

The Handbook of Artificial Intelligence

Volume IV

edited by

Avron Barr

Paul R. Cohen

and

Edward A. Feigenbaum

ADDISON-WESLEY PUBLISHING COMPANY, INC.

Reading, Massachusetts Menlo Park, California New York
Don Mills, Ontario Wokingham, England Amsterdam Bonn
Sydney Singapore Tokyo Madrid San Juan

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial capital letters.

Library of Congress Cataloging-in-Publication Data
(Revised for volume 4)

The Handbook of artificial intelligence.

Includes bibliographies and indexes.

Vol. 4 edited by Avron Barr, Paul R. Cohen, and
Edward A. Feigenbaum.

Vol. has imprint: Reading, Mass.: Addison-Wesley Pub. Co.

I. Artificial intelligence. I. Barr, Avron, 1949-

II. Feigenbaum, Edward A. III. Cohen, Paul R.

Q335.H36

006.3

80-28621

ISBN 0-201-51819-8 (hrd.)

ISBN 0-201-51731-0 (pbk.)

Copyright © 1989 by Addison-Wesley Publishing Company

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America. Published simultaneously in Canada.

Cover design by Copenhaver Cumpston

Text design by Dianne Kanerva

Set in 10-point Century Schoolbook by DEKR Corporation

ABCDEFGHIJ-DO-89

First printing, December, 1989

Acknowledgments

The editors and authors thank the following individuals.

Michael B. Albers
Mohamed Bachiri
David Barstow
Ross Beveridge
Inderpal Bhandari
Harold Brown
Jim Daniell
Clive Dym
Bob Engelmores
Larry Eshelman
Allen Goldberg
Cordell Green
David Hammock
Mehdi Harandi
Neil Iscoe
Yoshiteru Ishida
Peter Jackson
Philip Johnson
Laura Jones
Richard Jullig

Herve Lambert
G. E. Light
Virginia G. Lowry
Mitchell Lubars
Roy Maxion
Ian Miller
Steven Painter
Derek Partridge
Liam Peyton
Thomas Pressburger
Eric Schoen
Jos Schreinemakers
Linda Shapiro
Jay Showalter
Douglas Smith
David Steier
Marilyn Stelzner
Eswaran Subramanian
Bob Woodham
Robert Young

Editors

**Avron Barr
Paul R. Cohen
Edward A. Feigenbaum**

Contributors

**H. Penny Nii
Edmund H. Durfee
Victor R. Lesser
Daniel D. Corkill
Bruce G. Buchanan
Reid G. Smith
James Allen
Michael R. Lowry
Raul Duran
Yumi Iwasaki
Alfred Round
Robert M. Haralick
Alan K. Mackworth
Steven L. Tanimoto**

PREFACE

The Handbook of Artificial Intelligence began as a seminar directed by Edward Feigenbaum at Stanford University in the Spring of 1975. The idea was that if each student wrote eight or ten short articles, this would be sufficient to cover much of the field. Like many early AI projects, the *Handbook* had big goals but was somewhat naive about what it would take to achieve them. Ultimately it took more than one hundred graduate students and researchers, roughly 1500 pages, and seven years to represent most of AI at a level of detail that was accessible but not superficial. And we almost missed the boat: The early 1980s were certainly the last opportunity to survey the whole field with a successful balance of breadth, accessibility, and depth. By then, AI was already growing rapidly, like our ubiquitous search trees, getting wider and deeper as scientists extended the field and specialized areas within it. On the commercial front, AI was making front page news.

If we have any illusions that the *Handbook* project was finished in 1982, they were quickly dispelled. First, the books were remarkably successful, which convinced us that they filled a niche. Second, Addison-Wesley took over the *Handbook* and encouraged us to continue the project. Finally, the books had a surprisingly long life given the rapid progress of AI research. It became clear that by the mid-1980s, people were reading the *Handbook* to get a snapshot of the state of AI circa 1980. This convinced us that future volumes of the *Handbook* should not attempt to revise earlier volumes, but should instead summarize the field as we now understand it, circa 1990.

For this reason, we stress that although several chapters in this volume cover topics discussed in previous volumes, they are not revisions of the previous chapters. Computer vision, for example, was covered at length in Volume III by Takeo Kanade and his colleagues. But vision is a huge area of research and the chapter in this volume, by Robert Haralick, Alan Mackworth and Steven Tanimoto, offers a different perspective. Similarly, new distinctions and perspectives underlie the chapter by Michael Lowry and Raul Duran; in Volume II, the topic was called Automatic Programming, here it is Knowledge-based Software Engineering.

Two other chapters synthesize recent work in new frameworks. In his chapter on Natural Language Understanding, James Allen integrates work on unification grammars, semantic processing, contextual

knowledge, and discourse analysis. Some of the most dramatic changes since the original volumes have happened in the area of Expert Systems. The original volumes organized the discussions of expert systems not in terms of architectures, or knowledge representations, or any other underlying principles, but in terms of three applications areas: science, medicine, and education. The new chapter, by Bruce Buchanan and Reid Smith, offers a framework of fundamental principles and issues, within which specific expert systems serve as illustrations.

This volume also includes four chapters on topics that were nascent or nonexistent when we published the previous volumes. Yumi Iwasaki has written a broad, analytical survey of work in Qualitative Physics, an area that is becoming increasingly important as AI systems begin to interact with physical devices in real environments. Alfred Round's chapter on Simulation has similar concerns but a different orientation. It describes efforts to integrate numerical simulation and AI techniques. This work is representative of many research ventures that are forging new technologies at the intersections of AI and other fields.

Edmund Durfee, Victor Lesser, and Daniel Corkill discuss the problem of Cooperative, Distributed AI systems. Researchers in this challenging new area are trying to develop coordinated collections of spatially and functionally distributed agents that work simultaneously on problems that are too large for individual systems to solve efficiently. H. Penny Nii's chapter on Blackboard Systems should perhaps be on everyone's list of basic readings, so pervasive are blackboard architectures in AI today (e.g., most of the systems described in Durfee, Lesser, and Corkill's chapter are based on blackboards).

In the early days of the *Handbook* project, individual articles were contributed by graduate students and assembled into chapters by the editors. Often the editors rewrote the articles, making us de facto authors. By the time the third volume was being compiled, however, we had stopped soliciting individual articles. Nearly all the chapters in Volume III were produced by one or two experts. This gave us longer, more integrated chapters, although the basic form continued to be one or more overview articles followed by discussions of individual systems.

In Volume IV, we went one step further and commissioned signed chapters by some of the best people in the field. We asked them to describe the state of the art, the significant developments and the open questions in their areas. We bid authors to strive for breadth (albeit within one area of AI) and depth, although we secretly believed they would have to sacrifice breadth for depth. Remarkably, this didn't happen. Instead, many sacrificed the "Handbook format" of the first three volumes. To see why this is significant, recall the structure of previous *Handbook* chapters: an introductory section was often followed by an "issue" section, and then by several sections—the bulk of every chapter—describing

systems. In this volume, in contrast, the bulk of every chapter is devoted to issues, principles, and theory. Although systems still figure prominently, much more text is given to their analysis. It reflects a significant maturation of AI that this volume of the *Handbook* is less concerned with the systems we build than with what we have learned and have still to learn by building them.

Avron Barr
Paul Cohen
Ed Feigenbaum

CONTENTS OF VOLUME IV

Preface / xiii

XVI: Blackboard Systems / 1

H. Penny Nii

A. Overview / 3

B. Blackboard Model of Problem Solving / 4

1. The Blackboard Model / 4

2. The Blackboard Framework / 11

3. Perspectives / 16

4. Summary / 16

C. Evolution of Blackboard Architectures / 18

1. Prehistory / 18

2. The HEARSAY Project / 20

3. The HASP Project / 24

D. Blackboard Application Systems / 27

1. HEARSAY-II / 27

2. HASP/SIAP / 36

3. TRICERO / 50

4. PROTEAN / 56

5. Summary / 65

E. Summary: Elements of Blackboard Architecture / 67

1. Blackboard Systems and Task Characteristics / 68

2. "Problem Solving" Revisited: Search vs. Recognition / 70

- 3. Component Design / 74

XVII: Cooperative Distributed Problem Solving / 83

Edmund H. Durfee, Victor R. Lesser, Daniel D. Corkill

A. Overview / 85

B. An Example of CDPS / 95

C. Important CDPS Approaches and Empirical Investigations / 106

1. Negotiation / 107

2. Functionally Accurate Cooperation / 116

3. Organizational Structuring / 122

4. Multiagent Planning / 134

5. Sophisticated Local Control / 137	
6. Formal Frameworks / 143	
D. Conclusion / 146	
XVIII: Fundamentals of Expert Systems / 149	
Bruce G. Buchanan, Reid G. Smith	
A. Overview / 151	
B. Fundamental Principles / 160	
1. Representation of Knowledge / 161	
2. Reasoning Methods / 167	
3. Knowledge Base Development / 173	
4. Explanation / 174	
5. System-building Tools/Shells / 175	
6. Validation / 177	
7. Reasons for Using the Methods of Expert Systems / 178	
C. State of the Art / 181	
1. Size of System / 181	
2. Type of System / 183	
3. Some Observed Limitations / 184	
D. Design Principles and Summary / 189	
1. Design Principles / 189	
2. Summary / 191	
XIX: Natural Language Understanding / 193	
James Allen	
A. Overview / 195	
B. Unification Grammars / 198	
C. Semantic Interpretation / 206	
D. Semantic Interpretation Strategies / 213	
E. Modeling Context / 223	
F. Discourse Structure / 233	
G. Conclusion / 238	
XX: Knowledge-based Software Engineering / 241	
Michael Lowry, Raul Duran	
A. Overview / 243	
B. Specification Acquisition / 253	
1. Knowledge-based Specification Acquisition / 253	
2. Specification Languages / 255	
3. Specification Acquisition Methodologies / 258	
4. Specification Validation / 261	
5. Specification Maintenance / 263	
6. Recovering Specifications from Code / 265	

- C. Program Synthesis / 268
 - 1. Historical Perspective / 269
 - 2. Transformational Approach / 272
 - 3. Deductive Approach / 274
 - 4. Basic Rules / 275
 - 5. Large-grained Rules / 281
 - 6. Reusing Derivations / 282
 - 7. Basic Search Techniques / 284
 - 8. Knowledge-intensive Search Techniques / 286
 - D. Systems for Specification Acquisition / 292
 - 1. IDeA / 292
 - 2. Explainable Expert Systems / 294
 - 3. DRACO / 295
 - 4. The Requirements Apprentice / 296
 - 5. KATE / 298
 - 6. Ozym / 299
 - 7. Watson / 301
 - E. Program Synthesis Systems / 303
 - 1. CIP / 303
 - 2. Designer / 304
 - 3. KIDS / 307
 - 4. MEDUSA / 310
 - 5. KBEmacs / 311
 - 6. REFINE / 313
 - 7. SETL / 315
 - 8. STRATA / 316
 - 9. ELF / 318
 - 10. PHiNix / 320
 - F. Further Readings / 322
- XXI: Qualitative Physics / 323
- Yumi Iwasaki
- A. Overview / 325
 - B. Qualitative Calculus / 339
 - C. Reasoning About Behavior Using Qualitative Calculus / 350
 - 1. Qualitative Behavior and Qualitative States / 350
 - 2. State Transitions / 351
 - 3. Difficulties in Qualitative Prediction / 357
 - D. ENVISION / 362
 - 1. Device Model / 362
 - 2. Predicting Behavior / 366
 - 3. Conclusion / 368
 - E. Qualitative Process Theory / 371
 - 1. Representation of Objects / 372

2. Process Representation /	373
3. Predicting Behavior /	376
4. Conclusion /	380
F. QSIM /	382
G. Causal Ordering /	392
H. Causal Action/Event-based Approaches /	403
1. Commonsense Algorithm /	403
2. Functional Representation of Devices /	406
3. Consolidation /	409
4. Conclusion /	411
XXII: Knowledge-based Simulation /	415
Alfred Round	
A. Overview /	417
B. The Evolution of Knowledge-based Simulation /	419
1. An Overview of Simulation /	419
2. The Limitations of Numerical Simulation /	424
3. Object-oriented Language for Simulation /	425
C. Applications of Knowledge-based Simulation /	437
1. The Design of Flexible Manufacturing Systems /	437
2. Planning Therapies for Cancer Treatment /	441
3. Evaluating Business Proposals /	444
4. Solving Problems in Molecular Genetics /	448
D. The Design of Knowledge-based Simulation Systems /	452
1. Sequential Integrated Systems /	452
2. Parallel Integrated Systems /	455
3. Intelligent Front Ends for Building Numerical Simulation /	459
4. Rule-driven Simulation /	462
E. Qualitative Aspects of Knowledge-based Simulation /	464
1. Simplification of Processes /	464
2. Aggregation of Processes /	467
3. Multiple Levels of Abstraction /	470
4. Multiple Levels of Precision /	476
F. Real-World Applications of Knowledge-based Simulation /	483
1. COMAX: Knowledge-based Simulation for Cotton Crop Management /	483
2. SimKit: An Integrated, Knowledge-based Environment for Simulation /	487
3. ABLE: Knowledge-based Control for Particle Accelerators /	493
4. Forecast Pro: Intelligent Prediction of Business Trends /	500
G. Issues in the Development and Use of Knowledge-based Simulation /	504
1. Simulation, Inferencing, and Time /	504
2. The Development of Knowledge-based Simulation Applications /	510

**3. The Validation of Knowledge-based Simulation
Applications / 513**

H. Conclusion / 518

XXIII: Computer Vision Update / 519

Robert M. Haralick, Alan K. Mackworth, Steven L. Tanimoto

A. Overview / 521

B. Low-level Vision / 523

1. Segmentation Techniques / 523

2. Edges / 534

3. Stereo / 536

4. Mathematical Morphology for Image Analysis / 539

C. Computational Vision Advances / 547

1. Shape Representation and Analysis / 547

2. Criteria for Shape Representation / 547

3. Object Recognition / 558

4. Constraint Satisfaction / 560

D. Vision Architecture / 565

Bibliography / 583

Name Index / 623

Subject Index / 643

Chapter XVI

Blackboard Systems

H. Penny Nii—Stanford University

CHAPTER XVI: BLACKBOARD SYSTEMS

- A. *Overview* / 3
- B. *Blackboard Model of Problem Solving* / 4
 - 1. *The Blackboard Model* / 4
 - 2. *The Blackboard Framework* / 11
 - 3. *Perspectives* / 16
 - 4. *Summary* / 16
- C. *Evolution of Blackboard Architectures* / 18
 - 1. *Prehistory* / 18
 - 2. *The HEARSAY Project* / 20
 - 3. *The HASP Project* / 24
- D. *Blackboard Application Systems* / 27
 - 1. *HEARSAY-II* / 27
 - 2. *HASP/SIAP* / 36
 - 3. *TRICERO* / 50
 - 4. *PROTEAN* / 56
 - 5. *Summary* / 65
- E. *Summary: Elements of Blackboard Architecture* / 67
 - 1. *Blackboard Systems and Task Characteristics* / 68
 - 2. *"Problem Solving" Revisited: Search vs. Recognition* / 70
 - 3. *Component Design* / 74

A. OVERVIEW

HISTORICALLY THE BLACKBOARD MODEL arose from abstracting features of the HEARSAY-II (Erman et al., 1980) speech-understanding system developed between 1971 and 1976. HEARSAY-II was able to respond to spoken commands and queries about computer science abstracts stored in a database. From an informal summary description of the HEARSAY-II program, the HASP system was designed and implemented between 1973 and 1975. The domain of HASP was ocean surveillance, and its task was the interpretation of continuous passive sonar data. (*Domain* refers to a particular area of discourse, for example, chemistry. *Task* refers to a goal-oriented activity within the domain, for example, to analyze the molecular composition of a compound.) HASP, as the second example of a blackboard system, not only added credibility to the claim that a blackboard approach to problem solving was general, but it also demonstrated that it could be abstracted into a robust model of problem solving. Subsequently many application programs have been implemented whose solutions were formulated using the blackboard model. Because the characteristics of the application problems differed and the interpretation of the blackboard model varied, the design of these programs differed considerably. However, the blackboard model of problem solving has not undergone any substantial changes in the last fifteen years.

B. BLACKBOARD MODEL OF PROBLEM SOLVING

A *problem-solving model* is a scheme for organizing reasoning steps and domain knowledge to construct a solution to a problem. *Reasoning* refers to a computational process whereby needed information is inferred from what is already known.

For example, in a *backward-reasoning model*, problem solving begins by reasoning backward from a goal to be achieved toward an initial state (data). More specifically, in a *rule-based backward-reasoning model*, knowledge is organized as if-then rules and modus ponens inference steps are applied to the rules from a goal rule back to an "initial-state rule" (a rule whose premise clause matches input data; see Article III.C4, in Vol. I). An excellent example of this approach to problem solving is the MYCIN program (Shortliffe, 1976; see Article VIII.B1, in Vol. II). In a *forward-reasoning model*, however, the inference steps are applied from an initial state toward a goal. The OPS system exemplifies such a reasoning model (Forgy and McDermott, 1977). In an *opportunistic reasoning model*, pieces of knowledge are applied either backward or forward at the most "opportune" time (see Article XI.C, Vol. III).

Put another way, the central issue of problem solving is the question: "What pieces of knowledge should be applied when and how?" A problem-solving model provides a conceptual framework for organizing knowledge and a strategy for applying that knowledge.

B1. The Blackboard Model

THE BLACKBOARD MODEL of problem solving is a highly structured, special case of opportunistic problem solving. In addition to opportunistic reasoning as a knowledge-application strategy, the blackboard model prescribes the organization of the domain knowledge and all the input and intermediate and partial solutions needed to solve the problem. We refer to all possible partial and full solutions to a problem as its *solution space* (see Article II.A, Vol. I).

In the blackboard model, the solution space is organized into one or