

何炎祥 郑振楣 石树刚 编著

面向对象数据库

MIAN XIANG DUI XIANG

SHU JU KU

武汉大学出版社

面向对象数据库

何炎祥 郑振楣 石树刚 编著

武汉大学出版社

1995 年 6 月

图书在版编目(CIP)数据

面向对象数据库/何炎祥等编著. —武汉:武汉大学出版社,
1995.12

ISBN 7-307-02068-8

I. 面…

II. 何…

III. 数据库系统

IV. TP31

武汉大学出版社出版

(430072 武昌 珞珈山)

武汉华运印刷厂印刷

新华书店湖北发行所发行

1995年12月第1版 1996年3月第2次印刷

开本:850×1168 1/32 印张:9

字数:229千字 印数:1001-2000

ISBN 7-307-02068-8/TP·61 定价:9.80元

序 言

面向对象方法

面向对象方法(OO 方法)是一种方法学。OO 方法学的出发点和基本目标是使我们分析、设计和实现一个系统的方法尽可能接近我们认识一个系统的方法,也就是使描述问题的问题空间和解决问题的方法空间在结构上尽可能一致。因此,它涉及到分析方法、设计方法、思维方法和程序设计方法。

人们对一个系统或事物的认识是一个渐进过程,是在继承了以前有关知识的基础上,经过多次反复而逐步深化的过程。在这种认识过程中既含有从一般到特殊的演绎,又包括了从特殊到一般的归纳,例如,自上而下方法就是一种演绎方法,自下而上方法就是一种归纳方法。OO 方法既提供了从一般到特殊的演绎手段(如继承),也提供了从特殊到一般的归纳形式(如类),从而说明它比较自然地模拟了人类认识世界的方式,因而是一个很好的认知方法。

OO 方法学认为,任何事物都是对象(object),每个对象都有自己的运动规律和内部状态,每个对象都属于某个对象类,复杂对象可以由较简单的对象构成。不同对象的组合及相互作用便构成了我们要研究和分析的客观世界。

对象由一组数据和施加于这组数据上的一组操作构成。对象的每个操作称为“方法”(method)。对象的功能可以通过为其定义的一组方法来描述。对象的结构特征是由它的属性(attribute)表

示的。对象间的相互作用是通过“消息传递”(message passing)实现的。一个对象通过向另一个对象发送消息去激活它的某个方法,对象的每个方法都对应且仅对应一条消息。方法(操作)的具体实现细节,则封装在相应对象类的定义中,对外界是隐藏的。

一组具有相同结构特征和行为特征的对象构成一个类(class)。类中的所有对象都是其实例(instance)。可按类、超类或父类(superclass)、子类(subclass)的概念构成类层次结构。在这种层次结构中,一个类可继承其父类的全部描述(事实上可继承层次结构中位于其上面的所有类的全部描述)。

封装性(encapsulation)是一个源自抽象数据类型(ADT)的概念。对象具有封装性是指其内部细节对用户是隐藏的,用户只能见到对象封装界面上的信息,而且只能通过对象设计者提供的消息访问该对象。

多态性(polymorphism)是一个与类相关的概念,同一类的所有对象在收到同一条消息时,将采取同样的动作;不同类的对象在接收到同一条消息时可能采取不同的动作。不同对象对同一条消息采取不同的动作,这种情形就称为多态性。一般而言,多态性是指事物以多种形态存在,在程序设计语言中,则指数据具有多种类型,或一个名字可具有多种语义。

把程序的各种成分有机地组织起来的过程称为联编(binding)。联编若在程序的第一次运行前完成,则称为静态联编(static binding)或早期联编(early binding)。发生在程序运行时的联编称为动态联编(dynamic binding)或滞后联编(late binding)。要支持多态性,有时会要求采用动态联编技术。

面向对象语言

许多程序设计语言对面向对象语言(OOL)的形成起了积极作用,而OOL的研究又丰富和促进了OO方法的发展。80年代以

来, OOL 的研究形成了两大类, 一类是纯 OOL, 如 Smalltalk、Actor、Eiffel 等; 另一类是混合型 OOL, 即在传统的过程或程序设计语言中增加 OOL 的成分, 如 C++、Objective-C、OOPascal 等。

一般说来, 纯 OOL 的成分几乎都是“对象”, 这类语言侧重于方法的研究并强调开发快速原型的能力。而混合型 OOL 则比较强调运行效率及使传统程序设计者接受面向对象的思想。例如 LISP 与 OO 方法结合形成了混合型面向对象 LISP, 如 LOOPS、CLOS(Common Lisp Object System)、Flavors 等。在程序设计方面, C++ 比 Smalltalk 应用得更广泛, 效率也高得多。

Wegner 根据现有的 OOL 的特征来分类语言。他认为, 在现有的 OOL 中, 总共出现了 7 种语言特征:

对象 类 继承性 数据抽象

强类型化 并发性 持久性

现在还没有一种语言能支持所有这些特征。这 7 种特征有 128 种组合方案, 每一种组合方案都可以作为一种语言设计方案, 但一般仅考虑下述 6 种:

- 基于对象的语言, 支持对象
- 基于类的语言, 对象是类的实例
- 面向对象的语言, 支持类的继承性
- 面向对象的数据抽象语言, 支持数据抽象
- 面向对象的强类型化语言, 在编译时进行类型检查
- 支持并发性和持久性的面向对象的强类型化语言

面向对象应用

将 OO 方法应用到不同领域后, 就得到一系列相应领域的 OO 技术和应用, 例如:

- 面向对象分析(OOA)

- 面向对象设计(OOD)
- 面向对象实现(OOI)
- 面向对象语言(OOL)
- 面向对象数据库系统(OODBS)
- 面向对象操作系统(OOOS)
- 面向对象程序设计范式(OOPF)
- 面向对象人工智能(OOAI)
- 面向对象知识工程(OOKE)
- 面向对象软件工程(OOSE)
- 面向对象程序设计环境(OOPE)
- 面向对象智能决策支持系统(OOIDSS)
- 面向对象多媒体系统(OOMS)
- 面向对象地理信息系统(OOGIS)
- OO 体系结构、OO 仿真系统、OO 灵境或虚拟现实(virtual reality)系统等等。

面向对象数据库系统

Zdonik 和 Maier 认为,一个具备如下特征的 DBMS (DataBase Management System),才能称作是面向对象数据库系统(OODBS):

- 1) 能够支持数据库独立于应用程序而存在,支持数据库独立于操作数据库的系统而存在;
- 2) 提供概念或逻辑一级的数据抽象;
- 3) 支持数据库的查询和修改;
- 5) 支持数据库的物理重组,以适应不同应用对信息组织、信息利用和性能的不同要求;
- 6) 支持数据库控制功能,如保证语义的完整性、数据的安全性,并发控制和数据库恢复等;
- 7) 支持对象标识;

- 8) 支持复杂对象;
- 9) 支持对象的封装性。

Atkinson 等人于 1989 年在第一届国际演绎数据库和 OODB 会议上发表的论文“OODB 宣言”中总结了 OODBS 的一系列特征,并将它们分为“必备”、“任选”和“开放”三类。他们的观点是,只有具备下列所有“必备”特征的 DBMS 才称之为一个 OODBS:

1) 支持对象标识(对象的存在不依赖于其值,只依赖于它的标识),每个对象都有一个对象标识,对象间能根据对象标识而相互区分;

- 2) 支持复杂对象;
- 3) 支持类或类型;
- 4) 支持类或类型的层次结构,以获得继承性;
- 5) 支持封装性;
- 6) 支持重置、过载和动态联编;

7) 计算完备性 OODBS 的 DML (Data Management Language) 应具有足够强的表达能力,可以表示所有的可计算函数;

8) 可扩充性 系统应提供定义新类型的机制,用户定义的新类型应与系统预定义的类型之间没有差别,具有完全相同的操纵方式;

9) 支持持久性 对象的存活期与创建它的进程无关。对象的持久性不应依赖于其类型,每个对象都可(不经显示的转换)变成持久性对象;

10) 提供二级存储管理 包括索引管理、聚簇数据管理、对象缓冲、存取路径选择和查询优化,使对象的物理存储组织形式对用户透明;

11) 支持并发性 使得 OODBS 中的对象可被多个用户共享而不影响其一致性和完整性;

12) 支持数据库恢复 包括从硬件和软件故障中恢复的机制;

13) 系统应提供一种高级的独立于应用的查询功能。

简言之,一个 OODBS 首先应是一个 DBMS,同时又必须具备 OO 的特征。

OODBS 是数据库发展进程中的一个里程碑。进入 90 年代,OODBS 被人们广泛的研究,从而得到了迅速的发展。我们武汉大学数据库组以极大的兴趣关注这一进展,并积极开展有关方面的研究和开发工作,80 年代末就在研究生课程中增加了 OODBS 的内容,并完成了 OODBS 的原型系统——WOODBS。90 年代初结合 GIS 的应用,在 WOODBS 的基础上又开发了 SamBase,形成了一个面向对象的信息系统开发平台。多年来的教学和科研工作实践,使得我们更加喜爱它,更加觉得需要进一步发展它。为此,我们编写了此书,以期抛砖引玉,使计算机界的同行和有兴趣的读者与我们共同努力,以推动我国数据库的发展。

本书的组织方式和主要内容

本书共分十七章,第一章简单介绍了数据库技术的演变,面向对象方法及 OODB 的基本概念和基本特征。第二章介绍了 OO 核心数据模型和对核心模型的语义扩充。第三、四、十三章讨论了 OODBS 的基本接口,与其它数据库的关系,以及 OO 程序设计与数据库的集成技术。第六、七、十章介绍了查询模型、查询语言,以及查询优化和查询处理技术。第五、八、十一和十二章讨论了模式修改、授权机制、事务管理和进一步的语义扩充等内容。第九、十四章讨论了存储结构、存取策略和 OODBS 的基本结构。第十五、十六章概述了若干著名的 OODBS,并扼要介绍了 SamBase 的设计和实现思想。最后,第十七章给出 OODBS 发展中还需进一步研究的几个问题。

本书注重 OODBS 的基本概念、基本设计原理、基本实现方法以及内容的系统性、完整性和逻辑性,并特别强调理论与实际的结合。书后列有大量参考文献供读者查阅。本书可作为高等院校高年级学生、研究生的教科书,也可供有关科技工作者阅读参考。

致谢

在本书编写过程中,蔡一炜、陈燕涛、孙祥胜付出了辛勤的劳动。郭敏、林宏轩、江旻、李大庆、万庆伟、张彤、李涛、李立、庄严、朱坎、陈燕涛、严乐安、肖刚、易文根、高媛、孙祥胜、王延彬先后参加了 WOODBS 和 SamBase 的设计和实现工作,为本书提供了必要的素材。金丽莉编辑对本书的出版给予了全心的支持,在此一并表示深深的谢意。

虽然 OODBS 的研究已经取得了许多成果,但还有不少问题仍处于研究和探索阶段。由于水平所限,书中不足之处恳请同行和广大读者不吝指正。

作者

于武昌珞珈山

1995.5

内容简介

本书是一部论述 OODBS 的基本特征、数据模型、设计原理和主要实现方法的专著。内容包括:数据库技术的发展和演变,OO 核心数据模型,对核心模型的语义扩充,模式演变和修改,版本控制,查询优化和查询处理,授权,存储结构和存取策略,事务管理(并发控制和恢复),OODBS 基本结构和 OODB 基本界面,OO 程序设计与数据库的集成,以及 OODBS 的研究发展方向等。为便于理解,我们选用 ORION 数据库系统为背景贯穿全书,并扼要介绍了 SamBase 的设计和实现思想。本书注重 OODBS 的基本概念、基本设计原理和方法,以及内容的系统性、完整性和逻辑性,并强调理论与实际的结合。

本书可供高等院校的高年级学生、研究生、教师作为教学用书,也可供有关科技工作者阅读和参考。

目 录

序言	1
第一章 引论	1
1.1 数据库技术的演变	1
1.2 下一代数据库	1
1.2.1 传统数据库系统的特征	4
1.2.2 下一代数据库系统的特征	6
1.3 面向对象方法和面向对象数据库	7
1.3.1 面向对象方法	7
1.3.2 面向对象数据库	9
第二章 数据模型	11
2.1 核心数据模型	12
2.1.1 核心建模概念	12
2.1.2 核心概念的进一步讨论	13
2.1.2.1 对象和对象标识符	13
2.1.2.2 属性和方法	17
2.1.2.3 类	17
2.1.2.4 类层次和继承	18
2.2 对核心模型的语义扩充	20
2.2.1 复合对象	21
2.2.2 版本	23
第三章 OODBS 的基本接口	25
3.1 消息传递	29

3.2	DDL	29
3.3	DML	30
3.4	DCL	32
第四章	与非面向对象数据库的关联	33
4.1	概念的重叠	33
4.1.1	设计数据库	33
4.1.2	层次及网络数据库	34
4.1.3	可扩充数据库	35
4.1.4	语义数据库	35
4.1.5	关系数据库	36
4.2	缺乏标准	37
第五章	模式修改	39
5.1	分类	40
5.2	模型	42
5.2.1	模式修改的不变式	43
5.2.2	单模式修改规则	45
5.3	语义	48
5.3.1	软改变	48
5.3.2	硬改变	51
5.4	实现	52
5.5	方法无效性	54
5.6	模式版本	55
第六章	查询模型	57
6.1	单操作查询	57
6.1.1	模式图	58
6.1.2	查询图	59
6.1.3	复合类层次结构中的环形分枝	62
6.1.4	查询结果	65
6.2	多操作查询	69

6.2.1 连接	69
6.2.2 集合操作	73
6.3 其它查询模型	74
第七章 查询语言	75
7.1 查询语言概述	75
7.2 单操作查询	76
7.2.1 无环查询	77
7.2.2 有环查询	85
7.2.3 方法	89
7.2.4 查询结果	91
7.3 多操作查询	92
7.3.1 连接	92
7.4 集合操作	94
第八章 授权	96
8.1 关系数据库的授权模型	97
8.2 面向对象概念的影响	98
8.3 一个基本授权模型	100
8.3.1 授权主体层次	101
8.3.2 授权类型层次	103
8.3.3 授权对象的隐含授权	103
8.3.4 操作	106
8.4 基本授权模型的扩充	106
8.4.1 类相关授权的授权对象	107
8.4.2 类层次	108
8.5 授权目录	110
8.5.1 授权主体层次	111
8.5.2 授权对象	111
第九章 存储结构和存取策略	114
9.1 实例的存储结构	114

9.2 类对象的存储结构	116
9.3 数据库模式的存储结构	117
9.4 磁盘和页面组织	118
9.5 聚簇	120
9.6 实例存取策略	122
9.6.1 散列	122
9.6.2 索引	123
9.6.2.1 类层次索引	123
9.6.2.2 嵌套属性索引	125
第十章 查询优化和查询处理	129
10.1 OO 查询和关系查询的结构类似性	129
10.2 查询处理技术	133
10.3 对关系查询处理技术的必要修改和扩充	135
第十一章 事务管理	137
11.1 传统事务管理	138
11.1.1 并发控制	138
11.1.1.1 OO 概念的特点	138
11.1.1.2 ORION 系统中的并发控制	140
11.1.2 恢复	147
11.1.3 虚拟事务	149
11.2 长事务管理	150
11.2.1 长期等待	150
11.2.1.1 对象的版本	151
11.2.1.2 模式的版本	151
11.2.1.3 组事务	152
11.2.1.4 修改通知	152
11.2.1.5 不可串行化事务	153
11.2.1.6 软锁	153
11.2.2 恢复	154

第十二章 语义扩展	155
12.1 版本.....	155
12.1.1 语义.....	156
12.1.2 消息.....	157
12.1.3 实现.....	159
12.1.4 查询处理.....	160
12.1.5 模式演变.....	161
12.1.6 授权.....	162
12.2 复合对象.....	164
12.2.1 语义.....	164
12.2.2 消息.....	166
12.2.3 实现.....	169
12.2.4 模式演变.....	170
12.2.5 授权.....	171
12.2.6 并发控制.....	174
12.2.7 版本.....	180
第十三章 OO 程序设计与数据库的集成	183
13.1 概述.....	183
13.2 内存对象管理.....	185
13.2.1 对象缓冲.....	185
13.2.2 驻留对象描述符.....	187
13.3 对象缓冲对数据库系统体系结构的影响.....	192
13.3.1 刷新对象缓冲.....	192
13.3.2 查询处理.....	193
第十四章 OODBS 基本结构	195
14.1 概述.....	195
14.2 进程结构和通信子系统.....	198
14.3 对象子系统.....	201
14.4 事务子系统.....	201

14.5 存储子系统.....	203
14.5.1 内存管理.....	203
14.5.1.1 双缓冲技术.....	203
14.5.1.2 对象的获取和刷新.....	205
14.5.2 二级存储管理.....	205
14.5.3 查询计算.....	206
第十五章 OODBS 概观	210
15.1 商品化系统.....	210
15.1.1 GemStone	210
15.1.2 Vbase	212
15.1.3 Statice	213
15.2 工业企业界研究原型.....	214
15.2.1 IRIS	214
15.2.2 O2	216
15.2.3 Jasmin	216
15.3 学院界研究原型.....	217
15.3.1 ENCORE/ObServer	217
15.3.2 POSTGRES	218
15.3.3 AVANCE	219
15.3.4 OZ+	219
15.3.5 EXTRA	220
第十六章 SamBase 的设计与实现	221
16.1 引言.....	221
16.2 SamBase 对象模型	224
16.2.1 对象结构.....	224
16.2.1.1 属性类型构造符.....	225
16.2.1.2 属性.....	226
16.2.1.3 简单对象和复杂对象.....	227
16.2.1.4 对象存储.....	228