

面向对象与数据库技术的结合

北京工业大学计算机学院 毛国君



面向对象数据库技术在近十年来得到了迅速的发展,但是它的发展集中表现在面向对象和数据库技术结合方法的多样性上。本文对面向对象和数据库技术相结合的主要方法、现状及问题进行综述,并期望通过实例进一步阐述这一领域的一些较成熟技术。

引言

八十年代是数据库尤其是关系数据库广泛应用的黄金时期。但是随着数据库应用的不断深入,对图形、图象、超文本等复杂数据类型的处理成为新的课题。因此,近十年来,把其他技术渗透到数据库系统中的研究和实践得到充足的发展。在这一发展中,最引人注目、成果最丰富的可谓是面向对象方法和数据库技术的结合。经历了十多年的风风雨雨,面向对象数据库系统不仅理论上已经取得了一批重要成果,而且在原型系统和产品实践上也已探索出一些实用的方法。本文就面向对象和数据库技术结合方法的主要流派和发展情况进行综述,并通过实例从不同侧面分析它们的结构和技术特点。

数据库与面向对象技术的结合方法

面向对象数据库概念从提出之日起就是一个颇有争议的话题。它的争议不是在两者是否需要结合的问题上,而是在两者如何结合上。追溯面向对象数据库技术发展历程,可以大致归纳为如下几种解决思想。

1. 对象-关系数据库

传统的关系型数据库(RDB)是建立在关系代数基础上的,一个用户记录是由若干个具有特定数据类型的字段组成的。这些数据类型被限制在有关字符、数字等简单类型上。随着信息处理的多样化,有关图象、图形、声音、页面、档案、地理信息以及影象等信息的处理成为必需。究其原因,是传统的RDB的数据类型是不开放的,用户只能使用系统规定的数据类型而无法扩充和定义。为此,一种称为对象-关系数据库(ORDB)方法被提出,旨在扩充关系型数据库使之支持抽象数据类型(ADT)的定义和使用。

SQL语言是传统的RDB系统的操作语言,无论是在标准化还是成熟性方面都是无可挑剔的。但是,新的ADT的引入已经超出传统的SQL所及的范围。幸运的是,ADT

的引入是完全基于面向对象思想的。一个ADT把数据和操作封装在一起,因此只要扩充SQL语言使之能够调用ADT的操作方法,就可以象内部原有的数据类型一样被用户使用。

2. 面向对象数据库

目前,在对面向对象数据库概念的理解上有两个层次。其一是广义的理解,即把使用面向对象技术的数据库系统都称谓面向对象数据库系统。另一种是狭义的理解,即以一种革命化的新方法在数据模型、查询语言及系统结构等方面,全面引进面向对象概念和技术。本小节讨论的是面向对象数据库系统的狭义概念,简称为OODB。1989年,在由Atkinson等发表的“面向对象数据库系统宣言”中,详细阐述了OODB系统的13条必备特征、8条可选特征和其他特征[5]。这是首次比较全面的对OODB系统的性质进行阐述,对后来的发展起了重要的作用。简言之,OODB系统必须解决两个基本问题:

(1) 支持核心的面向对象数据模型和诸如封装性、继承性、多态性等性质。

(2) 扩充传统数据库的语义、语法,使OODB与面向对象数据模型融合,成为独立于应用程序的新型数据服务手段。

3. 演绎面向对象数据库

传统的演绎数据库(DDB)是基于关系模式的,通过演绎推理达到对关系数据库中数据的挖掘和利用[3]。它面临处理复杂数据的新的挑战。面向对象技术已经成为解决复杂数据类型的一种有效方法,但是,OODB系统的发展表明它的形式语义、模式演化等问题仍未取得期望的结果。演绎面向对象数据库(DOODB)旨在把两者有机结合起来,为面向对象数据库的实现提供一种新的途径。它要解决的两个主要问题是:

(1) 复杂对象模型及相应逻辑语言。它是从DDB出发,增加对复杂数据类型的支持。为此,必须解决复杂对

象模型的建立及相应逻辑查询语言的研制。

(2) 面向对象的演绎推理方法。复杂对象模型及相应逻辑语言确定了 OODB 系统的数据组织和数据使用的形式, 研究其演绎推理方法才能解决 OODB 内部实现机制。因此, 建立在面向对象数据模型基础上的规则演绎实用方法的研究也是一个重要问题。

4. 结合其他技术的面向对象数据库方法

随着数据库技术应用的日益普及和扩展, 把诸如组件、多媒体、AI 等技术应用到数据库系统中来建立新型面向对象数据库系统, 得到了良好的发展。现择取几个介绍如下。

(1) 瘦型 (Lean) 数据库系统。传统数据库管理系统是采用整块式结构设计的, 用户无法选择 DBMS 的功能, 由此不仅系统的开销大, 而且缺乏扩充性。近些年, 瘦型 (Lean) 或轻便型 (Light-Weight) 数据库系统倍受关注。面向对象的数据库组件技术成为用户构造新型轻便数据库系统的可靠方案。

(2) 多媒体数据库。多媒体应用系统的崛起是九十年代信息系统的重要标志, 而多媒体数据库研究更是一大热门。使用面向对象技术处理多媒体数据成为新的有效的方法。关于它的介绍很多, 这里不在赘述。

(3) 支持持久对象的程序设计语言。事实上, 数据库中的数据和普通程序设计语言依赖的数据的主要区别在于数据的持久性, 即数据库中的数据存储在外部存储器中进而可以重复利用, 而普通程序设计语言的变量数据暂存在内存中进而缺乏重复利用的机制。在面向对象程序设计语言中, 如果引进“持久对象”的概念, 就可以增加数据库操作功能。这种方法的倡导者认为, 依靠面向对象程序设计语言的类型系统和编程模式, 增加强制数据成为持久的、可共享的数据结构机制, 就可以实现持久对象数据重复利用。

(4) 扩充数据库系统的工具包。有人认为任何的 DBMS 都不可能满足所有用户的要求, 因此把完成通用功能的 DBMS 的核与实现特殊应用的工具包区别开来, 单独设计。这类工具包主要使用面向对象等的技术, 在查询语言、存储组织、事务机制和数据类型等不同层次上对 DBMS 进行扩充。基于这种方法的代表性工作是 EXODUS, 它引进了“对象存储器”的概念, 并提供了一种支持持久对象的程序设计语言 E。

各种方法的发展现状分析

上一节列举了面向对象与数据库技术的结合方法,

这些方法都从不同角度探讨了两者结合的可能空间和发现了丰富的实用方法及 (模型) 系统。勿容置疑, 不论是在理论上还是在实践上, 它们对面向对象数据库的发展都起到了重要的作用。但是, 也必须看到它们发展的不平衡性和应用的互补性。本节对它们的发展情况和主要问题加以分析。

1. 对象-关系数据库

对象-关系数据库的早期工作可以追溯到八十年代末或九十年代初, 比较有代表性的是由 Stonebraker 等组成的 DBMS 委员会在 1990 年提出的“第三代数据库系统宣言”。这个宣言提出了“第三代数据库系统必须支持更加丰富的对象结构和规则”的观点。在过去的十年中, 除了理论研究和原型系统外, 一些优秀的实用系统已经问世。对象-关系数据库系统可以说是面向对象数据库领域中商品化最好的。比较著名的有 UniSQL, IBM DB2/CS V2.0, INFORMIX Universal Server, CA Ingres 等。

2. 面向对象数据库

OODB 系统的现状可以概括为研究异常活跃、商品化进程步履艰难。值得欣慰的是, 在基本对象模型、面向对象查询语言、连接与索引机制等方面已经取得了一批有意义的研究成果, 一些原型和产品也已问世。比较著名的有 Gemstone, Ontos, O2, ObjectStore, Versant 等。令人遗憾的是, OODB 在产品研制方面仍然不象 RDB 那样规范和成熟, OODB 的标准化问题仍不能令人满意。即使已提出的 ODMG-93 标准, 对厂商仍然缺乏约束力和指导作用。

3. 演绎面向对象数据库

在复杂对象模型及相应逻辑语言方面, LPS 和 LDL 是两个有代表性的范例。LPS [3] 基于二类逻辑, 支持两类对象——单个对象和集合对象。LPS 主要使用全称谓词来存取集合类型数据, 不允许集合元素的嵌套。由 MCC 公司设计的 LDL [4] 语言虽然和 LPS 有许多类似之处, 但提出了一些新的概念和方法, 因此功能更强大。在 LDL 中, 集合类型可以支持两类使用形式——枚举和聚合。尤其是聚合可以嵌套, 使得 LDL 具有很强的集合操作能力。

IQL 和 LOGRES 是两个面向对象的演绎查询语言 [2, 3, 4]。IQL 的突出贡献可谓对象标识 (OID) 的引入。Abiteboul 认为 OID 是面向对象数据库查询语言的基本原语, 是一个类型化的指针。在 IQL 中, OID 被作为有向图的编码来使用。通过控制指针的使用, IQL 可以严格表达数据库操作。LOGRES 对 IQL 的最大改进是数据函数的

使用,其直接的好处就是可以直接支持继承、多态等机制。

4. 结合其他技术的面向对象数据库

数据库组件技术的主要研究集中在 DBMS 的开放体系结构; DBMS 对对象数据的支持; DBMS 对操作系统、操作语言等的协同工作。近几年,以数据库组件技术为中心的移动数据库市场呈现一派繁荣景象。Sybase 公司由于 Adaptive Server Anywhere 的推出得到了应得的实惠,可谓移动数据库的一面旗帜。支持持久对象的程序设计语言的主要贡献不是表现在商业意义上,因为至今离商业实用还有很大距离。但是,它的一批研究成果(如持久对象模型、导航式程序设计界面等)已经为其他方法所借鉴。

扩充数据库系统的工具包方法也比期望的商业效果要差,这主要是它的绝对开放性和广泛包容性理念导致灵活性和简洁性的降低,关系-对象数据库系统的冲击(关系-对象数据库系统已经为大多数用户提供足够的扩充性)也是制约其发展的原因之一。不过,近几年出现的面向对象的客户包装层方法倍受关注。这种方法主要是在关系数据库之上增加一个对象包装层,旨在支持用户在关系数据库基础上开发面向对象的客户端应用。HP, Next, Persistence Software 等公司都推出了这类产品。

面向对象与数据库技术结合的产品实例

本节介绍两个产品实例,主要阐述它们比较有代表性的特点,从中可以帮助我们更好地了解面向对象数据库技术的概念、方法和趋势。

1. UniSQL/X 的面向对象数据模型

UniSQL 公司在面向对象与数据库技术结合方面作出了贡献,其中有代表性的产品是 UniSQL/X 和 UniSQL/M。UniSQL/X 把关系模型统一在面向对象模型基础上,支持对象封装、继承等机制。UniSQL/M 在 UniSQL/X 基础上,增加了存取外部数据库的手段,是一个适合联邦数据库系统的模型和查询语言。

(1) 关系模型与面向对象模型的统一

关系模型	面向对象模型
关系	类
元组	实例
属性	数据成员
过程 *	方法
子关系 *	子类
父关系 *	超类

(2) 关系的嵌套。尽管关系代数中支持子关系/父关系概念,但是传统的数据库中的关系模型不允许关系的嵌套,这导致的直接问题就是无法满足复杂数据类型的处理。UniSQL/X 在兼容简单数据类型前提下,扩充了对关系的嵌套的支持。下面的例子可以说明这个问题。

```
CREATE TABLE Student (
    S# CHAR (6), Name CHAR (10), Age
    INTEGER, Hobby Fan);
CREATE TABLE Fan (
    F# CHAR (4), Name CHAR (20));
```

(3) 数据封装。数据的封装性是面向对象数据模型的基础,因此必须扩充传统的数据库中关系模型使之引入数据集的过程定义。下面的例子可以说明 UniSQL/X 实现机制。

```
CREATE TABLE Student (
    S# CHAR (6), Name CHAR (10), Age
    INTEGER, Hobby Fan)
```

```
PROCEDURE GetHobbyName CHAR;
GetHobbyName 过程为索取学生的爱好信息提供了对外接口。
```

(4) 对象继承。UniSQL/X 支持对象的继承机制(单重继承和多重继承),下面是一个单重继承的例子。

```
CREATE TABLE Student (
    S# CHAR (6), Hobby Fan) AS CHILD OF
Person;
CREATE TABLE Person (
    Name CHAR (10), Sex CHAR (2), Age
    INTEGER);
```

(5) 数据操纵方法。事实上,上面已经建立了一个基于关系和面向对象统一的数据模型基础上的数据库结构(图 1 所示)。

下面是 UniSQL/X 建立在上面数据库结构基础上的两个查询例子。

例 1 查找男同学中爱好网球的信息

```
SELECT *
FROM Studet
WHERE Sex= '男' AND Hobby. Name=
'Tennis';
```

例 2 查找学生中年齡大于 20 岁的姓名、年龄、爱好信息

```
SELECT Name, Age, Hobby. Name,
```

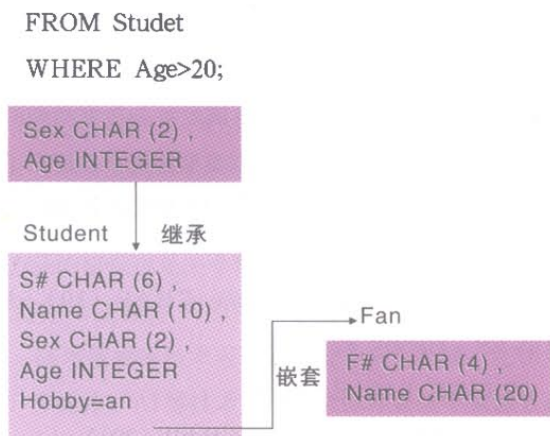


图 1

2. INFORMIX Universal Server

上一小节从数据库结构及查询语言等角度介绍了 UniSQL/X, 本节将从面向对象数据库系统的体系结构、扩充技术等角度介绍 INFORMIX Universal Server 系统。

(1) INFORMIX Universal Server 的体系结构。如图 2 所示, INFORMIX Universal Server 主要包括如下部分:

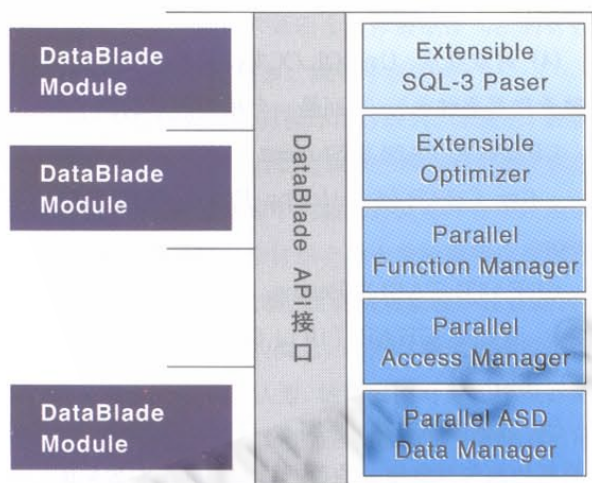


图 2 INFORMIX Universal Server 体系结构示意图

◆ 词法分析器 (Paser): 检查语句的合法性。特别地, 当有 DataBlade 模块加入后, 词法分析器将根据定义确定新的数据类型和函数等。

◆ 优化器 (Optimizer): 搜索查找语句的最佳路径等优化性工作。特别地, 支持 DataBlade 模块加入后的优化检索机制。

◆ 函数管理器 (Function Manager): 找出并执行查询语句中用到的函数。特别地, 支持 DataBlade 模块加入

后新函数的检索和执行。

◆ 存取方法 (Access Methods): 利用索引等技术提高数据存取效率。特别地, 当 DataBlade 模块引入复杂数据类型时, 提供高速二维索引的专用存取方法。

◆ 数据管理器 (Data Manager): 负责数据的物理输入/输出。特别地, 在 DataBlade 模块加入后, 可以快速定位到复杂数据类型的存储位置等。

◆ API 接口: 支持独立开发的面向对象模块的加入。

(2) INFORMIX Universal Server 的可扩充技术。INFORMIX 公司在收购了 Illustra 公司后, 全面继承并发展了它的扩充技术。再加之 INFORMIX 特有的 ASD (可伸缩体系结构) 技术, 使它推出的 INFORMIX Universal Server 成为面向对象数据库技术产品的佼佼者。

◆ 数据类型的扩充: 可以用面向对象方法 (如 C++, JAVA) 定义新的数据类型, 数据库中的字段可以是 HTML 文档、图象等复杂类型。在定义新的数据类型时, 实现在结构、规则、算子、函数和聚集等方面达到全方位的继承机制。

◆ 支持对象方法: 用户定义的函数是作为对象方法来使用的, 在封装性的前提下提供对外访问接口。

◆ 扩充存取方法: 随着新的数据类型的加入, 数据存取效率成为关键问题。它的即插即用的存取方法提供了一种随应用的发展扩充存取方法的灵活而高效的扩充机制。■

参考文献

- 1 Kim W., 面向对象数据库系统的研究方向, 计算机科学, Vol.25, No.5, 77-80, 1998
- 2 李勇等, 面向对象语言中的子类型关系, 计算机科学, Vol.26, No.3, 10-14, 1999
- 3 王浩等, 演绎的面向对象数据库研究, 计算机研究与发展, Vol.35, No.4, 289-294, 1998
- 4 王修伦等, 扩展演绎数据库技术, 计算机科学, Vol.25, No.5, 77-80, 1998
- 5 王燕, 面向对象的理论与 C++ 实践, 清华大学出版社, 1997
- 6 Shaw, G.M. and Zdonik S.B., An object-oriented query algebra, Data Engineering, Vol.12, No.3, 29-36, 1989
- 7 Informix 技术资料