The Soar Papers

Artificial Intelligence

Patrick Henry Winston, founding editor

J. Michael Brady, Daniel G. Bobrow, and Randall Davis, current editors

Artificial intelligence is the study of intelligence using the ideas and methods of computation. Unfortunately, a definition of intelligence seems impossible at the moment because intelligence appears to be an amalgam of so many information-processing and information-representation abilities.

Of course psychology, philosophy, linguistics, and related disciplines offer various perspectives and methodologies for studying intelligence. For the most part, however, the theories proposed in these fields are too incomplete and too vaguely stated to be realized in computational terms. Something more is needed, even though valuable ideas, relationships, and constraints can be gleaned from traditional studies of what are, after all, impressive existence proofs that intelligence is in fact possible.

Artificial intelligence offers a new perspective and a new methodology. Its central goal is to make computers intelligent, both to make them more useful and to understand the principles that make intelligence possible. That intelligent computers will be extremely useful is obvious. The more profound point is that artificial intelligence aims to understand intelligence using the ideas and methods of computation, thus offering a radically new and different basis for theory formation. Most of the people doing work in artificial intelligence believe that these theories will apply to any intelligent information processor, whether biological or solid state.

There are side effects that deserve attention, too. Any program that will successfully model even a small part of intelligence will be inherently massive and complex. Consequently, artificial intelligence continually confronts the limits of computer-science technology. The problems encountered have been hard enough and interesting enough to seduce artificial intelligence people into working on them with enthusiasm. It is natural, then, that there has been a steady flow of ideas from artificial intelligence to computer science, and the flow shows no sign of abating.

The purpose of this series in artificial intelligence is to provide people in many areas, both professionals and students, with timely, detailed information about what is happening on the frontiers in research centers all over the world.

J. Michael Brady
Daniel Bobrow
Randall Davis

The Soar Papers

Research on Integrated Intelligence

Volume One

Edited by

Paul S. Rosenbloom, John E. Laird, and Allen Newell

THE MIT PRESS
CAMBRIDGE, MASSACHUSETTS
LONDON, ENGLAND

Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

The Soar papers : research on integrated intelligence / edited by Paul S. Rosenbloom, John E. Laird, and Allen Newell.
p. cm.
Includes bibliographical references and index.
ISBN 0-262-18152-5 (hc)—ISBN 0-262-68071-8 (pbk)
1. Artificial intelligence. I. Rosenbloom, Paul S. II. Laird, John, 1954- . III. Newell, Allen.
Q335.S63 1992
006.3--dc20 91-48463
CIP

This book is dedicated to:

the memory of Allen Newell, who inspired so much;

the Soar community, who contributed so much;

Elaine, Ann, and Noel, who tolerated and supported so much.

Contents

Volume One: 1969-1988

1986

1987

1988

Volume Two: 1989-1991

1989

1990

Acknowledgments

The editors and the publishers are grateful to the following:

***Academic Press,* for**

Rosenbloom, P. S., Laird, J. E., and Newell, A., "The Chunking of Skill and Knowledge." Reprinted from *Working Models of Human Perception*, edited by B. A. G. Elsendoorn and H. Bouma, pp. 391-410. Copyright © 1989, Academic Press, London. Reprinted with permission.

Bösser, T., "A Discussion of 'The Chunking of Skill and Knowledge' by Paul S. Rosenbloom, John E. Laird and Allen Newell." Reprinted from *Working Models of Human Perception*, edited by B. A. G. Elsendoorn and H. Bouma, pp. 411-418. Copyright © 1989, Academic Press, London. Reprinted with permission.

The American Association for Artificial Intelligence for

Rosenbloom, P. S. and Newell, A. "Learning by Chunking: Summary of a Task and a Model'. Reprinted from *Proceedings of the National Conference on Artificial Intelligence*, pp. 255-257. Copyright © 1982, American Association for Artificial Intelligence, Menlo Park, California. Reprinted with permission.

Laird, J. E., Rosenbloom, P. S., and Newell, A., "Towards Chunking as a General Learning Mechanism." Reprinted from *Proceedings of the National Conference on Artificial Intelligence*, Austin, Texas, pp. 188-192. Copyright © 1984, Association for Artificial Intelligence, Menlo Park, California. Reprinted with permission.

Laird, J. E., Rosenbloom, P. S., and Newell, A., "Overgeneralization During Knowledge Compilation in Soar." Reprinted from *Proceedings of the Workshop on Knowledge Compilation*, edited by T. G. Dietterich. Copyright © 1986, Association for Artificial Intelligence, Menlo Park, California. Reprinted with permission.

Rosenbloom, P. S., and Laird, J. E., "Mapping Explanation-Based Generalization onto Soar." Reprinted from *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, Pennsylvania, pp. 561-567. Copyright © 1986, Association for Artificial Intelligence, Menlo Park, California. Reprinted with permission.

Rosenbloom, P. S., Laird, J. E., and Newell, A., "Knowledge Level Learning in Soar." Reprinted from *Proceedings of the Sixth National Conference on Artificial Intelligence*, Seattle, Washington, pp. 499-504. Copyright © 1987, Association for Artificial Intelligence, Menlo Park, California. Reprinted with permission.

Flynn, R., "Placing Soar on the Connection Machine." Reprinted from *How Can Slow Components Think So Fast? AAAI 1988 Mini-Symposium*. Copyright © 1988, American Association for Artificial Intelligence, Menlo Park, California. Reprinted with permission.

Laird, J. E., "Recovery from Incorrect Knowledge in Soar." Reprinted from *Proceedings of the Seventh National Conference on Artificial Intelligence*, St. Paul, Minnesota, pp. 618-623. Copyright © 1988, Association for Artificial Intelligence, Menlo Park, California. Reprinted with permission.

Nayak, P., and Gupta, A., and Rosenbloom, P. S., "Comparison of the Rete and Treat Production Matchers for Soar. Reprinted from *Proceedings of the Seventh National Conference on Artificial Intelligence*, St. Paul, Minnesota, pp. 693-698. Copyright © 1988, Association for Artificial Intelligence, Menlo Park, California. Reprinted with permission.

Steir, D. M. and Newell, A., "Integrating Multiple Sources of Knowledge into Designer-Soar: An Automatic Algorithm Designer. Reprinted from *Proceedings of the Seventh National Conference on Artificial Intelligence*, St. Paul, Minnesota, pp. 8-13. Copyright © 1988, Association for Artificial Intelligence, Menlo Park, California. Reprinted with permission.

Rosenbloom, P. S., "Beyond Generalization as Search: Towards a Unified Framework for the Acquisition of New Knowledge, pp. 17-21. Reprinted from *Proceedings of the AAAI Symposium on Explanation-Based Learning*, edited by G. F. DeJong. Copyright © 1988, American Association for Artificial Intelligence, Menlo Park, California. Reprinted with permission.

Etzioni, O., and Mitchell, T. M., "A Comparative Analysis of Chunking and Decision-Analytic Control." Reprinted from the *Proceedings of the 1989 AAAI Spring Symposium on AI and Limited Rationality*, Stanford, California. Copyright © 1989, American Association for Artificial Intelligence, Menlo Park, California. Reprinted with permission.

Laird, J. E. and Rosenbloom, P. S., "Integrating Execution, Planning, and Learning in Soar for External Environments." Reprinted from *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston, Massachusetts, pp. 1022-1029. Copyright © 1990, Association for Artificial Intelligence, Menlo Park, California. Reprinted with permission.

Rosenbloom, P. S. and Aasman, J., "Knowledge Level and Inductive Uses of Chunking (EBL)." Reprinted from *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston, Massachusetts, pp. 821-827. Copyright © 1990, Association for Artificial Intelligence, Menlo Park, California. Reprinted with permission.

Tambe, M. and Rosenbloom, P. S., "A Framework for Investigating Production System Formulations with Polynomially Bounded Match." Reprinted from *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston, Massachusetts, pp. 693-700. Copyright © 1990, Association for Artificial Intelligence, Menlo Park, California. Reprinted with permission.

Unruh, A., and Rosenbloom, P. S., "Two New Weak Method Increments for Abstraction." Reprinted from *Working Notes of the AAAI-90 Workshop on Automatic Generation of Approximations an d Abstractions*, edited by T. Ellman, pp. 78-86. Copyright © 1990, American Association for Artificial Intelligence, Menlo Park, California. Reprinted with permission.

***The Association for Computing Machinery*, for**
Tambe, M.; Kalp, D.; Gupta, A.; Forgy, C. L.; Milnes, B. G.; and Newell, A., "Soar/PSM-E: Investigating Match Parallelism in a Learning Production System." Reprinted from *Proceedings of the ACM/SIGPLAN Symposium on Parallel Programming: Experience with Applications, Languages, and Systems*, July , pp. 146-160. Copyright © 1988, Association for Computing Machinery, New York, New York. Reprinted with permission.

Howes, A. and Young, R. M., "Predicting the Learnability of Task-Action Mappings." Reprinted from *Proceedings of CHI '91 Human Factors in Computing Systems*, ACM Press, New Orleans. Copyright © 1991, Association for Computing Machinery, New York, New York. Reprinted with permission.

Newell, A., Yost, G. R., Laird, J. E., Rosenbloom, P. S., & Altmann, E., "Formulating the Problem Space Computational Model." Reprinted from *Carnegie Mellon Computer Science: A 25 Year Commemorative*, edited by R. F. Rashid, pp. 255-293. Reading, Massachusetts: Addison-Wesley. Copyright © 1991, Association for Computing Machinery, New York, New York. Reprinted with permission.

Young, R. M. and Whittington, J., "Using a Knowledge Analysis to Predict Conceptual Errors in Text-Editor Usage." Reprinted from *Proceedings of CHI '90*, April 1990, pp. 91-97. Copyright © 1990, Association for Computing Machinery, New York, New York. Reprinted with permission.

The Cognitive Science Society and Lawrence Erlbaum Associates for
Lewis, R. L., Huffman, S. B., John, B. E., Laird, J. E., Lehman, J. F., Newell, A., Rosenbloom, P. S., Simon, T., and Tessler, S. G., "Soar as a Unified Theory of Cognition: Spring 1990." Reprinted from *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, Cambridge, Massachusetts, pp. 1035-1042. Copyright © 1990, Cognitive Science Society Incorporated, Pittsburgh, Pennsylvania. Reprinted with permission.

Wiesmeyer, M. and Laird, J., "A Computer Model of 2D Visual Attention and Search." Reprinted from *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, Cambridge, Massachusetts. Copyright © 1990,

Cognitive Science Society Incorporated, Pittsburgh, Pennsylvania. Reprinted with permission.

Cho, B., Rosenbloom, P. S., and Dolan, C. P., "Neuro-Soar: A Neural-Network Architecture for Goal-Oriented Behavior." Reprinted from the *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pp. 673-677. Copyright © 1991, Cognitive Science Society Incorporated, Pittsburgh, Pennsylvania. Reprinted with permission.

Lehman, J. F., Lewis, R. L., and Newell, A., "Integrating Knowledge Sources in Language Comprehension." Reprinted from the *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*. Copyright © 1991, Cognitive Science Society Incorporated, Pittsburgh, Pennsylvania. Reprinted with permission.

Miller, C. S., and Laird, J. E., "A Constraint-Motivated Lexical Acquisition Model." Reprinted from the *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*. Copyright © 1991, Cognitive Science Society Incorporated, Pittsburgh, Pennsylvania. Reprinted with permission.

Polk, T. A. and Newell, A., "Modeling Human Syllogistic Reasoning in Soar." Reprinted from the *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*, Montreal, pp. 181-187. Copyright © 1988, Cognitive Science Society Incorporated, Pittsburgh, Pennsylvania. Reprinted with permission.

Lewis, R. L., Newell, A., and Polk, T. A., "Toward a Soar Theory of Taking Instructions for Immediate Reasoning Tasks." Reprinted from *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, pp. 514-512. Copyright © 1989, Cognitive Science Society, Inc., Pittsburgh, Pennsylvania. Reprinted with permission.

Polk, T. A., Newell, A., and Lewis, R. L., "Toward a Unified Theory of Immediate Reasoning in Soar." Reprinted from *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, pp. 506-513. Copyright © 1989, Cognitive Science Society, Inc., Pittsburgh, Pennsylvania. Reprinted with permission.

Ruiz, D. and Newell, A., "Tower-Noticing Triggers Strategy-Change in the Tower of Hanoi: A Soar Model." Reprinted from *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, pp. 522-529. Copyright © 1989, Cognitive Science Society, Inc., Pittsburgh, Pennsylvania. Reprinted with permission.

Elsevier Science Publishers for
Newell, A., "The Knowledge Level." Reprinted from *Artificial Intelligence*, Volume 18, pp. 87-127. Copyright © 1982, Elsevier Science Publishers B.V., Amsterdam, The Netherlands. Reprinted with permission.

Laird, J. E., Newell, A., & Rosenbloom, P. S., "Soar: An Architecture for General Intelligence." Reprinted from *Artificial Intelligence*, Volume 33, pp. 1-64. Copyright © 1987, Elsevier Science Publishers B.V., Amsterdam, The Netherlands. Reprinted with permission.

Norman, D. A., "Approaches to the Study of Intelligence," Reprinted from *Artificial Intelligence*, Vol. 47, pp. 327-346. Copyright © 1991, Elsevier Science Publishers B.V., Amsterdam, The Netherlands. Reprinted with permission.

Rosenbloom, P. S., Laird, J. E., Newell, A., and McCarl, R., "A Preliminary Analysis of the Soar Architecture as a Basis for General Intelligence." Reprinted from *Artificial Intelligence*, Volume 47, pp. 289-325. Copyright © 1991, Elsevier Science Publishers B.V., Amsterdam, The Netherlands. Reprinted with permission.

Rosenbloom, P. S., "A Symbolic Goal-Oriented Perspective on Connectionism and Soar." Reprinted from *Connectionism in Perspective*, edited by R. Pfeifer, Z. Schreter, F. Fogelman-Soulie, and L. Steels, pp. 245-263.Copyright © 1989, Elsevier Science Publishers B.V., Amsterdam, The Netherlands. Reprinted with permission.

Rosenbloom, P. S., Laird, J. E., and Newell, A., "Meta-Levels in Soar." Reprinted from *Meta-Level Architectures and Reflection*, edited by P. Maes and D. Nardi, pp. 227-240. Copyright © 1988, Elsevier Science Publishers B.V., Amsterdam, The Netherlands. Reprinted with permission.

Todd R. Johnson for
Johnson, T. R., Smith, J. W., and Chandrasekaran, B., "Task-Specific Architectures for Flexible Systems." Reprinted with permission.

The Institute of Electrical and Electronics Engineers, Inc. for
Rosenbloom, P. S., Laird, J. E., McDermott, J., Newell, A. and Orciuch, E., "R1-Soar: An Experiment in Knowledge-Intensive Programming in a Problem-Solving Architecture." Reprinted from *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 7, pp. 561-569. Copyright © 1985, The Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey. Reprinted with permission.

Steier, D.M., "Intelligent Architectures for Integration." Reprinted from *Proceedings of the IEEE Conference on Systems Integration*, August 1990. (A shorter version of this paper appeared in the proceedings.) Copyright © 1990, The Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey. Reprinted with permission.

International Joint Conferences on Artificial Intelligence, Inc. for
Golding, A. R., Rosenbloom, P. S., and Laird, J. E., "Learning General Search Control from Outside Guidance." Reprinted from *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pp. 334-337. Copyright © 1987, International Joint Conferences on Artificial Intelligence, Inc. Copies of this and other IJCAI proceedings are available from Morgan Kaufmann Publishers, Inc., 2929 Campus Drive, San Mateo, California 94403. Reprinted with permission.

Steier, D., "CYPRESS-Soar: A Case Study in Search and Learning in Algorithm Design." Reprinted from *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pp. 327-330. Copyright © 1987, International Joint Conferences on Artificial Intelligence, Inc. Copies of this and other IJCAI proceedings are available from Morgan Kaufmann Publishers, Inc., 2929 Campus Drive, San Mateo, California 94403. Reprinted with permission.

Unruh, A. and Rosenbloom, P. S., "Abstraction in Problem Solving and Learning." Reprinted from *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 681-687. Copyright © 1989, International Joint Conferences on Artificial Intelligence, Inc. Copies of this and other IJCAI proceedings are available from Morgan Kaufmann Publishers, Inc., 2929 Campus Drive, San Mateo, California 94403. Reprinted with permission.

Yost, G. R. & Newell, A., "A Problem Space Approach to Expert System Specification," *Reprinted from Proceedings of Eleventh International Joint Conference on Artificial Intelligence*, pp. 621-627. Copyright © 1989, International Joint Conferences on Artificial Intelligence, Inc. Copies of this and other IJCAI proceedings are available from Morgan Kaufmann Publishers, Inc., 2929 Campus Drive, San Mateo, California 94403. Reprinted with permission.

Bonnie E. John, Carnegie Mellon University, and Nintendo of America, Inc. for
John, B. E., Vera, A. H., and Newell, A., "Towards Real-Time GOMS," Abbreviated version of "Towards Real-Time GOMS," by John, B. E., Vera, A. H., and Newell, A.. Reprinted from Carnegie Mellon University School of Computer Science Technical Report CMU-CS-90-195. The illustration of the first screen from Super Mario Brothers 3® where Mario starts World 1, Level 1, and the photographs from pages 6 and 11 from the Super Mario Brothers 3® Instruction Booklet are copyright © 1991, Nintendo. Used with permission of Nintendo of America Inc.

Kluwer Academic Publishers for
Laird, J. E., Rosenbloom, P. S., and Newell, A., "Chunking in Soar: The Anatomy of a General Learning Mechanism," Reprinted from *Machine Learning*, Volume 1, pp. 11-46. Copyright © 1986, Kluwer Academic Publishers, Norwell, Massachusetts. Reprinted with permission.

Tambe, M., Newell, A., and Rosenbloom, P. S., "The Problem of Expensive Chunks and Its Solution by Restricting Expressiveness." Reprinted from *Machine Learning*, Volume 5, pp. 299-348. Copyright © 1990, Kluwer Academic Publishers, Norwell, Massachusetts. Reprinted with permission.

Lawrence Erlbaum Associates, Inc. for
Newell, A., "Reasoning, Problem Solving and Decision Processes: The Problem Space as a Fundamental Category." Reprinted from *Attention and Performance VIII*, edited by R. Nickerson. Copyright © 1980, Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey. Reprinted with permission.

Newell, A. and Rosenbloom, P. S., "Mechanisms of Skill Acquisition and the Law of Practice." Reprinted from *Cognitive Skills and their Acquisition*, edited by J. R. Anderson. Copyright © 1981, Lawrence Erlbaum Associates, Inc., Hillsdale, N.new Jersey. Reprinted with permission.

Rosenbloom, P. S., Newell, A., and Laird, J. E., "Towards the Knowledge Level in Soar: The Role of the Architecture in the Use of Knowledge." Reprinted from *Architectures for Intelligence*, edited by K. VanLehn, pp. 75-111. Copyright © 1991, Lawrence Erlbaum Associates, Inc., Hillsdale, N.new Jersey. Reprinted with permission.

Ajay K. Modi for
Modi, A. K. and Westerberg, A.W., "Integrating Learning and Problem Solving within a Chemical Process Designer." Presented at the Annual Meeting of the American Institute of Chemical Engineers. Reprinted with permission.

Morgan Kaufmann Publishers for
Rosenbloom, P. S. and Newell, A., "The Chunking of Goal Hierarchies: A Generalized Model of Practice." Reprinted from *Machine Learning: An Artificial Intelligence Approach, Volume II*, edited by R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, pp. 247-288. Copyright © 1986, Morgan Kaufmann Publishers, Inc., Los Altos, California. Reprinted with permission.

Steier, D. M., Laird, J. E., Newell, A., Rosenbloom, P. S., Flynn, R., Golding, A., Polk, T. A., Shivers, O. G., Unruh, A., and Yost, G. R., "Varieties of Learning in Soar: 1987." Reprinted from *Proceedings of the Fourth International Workshop on Machine Learning*, edited by P. Langley, pp. 300-311. Copyright © 1987, Morgan Kaufmann Publishers, Inc., Los Altos, California. Reprinted with permission.

Laird, J. E., Hucka, M., Yager, E.S., and Tuck, C.M., "Correcting and Extending Domain Knowledge Using Outside Guidance." Reprinted from *Proceedings of the Seventh International Conference on Machine Learning*, pp. 235-243. Copyright © 1989, Morgan Kaufmann Publishers, Inc., Los Altos, California. Reprinted with permission.

Simon, T., Newell, A., and Klahr, D., "Q-Soar: A Computational Account of Children's Learning about Number Conservation." Reprinted from *Computational Approaches to Concept Formation*, edited by D. Fisher and M. Pazzani. Copyright © 1991, Morgan Kaufmann, Los Altos, California. Reprinted with permission.

Operations Research Society of America for
Prietula, M. J., Hsu, W., Steier, D. M., and Newell, A., "Applying an Architecture for General Intelligence to Reduce Scheduling Effort." Reprinted from *ORSA Journal on Computing*. Copyright © 1992, Operations Research Society of America, Baltimore, Maryland. Reprinted with permission.

Mike Prietula for
Vicinanza, S. and Prietula, M.J., "A Computational Model of Musical Creativity (Extended abstract)." Prepared for the AI and Music Workshop, Eleventh International Joint Conference on Artificial Intelligence. Reprinted with permission.

Sage Publications, Inc. for
Carley, K. and Wendt, K., "Electronic Mail and Scientific Communication: A Study of the Soar Extended Research Group." Reprinted from *Knowledge: Creation, Diffusion, Utilization*, Volume 12, pp. 406-440. Copyright © 1991, Sage Publications, Inc., Newbury Park, California. Reprinted with permission.

Stanford University for
Scales, D. J., "Efficient Matching Algorithms for the Soar/Ops5 Production System," Technical Report KSL-86-47. Reprinted with permission.

David Steier for
"But How Did You Know To Do That?": What a Theory of Algorithm Design Process Can Tell Us," Engineering

Design Research Center, Carnegie Mellon University, August, 1989. Reprinted with permission.

R. Washington for
Washington, R. and Rosenbloom, P. S., "Applying Problem Solving and Learning to Diagnosis." Department of Computer Science, Stanford University. Reprinted with permission.

John Wiley and Sons for
Newell, A., "Heuristic Programming: Ill-Structured Problems." Reprinted from *Progress in Operations Research, III,* edited by J. Aronofsky, pp. 360–414. Copyright © 1969, John Wiley and Sons, New York, 1969. Reprinted with permission.

The University of Michigan for
Congdon, C. B., "Learning Control Knowledge in an Unsupervised Planning Domain," Artificial Intelligence Laboratory, The University of Michigan. Reprinted with permission.

Laird, J. E., "Extending Problem Spaces to External Environments," Artificial Intelligence Laboratory, The University of Michigan.

Amy Unruh for
Unruh, A., Rosenbloom, P. S., and Laird, J. E., "Dynamic Abstraction Problem Solving in Soar." Reprinted from Proceedings of the Third Annual Aerospace Applications of Artificial Intelligence Conference, Dayton, Ohio, pp. 245-256. Reprinted with permission.

Gregg Yost for
Yost, G. R. and Newell, A., "Learning New Tasks in Soar," Department of Computer Science, Carnegie-Mellon University. Reprinted with permission.

Introduction

The Soar project is an attempt to develop and apply a unified theory of human and artificial intelligence. At the core of this effort is an investigation into the *architecture*—the fixed base of tightly-coupled mechanisms—underlying intelligent behavior. This architecture then forms the basis for wide-ranging investigations into basic intelligent capabilities—such as problem solving, planning, learning, knowledge representation, natural language, perception, and robotics—as well as applications in areas such as expert systems and psychological modeling. This is a true cognitive-science enterprise, where human and artificial evidence and criteria are constantly intermingled in service of progress in both areas [u:91].

Since the project's official inception in 1983, it has grown from a three-man effort to a geographically distributed, interdisciplinary community of more than ninety researchers (as of early 1992). To a first approximation, the community consists of computer scientists and psychologists in the United States and Europe; however, there is representation from other locales (such as Canada) and disciplines, primarily within engineering and the social sciences. What binds this community together is a commitment to develop, study and use a single software architecture—the Soar architecture—as the basis for intelligent behavior.

To date, this shared enterprise has produced over one hundred and sixty articles and books across the board on Soar and intelligent behavior. The wide range of topics covered by this body of scientific literature has led to an equally wide range of publication methods, and thus to a situation where investigators within individual disciplines have trouble locating the full set of publications of interest. The principal purpose of the present volumes is not to add further to this literature and problem, but to help solve it by bringing together in one place a relatively comprehensive set of core papers from the Soar community. Towards this end we have selected sixty-eight articles for inclusion out of those available by early 1991. This includes articles previously published in journals, conferences, workshops, and books, as well as some technical reports and unpublished papers.

In selecting the set of articles to be included, one criterion was whether an article makes a principal contribution on its own, independent of the other articles. Though this still leaves some overlap among the articles, it is primarily limited to the introductory descriptions of Soar that, by necessity, occur in nearly all of the articles. Many of these descriptions actually provide distinct ways of viewing Soar—for example, as a hierarchy of cognitive levels, or as a general goal-oriented system—and thus maintain some degree of independent utility, while the remainder tend to be short. We ask the reader's forbearance in this matter.

Most published collections of papers fall within one of two categories. Either they contain papers by unrelated groups of researchers covering the state of the art in some generic topic area, or they contain papers by members of a single tightly-knit project providing a retrospective on the project's accomplishments. The present collection fits into a niche that partially overlaps these two traditional categories, but also covers territory between them. The contributors are members of the Soar community, comprising a set of tightly-knit projects bound together in a looser confederation. The topic is simultaneously quite narrow—it is focused on Soar—and unusually broad, attempting to cover the full span of intelligent behavior. The perspective combines state-of-the-art papers with their historical antecedents, including some important antecedents (by community members) that predate the construction of Soar. Rather than serving primarily a retrospective function, this should provide a window onto an ongoing and vital research enterprise.

In broadest terms, the intended audience for these volumes is researchers interested in intelligent behavior. Because the prime focus of Soar is on integrating together the capabilities required for intelligent behavior, these volumes should be of most interest to researchers studying unified theories of cognition, cognitive architectures, intelligent agents, integrated architectures for intelligence, etc. However, many of the articles also make contributions in more limited areas; particularly in learning, problem solving, expert systems, robotics, production systems, immediate reasoning, and human-computer interaction. Though these volumes assume no background on Soar—the relevant introductory articles are included—many of the articles do assume some amount of technical background in artificial intelligence or cognitive psychology.

The structure of these volumes is atypical for a collection of papers. The standard approach is to divide the articles into parts according to each article's principal contribution. The index, or discursive front matter, is then used to deal with hidden secondary contributions. Such an approach was not taken for this collection because it loses the logical (and historical) flow of the material—which is particularly important when following the development of a single system such as Soar—and because too many of the articles make principal contributions in multiple areas (in addition to secondary contributions). The collection is therefore physically organized around chronological parts, with an alphabetic ordering by authors used within each part. Volume One covers the direct precursors to Soar (prior to 1983) and 1983 through 1988. Volume Two covers 1989 through the early part of 1991.

The remainder of this introduction comprises sections discussing Soar's direct precursors, its architecture, implementation issues, the capabilities it supports, the domains in which it has been applied, its use in psychological modeling, perspectives on it and the accompanying research effort, and its use as a programmable system. These sections correspond to what would be part introductions in a more traditional organization, and provide a conceptual road map for the collection through citation of the key articles that cover these topics. Each citation is a two part code of the form *number:year*. The *number* part specifies where the article falls in the alphabetical listing for its year. The *year* part specifies the year of publication for published papers, and the year of completion for unpublished papers. So, for example, the citation 9:89 indicates the ninth article alphabetically in 1989. Occasionally there was a very long gap between completion of an article and when it was finally published. In such cases, a pair of years is specified, as in 1:88/91. The first specifies the year of completion (1988) while the second specifies the year of publication (1991). To maintain the logical flow of the articles, such an article is included in the part corresponding to the first—that is, completion—date. So, citation 1:88/91 appears as the first article in the 1988 part.

Immediately following the citations to articles in the collection, and occasionally instead of them, citations are also included to Soar articles (and books) not in the collection. Though these external citations are not an exhaustive set, they do include a number of the most relevant ones through part of 1992. External citations are structurally similar to internal ones, except that external articles are ordered within years by letters rather than numbers. So, for example, b:90 is the second additional article listed in 1990. To further differentiate internal and external citations, where there are both, they are always printed in separate lists.

Following the introduction is a complete bibliography of the included articles, along with their associated citations. Enclosed in square brackets at the end of each entry is the volume and page within this collection where the article can be found. Following the bibliography of included articles is a bibliography for the external citations. Then come the actual articles included in this volume, organized into their chronological parts.

Direct Precursors

We are now in a position to seriously attempt the construction of integrated intelligent systems only because of the steady background of progress in understanding the individual components of intelligent behavior. The development of Soar owes much to this rich background. The direct precursors cover the portion of this background mostly closely tied to Soar; in particular, the background research by the developers of Soar, and some of their close colleagues, that is on the direct line to the development of Soar. This is in no way intended to deny the influence of other work on Soar, but is instead consistent with this collection's focus on Soar and the community of researchers that have grown up around it.

The initial driver in the development of Soar was to combine a flexible problem solving capability based on *problem spaces* and *weak methods* with the recognitional use of knowledge that is provided by *production systems.* Because of their strong influence on the development of Soar, and their current lack of accessibility (in particular to researchers in artificial intelligence), classic papers on problem spaces [1:80] and weak methods [1:69] are included in this collection. The other classic precursor on these topics is [a:72], but that is a book unto itself.

The critical production-system precursors fall into four groups. The first group covers the initial development of production systems for cognitive modeling [a:73]. The second group covers the Ops5 production-system language [a:81], which is the original basis for Soar's current memory architecture. The third group covers the instructable production system project [a:78, a:80], which is the seed out of which the Soar project grew. The fourth group covers a production-system language, Xaps2 [2:82/87], that was both the basis for the first implementation of Soar, and the basis for the implementation of the earliest version of Soar-style chunking.

Soar's chunking mechanism arose out of earlier work on models of human practice. This work progressed from an analysis of the data on human practice — in particular, the power-law shape of practice curves—and the development of an abstract chunking model [1:81]; to the development of a task-specific, production-system implementation of chunking [2:82/87]; to a task-independent, goal-based implementation [2:83/86] [d:83]. It was a variant of this last approach that was then implemented in Soar.

The *knowledge level* did not play a large role in the early development of Soar, but it has played an increasing role in our recent understanding of the system. The seminal paper on the knowledge level [1:82] is included in this collection.

Architecture

In computer science, the *architecture* of a system is the fixed structure that provides a system that can be programmed. It often coincides with the boundary between hardware and software, but it need not—for example, when one programming language is built on top of another. In cognitive science, the term has been extended to refer to the architecture of the mind; that is, the fixed structure underlying the flexible domain of cognitive processing. Architectures proposed as the basis for human cognition are termed *cognitive architectures*, while those proposed as the basis for artificial cognition are referred to as *architectures for integrated intelligent systems* (or *architectures for intelligent agents*, or *architectures for general intelligence)*. Introductions to the former can be found in [5:89] [d:90, n:92] and a brief note on the latter can be found in [j:89]. Our ultimate goal for the Soar architecture is that it serve as a basis for both human and artificial cognition. There is no generally accepted term for such a combination, so how Soar is described usually varies by context.

Since the construction of the first major version of Soar (Soar1) in 1982, the architecture has always been embodied as runnable code. During this period, revisions to the theory and implementation have led to five additional major versions of Soar—Soar2 in 1983, Soar3 in 1984, Soar4 in 1986, Soar5 in 1989, and Soar6 in 1992. Soar1 through Soar5 were all implemented in Lisp, while Soar6 is in C.

The prime rationale for Soar1 [1:83] was the development of a *universal weak method*—that is, a system capable of exhibiting a wide range of problem-solving strategies—through the combination of (multiple) problem spaces and production systems (in this case, the Xaps2 production system [2:82/87]). Goals, problem spaces, states and operators were symbolized in the production system's working memory. Processing was driven by an elaboration-decision-application cycle. Control knowledge was brought to bear during the elaboration phase via the parallel firing of productions, with the results being integrated together during the decision phase via a voting scheme. Soar1 operators were applied when they were selected.

Soar2 [b:83] was developed to explain how and why subgoals arise. Soar1's deliberate generation strategy was augmented by a *universal-subgoaling* mechanism that automatically created subgoals for impasses (then referred to as *difficulties)* in decision making. To facilitate the detection and diagnosis of impasses, the vote-based decision procedure was replaced by one based on symbolic preferences. The processing cycle was also simplified by dropping the separate application phase—operator application now occurred by a combination of elaboration and decision. Soar2, along with all later versions of Soar, utilized a modified version of the Ops5 production system [a:81] rather than Xaps2.

Soar3 completed the transition started with Soar2 by making subgoal generation and termination completely automatic. Soar1's deliberate generation strategy was removed, and automatic subgoal termination was added. In the process, working memory was extended to contain a stack of problem-solving contexts, rather than just the current one, and production firing and subgoal termination were allowed to proceed in parallel for all of these contexts. The other major addition in Soar3 was learning. A chunking mechanism was constructed that built new productions based on the results of subgoal-based problem-solving experience [1:84, 1:86].

Soar4 [2:87, 10:89/91] was the first publicly-released, and officially documented [c:86], version of the architecture. It embodied a number of refinements, particularly in chunking, but its primary reason for separation from Soar3 was its public status. An early detailed description of Soar4 can be found in [2:87]. A later abstract description of Soar4 can be found in [10:89/91], along with an analysis of Soar4 as an architecture for general intelligence.[1]

Soar5 was developed to support the *single-state principle* (that only one state should be available per problem space) [d:90], to eliminate the large amounts of state copying that occurred during operator application in the earlier architectures, and to support interaction with the outside world [4:91]. In the process, a number of significant modifications were made to the architecture, including the use of preferences for all modifications to working memory (enabling destructive state modification, among other things), the distinction between persistent and ephemeral working-memory elements (which is supported by a justification-based truth-maintenance system), and the development of input and output interfaces. Soar5 is publicly distributed with a completely new manual [b:90].

Soar6 is the first complete reimplementation of the Soar architecture. Functionally it is quite close to Soar5; however, it provides improvements in efficiency through the more careful selection of algorithms, and should provide significant improvements in robustness and maintainability by bringing the implementation up to the level of modern software engineering practice. Also, in a radical departure, Soar6 is written in C rather than in Lisp to improve portability and to further improve efficiency. Portability is of particular concern in supporting the large and rather diffuse Soar community, and in supporting investigations into the implementation of Soar on parallel machines. The development of Soar6 is being driven by a specification that provides a formal implementation-free description of the Soar architecture.

Implementation Issues

Though the development of Soar6 is almost totally driven by implementation concerns such as robustness, maintainability, efficiency, and portability, it is by no means the first or only investigation of these issues with respect to Soar. Throughout the evolution of the Soar architecture there has been an ongoing concern with implementation issues. The three such issues that have led to significant research—as opposed to simply coding effort—are efficiency, boundedness, and scalability.

Efficiency is a constant concern because it has such a large impact on the size and complexity

[1] Though the description in [10:89/91] is mostly of Soar4, there are aspects of Soar5 in it as well.

of the systems that can be developed within Soar. Thus a good deal of effort has gone into optimizing its innermost loop—the production matcher. Some of this has been concerned with improving the existing serial algorithms, while the remainder has been concerned with developing and implementing parallel algorithms. On the serial side, a set of enhancements have been made to the implementation of the Rete algorithm that came with the Lisp version of Ops5 [4:86], and then this implementation was compared with an implementation based on the Treat algorithm [4:88]. On the parallel side, matchers for Soar have been investigated for both the Encore Multimax [9:88] and the Connection Machine [2:88]. The Encore implementation was based on earlier work on parallel implementations of Ops5 [a:83, a:84, a:86, b:86]. The Connection Machine implementation was substantially different, and only partially completed.

Boundedness interacts with efficiency, but is fundamentally distinct. It consists of the ability to limit the resources, such as time, utilized by architectural processes. An appropriate limit may sometimes be a fixed value, such as one second, but at other times might be a restriction on the computational complexity of the process, such as restricting the match time of a production to be linear in the number of conditions. Boundedness did not start out as a major issue for Soar, but has become of increasing concern because of its impact on real-time behavior, on the complexity of parallel implementations, and on the utility of learned rules. As with efficiency, the research effort here has so far been focused on bounding the time required to match productions to working memory [10:90, 11:90] [n:89, g:90].

Though a number of reasonably large systems have been constructed in Soar over the years—for example, Neomycin-Soar [10:88] which reached 4869 productions and NL-Soar [5:91] [n:91] which reached 1838 productions—only recently has scalability begun to be explicitly addressed as a research issue [e:92].

Capabilities

The architecture provides a set of fixed mechanisms that directly perform a limited number of low-level functions, such as creating, matching and executing productions; selecting problem space components; and detecting the existence and disappearance of impasses. What it doesn't do is provide directly all of the higher-level capabilities required of an intelligent system, such as the use of a wide range of problem solving and planning methods, or the performance of skill and knowledge acquisition. These capabilities arise, if at all, from combining the architecture with the appropriate knowledge. The architecture may directly provide aspects of the capability—for example, the storage aspect of a learning capability—but for the rest, the most that can be hoped is that the architecture effectively supports the requisite use of the appropriate knowledge.

Soar-based research on higher-level capabilities has both a top-down thread and a bottom-up thread. The top-down thread focuses on understanding how Soar can provide the full range of higher-level capabilities that have independently been assessed as important in producing intelligent behavior. The principle issues raised in this work are: whether Soar can produce the capability; how easily, naturally, and elegantly Soar can do so; what role the architecture plays in producing

(or hindering the production of) the capability; and what, if any, additional action a Soar-based version of the capability provides over an isolated implementation of it. The bottom-up thread—what is sometimes referred to as "listening to the architecture"—focuses on the understanding of the, possibly novel, capabilities that are suggested by experience with Soar and its use. The principle issues raised in this work are: what the capability is, how it is produced, and what it is good for. Of course, even in top-down research, there is usually a bottom-up aspect—as when a Soar-based implementation of a known capability takes on an interestingly different shape because of the particular functions and constraints provided by the architecture—and vice versa (e.g., when it is discovered that a particular interaction among Soar's mechanisms provides a capability that maps quite well onto one that is well known in the literature, but with a novel way of producing it).

Capability research in Soar also has, at least conceptually, both an isolative thread and an integrative thread. The isolative thread investigates individual capabilities divorced from each other. The integrative thread investigates how capabilities combine—both with other capabilities of the same type (e.g., the combination of multiple problem solving methods) and with capabilities of other types (e.g., the combination of abstraction planning and explanation-based learning). While some of the capability research in Soar does follow one of these pure threads, most of the research actually intertwines them in a particular way, focusing on the production of an individual capability plus that capability's relationship to Soar's various architectural mechanisms. So, for example, work on abstraction focuses on how abstraction is produced, and how it interacts with impasses, chunking, etc. It does not, to nearly the same extent, focus on abstraction's integration with other high-level capabilities, except for some planning or problem solving capability to which the abstraction must be tied.

The high-level capabilities investigated so far in Soar are most naturally partitioned into four general types: problem solving and planning, learning, external interaction, and knowledge representation. This ordering roughly corresponds to the ordering in which these capability types began to be addressed in Soar, and is the order in which we go through them here.

Problem solving and planning capabilities were central to the early development of Soar, and still attract considerable research effort. The original focus was on weak problem-solving methods. In particular, a *universal weak method* was developed that could, with the addition of small increments of knowledge, exhibit the range of standard AI search methods [1:83] [c:83, b:83]. Other major thrusts in this area include investigations of goals [b:83, e:91, a:92], abstraction [6:87, 14:89, 12:90, 8:90] [k:88], planning [4:90, 8:90] [i:88, f:89, i:91, c:92, d:92, m:92], generic tasks [2:90] [c:88, j:88, h:89, g:91, j:91, o:92], abduction [f:91, j:91, l:91], and mental models [5:88, 7:89, 3:89]. There has also been a continuing low-level interest in analogy and case-based reasoning [1:87, 15:89] [a:87, f:90, c:92], plus more recent thrusts in multi-tasking [b:92] and multi-agent problem solving [a:91].

Overviews of the learning research in Soar can be found in [1:86, 5:87] [e:86, c:87]. At the grossest level, this research can be partitioned into work on skill acquisition, knowledge acquisition, and integrated learning. Most of the early work was on skill acquisition; in particular, on learning to perform existing tasks more quickly. Our work with chunking began with its application as a model of human practice [1:81, 2:82/87, 2:83/86, 9:89] [b:82, d:83], which then grew into investigations of the acquisition of control, (macro-)operator, and reactive knowledge [1:84, 1:86, 1:87, 2:89] [i:91]. This expansion into core machine learning topics raised the question of the relationship of chunking to explanation-based learning [3:86] [c:90]. It also forced us to address the utility question—that is, whether chunking was actually speeding up Soar when measured in terms of real time—and led to the identification of the problem of expensive chunks, and to a space of possible solutions [10:90, 11:90] [m:88, n:88, n:89, g:90, v:91, w:91].

Investigations of knowledge acquisition began by looking at simple forms of associative learning [3:87, 9:89, 6:88, 11:89/91, 7:90, 1:90] [c:87]. On the cognitive side this focused on forms of verbal learning, such as learning to recognize and recall words and nonsense syllables, while on the AI side it focused on demonstrating how chunking could perform knowledge-level learning. More complex forms of knowledge acquisition then began to appear in work on inductive generalization and concept formation [7:90, 6:91, 14:89, 5:90, 8:91] [c:84, j:88, i:92]. At the most complex end comes work on the acquisition, extension and correction of domain models (that is, problem spaces) [2:86, 3:88, 11:88, 3:89, 3:90, 1:90] [b:88, k:88, f:92].

Investigations of integrated learning can be found in [5:87, 9:89, 6:88, 7:90] [j:88]. Most of this work focuses on combining the use of chunking as a skill acquisition mechanism with its use as the storage mechanism underlying knowledge acquisition.

The study of external interaction in Soar is much less mature than either the study of problem solving and planning or the study of learning. It did not really come into its own until the development of the interaction framework in Soar5, and Soar still does not have a stable and well-developed perceptual-motor system. However, work has been progressing along a number of fronts in getting Soar to interact with people, with the physical world, and with other software systems. Research on interaction with people is focused on ways they can communicate to Soar other than via the standard programming model of the person directly creating new structures in Soar's memory. This has consisted of studies of in-context advice taking [1:87, 3:90] [m:91], instruction taking [11:88, 3:89], and natural language comprehension [5:91, 3:89, 5:90] [b:84, g:89, o:89, n:91]. Research on interaction with the physical world has focused on low-level work on perceptual attention and search [9:91] [i:90, p:92], goal-directed visual processing [g:92, i:91], robotics (hand-eye and mobile-hand systems) [3:90, 4:90, 4:91] [m:91, i:89], and interruption [4:90] [o:88, f:89, b:92]. Research on interaction with other software systems is still in its formative stages, but there has been work on both low-level communication details and high-level analyses of issues [p:91].

Knowledge representation has not traditionally been a major explicit focus in Soar. Since its inception, Soar has been based on objects as sets of attribute-value pairs, long-term knowledge as sets of productions, and domain models as sets of problem spaces. On top of this basic structure sits default knowledge which defines the top problem solving context, default responses to impasses, monitoring abilities, and default spaces (and concepts) that facilitate look-ahead search and operator subgoaling [b:90]. Each system-building effort then starts with this, and acts as a (mostly) implicit study of the specifics of representing the types of knowledge required for the domains and methods of interest. Citations to this work are thus best found by looking under the

respective domains and capabilities. One explicit study of the representational issues for procedural, episodic, and declarative knowledge in Soar can be found in [11:89/91].

Domains

An important aspect of generality in an intelligent system is the range of task domains in which it can effectively behave. Thus a significant thread throughout the development of Soar has been its application in diverse domains. Befitting the early emphasis on weak problem solving methods, the earliest applications of Soar were in classic toy tasks, such as puzzles and games [1:83, 1:84, 1:86, 2:87] [b:83, b:87]. Even with the increased emphasis now placed on more practical knowledge-intensive and interactive domains, toy tasks continue to be of interest because of their ability to capture in particularly clear ways the essence of many hard search problems [3:88, 1:90, 3:90, 4:90, 12:90] [b:87, b:88].

The shift to more knowledge-intensive domains began with the implementation of the R1-Soar computer-configuration system [1:85]. Since its initial development, R1-Soar has been used as the basis for experiments in skill acquisition [2:87], abstraction [6:87, 14:89], and task acquisition [16:89]. The development of R1-Soar was soon followed by a sequence of systems in the area of algorithm and software design [4:87, 8:88, 13:89] [k:89, a:88]; and then by a set of systems that focused on other design domains [4:89, 15:89] [j:88, k:88], medical diagnosis [10:88], blood banking [a:90, g:91, l:91], and factory scheduling [6:90/92] [f:88, d:89, e:89]. The first in what is to be a series of articles on the Soar approach to building knowledge-intensive systems can be found in [o:92].

The most recent shift in application focus has been to domains that require tight coupling between Soar and its environment, whether this environment be the physical world or other software systems (or people, for that matter, though no substantial domains have shown up here yet). The physical-world domains investigated so far are still quite simple; in particular, block alignment by a hand-eye system [3:90] [m:91] and cup collection and disposal by a mobile-hand system [4:90]. The software domains being investigated are more varied, including databases [p:91], symbolic mathematics packages [p:91], chemical process simulators [p:91], drawing packages [p:91], tutorial environments [p:91, x:91], building-design tools [9:90] [p:91], and physical- world simulators.

In addition to the domains listed in this section, there has also been work on algorithmic domains, such as multi-column subtraction [10:89/91], and a significant number of applications in the context of psychological modeling (which is the focus of the next section).

Psychological Modeling

One of the grand challenges for Soar is to develop it into a *unified theory of cognition* that, through the structure of its architecture and the content of its problem spaces, can veridically produce the full range of human cognitive behavior. The major statement of this grand challenge, and of Soar's status with respect to it, can be found in [d:90]. Additional overviews and status reports can be found in [5:90] [p:88, q:88, h:92, j:92], and an attempt to extend this idea into the social realm can be found in [b:91].

The work covered in papers included in this collection provides a patchwork of results in the general areas of perception, routine cognitive skill, reasoning, problem solving, learning, and development. The principal focus so far in perception has been on modeling visual attention [9:91] [i:90, p:92]. Some issues of perception also arise in the work on routine cognitive skill; however, the emphasis there is much more on high-level aspects of perception and motor control, and their relationship to cognition. This work to date has focused mostly on using Soar as the substrate for, and extensions to, two general human-computer interaction (HCI) formalisms: GOMS (Goals, Operators, Methods, and Selection rules) [5:90, 3:91] and PUM (Programmable User Models) [13:90, 2:91]. Domains covered in this work include highly interactive ones, such as on-line browsers [5:90] [k:92] and video games [3:91], plus the more traditional domain of text editing [13:90, 2:91]. Other work on routine cognitive skill includes efforts to model calculator use [c:91], automobile driving [b:92] and the NASA Test Director at the Kennedy Space Center [k:91].

With reasoning, the emphasis shifts to pure cognitive processing. The work so far in human reasoning has concentrated on extended mental-model-based approaches to classic tasks that reveal ways in which people do not follow the standard rules of logic. Categorical syllogisms have received the most attention, however there has also been work on relational reasoning, conditional reasoning, and the Wason selection task [5:88, 7:89, 3:89]. The work on problem solving has focused on modeling human protocols in non-routine domains ranging from simple puzzles, such as Towers of Hanoi [12:89] and Cryptarithmetic [d:90], to more pragmatic domains such as instructional design [q:91].

Much of the Soar-based research on human learning has concerned relatively low-level phenomena such as practice phenomena—in particular, modeling the power law of practice [1:81, 2:82/87, 2:83/86, 9:89]—and the beginnings of a model of classical verbal learning [3:87, 9:89]. However, there has been some work on higher level phenomena, such as concept acquisition—in the context of series completion [5:90] and lexical acquisition [6:91] [i:92]—and strategy change [12:89]. At the highest end, learning shades into developmental transitions, with work on quantity conservation [8:91] and the balance beam [d:90].

Perspectives

The most basic way to view Soar is as a symbol system; that is, as a collection of symbols and mechanisms for their processing [5:89, 2:87] [e:90, o:92]. However, this view—though accurate—by itself misses much of what gives Soar its particular character. The first missing aspect is the fine structure of how Soar's mechanisms form a tightly coupled hierarchy of layers—memory, decision, and goal—in which each layer forms the inner loop of the layer above it [10:89/91] [d:90]. These layers increase progressively in both complexity and time scale from the bottom to the top of the hierarchy.

The second missing aspect is the identification of the class(es) of symbol systems within which Soar should be situated. This is not a completely straightforward task for Soar because of its range of capabilities and application domains. It is also not a completely bounded task, as new developments are periodically increasing the set of classes. Nonetheless, various analyses have identified at least the following classes: a general problem solver (or goal-oriented system) [1:83, 8:89], a learning system [5:87], a cognitive architecture [5:89], an expert system shell [1:85], a meta-level (reflective) system [7:88], and a hybrid planning system [4:90, 8:90].

The third missing aspect is the comparison of Soar with other integrated architectures. So far, not many such comparisons exist in the published literature (as opposed to the larger number of informal comparisons that are generated during courses on integrated architectures). However, written comparisons do at least exist for GPS [j:92], Act* [5:89], and the constructive-integration model [h:90].

The fourth missing aspect is that Soar provides more than just a symbol system. Above the symbol system, it provides a problem-space system that is based on the symbol system, but has its own distinct computational model [7:91] [b:90, e:90, o:92]. Further up—above the problem-space system — Soar provides an approximation to a knowledge-level system [11:89/91, 7:91] [1:82, d:90, n:92, o:92]. There is currently no theoretically interesting system under Soar's symbol system; however, some thought has been given to the relationship of Soar to lower neural systems [8:89] [d:90, n:92], and work has begun on a neural-network basis for Soar [1:91].

The fifth missing aspect is how Soar is viewed by researchers from outside the community directly involved in its development and use. Several external commentaries on Soar originated as invited responses to papers presented at workshops and symposia (though the commentaries often address more general issues than those specifically raised in the papers at hand, as well as often addressing other systems in addition to Soar). In particular, a commentary based on [9:89] can be found in [1:89], one based on [10:89/91] can be found in [6:89/91], and two based on [11:89/91] can be found in [d:91] and [r:91]. Other commentaries have taken the form of book reviews: [d:87] and [h:88] are reviews of [d:86], while [h:91] and [o:91] are reviews of [d:90]. In addition to commentaries, several researchers have compared individual Soar capabilities directly with other alternatives. In particular, Soar's learning abilities have been compared with decision-analytic control [2:89], backpropagation [b:88], and a recursion-controlled version of explanation-based generalization [c:90].

The sixth, and final, missing aspect is the view of Soar, and the Soar research effort, from historical and sociological perspectives. On the historical side, this introduction covers much of the development of Soar, plus a bit about its prehistory. Additional details on its prehistory can also be found in [d:86]. On the sociological side, a study of the development of the Soar community can be found in [1:88/91].

Using Soar

Ideally Soar should be "programmable" as a knowledge-level system. Adjustments and enhancements of behavior should be possible by simply conveying the appropriate knowledge to the system. Though such an activity can be thought of as programming at the knowledge level, it really has more in common with acts of communication than with acts of programming. Failing this, Soar should be programmable as a problem-space system. Augmentations of its store of problem spaces should be directly derivable from provided descriptions. Programming a problem-space system is primarily a modeling activity, in which systems are constructed by describing the objects and actions in the domains of interest. To the extent that Soar fails to be programmable as a problem-space system, it must be programmed as a symbol-level system. Here the requisite activity is closer to classical production-system programming, though the myriad of ways in which Soar differs from such classical systems—including the absence of conflict resolution and the provision of preferences, subgoaling, chunking, etc.—dictates that programming Soar at the symbol level is still a rather distinct activity.

In reality, Soar is not currently programmable at the knowledge level, and it is far from clear how to provide such a capability. However, there has been progress towards the ability to program it as a problem-space system. A front end, called TAQL (Task AcQuisition Language), has been developed that provides a problem-space language which can be automatically compiled into Soar productions [y:91]. When using TAQL, its manual should be used in conjunction with the Soar manual [b:90]. The Soar manual provides an overview of the problem space level, details of Soar's symbol level structures and mechanisms, an introduction to encoding tasks in Soar, a description of the default knowledge that comes with the system, descriptions of the user-accessible variables and functions, a glossary, and production templates for common problem-space operations (which are useful when programming Soar at the symbol level). When programming at the symbol level, the Soar manual should be sufficient on its own.

In addition to TAQL, the usability of Soar has been enhanced by the construction of the Developers Soar Interface (DSI). The DSI provides GnuEmacs editor modes for Soar [t:91] and TAQL [s:91], plus an X-based graphical display interface [l:92].[2]

Conclusion

This brings us to the end of the road map. Following the presentation of the two bibliographies—of included and additional articles—come the Soar papers.

[2] An earlier attempt to build such an interface for Soar is described in [f:86, g:86].

Bibliography of Included Articles

Prior to 1983

1:69 Newell, A., "Heuristic programming: Ill-structured problems," in *Progress in Operations Research, III*, Aronofsky, J., ed., Wiley, New York, 1969, pp. 360-414. [One: 3]

1:80 Newell, A., "Reasoning, problem solving and decision processes: The problem space as a fundamental category," in *Attention and Performance VIII*, R. Nickerson, ed., Erlbaum, Hillsdale, N.J., 1980. [One: 55]

1:81 Newell, A. & Rosenbloom, P. S., "Mechanisms of skill acquisition and the law of practice," in *Cognitive Skills and their Acquisition*, J. R. Anderson, ed., Erlbaum, Hillsdale, NJ, 1981, pp. 1-55. [One: 81]

1:82 Newell, A., "The knowledge level," *Artificial Intelligence*, Vol. 18, 1982, pp. 87-127. [One: 136]

2:82/87 Rosenbloom, P. S. & Newell, A., "Learning by chunking: A production-system model of practice," in *Production System Models of Learning and Development*, D. Klahr, P. Langley, R. Neches, eds., Bradford Books/The MIT Press, Cambridge, MA, 1987, pp. 221-286. [One: 177]

1983 – 1985

1:83 Laird, J. E. & Newell, A., "A Universal Weak Method," Tech. report #83-141, Carnegie-Mellon University Computer Science Department, June 1983. [One: 245]

2:83/86 Rosenbloom, P. S. & Newell, A., "The chunking of goal hierarchies: A generalized model of practice," in *Machine Learning: An Artificial Intelligence Approach*, Volume II, R. S. Michalski, J. G. Carbonell, & T. M. Mitchell, eds., Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1986, pp. 247-288. [One: 293]

1:84 Laird, J. E., Rosenbloom, P. S., & Newell, A., "Towards chunking as a general learning mechanism," *Proceedings of the National Conference on Artificial Intelligence*, AAAI, 1984, pp. 188-192. [One: 335]

1:85 Rosenbloom, P. S., Laird, J. E., McDermott, J., Newell, A. & Orciuch, E., "R1-Soar: An experiment in knowledge-intensive programming in a problem-solving architecture," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 7, 1985, pp. 561-569. [One: 340] 1986

1986

1:86 Laird, J. E., Rosenbloom, P. S., & Newell, A., "Chunking in Soar: The anatomy of a general learning mechanism," *Machine Learning*, Vol. 1, 1986, pp. 11-46. [One: 351]

2:86 Laird, J. E., Rosenbloom, P. S., & Newell, A., "Overgeneralization during knowledge compilation in Soar," *Proceedings of the Workshop on Knowledge Compilation*, T. G. Dietterich,ed., AAAI/Oregon State University, 1986, pp. 46-57. [One: 387]

3:86 Rosenbloom, P. S., & Laird, J. E., "Mapping explanation-based generalization onto Soar," *Proceedings of the Fifth National Conference on Artificial Intelligence*, AAAI, 1986, pp. 561-567. [One: 399]

4:86 Scales, D. J., "Efficient Matching Algorithms for the Soar/Ops5 Production System," Tech. report KSL-86-47, Knowledge Systems Laboratory, Department of Computer Science, Stanford University, 1986. [One: 407]

1987

1:87 Golding, A. R., Rosenbloom, P. S., & Laird, J. E., "Learning general search control from outside guidance," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, IJCAII, 1987, pp. 334-337. [One: 459]

2:87 Laird, J. E., Newell, A., & Rosenbloom, P. S., "Soar: An architecture for general intelligence," *Artificial Intelligence*, Vol. 33, 1987, pp. 1-64. [One: 463]

3:87 Rosenbloom, P. S., Laird, J. E., & Newell, A., "Knowledge level learning in Soar," *Proceedings of Sixth National Conference on Artificial Intelligence*, AAAI, 1987, pp. 499-504. [One: 527]

4:87 Steier, D., "Cypress-Soar: A case study in search and learning in algorithm design," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, IJCAII, 1987, pp. 327-330. [One: 533]

5:87 Steier, D. M., Laird, J. E., Newell, A., Rosenbloom, P. S., Flynn, R., Golding, A., Polk, T. A., Shivers, O. G., Unruh, A., & Yost, G. R., "Varieties of Learning in Soar: 1987," *Proceedings of the Fourth International Workshop on Machine Learning*, P. Langley,ed., Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1987, pp. 300-311. [One: 537]

6:87 Unruh, A., Rosenbloom, P. S., & Laird, J. E., "Dynamic abstraction problem solving in Soar," *Proceedings of the Third Annual Aerospace Applications of Artificial Intelligence Conference*, Dayton, OH, 1987, pp. 245-256. [One: 549]

1988

1:88/91 Carley, K. & Wendt, K., "Electronic Mail and Scientific Communication: A study of the Soar extended research group," *Knowledge: Creation, Diffusion, Utilization*, Vol. 12, 1991, pp. 406-440. [One: 563]

2:88 Flynn, R., "Placing Soar on the Connection Machine," Digital Equipment Corporation, September, 1988. An extended abstract of this paper was distributed at the AAAI Mini-Symposium, How Can Slow Components Think So Fast?. [One: 598]

3:88 Laird, J. E., "Recovery from incorrect knowledge in Soar," *Proceedings of the Seventh National Conference on Artificial Intelligence*, AAAI, 1988, pp. 618-623. [One: 615]

4:88 Nayak, P., & Gupta, A., & Rosenbloom, P. S., "Comparison of the Rete and Treat production matchers for Soar (a summary)," *Proceedings of the Seventh National Conference on Artificial Intelligence*, AAAI, 1988, pp. 693-698. [One: 621]

5:88 Polk, T. A. & Newell, A., "Modeling human syllogistic reasoning in Soar," *Proceedings of the 10th Annual Conference of the Cognitive Science Society*, 1988, pp. 181-187. [One: 627]

6:88 Rosenbloom, P. S., "Beyond generalization as search: Towards a unified framework for the acquisition of new knowledge," *Proceedings of the AAAI Symposium on Explanation-Based Learning*, G. F. DeJong,ed., AAAI, Stanford, CA, 1988, pp. 17-21. [One: 634]

7:88 Rosenbloom, P. S., Laird, J. E., & Newell, A., "Meta-levels in Soar," in *Meta-Level Architectures and Reflection*, P. Maes & D. Nardi, eds., North Holland, Amsterdam, 1988, pp. 227-240. [One: 639]

8:88 Steier, D. M. & Newell, A., "Integrating multiple sources of knowledge into Designer-Soar, an automatic algorithm designer," *Proceedings of the Seventh National Conference on Artificial Intelligence*, AAAI, 1988, pp. 8-13. [One: 653]

9:88 Tambe, M., Kalp, D., Gupta, A., Forgy, C.L., Milnes, B.G., & Newell, A., "Soar/PSM-E: Investigating match parallelism in a learning production system," *Proceedings of the ACM/SIGPLAN Symposium on Parallel Programming: Experience with Applications, Languages, and Systems*, July 1988, pp. 146-160. [One: 659]

10:88 Washington, R. & Rosenbloom, P. S., "Applying Problem Solving and Learning to Diagnosis," Department of Computer Science, Stanford University. Unpublished. [One: 674]

11:88 Yost, G. R. & Newell, A., "Learning new tasks in Soar," Department of Computer Science, Carnegie-Mellon University. Unpublished. [One: 688]

1989

1:89 Bösser, T., "A discussion of 'The chunking of skill and knowledge' by Paul S. Rosenbloom, John E. Laird & Allen Newell," in *Working Models of Human Perception*, B. A. G. Elsendoorn & H. Bouma, eds., Academic Press, London, 1989, pp. 411-418. [Two: 705]

2:89 Etzioni, O., & Mitchell, T. M., "A comparative analysis of chunking and decision-analytic control," *Proceedings of the 1989 AAAI Spring Symposium on AI and Limited Rationality*, Stanford, 1989. [Two: 713]

3:89 Lewis, R. L., Newell, A., & Polk, T. A., "Toward a Soar theory of taking instructions for immediate reasoning tasks," *Proceedings of the 11th Annual Conference of the Cognitive Science Society*, 1989, pp. 514-521. [Two: 719]

4:89 Modi, A.K. & Westerberg, A.W., "Integrating learning and problem solving within a chemical process designer," Presented at the Annual Meeting of the American Institute of Chemical Engineers. [Two: 727]

5:89 Newell, A., Rosenbloom, P. S., & Laird, J. E., "Symbolic architectures for cognition," in *Foundations of Cognitive Science*, M. I. Posner, ed., Bradford Books/MIT Press, Cambridge, MA, 1989, pp. 93-131. [Two: 754]

6:89/91 Norman, D. A., "Approaches to the study of intelligence," *Artificial Intelligence*, Vol. 47, 1991, pp. 327-346. [Two: 793]

7:89 Polk, T. A., Newell, A., & Lewis, R. L., "Toward a unified theory of immediate reasoning in Soar," *Proceedings of the 11th Annual Conference of the Cognitive Science Society*, 1989, pp. 506-513. [Two: 813]

8:89 Rosenbloom, P. S., "A symbolic goal-oriented perspective on connectionism and Soar," in *Connectionism in Perspective*, R. Pfeifer, Z. Schreter, F. Fogelman-Soulie, & L. Steels, eds., Elsevier (North-Holland), Amsterdam, 1989, pp. 245-263. [Two: 821]

9:89 Rosenbloom, P. S., Laird, J. E., & Newell, A., "The chunking of skill and knowledge," in *Working Models of Human Perception*, B. A. G. Elsendoorn & H. Bouma, eds., Academic Press, London, 1989, pp. 391-410. [Two: 840]

10:89/91 Rosenbloom, P. S., Laird, J. E., Newell, A., & McCarl, R., "A preliminary analysis of the Soar architecture as a basis for general intelligence," *Artificial Intelligence*, Vol. 47, 1991, pp. 289-325. [Two: 860]

11:89/91 Rosenbloom, P. S., Newell, A., & Laird, J. E., "Towards the knowledge level in Soar: The role of the architecture in the use of knowledge," in *Architectures for Intelligence*, K. VanLehn, ed., Erlbaum, Hillsdale, NJ, 1991, pp. 75-111. [Two: 897]

12:89 Ruiz, D. & Newell, A., "Tower-noticing triggers strategy-change in the Tower of Hanoi: A Soar model," *Proceedings of the Annual Conference of the Cognitive Science Society*, 1989, pp. 522-529. [Two: 934]

13:89 Steier, D., ""But How Did You Know To Do That?": What a theory of algorithm design process can tell us," Engineering Design Research Center, Carnegie Mellon University. Unpublished. [Two: 942]

14:89 Unruh, A. & Rosenbloom, P. S., "Abstraction in Problem Solving and Learning," *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, IJCAII, 1989, pp. 681-687. [Two: 959]

15:89 Vicinanza, S. & Prietula, M.J., "A computational model of musical creativity (Extended abstract)," Prepared for the AI and Music Workshop, Eleventh International Joint Conference on Artificial Intelligence. [Two: 974]

16:89 Yost, G. R. & Newell, A., "A problem space approach to expert system specification," *Proceedings of Eleventh International Joint Conference on Artificial Intelligence*, IJCAII, 1989, pp. 621-627. [Two: 982]

1990

1:90 Congdon, C.B., "Learning Control Knowledge in an Unsupervised Planning Domain," Artificial Intelligence Laboratory, The University of Michigan. Unpublished. [Two: 991]

2:90 Johnson, T. R., Smith, J. W., & Chandrasekaran, B., "Task-specific architectures for flexible systems," Submitted to *Applied AI*. [Two: 1004]

3:90 Laird, J.E., Hucka, M., Yager, E.S., & Tuck, C.M., "Correcting and extending domain knowledge using outside guidance," *Proceedings of the Seventh International Conference on Machine Learning, 1990*, pp. 235-243. [Two: 1027]

4:90 Laird, J. E., & Rosenbloom, P. S., "Integrating execution, planning, and learning in Soar for external environments," *Proceedings of the Eighth National Conference on Artificial Intelligence*, AAAI, 1990, pp. 1022-1029. [Two: 1036]

5:90 Lewis, R. L., Huffman, S. B., John, B. E., Laird, J. E., Lehman, J. F., Newell, A., Rosenbloom, P. S., Simon, T., & Tessler, S. G., "Soar as a unified theory of cognition: Spring 1990," *Proceedings of the 12th Annual Conference of the Cognitive Science Society, 1990*, pp. 1035-1042. [Two: 1044]

6:90/92 Prietula, M. J., Hsu, W., Steier, D. M., & Newell, A., "Applying an architecture for general intelligence to reduce scheduling effort," *ORSA Journal on Computing, 1992*, In press. [Two: 1052]

7:90 Rosenbloom, P. S. & Aasman, J., "Knowledge level and inductive uses of chunking (EBL)," *Proceedings of the Eighth National Conference on Artificial Intelligence*, AAAI, 1990, pp. 821-827. [Two: 1096]

8:90 Rosenbloom, P. S., Lee, S., & Unruh, A., "Responding to impasses in memory-driven behavior: A framework for planning," *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling, and Control, DARPA, 1990*, pp. 181-191. [Two: 1103]

9:90 Steier, D.M., "Intelligent architectures for integration," *Proceedings of the IEEE Conference on Systems Integration*, August 1990, A shorter version of this paper appeared in the proceedings. [Two: 1114]

10:90 Tambe, M., Newell, A., & Rosenbloom, P. S., "The problem of expensive chunks and its solution by restricting expressiveness," *Machine Learning*, Vol. 5, 1990, pp. 299-348. [Two: 1123]

11:90 Tambe, M. & Rosenbloom, P. S., "A framework for investigating production system formulations with polynomially bounded match," *Proceedings of the Eighth National Conference on Artificial Intelligence*, AAAI, 1990, pp. 693-700. [Two: 1173]

12:90 Unruh, A., & Rosenbloom, P. S., "Two new weak method increments for abstraction," *Working Notes of the AAAI-90 Workshop on Automatic Generation of Approximations and Abstractions*, T. Ellman,ed., AAAI, 1990, pp. 78-86. [Two: 1181]

13:90 Young, R.M. & Whittington, J., "Using a knowledge analysis to predict conceptual errors in text-editor usage," *Proceedings of CHI'90, 1990*, pp. 91-97. [Two: 1190]

1991

1:91 Cho, B., Rosenbloom, P. S., & Dolan, C. P., "Neuro-Soar: A neural-network architecture for goal-oriented behavior," *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, Erlbaum, Hillsdale, NJ, 1991, pp. 673-677. [Two: 1199]

2:91 Howes, A. & Young, R. M., "Predicting the learnability of task-action mappings," *Proceedings of CHI'91 Human Factors in Computing Systems*, ACM Press, 1991, pp. 113-118. [Two: 1204]

3:91 John, B. E., Vera, A. H., & Newell, A., "Towards Real-Time GOMS," Abbreviated version of "Towards Real-Time GOMS," by John, B. E., Vera, A. H., & Newell, A. Carnegie Mellon University School of Computer Science Technical Report CMU-CS-90-195. [Two: 1210]

4:91 Laird, J. E., "Extending problem spaces to external environments," Artificial Intelligence Laboratory, The University of Michigan. Unpublished. [Two: 1294]

5:91 Lehman, J. F., Lewis, R. L., & Newell, A., "Integrating knowledge sources in language comprehension," *Proceedings of the Thirteenth Annual Meeting of the Cognitive Science Society*, Erlbaum, Hillsdale, NJ, 1991, pp. 461-466. [Two: 1309]

6:91 Miller, C. S., & Laird, J. E., "A constraint-motivated lexical acquisition model," *Proceedings of the Thirteenth Annual Meeting of the Cognitive Science Society*, Erlbaum, Hillsdale, NJ, 1991, pp. 827-831. [Two: 1315]

7:91 Newell, A., Yost, G. R., Laird, J. E., Rosenbloom, P. S., & Altmann, E., "Formulating the problem space computational model," in *CMU Computer Science: A 25th Anniversary Commemorative*, R. F. Rashid, ed., ACM Press/Addison-Wesley, 1991, pp. 255-293. [Two: 1321]

8:91 Simon, T., Newell, A., & Klahr, D., "A computational account of children's learning about number conservation," in *Concept Formation: Knowledge & Experience in Unsupervised Learning*, Fisher, D.H., Pazzani, M.J & Langley, P., eds., Morgan Kaufmann, Los Altos, CA, 1991. [Two: 1360]

9:91 Wiesmeyer, M. & Laird, J.E., "Attentional Modeling of Object Identification and Search," Tech. report CSE-TR-108-91, Department of Electrical Engineering and Computer Science, University of Michigan, 1991, A portion of this article appeared under the title "A computer model of 2D visual attention, in *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society, 1990*. [Two: 1400]

Bibliography of Additional Articles

Prior to 1983

a:72 Newell, A. & Simon, H. A., *Human Problem Solving*, Prentice-Hall, Englewood Cliffs, 1972.

a:73 Newell, A., "Production systems: Models of control structures," in *Visual Information Processing*, Chase, W. G., ed., Academic Press, New York, 1973, pp. 463-526.

a:78 Rychener, M. D. & Newell, A., "An instructable production system: Basic design issues," in *Pattern-Directed Inference Systems*, Waterman, D. A. & Hayes-Roth, F., eds., Academic Press, New York, 1978, pp. 135-153.

a:81 Forgy, C. L., "OPS5 User's Manual," Tech. report CMU-CS-81-135, Computer Science Department, Carnegie-Mellon University, July 1981.

a:82 Forgy, C. L., "Rete: A fast algorithm for the many pattern/many object pattern match problem," *Artificial Intelligence*, Vol. 19, 1982, pp. 17-37.

b:82 Rosenbloom, P. S. & Newell, A., "Learning by chunking: Summary of a task and a model," *Proceedings of the National Conference on Artificial Intelligence*, AAAI, 1982, pp. 255-257.

1983–1985

a:83 Gupta, A. & Forgy, C., "Measurements on Production Systems," Tech. report CMU-CS-83-167, Computer Science Department, Carnegie Mellon University, December 1983.

b:83 Laird, J. E., Universal Subgoaling, PhD dissertation, Carnegie-Mellon University, 1983, (Available in Laird, J. E., Rosenbloom, P. S., & Newell, A. *Universal Subgoaling and Chunking: The Automatic Generation and Learning of Goal Hierarchies*, Hingham, MA: Kluwer, 1986).

c:83 Laird, J. E. & Newell, A., "A universal weak method: Summary of results," *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, IJCAII, 1983, pp. 771-773.

d:83 Rosenbloom, P. S., The Chunking of Goal Hierarchies: A Model of Practice and Stimulus-Response Compatibility, PhD dissertation, Carnegie-Mellon University, 1983, (Available in Laird, J. E., Rosenbloom, P. S., & Newell, A. *Universal Subgoaling and Chunking: The Automatic Generation and Learning of Goal Hierarchies*, Hingham, MA: Kluwer, 1986).

a:84 Gupta, A., "Implementing OPS5 Production Systems on DADO," Tech. report CMU-CS-84-115, Computer Science Department, Carnegie Mellon University, March 1984.

b:84 Powell, L., "Parsing the Picnic Problem with a Soar3 Implementation of Dypar-1," Department of Computer Science, Carnegie-Mellon University. Unpublished.

c:84 Saul, R. H., "A SOAR2 Implementation of Version-Space Inductive Learning," Department of Computer Science, Carnegie-Mellon University. Unpublished.

1986

a:86 Gupta, A., Parallelism in Production Systems, PhD dissertation, Computer Science Department, Carnegie Mellon University, March 1986, Also available in *Parallelism in Production Systems*, Morgan-Kaufman, 1987.

b:86 Gupta, A., Forgy, C., Newell, A., and Wedig, R., "Parallel algorithms and architectures for production systems," *Proceedings of the Thirteenth International Symposium on Computer Architectures*, June 1986, pp. 28-35.

c:86 Laird, J. E., "Soar User's Manual (Version 4)," Tech. report ISL-15, Xerox Palo Alto Research Center, 1986.

d:86 Laird, J. E., Rosenbloom, P. S., & Newell, A., *Universal Subgoaling and Chunking: The Automatic Generation and Learning of Goal Hierarchies*, Kluwer Academic Publishers, Hingham, MA, 1986.

e:86 Rosenbloom, P. S., Laird, J. E., Newell, A., Golding, A., & Unruh, A., "Current research on learning in Soar," in *Machine Learning: A Guide to Current Research*, T. M. Mitchell, J. G. Carbonell, & R. S. Michalski, eds., Kluwer Academic Press, Boston, MA, 1986, pp. 281-290.

f:86 Unruh, A., "Comprehensive Programming Project: An Interface for Soar," Department of Computer Science, Stanford University. Unpublished.

1987

a:87 Faieta, B., "Implementing PUPS analogy in SOAR," Unpublished progress Note.

b:87 Faieta, B., "Analysis of learning to play GO using the Soar model," Unpublished final project.

c:87 Laird, J. E., & Rosenbloom, P. S., "Research on learning in Soar," *Proceedings of the Second Annual Artificial Intelligence Research Forum*, NASA Ames Research Center, Palo Alto, CA, 1987, pp. 240-253.

d:87 Lewis, C., "The search for cognitive harmony," *Contemporary Psychology*, Vol. 32, 1987, pp. 427-428, Review of Universal Subgoaling and Chunking: The Automatic Generation and Learning of Goal Hierarchies.

1988

a:88 Adelson, B., "Modeling software design in a problem-space architecture," *Proceedings of the Annual Conference of the Cognitive Science Society*, August 1988, pp. 174-180.

b:88 Crawford, A., "On Soar and PDP models: Learning in a probabilistic environment," Honors Thesis, Computer Science Department, Harvard University.

c:88 Flynn, R., "Restructuring Knowledge in Soar: Searching for the right paradigm-specific representations," Digital Equipment Corporation, September, 1988. Reprinted and available from Carnegie Mellon University. Unpublished.

d:88 Gupta, A. & Tambe, M., "Suitability of Message Passing Computers for Implementing Production Systems," *Proceedings of the Seventh National Conference on Artificial Intelligence*, AAAI, 1988, pp. 687-692.

e:88 Gupta, A., Tambe, M., Kalp, D., Forgy, C. L., and Newell, A., "Parallel implementation of OPS5 on the Encore Multiprocessor: Results and analysis," *International Journal of Parallel Programming*, Vol. 17, No. 2, 1988, pp. 95-124.

f:88 Hsu, W., Prietula, M., & Steier, D., "Merl-Soar: Applying Soar to scheduling," *Proceedings of the Workshop on Artificial Intelligence Simulation*, The National Conference on Artificial Intelligence, 1988, pp. 81-84.

g:88 Kalp, D., Tambe, M., Gupta, A., Forgy, C., Newell, A., Acharya, A., Milnes, B., & Swedlow, K., "Parallel OPS5 User's Manual," Tech. report CMU-CS-88-187, Computer Science Department, Carnegie Mellon University, November 1988.

h:88 Mostow, J., "Review of Universal Subgoaling and Chunking: The Automatic Generation and Learning of Goal Hierarchies," *American Scientist*, Vol. 76, 1988, pp. 410.

i:88 Reich, Y., "Learning Plans as a Weak Method for Design," Department of Civil Engineering, Carnegie Mellon University. Unpublished.

j:88 Reich, Y. & Fenves, S., "Integration of Generic Learning Tasks," Department of Civil Engineering, Carnegie Mellon University. Unpublished.

k:88 Reich, Y. & Fenves, S., "Floor-System Design in Soar: A Case Study of Learning to Learn," Tech. report EDRC-12-26-88, Engineering Design Research Center, Carnegie Mellon University, March 1988.

l:88 Rosenbloom, P. S. & Newell, A., "An integrated computational model of stimulus-response compatibility and practice," in *The Psychology of Learning and Motivation, Volume 21*, G. H. Bower, ed., Academic Press, 1988, pp. 1-52.

m:88 Tambe, M. & Newell, A., "Some chunks are expensive," *Proceedings of the Fifth International Conference on Machine Learning*, J. Laird,ed., 1988, pp. 451-458.

n:88 Tambe, M., & Newell, A., "Why Some Chunks are Expensive," Tech. report CMU-CS-88-103, Computer Science Department, Carnegie-Mellon University, 1988.

o:88 van Berkum, J. J. A., "Cognitive Modeling in Soar," Tech. report WR 88-01, Traffic Research Center, University of Groningen, July 1988.

p:88 Waldrop, M. M., "Toward a unified theory of cognition," *Science*, Vol. 241, 1988, pp. 27-29.

q:88 Waldrop, M. M., "Soar: A unified theory of cognition?," *Science*, Vol. 241, 1988, pp. 296-298.

1989

a:89 Acharya, A. & Tambe, M., "Efficient implementations of production systems," *VIVEK: A quarterly in artificial intelligence*, Vol. 2, No. 1, 1989, pp. 3-18.

b:89 Acharya, A. & Tambe, M., "Production systems on message passing computers: Simulation results and analysis," *Proceedings of the International Conference on Parallel Processing*, 1989, pp. 246-254.

c:89 Harvey, W., Kalp, D., Tambe, M., McKeown, D., & Newell, A., "Measuring the Effectiveness of Task-Level Parallelism for High-Level Vision," Tech. report CMU-CS-89-125, School of Computer Science, Carnegie Mellon University, March 1989.

d:89 Hsu, W.L., Newell, A., Prietula, M.J., & Steier, D.M., "Sharing scheduling knowledge between intelligent agents (Extended abstract)," *Proceedings of the AAAI-SIGMAN Workshop on Manufacturing Scheduling*, Eleventh International Joint Conference on Artificial Intelligence, 1989.

e:89 Hsu, W., Prietula, M., & Steier, D., "Merl-Soar: Scheduling within a general architecture for intelligence," *Proceedings of the Third International Conference on Expert Systems and the Leading Edge in Production and Operations Management*, May 1989, pp. 467-481.

f:89 Hucka, M., "Planning, Interruptability, and Learning in Soar," Department of Electrical Engineering and Computer Science, University of Michigan. Unpublished.

g:89 Huffman, S. B., "A natural-language system for interaction with problem-solving domains: Extensions to NL-Soar," Unpublished report on directed-study research.

h:89 Johnson, T. R., Smith, J. W. Jr., & Chandrasekaran, B., "Generic Tasks and Soar," *Working Notes of the AAAI Spring Symposium on Knowledge System Development Tools and Languages*, Stanford, CA, 1989, pp. 25-28.

i:89 Laird, J. E., Yager, E. S., Tuck, C. M., & Hucka, M., "Learning in tele-autonomous systems using Soar," *Proceedings of the NASA Conference on Space Telerobotics*, Pasadena, CA, 1989, pp. 415-424.

j:89 Newell, A, "The quest for architectures for integrated intelligent systems (Extended abstract)," Presented at the Eleventh International Joint Conference on Artificial Intelligence.

k:89 Steier, D.M., Automating Algorithm Design Within an Architecture for General Intelligence, PhD dissertation, School of Computer Science, Carnegie Mellon University, March 1989.

l:89 Tambe, M. & Acharya, A., "Parallel implementations of production systems," *VIVEK: A quarterly in artificial intelligence*, Vol. 2, No. 2, 1989, pp. 3-22.

m:89 Tambe, M., Acharya, A., & Gupta, A., "Implementation of Production Systems on Message Passing Computers: Techniques, Simulation Results and Analysis," Tech. report CMU-CS-89-129, School of Computer Science, Carnegie Mellon University, April 1989.

n:89 Tambe, M. & Rosenbloom, P. S., "Eliminating expensive chunks by restricting expressiveness," *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, IJCAII, 1989, pp. 731-737.

o:89 Wu, H. J., "Coherenece and ambiguity resolution," Artificial Intelligence Laboratory, The University of Michigan. Unpublished.

1990

a:90 Johnson, K.A., Smith, J.A., & Smith, P., "Design of a blood bank tutor," *Proceedings of the AAAI Mini-Symposium on Artificial Intelligence in Medicine*, March 1990, pp. 102-104.

b:90 Laird, J. E., Congdon, C. B., Altmann, E., & Swedlow, K. R., "Soar User's Manual: Version 5.2," Tech. report, Electrical Engineering and Computer Science, The University of Michigan, 1990, Also available from The Soar Project, School of Computer Science, Carnegie Mellon University, CMU-CS-90-179.

c:90 Letovsky, S., "Operationality Criteria for Recursive Predicates," *Proceedings of the Eighth National Conference on Artificial Intelligence*, AAAI, 1990, pp. 936-941.

d:90 Newell, A., *Unified Theories of Cognition*, Harvard University Press, Cambridge, Massachusetts, 1990.

e:90 Piersma, E. H., "Constructive Criticisms on Theory and Implementation: The Soar6 Computational Model," Tech. report WR 90-01, Traffic Research Center, University of Groningen, 1990.

f:90 Tambe, M., "Understanding Case-based Reasoning in Comparison with Soar: A preliminary report," School of Computer Science, Carnegie Mellon University. Unpublished.

g:90 Tambe, M. & Rosenbloom, P. S., "Production system formulations with non-combinatoric match: Some results and analysis," School of Computer Science, Carnegie Mellon University. Unpublished.

h:90 Wharton, C., & Lewis, C., "Soar and the Constructive-Integration Model: Pressing a Button in Two Cognitive Architectures," Tech. report CU-CS-466-90, Department of Computer Science, University of Colorado at Boulder, March 1990.

i:90 Wiesmeyer, M. & Laird, J., "A computer model of 2D visual attention," *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, Erlbaum, Hillsdale, NJ, 1990, pp. 582-589.

1991

a:91 Carley, K., Kjaer-Hansen, J., Prietula, M. & Newell, A., "Plural-Soar: A Prolegomenon to Artificial Agents and Organizational Behavior," Carnegie Mellon University and Joint Research Center, Ispra. Unpublished.

b:91 Carley, K. and Newell, A., "The Nature of the Social Agent." School of Computer Science and Department of Psychology, Carnegie Mellon University. Unpublished.

c:91 Churchill, E.F. & Young, R.M., "Modeling Representations of Device Knowledge in Soar," *AISB-91 Artificial Intelligence and Simulation of Behaviour*, Springer-Verlag, 1991, pp. 247-255.

d:91 Clancey, W. J., "The frame of reference problem in the design of intelligent machines," in *Architectures for Intelligence*, K. VanLehn, ed., Erlbaum, Hillsdale, NJ, 1991, pp. 357-423.

e:91 Covrigaru, A. A. & Lindsay, R. K., "Deterministic autonomous systems," *AI Magazine*, Vol. 12, 1991, pp. 110-117.

f:91 DeJongh, M., Causal Processes in the Problem Space Computational Model: Integrating Multiple Representations of Causal Processes in Abductive Problem Solving, PhD dissertation, Laboratory for Knowledge-Based Medical Systems, Ohio State University, 1991.

g:91 DeJongh, M. & Smith, J. W., Jr., "Integrating Models of a Domain for Problem Solving," *Proceedings of the Eleventh International Conference on Expert Systems and Their Applications: Second Generation Expert Systems*, 1991, pp. 125-135, Avignon, France.

h:91 Feldman, J. A., "Cognition as Search," *Science*, Vol. 251, 1991, pp. 575, Review of *Unified Theories of Cognition*.

i:91 Huffman, S.B. & Laird, J.E., "Perception, Projection and Reaction in External Environments in Soar: Initial Approaches," Tech. report CSE-TR-118-91, Department of Electrical Engineering and Computer Science, University of Michigan, 1991.

j:91 Johnson, T.R., Generic Tasks in the Problem-Space Paradigm: Building Flexible Knowledge Systems While Using Task-Level Constraints, PhD dissertation, Laboratory for Knowledge-Based Medical Systems, Ohio State University, 1991.

k:91 John, B.E., Remington, R.W. & Steier, D.M., "An Analysis of Space Shuttle Countdown Activities:

Preliminaries to a Computational Model of the NASA Test Director," Tech. report CMU-CS-91-138, School of Computer Science, Carnegie Mellon University, 1991.

l:91 Johnson, T. R. & Smith, J. W., "A framework for opportunistic abductive strategies," *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, Erlbaum, Hillsdale, NJ, 1991, pp. 760-764.

m:91 Laird, J. E., Yager, E. S., Hucka, M., & Tuck, C. M., "Robo-Soar: An integration of external interaction, planning and learning using Soar," *Robotics and Autonomous Systems*, Vol. 8, 1991, pp. 113-129.

n:91 Lehman, J. F., Lewis, R.L. & Newell, A., "Natural Language Comprehension in Soar: Spring 1991," Tech. report CMU-CS-91-117, School of Computer Science, Carnegie Mellon University, 1991.

o:91 Lindsay, R., "What is Cognitive Science," *Psychological Science*, Vol. 2, 1991, pp. 228-311, Review of Unified Theories of Cognition.

p:91 Newell, A. & Steier, D., "Intelligent Control of External Software Systems," Tech. report EDRC 05-55-91, Engineering Design Research Center, Carnegie Mellon University, 1991.

q:91 Pirolli, P. & Berger, D., "The Structure of Ill-structured Problem Solving in Instructional Design," Tech. report DPS-5, University of California at Berkeley, 1991.

r:91 Pylyshyn, Z. W., "The role of cognitive architectures in theories of cognition," in *Architectures for Intelligence*, K. VanLehn, ed., Erlbaum, Hillsdale, NJ, 1991, pp. 189-223.

s:91 Ritter, F.E., "TAQL-mode Manual," The Soar Project, School of Computer Science, Carnegie Mellon University. Unpublished.

t:91 Ritter, F.E., Hucka, M., & McGinnis, T.F., "Soar-mode Manual," The Soar Project, School of Computer Science, Carnegie Mellon University. Unpublished.

u:91 Rosenbloom, P. S., "Climbing the hill of cognitive-science theory," *Psychological Science*, Vol. 2, 1991, pp. 308-311, Invited Commentary.

v:91 Tambe, M., Eliminating Combinatorics from Production Match, PhD dissertation, School of Computer Science, Carnegie Mellon University, 1991.

w:91 Tambe, M., Kalp, D. & Rosenbloom, P., "Uni-Rete: Specializing the Rete Match Algorithm for the Unique-attribute Representation," Tech. report CMU-CS-91-180, School of Computer Science, Carnegie Mellon University, 1991.

x:91 Ward, B., ET-Soar: Toward an ITS for Theory-Based Representations, PhD dissertation, School of Computer Science, Carnegie Mellon University, 1991.

y:91 Yost, G.R. & Altmann, E., "TAQL 3.1.3: Soar Task Acquisition Language User Manual." School of Computer Science, Carnegie Mellon University. Unpublished.

1992

a:92 Aasman, J. & Akyürek, A., "Flattening goal hierarchies," in *Soar: A Cognitive Architecture in Perspective*, Michon, J. A. & Akyürek, A., eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992, pp. 199-217.

b:92 Aasman, J. & Michon, J. A., "Multitasking in driving," in *Soar: A Cognitive Architecture in Perspective*, Michon, J. A. & Akyürek, A., eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992, pp. 169-198.

c:92 Akyürek, A., "On a Computational Model of Human Planning," in *Soar: A Cognitive Architecture in Perspective, Michon*, J. A. & Akyürek, A., eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992, pp. 81-108.

d:92 Akyürek, A., "Means-Ends Planning: An Example Soar System," in *Soar: A Cognitive Architecture in Perspective*, Michon, J. A. & Akyürek, A., eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992, pp. 109-167.

e:92 Doorenbos, B., Tambe, M., & Newell, A., "Learning 10,000 chunks: What's it like out there?," *Proceedings of the Tenth National Conference on Artificial Intelligence*, AAAI, 1992, pp. 830-836.

f:92 Huffman, S.B., Pearson, D.J. & Laird, J.E., "Correcting imperfect domain theories: A knowledge-level analysis," in *Machine Learning: Induction, Analogy and Discovery*, Chipman, S. & Meyrowitz, A., eds., Kluwer Academic Publishers, Hingham, MA, 1992, In press.

g:92 Johnson, T. R., Smith, J. W., Johnson, K., Amra, N., & DeJongh, M., "Diagrammatic Reasoning of Tabular Data," Tech. report OSU- LKBMS-92-101, Laboratory for Knowledge-based Medical Systems, Ohio State University, 1992.

h:92 Michon, J. A. & Akyürek, A., editors, *Soar: A Cognitive Architecture in Perspective*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992.

i:92 Miller, C.S. & Laird, J.E., "A simple, symbolic model for associative learning and retrieval," Artificial Intelligence Laboratory, The University of Michigan. Unpublished.

j:92 Newell, A., "Unified theories of cognition and the role of Soar," in *Soar: A Cognitive Architecture in Perspective*, Michon, J. A. & Akyürek, A., eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992, pp. 25-79.

k:92 Peck, V.A. & John, B.E., "Browser-Soar: A computational model of a highly interactive task," *Proceedings of CHI'92*, ACM Press, New York, 1992, pp. 165-172.

l:92 Ritter, F.E. & McGinnis, T.F., "SX: A manual for the Soar graphical display and interface in X windows," The Soar Project, School of Computer Science, Carnegie Mellon University. Unpublished.

m:92 Rosenbloom, P. S., Lee, S., & Unruh, A., "Bias in planning and explanation-based learning," in *Machine Learning Methods for Planning*, S. Minton, ed., Morgan Kaufmann, San Mateo, CA, 1992, In press.

n:92 Rosenbloom, P. S. & Newell, A., "Symbolic Architectures: Organization of Intelligence I," in *Exploring Brain Functions: Models in Neuroscience*, T. Poggio & D. A. Glaser, eds., John Wiley & Sons, Chichester, England, 1992, In press.

o:92 Smith, J.W. & Johnson, T.R., "Stratified Knowledge System Design and Implementation: The Soar Approach," *IEEE Expert*, 1992, In press.

p:92 Wiesmeyer, M., An Operator-Based Model of Human Covert Visual Attention, PhD dissertation, Department of Electrical Engineering and Computer Science, The University of Michigan, 1992.