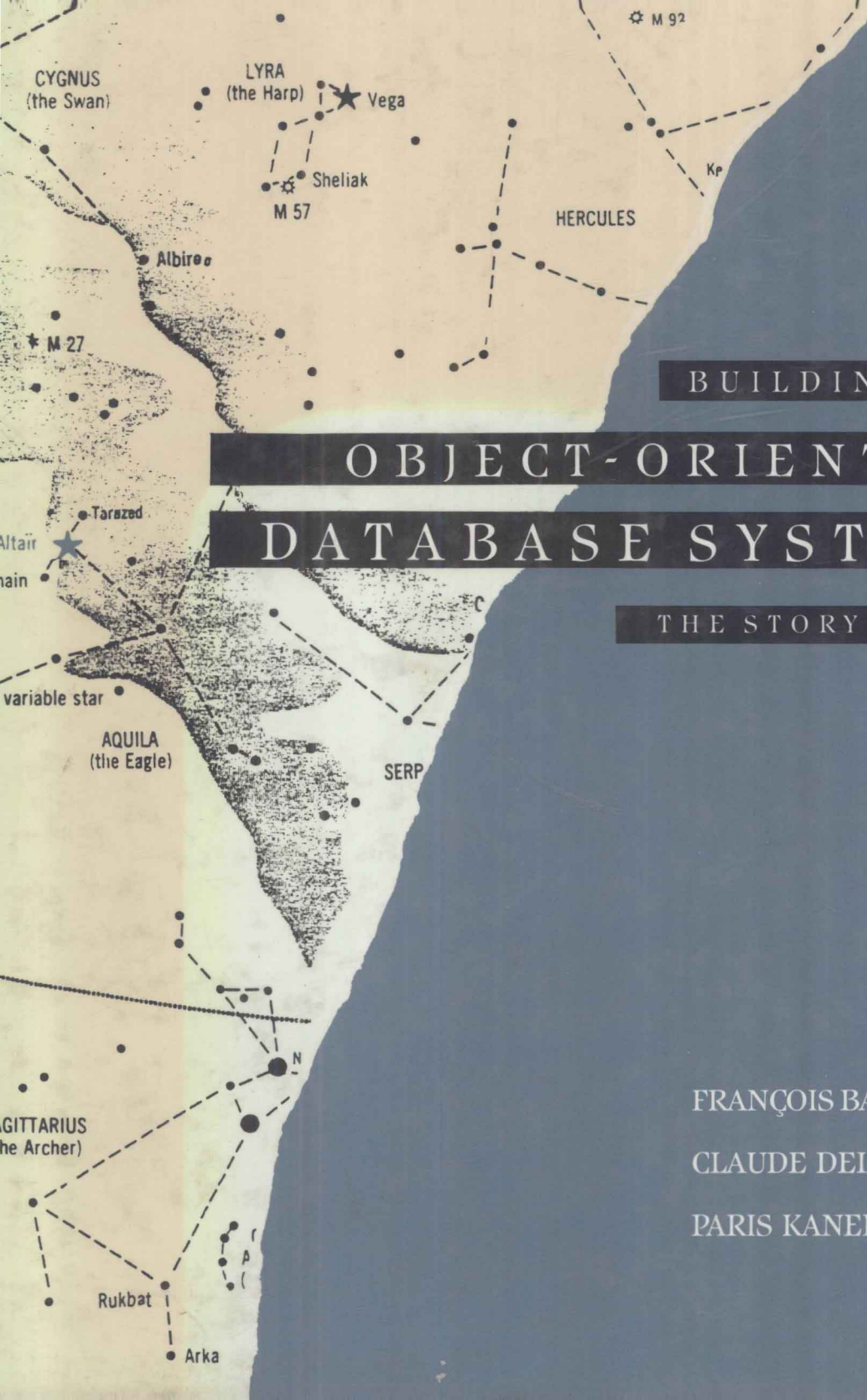

BUILDING AN OBJECT-ORIENTED DATABASE SYSTEM

THE STORY OF O_2

EDITED BY

FRANÇOIS BANCILHON

CLAUDE DELOBEL

PARIS KANELLAKIS

BUILDING AN OBJECT-ORIENTED DATABASE SYSTEM

THE STORY OF O_2

Edited by

Francois Bancilhon
Claude Delobel
Paris Kanellakis

MORGAN KAUFMANN PUBLISHERS
SAN MATEO, CALIFORNIA

Senior Editor: *Bruce M. Spatz*
Production Manager: *Yonie Overton*
Cover Design: *Wells Larson & Associates*
Copyeditor: *Tony Hicks*
Composition/Indexing: *SuperScript Typography*
Proofreaders: *Lynn Meinhardt, Gary Morris, Susan Festa*

Acknowledgements

Chapter 1: M. Atkinson, F. Bancilhon, D. DeWitt, K. Dittrich, D. Maier, S. Zdonik, "The Object-Oriented Database System Manifesto." © 1990, Elsevier North Holland. With permission of the authors and the publisher from *Proceedings of the First International DOOD Conference, Kyoto, Japan, December 1989*, edited by J. Kim, J. M. Nicholas, and S. Nishio. *Chapter 2:* O. Deux et al., "The Story of O_2." © 1990, IEEE. With permission of the authors and the publisher from *Transactions on Knowledge and Data Engineering*, 2(1), March 1990: 91–108. *Chapter 4:* C. Lécluse, P. Richard, and V. Vélez, "O_2, An Object-Oriented Data Model." © 1988, Association for Computing Machinery, Inc. With permission of the authors and the publisher from *Proceedings of the ACM SIGMOD Conference, Chicago, Illinois, June 1988. Chapter 5:* S. Abiteboul and P. Kanellakis, "Object Identity as a Query Language Primitive." © 1989, Association for Computing Machinery, Inc. With permission of the authors and the publisher from *Proceedings of the ACM SIGMOD Conference, Portland, Oregon, June 1989. Chapter 6:* S. Abiteboul, P. Kanellakis, and E. Waller, "Method Schemas." © 1990, Association for Computing Machinery, Inc. With permission of the authors and the publisher from *Proceedings of the 9th ACM SIGACT-SIGMOD-SIGART Conference on Principles of Database Systems, Nashville, Tennessee, April 1990. Chapter 7:* R. Zicari, "A Framework for O_2 Schema Updates." © 1991, IEEE. With permission of the author and the publisher from *Proceedings of the 7th IEEE International Conference on Data Engineering, April 8–12, 1991, Kobe, Japan. Chapter 10:* G. Barbedette, "LispO_2, A Persistent Object-Oriented Lisp." © 1989, Springer-Verlag. With permission of the author and the publisher from *Proceedings of the Second EDBT Conference, Venice, Italy, March 1989. Chapter 12:* S. Cluet, C. Delobel, C. Lécluse, and P. Richard, "Reloop, An Algebra-Based Query Language for an Object-Oriented Database System." © 1990, Elsevier North Holland. With permission of the authors and the publisher from the *Proceedings of the First International DOOD Conference, Kyoto, Japan, December 1989*, edited by J. Kim, J. M. Nicholas, and S. Nishio. *Chapter 20:* M. Cart and J. Ferrié, "Integrating Concurrency Control into an Object-Oriented Database System." © 1989, Springer-Verlag. With permission of the authors and the publisher from *Proceedings of the Second EDBT Conference, Venice, Italy, March 1989. Chapter 22:* D. Plateau, R. Cazalens, J. C. Mamou, and D. Tallot, "Building User Interfaces with the Looks Hyper-Object System." © 1990, Springer-Verlag. With permission of the authors and the publisher from *Eurographics Workshop on Object-Oriented Graphics, Königswinter, Germany, June 1990. Chapter 26:* G. Arango, "Self-Explained Toolboxes: A Practical Approach to Reusability." © 1990, TOOLS/90 Société des Outils du Logiciel. With permission of the author and the publisher from *Proceedings of the TOOLS 90 Conference, Paris, 1990. Chapter 28:* M. Scholl and A. Voisard, "Object-Oriented Database System for Geographic Applications: An Experience with O_2." © 1990, Springer-Verlag. With permission of the authors and the publisher from *International Workshop on Geographical Databases (Esprit Basic Research Series), Capri, Italy, May 16–17, 1991.*

Morgan Kaufmann Publishers, Inc.
Editorial Office:
2929 Campus Drive, Suite 260
San Mateo, CA 94403

96 95 94 93 92 5 4 3 2 1

Library of Congress Cataloging in Publication Data is available for this book.
ISBN 1-55860-169-4

BUILDING AN OBJECT-ORIENTED DATABASE SYSTEM

THE STORY OF O_2

THE MORGAN KAUFMANN SERIES IN
DATA MANAGEMENT SYSTEMS

Series Editor, Jim Gray

BUILDING AN OBJECT-ORIENTED DATABASE SYSTEM: THE STORY OF O_2
Edited by François Bancilhon (O_2 Technology),
Claude Delobel (O_2 Technology), and
Paris Kanellakis (Brown University)

TRANSACTION PROCESSING
Jim Gray (Digital Equipment Corporation) and
Andreas Reuter (Stuttgart University)

DATABASE TRANSACTION MODELS FOR ADVANCED APPLICATIONS
Edited by Ahmed K. Elmagarmid (Purdue University)

A GUIDE TO DEVELOPING CLIENT/SERVER SQL APPLICATIONS
Setrag Khoshafian (Portfolio Technologies, Inc.),
Arvola Chan (Versant Object Technology),
Anna Wong (CLaM Associates), and
Harry K. T. Wong (Nomadic Systems)

THE BENCHMARK HANDBOOK FOR DATABASE AND TRANSACTION PROCESSING SYSTEMS
Edited by Jim Gray (Digital Equipment Corporation)

CAMELOT AND AVALON: A DISTRIBUTED TRANSACTION FACILITY
Edited by Jeffrey L. Eppinger (Transarc Corporation),
Lily B. Mummert (Carnegie Mellon University),
and Alfred Z. Spector (Transarc Corporation)

DATABASE MODELING AND DESIGN: THE ENTITY-RELATIONSHIP APPROACH
Toby J. Teorey (University of Michigan)

READINGS IN OBJECT-ORIENTED DATABASE SYSTEMS
Edited by Stanley B. Zdonik (Brown University) and
David Maier (Oregon Graduate Center)

READINGS IN DATABASE SYSTEMS
Edited by Michael Stonebraker (University of California at Berkeley)

DEDUCTIVE DATABASES AND LOGIC PROGRAMMING
Jack Minker (University of Maryland)

Preface

Object-oriented database systems are new software systems integrating techniques from databases, object-oriented languages, programming environments, and user interfaces. We believe (and hope to convince the reader) that this particular synthesis of computer science ideas and software tools is larger than its parts. In fact, it is creating a new generation of database technology.

This book provides an in-depth perspective of the new technology through the description of a complete example prototype: the object-oriented database system O_2. The exposition ranges from the data model through the system implementation to applications. The format of the book is a commented and edited collection of papers that cover all aspects of the prototype software system O_2 (focusing on its V1 version). The authors are designers, implementors, and users of this system.

The articles collected here contain a wealth of essential details on all aspects of building an object-oriented database system. This is knowledge that can help researchers, database designers, and users to assess the nature and potential of the new technology.

Although they have long been successful in business, database systems have not been fully utilized for advanced applications such as office information systems (OIS) and computer-aided design (CAD). These applications have new requirements in design environments, transaction mechanisms, and complex or multimedia data types. O_2 has been built with such advanced applications in mind. It is not just an extension of a network or relational database system tailored to specialized applications, but represents an integrated approach to software engineering that combines object-oriented programming and database technology.

During the last decade, object-oriented programming concepts (such as classes of objects with methods and inheritance) and languages (such as Smalltalk or C++) have received a great deal of attention. One reason for the popularity of object-oriented paradigms is that data abstraction, modularity, and code reusability are key elements in building large software systems. Languages and programming environments that emphasize these three principles are bound to impact experimental computer science. However, one should bear in mind that the software engineering problems (which object-oriented languages were

designed to address) are very different from the problems that originally led to the development of database management systems.

Database systems evolved, quite independently from programming languages, because of the practical need for efficient manipulation of large amounts of structured information. The insight that data is an integrated resource which is independent of application programs has led to more than twenty-five years of database technology. Four particular themes have been central in database research and development: (1) data persistence beyond the scope of application programs, (2) very high level but reasonably fast query languages that are independent of the physical organization of the data, (3) efficient secondary storage management for large amounts of structured information, and (4) transaction management guaranteeing access by concurrent users, data integrity, security, and recovery from faults.

This overall emphasis of database technology on performance has made its integration with object-oriented programming a challenging task. For example, specific concrete types, such as records and lists, were given prominent roles in various data description languages (DDLs). This made good implementations possible, but resulted in reduced flexibility of data abstraction. Reasonably efficient but ad hoc data manipulation languages (DMLs) were developed. Unfortunately most were largely incompatible with the widely used programming languages. The resulting "impedance mismatch" between the database query language and the general-purpose or host language motivated the research on language integration in database systems.

There are some obvious ways of approaching the integration of host and query language. One approach is to add database features, such as persistence, to a widely used general-purpose language. Pascal/R and PS-Algol represent pioneering efforts in this direction. Another approach is to extend a successful query language toward the closest programming language. For example, the research on databases and logic programming was largely motivated by the potential uses of Prolog+database; after all, logic programming is the computing paradigm closest to relational query languages. From this emerged a number of interesting prototypes (e.g., LDL or NAIL!).

Here we should point out that adding database capabilities to the *appropriate* general-purpose language can bring significant benefits, beyond any improvement of the query/host language interface. If the programming language was designed a priori to support a rich set of data abstractions, modularity, and code reusability, then its persistent version is an excellent candidate for nonstandard database applications, such as OIS and CAD. For example, the potential of rich type systems with semantic features, such as inheritance, was illustrated in various prototype languages (e.g., ADABTBL, Galileo, Taxis, Trellis/Owl).

Beyond persistence, a database version of a language offers schema-management facilities. In this case, the schema is the persistent set of type declarations and comes with mechanisms for concurrency control, recovery, versioning, library management, and more generally for schema evolution. Object-oriented

languages were designed precisely for building and evolving large software systems; they facilitate the development of programming environments and user interfaces. In this light, adding persistence and other database functionalities to Smalltalk in the Gemstone system was a natural, but very important, step in demonstrating the feasibility of a new technology:

object-oriented database systems (OODBs).

At present there is a lot of experimental work under way which has resulted in prototype and even commercial systems claiming the OODB label (or claiming to incorporate major object-oriented programming concepts). There are also many proposed designs and some theoretical analysis. For example, proceeding alphabetically, and fully aware of the everchanging nature of this list, we can mention some of the better-known implemented systems: Cactis, Damokles, Encore/ObServer, Exodus, G-Base, GemStone, Iris, O_2, Ode, Ontos/VBase, Orion, Probe, Postgres, and Vision.

We will not attempt a detailed survey or classification of these systems. A new software technology is typically a creative synthesis of older ideas, tools, and concepts. Its multiple origins make reaching agreement on its precise specification impractical and even damaging to the diversity of the field. Therefore, the goal of this book is not to give a definition of "The OODB" but to clarify (by example) what seem to be the principal OODB components and the design choices made in building them.

The first chapter in this book, entitled "The Object-Oriented Database System Manifesto," was an attempt to outline a commonly accepted part of the OODB specifications. Historically it followed most of the other papers in this book, and it is in large part based on the lessons of building a number of OODBs. Since O_2 is one of these systems, the rest of the book makes concrete the manifesto's many and rather forcefully described golden rules.

The book presents a complete and consistent view of the Altaïr project—a five-year research and development effort to build O_2 that started in September 1986. All aspects of the project are described. For consistency, we focus on the V1 version of the prototype, which was operational in September 1989 and has been distributed to more than 30 sites. The V1 version followed an initial experimental V0 version and preceded the various commercial product versions. As this book goes to press, a commercial release of the industrial version of O_2 is available (since June 1991). Many of the functionalities are similar to the ones described here, although there are some differences.

The most interesting papers related to the O_2 project are presented in what we feel is the most sensible expository sequence. The material in the chapters consists (primarily but not exclusively) of papers which have appeared in the proceedings of internationally recognized computer science conferences. The papers were edited and reformated to make the presentation as uniform as possible.

The book is divided into six parts. Part 1 consists of two papers: the "Manifesto" and "The Story of O_2." It is intended to provide a good overall summary, and we hope it will entice the reader to venture further in the text.

The papers in parts 2 to 5 are related to each other and to the whole through short introductions. The introduction to part 2 contains a complete definition of the O_2 data model; the introduction to part 3 contains a detailed discussion of query/host language integration; and the introductions to parts 4 and 5 contain summaries of the key technical issues related to the system and the programming environment, respectively. The introductions also contain comments on the history of the O_2 contributions, and they close with a roadmap to each part's contents. Part 6 concludes the exposition with descriptions of two applications.

As editors we were faced with the hard task of choosing which parts to emphasize (through the selection of specific papers) from a large project with many contributors. For the data model we focused on the clean synthesis of object-oriented concepts and database complex structures that O_2 offers, on a novel analysis of the power of object identity, and on new ways of controlling schema updates. For the language part we emphasize the multilanguage aspect of O_2—a characteristic that distinguishes it from other efforts in the field. The papers in this part describe the integration with programming languages such as C, Basic, and Lisp, as well as the development of specific query languages. For the system part we have tried to present as many details as possible: on object manager, object clustering, distribution, alternative architectures, version management, and concurrency. The programming environment part is devoted to software engineering tools and the user interface; this reflects the revolutionary impact that high-resolution bitmap workstations have had on computing and the importance for any new database technology of a high-quality interface with the overall programming environment.

In the next chapter we use the golden rules of the object-oriented database manifesto as a way to introduce the material in this collection. If one reverts to the original historical sequence, these rules also provide an accurate set of conclusions. However, let the reader beware. Manifestos invariably reflect the experiences of their authors but also their biases.

We hope that the computer science experiment described in these pages will provide readers with the expertise to follow the final rule of the opening manifesto—"Thou shalt question the golden rules"—and to decide for themselves what object-oriented databases are or should be.

François Bancilhon
Altaïr

Claude Delobel
Université de Paris-Sud

Paris Kanellakis
Brown University

Acknowledgements

The O_2 system is the result of a group effort and this book is the collective work of all those who contributed to the Altaïr[1] project. The editors wish to thank the following people.

The technical staff and the researchers who participated in the design and/or implementation of the system: Gustavo Arango, Gilles Barbedette, Véronique Benzaken, Guy Bernard, Pascale Biriotti, Patrick Borras, Patrice Boursier, Philippe Bridon, René Cazalens, Sophie Cluet, Vineeta Darnis, Christine Delcourt, Anne Doucet, Denis Excoffier, Philippe Futtersack, Sophie Gamerman, Olivier Grémont, Constance Grosselin, Gilbert Harrus, Laurence Haux, John Ioannidis, Mark James, Geneviève Jomier, Jean Marie Larchevêque, Christophe Lécluse, Carol Lepenant, Didier Lévêque, Joëlle Madec, Jacques Madelaine, Jean-Claude Mamou, Jean-Baptiste N'dala, Patrick Pfeffer, Didier Plateau, Bruno Poyet, Michel Raoux, Philippe Richard, Michel Scholl, Dominique Stève, Didier Tallot, Fernando Vélez, and Roberto Zicari.

The students who participated in the project: Laurent Alonzo, Thomas Baudel, Yves Branwschweig, Yveline Cessou, Xavier Crinon, Fabrice Laurence, Sabine Letellier, Vincent Marfaing, Eli Nakdimon, Marc Poinot, and Yves-Henri Saliou.

The consultants who helped at many stages of the design: Michel Adiba, Malcolm Atkinson, Haran Boral, Peter Buneman, George Copeland, Joëlle Coutaz, David DeWitt, Gilles Kahn, Sacha Krakowiak, David Maier, and Marc Shapiro.

The colleagues who agreed to have their papers included in this book: Serge Abiteboul, Michéle Cart, Wojciech Cellary, Klaus Dittrich, Jean Ferrié, Stan Zdonik.

The administrative staff, thanks to whom the group was able to operate efficiently: Eve-Lyne Daneels, Florence Deshors, Hélène Gans, Karine Maillard, Pauline Turcaud, and Carole Viard.

[1]Altaïr is a consortium funded by IN2 (a Siemens subsidiary), INRIA (Institut National de Recherche en Informatique et Automatique), and LRI (Laboratoire de Recherche en Informatique, University of Paris-Sud and CNRS). It is a five-year research and development project started in September 1986. Bull joined the consortium in 1989. Its goal is to design and implement a next-generation database management system. The project is supported by the members of the consortium, by a Eureka convention, BD 11, and by two Esprit projects, FIDE and ITHACA.

The editors would also like to thank the Defense Advanced Research Projects Agency, the National Science Foundation, and the European Community Esprit projects for their support of the task of editing and commenting on (via introductions) the material in this volume.

Finally, detailed credits for the various research activities described in this book are listed under "Acknowledgements" at the end of each chapter.

Contents