

# WHYMX存储系统的设计方法

毛兆余 王能斌 董玮 徐庆云 杨传钧

(机电部第11研究所, 北京)

## 摘 要

The concept and design of the simplest common interface(SCI) are firstly presented in the paper. Then the construction of the storage system of WHYMX is simply introduced. Consequently, an approach is described for extending the storage system, several problems with the storage system expansion are also analysed.

目前,许多新的应用都要求数据库系统提供扩展功能,以便用户随时可以定义自己的数据类型及其操作。但是,传统的数据库系统无论在数据模型,还是在存储结构及其存取方法等方面,都不能有效地提供这种支持。尽管有些数据库系统允许用户增加一些自己定义的数据类型及其操作,但是这种扩展是通过查询语言或嵌入到应用程序中来实现的,况且这些数据库系统的基础DBMS在基本存储

方面并没有支持这种扩展,从而这种扩展方法在通用性上受到限制,而且还给查询及查询优化带来了困难,影响系统性能。因此,数据库管理系统怎样才能有效地支持用户的扩展呢?即允许用户在数据库系统上添加适当的存储结构和存取算法,以适应于自己扩展的数据类型及其操作。这已成为当今数据库领域研究的重点之一。

面向对象数据库管理系统的“扩展”之一是:存

## 参考文献

- [1] M. Atkinson et al, "The Object-Oriented Database System Manifesto", Proceedings of the 1st DOOD, Japan, Dec. 1989. 中译文见《计算机科学》1990No.3
- [2] J. Banerjee et al, "Data Model Issues for Object-Oriented Applications", ACM TODS, Jan. 1987
- [3] Goldberg and D. Robson, "Smalltalk-80, the language and its implementation", Addison-Wesley, 1983
- [4] S. F. Hudson and R. King, "Object-Oriented Database Support for Software Environments", Proceedings of the ACM SIGMOD Conference, May 1987
- [5] T. Kaehler, "Virtual Memory on a Narrow Machine for an Object-Oriented Language", OOPSLA'86
- [6] W. Kim, N. Ballou, H. T. Chou, D. Woelk, J. Banerjee, "Integrating an Object-Oriented Programming System with a Database System", OOPSLA'88, Sep 1988
- [7] D. Maier et al, "Development of an Object-Oriented DBMS", Proceedings of the OOPSLA'86, Sep. 1986
- [8] 田中克巳, "オブジェクト指向データベースシステム-その背景と概念", bit, Vol.20, No.6, 1988.
- [9] 李伟华, 王志英, 慈云桂, "面向对象数据库研究", 计算机工程与科学, 1990年第3期
- [10] 胡守仁, 慈云桂, 王志英, 李伟华, 奚建清, "通用的面向对象知识处理系统GOOK-PS", 第九届全国数据库会议论文集, 1990, 9, 上海

储结构及其算法的扩展,也就是说,怎样才能使用户把新的存储结构及其算法插入到系统中,并且在运行时怎样计算和执行它们,本文在这方面作了一些有益的探讨。

本文将介绍一个面向对象的、多用户数据库管理系统WHYMX中存储管理系统的可扩展性设计方法。首先,介绍最简单通用界面的概念和作用;其次,描述可扩展存储结构的设计方法;接着,描述存储算法的可扩展性;第四讨论存储系统扩展中所遇到的一些问题;最后一节是本文的结论。

### 一、最简单通用界面

要想使得DBMS可以扩展,DBMS必须是具有标准化界面的开放体系结构,而可扩展的关键在于:这些界面怎样设计。显而易见,我们应当避免把具有特定性的界面设计为DBMS中的界面,即DBMS中的界面都应是经过标准化的界面。因此,我们采用的方法是:1)找出一类易于实现,并具有良好的性能的算法;2)设计一个能够支持所有这类算法的界面。一般来说,如果这个界面所包含的这类算法越多,则这个界面就越能刻画这类算法的基本特征。由此可见,这样的界面具有一般性或通用性。因此,我们把这种标准化了的界面设计方法称为最简单通用界面(Simplest Common Interface,简称为SCI)方法。

最简单通用界面方法为DBMS的扩展提供了一条有用的途径,因为当一类算法都支持同一个界面时,该类算法之间可以相互替代。因此如果某一具体算法不再具有用户所期望的性能,则可以用另一个算法来替代,而且不必修改较高层模块。实践证明没有一个界面能够包括所有特点的算法。但是,由于SCI能刻画一类算法的公共特性,所以它有一个显著的优点,即它在增加一个新算法时,或者不需要变化,或者只需简单的调优变化,而不必做大量的修改。这样,通过逐个地把新的算法添加到SCI中,我们就可以构造一个SCI。

WHYMX存储系统使用了SCI设计方法。它包含有存储对象、文件对象、缓冲管理、

并发控制及恢复等算法的SCI。

### 二、存储系统的层次结构

SCI是可扩展存储系统必要的设计方法。例如,我们在构造WHYMX存储系统的缓冲管理时就使用了SCI方法<sup>[1]</sup>,该管理系统为存贮系统中较上层软件提供一个缓冲SCI界面。在该SCI界面中包含有读、写、占有及释放页等算法。尽管可交换性是我们所要考虑的重要而又公认的目标,但是该SCI还包含有若干个这些算法所依赖的较低层算法。注意,这些较低层算法的实现不应再分。为此,一种比较好的设计方法是采用层次结构体系,其中每层提供上层定义所需的基本算法。为了提供最大可扩展,每层都应该具有一个SCI。

众所周知,存贮结构是“概念到内部”的数据映射。尽管这些映射对于一特定的DBMS来说,可能是非常复杂的,但是它们可以由更基本的映射来定义。例如,WHYMX中抽象文件就是由文件对象来映射<sup>[2]</sup>,而文件对象则由索引和存贮对象(数据)这两个基本存贮结构来映射的,其中存贮对象又由小存贮对象和大存贮对象这两个最基本的存贮结构来定义。在WHYMX的存贮系统中,文件对象采用二级存贮形式的简单文件结构来实现,诸如物理顺序、索引顺序、B+树及堆等。此外,文件对象还可以通过基本转换映射到较低层的文件对象,诸如索引、压缩等。当然,较低层的文件对象可以用上述的简单文件和基本转换来实现。由此可见,这些概念产生了存贮结构的层次观点。

### 三、存贮结构的扩展

在面向对象数据库管理系统中存贮系统应该能够接纳用户添加的新存贮结构。存贮结构的扩展并不是指:简单地把一个新索引结构,例如R树、R+树及新的isam结构,增加到存贮系统上,注意,这点是非常重要的。正相反,存贮结构的扩展是指这样一种能力:即组合基本存贮结构及它们的算法到存贮系统上的能力。下面我们通过一个实例

来说明, WHYMX存贮系统中存贮结构的扩展是怎样实现的。

现在, 我们来看一个在抽象文件F上的检索操作。假设在F上有一个检索操作Retrieve(F, Q, O), 该操作将以O的顺序, 从抽象文件F中, 检索满足查询Q的对象。在WHYMX系统中, 该操作被当作一条消息发送给对象F, 对象F从相应的类结构中取出检索处理算法, 进行查询。对象F所属的类或为B+树类(该类的对象都以B+树实现, 用B+树方法进行检索<sup>[21]</sup>), 或为isam类(该类的对象都用isam结构实现, 用isam方法进行检索), 或用倒排文件类, 如此等等。为了使存贮结构扩展最简单, 数据库管理系统要能识别抽象文件F在存贮系统中是以什么方式实现的, 并使之成为相应类的一个对象。如果一个新的存贮结构被发现, 则只需为该结构产生一个新的类即可, 而检索这一结构的算法作为该新类结构中的操作结构, 相应的文件成为该类的对象, 并一同存于存贮系统中。

#### 四、存贮算法的扩展

存贮系统也必须能够接纳由用户或数据库管理员定义的新存贮算法。前面, 我们介绍了存贮结构的扩展, 现在, 我们通过另一个实例来说明存贮算法的扩展。

例如, 在倒排文件类中, 由三个标准检索算法组成了它的操作结构。倒排文件类说明如下:

```
CLASS Index
{
  Attribute      /*属性结构*/
  :
  Constraint     /*约束条件*/
  :
  Operation      /*操作结构*/
  Scan-Data-File (F, Q, O) ;
                  /*扫描数据文件*/
  Use-1-Index (F, Q, O) ;
                  /*使用一个索引文件*/
  Use-n-Index (F, Q, O);
```

/\*使用n个索引文件\*/

其中, 第一个算法扫描数据文件; 第二个算法使用一个索引文件来处理查询; 第三个算法是典型的存取多索引算法。如果用户想要插入一个新的存贮算法, 则必须修改类的操作结构, 把算法加入类的操作结构中即可。这与上节所描述的存贮结构的扩展方法有所不同。一种是增加新类, 而另一种是修改现有类的操作结构。此外, 在选择具体操作时所使用的的方法也有一些本质的差别。对于存贮结构来说, 把抽象文件映射成物理形式(即最基本的内部存贮结构)是由一些基本转换来完成的, 而这些基本转换的选择执行是在DBMS编译时确定的。对于算法的扩展来说, 最小代价算法的选择执行是在查询执行时完成的; 实际上, 这种选择依赖于查询。选择最小代价算法执行的过程是查询优化的工作。

在WHYMX系统中, 我们现在没有考虑查询优化, 尽管如此, 存贮算法的扩展还是可以用这种方法实现。

#### 五、存贮系统扩展的其它问题

在实际中, 一般只有高级用户才可能需要通过增加自己的存贮结构及其算法来扩充系统的功能。而且, 相对于数据类型和操作符的扩展来说, 用户增加新存贮结构及这些新存贮结构的操作算法的可能性要小得多。因此, WHYMX系统目前没有为这一任务提供一个良好的接口, 并且应当注意到这一任务目前由专家来完成。在本节中, 我们将简要地探讨一下用户在把存贮结构及其算法扩展到存贮系统中可能遇到的几个问题。

一般来说, 存贮算法将使用系统定义的缓冲区, 因此扩展存贮算法的实现者必须熟悉系统提供的缓冲管理子系统。他在编写自己的存贮算法时, 还必须调用缓冲管理子系统提供的读、写、占有及释放缓冲页的例程(即缓冲管理的SCI界面的构成例程)。此外, 还应当了解系统提供的锁管理机制<sup>[21]</sup>,

# 工程数据库中版本管理技术的研究<sup>\*</sup>)

蒋泽军 徐秋元 (西北工业大学)

聂培尧 (山东财政学院)

## 摘 要

In this paper, a method which is called the "Forward Version Management" is proposed by the author to hold the multiversion in an engineering database. This method can also generate some new database version in terms of the existing historic versioning data stored in an engineering database. Since several design versions can be stored simultaneously in a database, it is possible to hold the historic data and to support the backtracking in a design process. Therefore, it can meet the needs of "trial and error" procedure in designs.

在CAD/CAM系统中, 对于一个工程实体的设计, 由于性能描述方法的不同, 设计方案的差异, 以及对性能要求的差别, 在设计各个阶段将会形

因为他必须要对自己将增加的索引实现封锁。当然, 他可以利用系统提供的锁管理机制, 也可以编写自己所选择的锁。如果自己编写, 则在他的锁机制和系统的锁管理机制之间可能存在不可测试的死锁, 这一点是要特别注意的, 也是非常重要的。最后, 还必须掌握系统提供的恢复机制<sup>[3]</sup>, 或自己规定在各个索引页被强制成固定内存时, 所需要的排列顺序, 因为在事务失败或系统发生故障时, 这一顺序可能被用来保护索引的一致性。

本文首先介绍了最简单通用界面的概念及其设计。SCI为扩展WHYMX存贮系统提供了一种有用的形式。其次, 概述了WHYMX存贮系统的层次结构。可扩展存贮系统设计最好采用层次结构, 而且为了得到最大可扩展性, 每层都应该有一个SCI界面。接着, 本文描述了WHYMX的存贮系统中

成同一工程实体的不同设计版本<sup>[1,2]</sup>。

采用一般的商用数据库系统来管理这些不同的设计版本, 就需要建立多个数据库来管理多个设计

存贮结构及它们的操作算法的扩展方法。该方法无论是对用户, 还是对数据库管理员来说, 使用都极其方便。与其它面向对象的数据库管理系统中所使用的方法相比, 该方法显得更加灵活, 并且无需对DBMS重新编译。最后, 我们探讨了在扩展存贮结构和存贮算法时, 应当注意的几个问题。这些问题给用户扩展存贮系统带来了不便和困难, 同时这些问题有待于我们以后进一步解决。

## 参 考 文 献

- [1] 毛兆余, "WHYMX的存贮管理", 硕士论文, 1990.3
- [2] 毛兆余等, "WHYMX存贮系统中的对象和文件管理", 《第九届全国数据库学术会议论文集》, 上海, 1990.9
- [3] 徐庆云等, "面向对象的数据库系统的并发控制", 《第九届全国数据库学术会议论文集》, 上海, 1990.9

•) 国家自然科学基金资助课题。