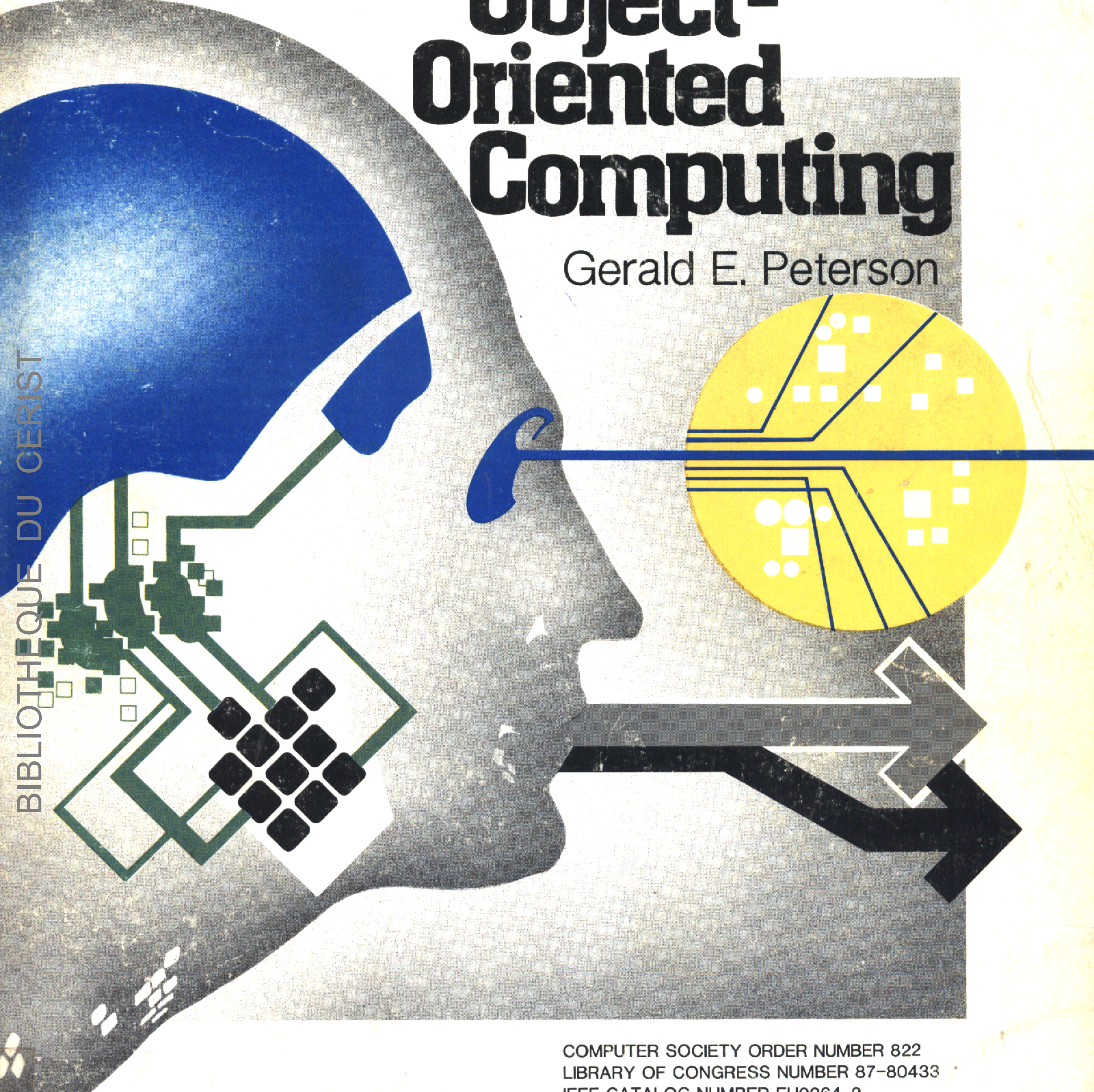
Volume 2: Implementations

TUTORIAL: Object-Oriented Computing

Gerald E. Peterson



BIBLIOTHEQUE DU CERIST

COMPUTER SOCIETY ORDER NUMBER 822
LIBRARY OF CONGRESS NUMBER 87-80433
IEEE CATALOG NUMBER EH0264-2
IEEE 0-8186-0822-6
SAN 264-620X

THE COMPUTER SOCIETY OF THE IEEE

IEEE THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

IEEE COMPUTER SOCIETY PRESS

[illegible]

C
1947

Volume 2: Implementations

TUTORIAL: Object-Oriented Computing

Gerald E. Peterson

COMPUTER SOCIETY ORDER NUMBER 822
LIBRARY OF CONGRESS NUMBER 87-80433
IEEE CATALOG NUMBER EH0264-2
IEEE 0-8186-0822-6
SAN 264-620X

THE COMPUTER SOCIETY OF THE IEEE

IEEE THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

COMPUTER SOCIETY PRESS

BIBLIOTHEQUE DU CERIST

4880

4880

Published by Computer Society Press of the IEEE
1730 Massachusetts Avenue, N.W.
Washington, D.C. 20036-1903

Cover designed by Jack I. Ballestero

Copyright and Reprint Permissions: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of U.S. copyright law for private use of patrons those articles in this volume that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 29 Congress Street, Salem, MA 01970. Instructors are permitted to photocopy isolated articles for noncommercial classroom use without fee. For other copying, reprint or republication permission, write to Director, Publishing Services, IEEE, 345 E. 47th St., New York, NY 10017. All rights reserved. Copyright © 1987 by The Institute of Electrical and Electronics Engineers, Inc.

Computer Society of the IEEE Order Number 822
Library of Congress Number 87-80433
IEEE Catalog Number EH0264-2
ISBN 0-8186-0822-6 (Paper)
ISBN 0-8186-4822-8 (Microfiche)
SAN 264-620X

Order from: Computer Society of the IEEE
Terminal Annex
Post Office Box 4699
Los Angeles, CA 90080

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331

Computer Society of the IEEE
13, Avenue de l'Aquilon
B-1200 Brussels
BELGIUM

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

[illegible]

BIBLIOTHEQUE DU CERIST

BIBLIOTHEQUE DU CERIST

Preface

Object-oriented computing is a style of computing in which data and associated procedures are encapsulated to form an *object*. An object is a useful computing entity existing at a higher level than procedures or data structures.

Object-oriented computing has gained considerable interest in the last few years. This interest is being fanned by the U. S. Department of Defense's push of Ada, a common (partially) object-oriented language.

This tutorial should be of interest to that group of computer professionals who have heard the term "object-oriented," know it is important, and want to obtain a more substantial understanding of the concept. This group includes

- software engineers
- AI professionals
- professors and students of programming language and computer architecture courses
- those who are building or proposing new computer languages or architectures

Most of the papers are at the level of the practicing computer engineer. Some papers, especially those in Chapters 1 and 2 of Volume 1, are accessible to the computer-knowledgeable layman.

All aspects of object-oriented computing are considered, including object-oriented languages, object-oriented design and development, examples of object-oriented programming, object-oriented databases, and object-oriented computer architectures.

Volume 1 presents the basic concepts of object-oriented programming and describes several object-oriented languages.

An overview of the fundamental ideas of object-oriented programming is given in Chapter 1.

Smalltalk is a language based entirely on the use of objects. Everything in the language, including integers, is an object. A great deal of knowledge about object-oriented computing can be obtained by studying Smalltalk. Chapter 2 is devoted exclusively to this language.

Other languages are supportive of the object philosophy. These languages include Ada, in which objects can be created as packages; Modula-2, in which objects are created as modules; and some dialects of Lisp. The manner in which the object-oriented philosophy can be incorported in these languages is considered in Chapters 3 and 4.

Volume 2 is devoted to the manner in which programs are implemented using object-oriented methods, and the manner in which object-oriented languages are themselves implemented.

It has been found that many problems decompose naturally into objects and messages that pass between them. Object-oriented development is the process of decomposing a problem into objects and messages and maintaining these structures in the implementation. Object-oriented development techniques and examples of their use are presented in Chapter 1.

Much insight into the nature of object-oriented programming can be obtained by studying examples where it was successfully put to use. Several examples of object-oriented programming are considered in Chapter 2.

The biggest problem with languages that fully support the object-oriented philosophy is efficient implementation. On a dedicated processor costing in the neighborhood of $100K, Smalltalk runs satisfactorily. On less expensive machines, however, Smalltalk implementations have been too slow to be practical for large programs. Chapter 3 contains many ideas about how to overcome this problem.

Several important concepts and issues that did not fit in other chapters are considered in Chapter 4.

Object-oriented computer architectures have also been addressing, which have guided these implementations. Several examples of object-based architectures are described in Chapter 6. These include the Intel iAPX 432 and the IBM System/38.

Gerald E. Peterson

Acknowledgments

The author wishes to thank the McDonnell Douglas Astronautics and Aerospace Information Services Companies for providing a pleasant climate in which this work could be undertaken. This tutorial resulted from studies which were undertaken in order to improve the manner in which software is developed at McDonnell Douglas Corporation.

BIBLIOTHEQUE DU CERIST

Table of Contents

Preface iii

Acknowledgments iv

Chapter 1: Object-Oriented Design 1

Object-Oriented Development 5
G. Booch (*IEEE Transactions on Software Engineering,* February 1986, pages 211-221)

Towards a General Object-Oriented Software Development Methodology 16
E. Seidewitz and M. Stark (*Proceedings of the First International Conference on Ada Programming Language Applications for the NASA Space Station,* 1986, pages D.4.6.1-D.4.6.14)

Modular Software Construction and Object-Oriented Design Using Ada 30
R.F. Sincover and R.S. Wiener (*Journal of Pascal, Ada, and Modula-2,* March/April 1984, pages 29-34)

Reusability: The Case for Object-Oriented Design 37
B. Meyer (*IEEE Software,* March 1987, pages 50-64)

Object-Oriented Design and Ballistics Software 52
M. Temte (*ACM Ada Letters,* November/December 1984, pages 25-36)

Chapter 2: Examples of Object-Oriented Programming 65

Programming Experiences 67
L. Tesler (*Byte,* August 1986, pages 195-205)

MACAPP: An Application Framework 72
K.J. Schmucker (*Byte,* August 1986, pages 189-193)

Intermedia: The Architecture and Construction of an Object-Oriented Hypermedia System and Applications Framework 76
N. Meyrowitz (*Proceedings of the ACM Conference on Object-Oriented Programming Systems, Languages and Applications,* 1986, pages 186-201)

A Smalltalk System for Algebraic Manipulation 92
S.K. Abdali, G.W. Cherry, and N. Soiffer (*Proceedings of the ACM Conference on Object-Oriented Programming Systems, Languages and Applications,* 1986, pages 277-283)

Objects, Icons, and Software-ICs 99
B. Cox and B. Hunt (*Byte,* August 1986, pages 161–176)

Smalltalk-80 Aids Troubleshooting System Development 109
J.H. Alexander and M.J. Freiling (*Systems & Software,* April 1985, pages 111-118)

Chapter 3: Implementing Object-Oriented Languages 115

The Smalltalk-80 Virtual Machine 117
G. Krasner (*Byte,* August 1981, pages 300–320)

What Price Smalltalk? 125
D. Ungar and D. Patterson (*Computer,* January 1987, pages 67-74)

Designing an Efficient Language 133
C.B. Duff (*Byte,* August 1986, pages 211-224)

BIBLIOTHEQUE DU CERIST

Chapter 4: Concepts and Issues in Object-Oriented Programming 141

Abstraction Techniques in Modern Programming Languages 146
M. Shaw (*IEEE Software*, October 1984, pages 10-26)

The Modular Structure of Complex Systems 162
D.L. Parnas, P.C. Clements, and D.M. Weiss
(*IEEE Transactions on Software Engineering*,
March 1985, pages 259-266)

Integrating Access-Oriented Programming into a Multiparadigm Environment 170
M.J. Stefik, D.G. Bobrow, and K.M. Kahn (*IEEE Software*,
January 1986, pages 10-18)

Object-Oriented Data 180
R.W. Peterson (*AI Expert*, March 1987, pages 26-31)

The Learnability of Object-Oriented Programming Systems 186
T. O'Shea, K. Beck, D. Halbert, and K. Schmucker (*Proceedings of the ACM Conference on Object-Oriented Programming Systems, Languages and Applications*, 1986, pages 502-504)

Using Prototypical Objects to Implement Shared Behavior in Object Oriented Systems 189
H. Lieberman (*Proceedings of the ACM Conference on Object-Oriented Programming Systems, Languages and Applications*, 1986, pages 214-223)

Chapter 5: Principles of Object-Based Architecture 199

Virtual Memory for an Object-Oriented Language 201
T. Kaehler (*Byte*, August 1981, pages 378-387)

The Object Model: A Conceptual Tool for Structuring Software 209
A.K. Jones (*Operating Systems—An Advanced Course*, edited by R. Bayer,
R.M. Graham, and G. Seegmuller, 1978, pages 7-16)

Programming Semantics for Multiprogrammed Computations 216
J.B. Dennis and E.C. Van Horn (*Communications of the ACM*,
March 1966, pages 143-155)

Capability-Based Addressing 229
R.S. Fabry (*Communications of the ACM*, July 1974, pages 403-412)

Object Management on Distributed Systems 239
R.H. Halstead, Jr. (*Proceedings of the 7th Texas Conference on Computing Systems*, 1978, pages 7-7–7-14)

Chapter 6: Examples of Object-Based Architecture 247

An Object-Oriented Command Language 251
R. Snodgrass (*IEEE Transactions on Software Engineering*,
January 1983, pages 1–8)

Design Considerations for the IBM System/38 259
F.G. Soltis and R.L. Hoffman (*Digest of Papers, COMPCON S'79*, 1979, pages 132-137)

iMAX: A Multiprocessor Operating System for an Object-Based Computer 265
K.C. Kahn, W.M. Corwin, T.D. Dennis, H. D'Hooge D.E. Hubka, L.A. Hutchins, J.T. Montague, and F.J. Pollack, and M.R. Gifkins (*Proceedings of the 8th Symposium on Operating Systems Principles*, 1981, pages 127-136)

The iMAX-432 Object Filing System 275
F.J. Pollack, K.C. Kahn, and R.M. Wilkinson (*Proceedings of the 8th Symposium on Operating Systems Principles*, 1981, pages 137-147)

BIBLIOTHEQUE DU CERIST

The Intel® 432 Ada Programming Environment 286
S. Zeigler, N. Allegre, D. Coar, R. Johnson, J. Morris, and G. Burns
(*Digest of Papers, COMPCON Spring '81,* 1981, pages 405-410)

Virtual Memory on a Narrow Machine for an Object-Oriented Language 292
T. Kaehler (*Proceedings of the ACM Conference on Object-Oriented Programming Systems, Languages and Applications,* September 1986, pages 87-106)

Author Biography 313

BIBLIOTHEQUE DU CERIST