selected groups of the population such as prisoners, drivers, and drug addicts, databases used by compilers and text editors, and information systems which are accessible by networks.

Acknowledgement

The authors wish to thank an anonymous referee for the helpful suggestions for improving the presentation of the algorithm.

REFERENCES

1. R. Fagin, J. Nievergelt, N. Pippenger and H. R. Strong, Extendible hashing – a fast access method for dynamic files, *ACM Transactions on Database Systems*, **4** (3), 315–344 (1979).
2. P. A. Larson, Dynamic hashing, *BIT*, **18** (2), 184–201 (1978).
3. Witold Litwin, Linear Hashing: a new tool for file and table addressing, *Proc. 6th International Conference on Very Large Database*, Montreal, 212–223 (1980).
4. P. A. Larson, Linear hashing with separators – a dynamic hashing scheme achieving one-access retrieval, *ACM Transactions on Database Systems*, **13** (3), 366–388 (1988).
5. Alan L. Tharp, *File Organization and Processing*, Wiley, New York 398 (1988).
6. Seng Fuat Ou, HSUFSP linear hashing, Internal Report, N.C. State University, Raleigh NC, U.S.A. (1990).
7. P. A. Larson, Personal Communication (1990).
8. Devore Jay L. *Probability and Statistics for Engineering and the Sciences*, Brooks/Cole Publishing Company, Monterey, California (1982).

Book Review

GORDON BLAIR, JOHN GALLAGHER, DAVID HUTCHISON and DOUG SHEPHERD
Object-Oriented Languages, Systems and Applications
Pitman Publishing, London, £22.95
0-273-03132-5

Object-Oriented Languages, Systems and Applications is a collection of chapters by various authors covering a variety of object-oriented topics. It aims to provide comprehensive coverage of the object-oriented paradigm and its applications, rather than concentrating on one application area.

The book is logically divided into four parts. The first five chapters (Part I) are concerned with basic concepts. The first chapter introduces the notion of 'object-orientation' and describes the structure of the book. A 'dependency chart' is provided showing the interrelationships between the chapters, and a guide is provided for readers who may wish to work through the book in stages or from different viewpoints, rather than from beginning to end. After this introductory chapter, there are three chapters on basic concepts. The first of these introduces the well-known object-oriented concepts of encapsulated objects, classes and inheritance. Chapter 3 discusses some variations on the traditional approaches. It introduces a number of techniques for behaviour sharing and evolution, including delegation, actors and photocopying, which are associated with classless systems. Chapter 4 introduces concepts associated with abstract data types (ADTs), and includes a useful section on the differences between types and classes. Chapter 5 then asks the question: 'What *are* Object-Oriented Systems?' A model of object-oriented computing, embracing four dimensions – encapsulation, classification, polymorphism and interpretation – is discussed. This is followed by an introduction to formal approaches to object orientation which seems a little out of place; since the subject is re-introduced in the final chapter, this material might have been better placed there.

The next five chapters (Part II) discuss the application of the object-oriented approach to the following areas respectively: programming languages, database systems, design methods, distributed systems and interactive user interfaces. These serve to illustrate the wide-ranging applicability of the object-oriented paradigm. Although these chapters are structured in different ways (having been written by different authors), they all introduce and compare different systems/languages, after covering general aspects and issues in the relevant application areas. Chapters 11 and 13 (Part III) describe three specific object-oriented products: an object-oriented processor, REKURSIV; an object-oriented language, BETA; and an object-oriented database system, Iris. The final chapter (Part IV) considers future directions for research. The authors stress the importance of embarking on a period of consolidation, obtaining practical experience and establishing consistent terminology and semantics. It is interesting to note that in chapter 5 the model of object-oriented computing which is presented is said to provide a 'more general interpretation [which] widens the scope of object orientation'. Some would argue that the scope should be narrowed, not widened! But perhaps this widening in scope is an essential prerequisite for an effective period of consolidation and subsequent narrowing of concepts and terminology. In this final chapter, specific areas of importance for the future are singled out: software engineering methodologies, formal methods, sharing and distribution, ODP standardization work and multimedia objects.

The book is oriented towards postgraduate computer scientists and computer professionals; with its broad coverage of object-oriented topics, it is certainly important reading for researchers in the field. The authors suggest that it could form the basis of a final-year undergraduate option course, but I feel that some awareness of the widely recognised object-oriented concepts is essential for a full appreciation of the more specialised material and the issues addressed.

A large number of authors have contributed to the book. Although this could have resulted in a very disjointed production, the editors have attempted to ensure that the coverage of particular topics is balanced, and have provided a guide to the structure of the work. However, while there are certain advantages in having different authors for the chapters on application areas and specific products, it might have been more appropriate for the chapters on basic concepts to be written by a single author. These chapters are not as integrated as they might be. It must be said, though, that there is a wealth of material in this book for anyone with a basic knowledge of object-oriented computing wishing to extend that knowledge.

ELIZABETH OXBORROW
Canterbury

Announcement

International Journal of Applied Intelligence

Special issue on
Distributed AI in Manufacturing. Guest editor Mark S. Fox, Carnegie Mellon University.

Papers are solicited for this issue on any topic that demonstrates the use of DAI in any facet of industrial organisation such as design, planning, production, distribution, field service, sales, marketing, finance, etc. The following are examples of topics.

- Concurrent design
- Control of autonomous automated guided vehicles
- Enterprise integration
- Group sales forecasting
- Distributed field service
- Distributed planning and scheduling

Dates to note: deadline for paper submissions March 1992; scheduled for publication 1 July 1992.

Please send your papers to:
Karen S. Cullen,
Kluwer Academic Publishers,
101 Philip Drive,
Norwell, MA 02061, USA