

Books

The IRM Imperative: Strategies for Managing Information Resources, James M. Kerr, John Wiley & Sons, Inc., New York, 1991. 282 pp. (ISBN 0-471-52434-4).

The notion or belief that information is an asset that needs rigorous management has been held for over 20 years. However, the idea that information is an influential competitive factor that contributes either to the success or failure of an enterprise is receiving recent attention, partly as a result of articles appearing in the *Harvard Business Review*, the *Sloan Management Review*, and other business-management and strategic-planning publications.

In his book entitled *The IRM Imperative*, James Kerr focuses on the management of information as a corporate asset. He examines technologies, tools, and related management scenarios for their value as competitive advantage differentiators. He even considers the infrastructural changes required by an enterprise that aspires to a more competitive marketing position.

Kerr allows the reader to explore a wide variety of information development and management approaches, technologies, and techniques. Among these are object-oriented programming, prototyping, information engineering, electronic data interchange, joint and rapid application development, CASE technologies, information systems planning, project life cycles, corporate networking, management information centers, decision support, and executive information systems. In addition, this is the first commercially published book to include an entire chapter on IBM's AD/Cycle* framework.

What makes Kerr's book stand out as an information management tutorial is the careful selection of a wide range of currently important infor-

mation systems management topics, presented as a series of introductory discussions. The author provides the reader with 12 independent but related chapters, similar to a well-planned conference proceeding. Each chapter ends with a summary, a list of key points, and a brief list of references. A three-page bibliography in the Appendix supplements these short reference lists.

In the Preface, Kerr correctly suggests that his book provides insight as to where the field of information systems is heading, while keeping business professionals abreast of the technologies and approaches that will become the backbone of application development in the years to come.

Readers familiar with the 1981 book by Synnott and Gruber, *Information Resource Management* (John Wiley & Sons, Inc., 1981), will recognize that several chapters in Kerr's book build upon the foundation these authors provided. Some bibliophiles might even consider Kerr's book the long-awaited second edition of Synnott and Gruber. It is well to keep in mind that 10 years ago, computers and telecommunications were primarily confined to the back office and were managed as you might any overhead activity. In the last 10 years, much has changed. To quote the author, "For the first time since the computer was introduced to the workplace, failure to automate may lead to business failures."

You may recognize the name of the author from his numerous publications in *ComputerWorld*, *Database Programming and Design*, and other trade publications. Kerr is a member of the teaching faculty at the Hartford Graduate Center in Hartford, Connecticut, which offers degrees both

©Copyright 1992 by International Business Machines Corporation.

in its own right and in affiliation with Rensselaer Polytechnic Institute.

Even though originally designed for the professional trade press, Kerr's book is being adopted as a textbook in some industrial and academic programs. In addition to being well received by information professionals in the U.S., the book is being translated into other languages, including Japanese.

Perhaps a supplemental instructor's guide might be considered for the next edition, to help develop education for managers and technicians in this dynamically changing environment of information resources management.

I highly recommend this easy-to-read book—not only to information professionals, but to business management professionals in general. It serves well as a basic building block for anyone interested in developing a successful business information strategy for a competitive enterprise today.

Karen Takle Quinn
IBM Santa Teresa Laboratory
San Jose
California

*Trademark or registered trademark of International Business Machines Corporation.

Introduction to Object-Oriented Databases, Won Kim, The MIT Press, Cambridge, MA, and London, England (1990). 234 pp. (ISBN 0-262-11124-1).

This is a comprehensive, very readable, and timely introduction to topics about object-oriented databases. Typically, a reader who wants to become familiar with the issues related to the design of object-oriented databases will have to refer to research papers, conference proceedings, and the object-oriented chapters in the most recent books on databases. The reader will have to negotiate through differences in terminology and presentation of various authors and discover the common principles of the object-oriented technology. Won Kim's book brings the important topics together using one consistent terminology, one consistent style in presenting object-oriented hierarchies, and few simple examples, which he uses throughout the book. Also, the choice to

base the descriptions and definitions on ORION, an object-oriented database the author codeveloped while working at MCC (Microelectronics and Computer Technology Corporation in Austin, Texas), is a fortunate one. This brings a degree of realism to the conceptual descriptions that designers and implementers of object-oriented databases will appreciate.

The book consists of 16 chapters. Data modeling and the semantics of object-oriented databases are covered in chapters 2 through 8. Design and implementation issues are covered in chapters 9 through 14. Chapter 15 surveys a representative subset of operational object-oriented databases, and the last chapter 16 summarizes topics of current and future research.

Chapter 2 introduces class hierarchies (IS-A relationship) and class composition hierarchies (IS-PART-OF relationship), and documents their orthogonal relationship. An even-handed evaluation of inheritance describes advantages but also states clearly its problems in terms of name conflicts, scoping, and violation of the encapsulation principle. The chapter finishes by introducing composite objects and version control; both are described in more detail in chapter 12. The next chapters introduce the programming interface for object-oriented databases, the taxonomy and rules for schema modification, and an extensive proposal for an object-oriented query language. Authorization, storage management, and transaction management round off the main part of the book.

Two observations about object-oriented technology will become quite evident while reading this book. First, the problem of semantic relativism, which cuts across various data models, does not become any easier with the object-oriented data model. The choice between the two hierarchies of the object-oriented data model and how to interleave them introduces just another set of variables for the database designer. As in other data models, no common rules exist that help to make this design choice. Second, the taxonomy and the rules needed for schema modification clearly show that inheritance violates the principle of encapsulation and consequently requires rules of this kind and complexity. The use of inheritance requires a trade-off between easy maintenance (no or limited inheritance) and reuse (extensive inheritance).

It is now becoming very useful to separate common database technology issues from data modeling issues. Examples of such common issues are storage management, transaction management, and long transaction management, which are problems that can be solved without great concern for the semantics of a particular data model. From a practitioner's point of view, I feel that the line between object-oriented data model semantics and these common database technology issues has not been drawn strongly and consistently enough. The author draws this line quite clearly on page 4 when long transaction management, long data support, and versioning are separated from data model issues like nested object support or support for general data types. But the detailed descriptions of storage structures (chapter 9), and transaction management (chapter 10) make it difficult for the reader to separate common database technology issues from the intrinsic problems that the object-oriented data model introduces to storage management or to transaction management.

Despite these concerns, which express a specific bias of the reviewer, I recommend this book to everyone who deals with the design or the implementation of object-oriented databases.

Franz Spickhoff
IBM Santa Teresa Laboratory
San Jose
California

Note—The books reviewed are those the Editor thinks might be of interest to our readers. The reviews express the opinion of the reviewers.