明性知识方面都很薄弱。关系数据库在模拟说明性知识中的某个简单的类型，即关系，在这方面是很有效的。例如，我们可以创建一个部件关系用来描述需用一部分来装配另一部分的概念。如部件 P1 在生产部件 P0 时用到，关系(P0, P1)就存在于这两部件间。关系的观点实际上是一个形式化的数学上的概念，数学家和计算机学家对它的特性已进行了透彻的研究。关系数据库是这个概念在实际中的实现，在关系数据库中，所有的知识都要表示为数据集和它们之间的关系。

然而，关系只是说明性知识中一种很简单的类型，它不能表示逻辑规则和约束。在以特征为基础的 CAD 系统中，一个高级逻辑约束方面的例子如下：当一个设计的几何图形是合法的并且这个设计是可制造的时侯，这个设计是合法的。程序设计语言 PROLOG 能够表示这样的观点，而且 PROLOG 和程序设计在 CAD 中是一个值得积极研究的领域。在商用数据库界，无论关系还是对象数据库目前对这种知识类型都无能为力。然而经过扩充，它们都能做更多工作。我们知道，关系体系可表示为逻辑程序设计体系的一个子集。好几种面向对象的程序设计语言(主要的研究项目)已经成功地把约束的概念和逻辑规则组合起来。关系和对象数据库都会很快得益于这些进展。

四、性能

软件开发人员最终从数据库系统引出的性能水平在主存中运行时是可与 C 语言相比的。无论关系还是对象数据库现在都没有达到这种性能水平。关系和对象数据库通常使用一个用户服务器体系结构，它常需要将那些应用程序中对数据库的调用通过网络传递，以访问驻留在服务器上的数据。这种体系结构对很多类型的数据库应用都很适用，但对于高性能工作站应用效率太低(尽管一些对象系统能够绕开这个问题)。这个问题只有在未来诸如为生产能力进行设计和以特征为基础方法的 CAD 发展时才会产生困难，在这种 CAD 系统中不仅需要图象数据，还需要来自其它子系统的数据。

虽然对象数据库还不能达到预期的内存处理速度，初步的测试表明它做了一些改进，超过了关系技术。一些对象数据库厂商把重点集中在工程工作站市场，尤其是 CAD 上，还特意针对这些应用的特殊要求设计了数据库产品。因为对象数据库技术相对较新，因而可以预言性能方面还可以继续改进。改进了的处理速度和简化了的应用开发相结合使对象数据库成为支持 CAD 和 CAE 的有力技术。

五、结论

这篇文章所讨论的技术问题说明，关系和对象数据库在制造系统的设计中都可扮演举足轻重的角色。但无论哪种技术都无法包容制造系统设计的所有方面。关系数据库更适合于报表密集型应用如 MRPII，而对象数据库则在工作站应用方面更具潜力，如 CAD 和 CAE，它们不是报表密集型的，而是要求有较高的性能。

最大的关系数据库厂商已经开始研制直接瞄准制造市场的内部程序。特别是 IBM 公司控制着关系数据库的大部分市场，并且它早就涉及了制造的各个方面，包括 CAD、CAM、MRP(原料需求规划)。IBM 已经围绕着它的系统应用体系结构 SAA 和 UNIX 的 AIX 实现编制了开发 CIM 体系结构方面的程序。在这些成果中，对象数据库看来并没有发挥作用。

从整个市场考虑，集成制造系统的设计者面临着极复杂的技术决策。在制造环境中工作的数据源管理人员必须一步步详细审查设计上的问题，以便在所有应用系统中使用各种最佳的软件技术。

参考文献：

[1] Jeff Gomsi, Mike DeSanti, "A Technical comparison of Relationaland Object-Oriented Data Bases for Manufacturing Applications", Data Resource Management (Winter, 1992), PP40-46.

[2] Bray, O.H., "A Data Resource Manager's Guide to the Role of DataBase Technology in CAD / CAM", Data Resource Management (Fall, 1990), PP39-46.