Object-Oriented Databases

D.N.CHORAFAS
H.STEINMANN

面向对象数据库

清华大学出版社
Prentice-Hall

Object-Oriented Databases
面向对象数据库

Dimitris N. Chorafas/Heinrich Steinmann

清华大学出版社
Prentice-Hall International Inc.

(京)新登字 158 号

内 容 提 要

这是一本从基本概念出发论述面向对象这一方法的著作。主题是面向对象(OO)数据库系统的表示法、控制和一致性。作者概要地介绍了通过对象的运用把数据处理、决策支持和知识工程综合起来的方法,深入讨论了继承性、数据抽象、封装、多态形式、元规则等。本书主要论题为:OO 环境结构化的方法;语义模型和对象方法在软件复用性中的作用;OO 程序设计和软件工程的主要途径;对象环境中的元程序设计;运用对象进行有效的原型开发;原型开发如何有助于分布式数据库方法;对象软件中的基本成份;空间和时态语义学实例;对象方法优于相关方法的原因;美国和日本最优秀的对象数据库管理系统;特殊的面向对象数据库管理软件的特性;与数据级并行性有关的对象处理;高性能计算和通信的挑战。

中国〔大陆〕版

Object-Oriented Databases

面向对象数据库

Dimitris N. Chorafas/Heinrich Steinmann

清华大学出版社
Prentice-Hall Inc. 出版
北京 清华园
清华大学印刷厂印刷
新华书店总店科技发行所发行

开本:787×1092 1/16 印张:21
1994 年 5 月第 1 版 1994 年 5 月第 1 次印刷
印数:0001—1000
ISBN 7-302-01497-3/TP · 602
定价:46.00 元

Foreword

An Introduction to Object-Oriented Databases

John Pfaltz, PH.D.

In the information sciences there has been a traditional separation between data and the algorithms that process such data. Over the years there have been efforts to unify this duality, as in intelligent systems written in LISP, which can execute their own data; but for the most part this duality is as firmly rooted in computer science as is the mind-body duality of Descartes in philosophy. It is firmly rooted in the architectures that we use: A memory register is not an arithmetic register.

In many applications, such as computation and traditional bookkeeping, this duality is of little consequence. The separation of the mathematical domain from the theorems and procedures that are valid over the domain, or of the corporate books and ledgers from the accountants and management who use them, have their counterparts in the computational model itself. But in newly emerging applications and disciplines, such as more general information processing or knowledge databases, the duality can have unfortunate consequences. We can begin to appreciate these consequences by asking ourselves, "What is information?" or "What is knowledge?."

This is not an easy question to answer. However, it is clear that pure data—that is, some sequence of bits stored on a disk—is neither information nor knowledge. Some might assert that information is data that generates, or modifies, an action in the external world; that is, it is "useful data" in the sense that it is actually used. For example, in a knowledge base, data must actually constrain or modify the behavior of the system in order to be considered "knowledge." In this view, obvious tautologies or irrelevant axioms are not knowledge, because they are never used. This kind of functional definition of information actively involves an external context and an understanding of its actual use in that context.

Alternatively, one can, as a more narrowly conceived database technician, attempt to divorce oneself from the actual use of the data and assert that data becomes information when it is made available, and presented to, a decision maker in a form that *could* be used in a decision process. But either view assumes that information involves both data and a user, and that the transfer between them is essential. This approach significantly extends the role of information management from just storing and retrieving data to processing it appropriately and engaging in at least half of the vital supplier/user interface.

In the authors' view, the fundamental issues facing information management in the 1990s are much broader than just storing bytes on gigabyte disks and retrieving them rapidly with well-specified query procedures. Such manipulation of passive data is just one end of a data processing spectrum. The center of this spectrum includes forms of *active* data that can modify the presentation of passive data and can help to shape and formulate appropriate queries from incompletely specified requests, and at its other extreme can be part of the decision-making process itself. Consequently, the reader should not expect to find just a manual of state-of-the-art object-oriented database techniques; this is not such a book. Instead, this book is more visionary. It explores and brings together in a single volume many of the new ideas regarding how we may expect information to be managed in the next decade. Some of the information management approaches covered are relatively well developed, with practical versions emerging, and about to emerge, in the commercial marketplace. Others are only in prototype form, and some are purely experimental. A few, we can expect, will turn out to be impractical or just downright "wrong." Nevertheless, given the current ferment in database management arising from new storage technologies, inexpensive distributed-processor workstations, and techniques of massively parallel processing, serious database professionals must become aware of these new concepts in order to evaluate their applicability in their own domain of application.

The authors have chosen the object-oriented approach to database management as their vehicle of development. This is a natural choice because in object-oriented programming one also seeks to blur the duality between data and process. Objects have state, that is, associated passive data values; but they also have behaviors, which are active. The object-oriented approach is not the only database model that attempts to fuse passive data with active processing to generate data. The reader should also be aware of a novel approach that David Gelernter has developed in his "Linda" system, and the approach to data parallel algorithms that Daniel Hillis has embodied in the "connection machine," in which each data element is a complete but tiny processor. Nevertheless, the authors' choice to focus on object-oriented methods is a wise one. The object-oriented approach has already shown itself to be extremely valuable in the development of human/machine interfaces in distributed networks. Thus it is a proven tool for implementing the most basic data transfer step that underlies all conceptions of information, and that we know will be fundamental in all future database applications. Moreover, there have been successful object-oriented implementations of CAD/CAM systems and of knowledge-based systems to which many database applications will have to interface. Finally, object-oriented programming is a relatively mature approach for which stable software has, and is being generated.

However, object orientation is not a single unified approach either to computing or to data representation. It is a general way of thinking that both colors, and is colored by, one's application area. For example, in expert and knowledge-based systems the concept of meta-objects is crucial; in operating and pure database systems they are irrelevant. How does one therefore present the object-oriented approach to database representation? One can elect to present in an all-inclusive manner the myriad developments that are emerging under the object-oriented database banner, or one can choose to filter the presentation so as to concentrate on those that are "most promising." The authors have chosen the former path, which, I believe, is the correct one; because the latter would have them imposing their personal view of the object-oriented database field. Still, by presenting the broad panorama of object orientation with all its possible interpretations, the authors are certain to upset virtually every reader who already has some familiarity with the field. For example, I find myself taking violent exception to a number of their assertions, but in every case, the assertion is one that is accepted by some subset of the computer science and data modeling communities, and which, for that reason should not be ignored.

Consequently, the reader is warned that he, or she, cannot approach

this provocative book with a passive mind. Not all of its ideas can be accepted at face value, as the authors themselves acknowledge in their discussion of object-oriented SQL in Chapter 8. The reader must actively test each idea, each concept, and each data management approach against their understanding of what is computationally practical and their experience with a particular field of application. One result of this critical approach will be a filter that dismisses some sections, even some chapters, as impractical or irrelevant for their purposes. But another result of such an active involvement will be a much better understanding of what individuals and companies can expect to be the information requirements of the future, what methodologies may be available, and which paths should be aggressively followed and developed.

On many fronts, new information management applications have pushed the relational database mode, and its associated implementations, to their limits. New database technologies based on the object-oriented model are coming to the fore. This book presents the best overview of ways to think about, and use, these object-oriented databases that is available today. Awareness of its contents is of vital importance to anyone who claims to be a database professional.

John Pfaltz, PH.D.
Professor of Information Science
and Director, Institute for Parallel Computation,
School of Engineering and Applied Science
University of Virginia

Preface

A popular illusion holds that books are inanimate objects belonging to cloistered shades and academic quiet, away from the world of action. According to this misconception, books are full of impractical theory therefore of little significance to business people.

Yet, throughout industry there is plenty of evidence that far from being of only theoretical potential books are frequently dynamic, capable of changing the direction of events. This is the case with the description of new scientific and technological developments which have the power to overtake the processes and concepts with which we have been dealing for years, and hence have become a second nature to us.

The advent of object-oriented solutions in information technology fits this description of a new wave of change. One of the critical advantages of object orientation is the simplification achieved through semantics, the ability to give meaning to our queries when communicating with the sprawling distributed databases which we have constructed and we use daily. The following domains offer the best opportunities for the implementation of this approach:

- Computer aided design (CAD)
- Computer aided manufacturing (CAM)
- Computer integrated manufacturing (CIM)
- Complex market-oriented operations (Forex, securities)
- Cross-functional projects (risk management)
- Cross-departmental projects (cost control)
- Cartographical implementations

- Computer assisted software engineering (CASE)
- Office automation (OA)

Object-oriented database systems, their representation, control, and consistency is the subject of the present book. Object representation is the basis of a new concept: The trend to intelligent distributed databases which will dominate the 1990s.

This reference under no condition means that all database management systems (DBMS), which so far have provided good services, will be scrapped. Companies are not going to discard over a couple of years their millions, often hundreds of millions, of dollars in programming and DBMS investments to join a revolution—even if object solutions were a revolution, which they are not.

Object-oriented approaches are an evolution, not a revolution, and clear-eyed companies are actively looking for better solutions to their problems. This is particularly true of operations in the nine domains just outlined—but the serious reader should be warned that object representation is no cure for all illnesses. It is just a step forward that is worth trying.

This book starts with the fundamentals. In a comprehensive manner requiring no prerequisite knowledge, Chapter 1 explains what is meant by objects, elaborating on the concepts of type and class. Chapter 2 focuses on inheritance, as well as the meaning of data abstraction, encapsulation, polymorphism, metarules, constraints, concurrency, and dynamic binding.

Chapter 3 builds upon the background provided by the first two chapters, explaining how to architecture an object-oriented environment. Its topics include the handling of object-based messages, referential integrity, versioning, and ad hoc queries. Chapter 4 presents the crucial issues associated with semantic modeling and conveys the role of object solutions on software reusability.

A pillar on which this whole approach rests is the difference between the record-based database management that we have had so far—whether on relational, network, or hierarchical principles—and object orientation. This change alone would justify an object database; the facts behind this statement are the subject of Chapter 5.

A great deal of work can be successfully done through object solutions that cannot be satisfactorily accomplished otherwise. Long transactions is a good example, and it is the theme of Chapter 6. This, however, brings under perspective the need to handle programming concepts.

Software engineering and database management have traditionally been separate disciplines. Classically, file management technology has con-

centrated on static issues of information storage; programming handled the dynamic software aspects.

However, since the development of relational database models and the evolution of fourth generation languages, the two disciplines have been combining towards systems that model both processes and data. This has been further underlined with object-oriented solutions.

Chapter 7 introduces the main approaches to object-oriented programming, including the new discipline characterizing software engineering. Metaprogramming in an object environment is the subject of Chapter 8, which also presents the Object SQL (OSQL) shell.

Since experimentation is key to successful implementation, the theme of Chapter 9 is that of prototyping by using objects, including graphical solutions, to aid in man-machine communication. In an implementation sense, Chapter 10 carries this concept further by discussing how prototyping can help distributed database solutions.

Given the fact that able approaches to the management of databases and object programming are interwoven, Chapter 11 devotes itself to the nuts and bolts of object software. Particular attention is being paid to C++ as an assembler level language.

The focal point of Chapter 12 is spatial and temporal semantics. This is by no means a theoretical issue. It involves the concept of computer-generated space as well as the modeling of temporal data and the patterns which exist in semantic representation. Chapter 13 presents practical examples on this very subject.

Chapter 14 discusses the limits of relational DBMS. First, the notion of a multidatabase is defined, then capitalizing on the background already built regarding long transactions, Chapter 15 demonstrates why object solutions perform better than relational ones.

Thereafter, interest is shifted to the notion of an object database management system—and why its functionality goes well beyond the relational capabilities which are dominant today. Chapter 16 presents the reasons for this new departure in database management—and the goals we are after.

The characteristics of specific object-oriented DBMS are explained in Chapters 17 to 20. Choosing among available commodity software is not easy; there are more than a dozen alternatives to select from:

- Chapter 17 discusses Ontos and Gemstone
- Chapter 18, Versant and Object Store
- Chapter 19, Iris and Pegasus

Chapter 20 presents the main object DBMS made in Japan: Mandrill by Hitachi, Odin by NEC, MIB by Mitsubishi. Also, the fundamental work on practical applications done by the Electrotechnical Laboratory.

The book concludes with the treatment of objects in connection to data level parallelism. Chapter 21 demonstrates that an object orientation is not just the better approach for the new generation of computers—it is the only one which can be effective. Parallelism is vital, and this should not be surprising. Nature itself is massively parallel.

Subsequently, Chapter 22 introduces the challenge of high performance computing and communications (HPCC). Taking a hypercube architecture as an example, it explains how Input/Output intensive applications can be handled in an able manner through object orientation—which amounts to new perspectives in computation.

The main emphasis of this book is practical: How can object-oriented concepts be put to fruitful use? At the same time, to enhance the reader's background, some of the presentation is directed at the ongoing research on object-oriented solutions and the results which have been obtained so far.

The book is addressed to information technology practitioners who want to upgrade their knowhow—as well as researchers who wish to understand the issues, tools, and architecture of the coming wave in database management. In addition, this book can be used by students who are taking courses in object-oriented systems, DBMS, or programming languages.

The authors feel indebted to a number of information technology scientists and to corporate and university researchers who contributed facts and ideas during the extensive research which has taken place throughout 1991 in the United States, Japan, and Europe. The companies which made a contribution and the participants to this research are identified in the Acknowledgments.

Particular indebtedness is felt by the authors to the object-oriented database and programming specialists who reviewed and commented on parts of this work. Dr. Dennis McLeod and Dr. John Pfaltz have been most helpful in this regard. To Eva-Maria Binder goes the credit for the artwork, typing and index.

Dimitris N. Chorafas
Heinrich Steinmann
January, 1993

Contents

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

CHAPTER 7

CHAPTER 8

CHAPTER 9

CHAPTER 10

CHAPTER 11

CHAPTER 12

CHAPTER 13

CHAPTER 14

CHAPTER 15

CHAPTER 16