Foundations of INTELLIGENT TUTORING SYSTEMS

edited by

Martha C. Polson
J. Jeffrey Richardson

LEA

Foundations of INTELLIGENT TUTORING SYSTEMS

Edited by

Martha C. Polson
J. Jeffrey Richardson
University of Colorado

Advisory Editor

Elliot Soloway
Artificial Intelligence Project
Yale University

LEA
LAWRENCE ERLBAUM ASSOCIATES PUBLISHERS
1988 Hillsdale, New Jersey Hove and London

Lawrence Erlbaum Associates, Inc., Publishers
365 Broadway
Hillsdale, New Jersey 07642

Production and interior design: Lori L. Baronian
Cover design: Beth Ann Schmalz and Lori L. Baronian

Library of Congress Cataloging-in-Publication Data

Foundations of intelligent tutoring systems / edited by Martha C. Polson, J. Jeffrey Richardson.
p. cm.
Bibliography: p.
Includes index.
Contents: Foundations of intelligent tutoring systems / Hugh L. Burns and Charles G. Capps—The expert module / John R. Anderson—The student modeling module / Kurt VanLehn—Curriculum and instruction in automated tutors / Henry M. Halff—The environment module of intelligent ttuoring systems / Richard R. Burton—The role of human-computer interaction in intelligent tutoring systems / James R. Miller—Pragmatic considerations in research, development, and implementation of intelligent tutoring systems / William B. Johnson—Evaluating ITS / David C. Littman and Elliot M. Soloway—Directions for research and applications / J. Jeffrey Richardson.
ISBN 0-8058-0053-0. ISBN 0-8058-0054-9 (pbk.)
1. Computer-assisted instruction. I. Polson, Martha C. (Martha Campbell) II. Richardson, J. Jeffrey.
LB1028.5F634 1988
371.3'9445—dc19 87-22455
CIP

Printed in the United States of America
10 9 8 7 6 5 4

Contents

Contributors vii
Preface xi

1

Foundations of Intelligent Tutoring Systems: An Introduction 1
Hugh L. Burns and Charles G. Capps

The Expert Module, *2*
The Student Diagnosis Module, *5*
The Curriculum and Instruction Module, *7*
The Instructional Environment, *9*
The Human–Computer Interface, *11*
ITS Pragmatics, *13*
ITS Evaluation, *15*
Toward Knowledge-Based Educational Systems, *17*
References, *18*

2

The Expert Module 21
John R. Anderson

Relationship of Expert Modules to Expert Systems, *23*
Black Box Models, *26*
Glass Box Expert Systems, *29*
Cognitive Models, *34*
Procedural Knowledge, *35*

Declarative Knowledge, *40*
Qualitative Process Models, *45*
Basic Research Issues, *48*
Near-Term Goals, *49*
References, *50*

3

Student Modeling **55**
Kurt VanLehn

The Student Modeling Problem, *56*
A Three-Dimensional Space of Student Models, *57*
Diagnostic Techniques, *64*
Research Issues, *74*
References, *76*

4

Curriculum and Instruction in Automated Tutors **79**
Henry M. Halff

Curriculum, *81*
Instruction, *89*
Research and Practice in Automated Tutors, *96*
Conclusion, *105*
References, *106*

5

The Environment Module of Intelligent Tutoring Systems **109**
Richard R. Burton

Introduction to the Environment Module, *109*
Issues of Instructional Environments, *115*
Tools for Building Instructional Environments, *129*
Future Technology and Instructional Environments, *136*
Research Opportunities in Instructional Environments, *137*
References, *139*

6

The Role of Human-Computer Interaction in Intelligent Tutoring Systems 143

James R. Miller

Introduction, *143*

Human-Computer Interaction: The State of the Art, *145*

Going Beyond Interface Technology: Toward Interface Semantics, *168*

Where Can We Begin?, *181*

References, *184*

7

Pragmatic Considerations in Research, Development, and Implementation of Intelligent Tutoring Systems 191

William B. Johnson

Introduction, *191*

Existing Intelligent Tutoring Systems for Training, *193*

Designing the System to Meet User Needs, *195*

Issues in Implementing ITSs, *199*

ITS Applications and Research, *203*

Summary, *206*

References, *206*

8

Evaluating ITSs: The Cognitive Science Perspective 209

David Littman and Elliot Soloway

Introduction, *209*

External Evaluation: The Cognitive Perspective, *211*

Internal Evaluation: The Architecture Perspective, *224*

Existing Evaluations Interpreted, *229*

Directions for Future Work, *234*

Conclusions, *241*

Acknowledgments, *241*

References, *241*

9

Directions for Research and Applications **243**
J. Jeffrey Richardson

Recommended Research, *243*
Recommended Applications, *248*
References, *252*

Appendix I: Selected Intelligent Tutoring Systems **253**

Glossary of ITS Terms **261**

Author Index **267**

Subject Index **273**

Contributors and Editors

John R. Anderson, ***Carnegie-Mellon University.*** Dr. Anderson is Professor of Psychology, Carnegie-Mellon University, holding the Walter Van Dyke Bingham Chair of Cognitive Science. Having received his Ph.D. in psychology from Stanford University in 1972, his current research is centered on ACT, a production-system capable of simulating intelligence behavior. His numerous research interests are now being brought together in a major new project at Carnegie-Mellon to develop intelligent computer-based tutors, with tutors for high-school geometry and introductory college LISP having been developed already.

Hugh L. Burns, ***United States Air Force.*** Hugh Burns, as a Lieutenant Colonel in the U.S. Air Force, serves as Chief, Intelligent Systems Branch, Air Force Human Resources Laboratory (AFHRL) Brooks AFB, San Antonio, Texas. He earned his Ph.D. at the University of Texas at Austin investigating rhetorical theory, computer-assisted instruction, and instructional discourse. Lt. Col. Burns participated in the Air Force's Project Forecast II-A "Think Tank" to identify the critical technologies for the next generation of Air Force needs and capabilities.

Richard R. Burton, ***Xerox Palo Alto Research Center.*** Dr. Burton is a Member of the Research Staff of the Xerox Palo Alto Research Center, working in the Intelligent Systems Laboratory. Dr. Burton's long involvement in intelligent tutoring systems began prior to the receipt of his Ph.D. in information and computer science from the University of California, Irvine, in 1976. He was collaborator on three foundational intelligent tutoring systems: BUGGY, WEST, and SOPHIE.

Charles G. Capps, ***United States Air Force.*** Charles Capps is a Lieutenant assigned to the Air Force Human Resources Laboratory (AFHRL) at Brooks AFB, Texas. He received his undergraduate degree from the University of Alabama and his Master's degree in clinical psychology at Michigan State University. Since being assigned to the Intelligence Systems branch at AFHRL, his research activities have focused on projects investigating student performance and instructional expertise within intelligent instructional systems.

Henry M. Halff, ***Halff Resources, Inc.*** Dr. Halff is well known in scientific, military, and government communities for his management of research programs in the rapidly advancing fields of educational technology and cognitive science, work performed as scientific officer of the Office of Naval Research. Recently, Dr. Halff founded Halff Resources, where he is currently involved in applications of artificial intelligence to computer-aided instruction. Dr. Halff received his Ph.D. in psychology in 1969 from the University of Texas.

William B. Johnson, ***Search Technology, Inc.*** Dr. Johnson is Senior Scientist, Search Technology, Inc., where he manages training research for the military, nuclear power, and aviation industries. Dr. Johnson is currently the knowledge engineer for the Office of Naval Research project on intelligent maintenance training. Holding a Ph.D. in education, 1980, from the University of Illinois, Dr. Johnson is also a certified private pilot, airframe and powerplant mechanic, and designated mechanical engineer. He served as consultant to the 1985 National Academy of Science Air Force Summer Study on Fault Isolation in Air Force Weapons and Support Systems. He is an expert in simulation-oriented computer-based instruction for fault diagnosis.

David C. Littman, ***Yale University.*** Dr. Littman is currently a doctoral candidate in the Artificial Intelligence Program of the Department of Computer Science. Dr. Littman received his first Ph.D. in Experimental Psychology from Cornell University in 1976. Dr. Littman has been involved in evaluating educational technology since 1977. His main interests are intelligent tutoring systems and artificial intelligence approaches to scientific reasoning.

James R. Miller, ***Microelectronics Computer Corporation.*** Dr. Miller is team leader of the MCC Human Interface Program where question answering and coaching are being investigated in order to enable the computer user and system to cooperate in achieving tasks stated by the user. Prior to joining MCC, Dr. Miller was Director of Sponsored Research, Computer-Thought Corporation, where he was responsible for the student model, on-line documentation system, and user interface of a knowledge-based training system for the Ada programming language. Dr. Miller holds a Ph.D. in Psychology from the University of California, Los Angeles, granted in 1978.

Martha Campbell Polson, *University of Colorado.* Dr. Polson is Assistant Director of the Institute of Cognitive Science at the University of Colorado, Boulder. Since receiving her Ph.D. from the University of Indiana in 1968, her research interests have spanned a number of areas. Prior to assuming her current position, she was a visiting scientist at the Training Systems Division of the Air Force Human Resources Laboratory where she became involved in the intelligent tutoring systems program and this project.

J. Jeffrey Richardson, *University of Colorado.* Dr. Richardson is Executive Director of the Center for Applied Artificial Intelligence, College of Business, University of Colorado. This center pursues AI applications in business, equipment maintenance, intelligent tutoring systems, and operations research. Dr. Richardson also heads the Colorado Institute for Artificial Intelligence, a state-wide university-industry consortium dedicated to enhancing the technology and human resources foundation for AI in Colorado. His principle research interests include AI applications in equipment maintenance and in training. Dr. Richardson received his Ph.D. in education from the University of Colorado in 1981 and worked at Denver Research Institute prior to moving to the University of Colorado.

Elliot M. Soloway, *Yale University.* Dr. Soloway is Associate Professor of computer science and psychology at Yale University where he is Director of the Cognition and Programming Project. Having received his Ph.D. in computer and information science at the University of Massachusetts in 1978, his work focuses on the cognitive modeling of computer programming. An important application of this is the development of intelligent tutorial systems for college-level programming instruction. Dr. Soloway is also Vice President of Computer-Teach, Inc., an educational software company.

Kurt VanLehn, *Carnegie-Mellon University.* Dr. VanLehn is Assistant Professor in the Department of Psychology at Carnegie-Mellon University, having recently moved there from his position as Research Associate at Xerox Palo Alto Research Center. Having received his Ph.D. in computer science from the Massachusetts Institute of Technology in 1983, Dr. VanLehn is a major contributor to the cognitive science literature pertaining to human skill acquisition, including pioneering work in acquiring procedural skills from lesson sequences and "felicity conditions" for human skill acquisition. Further contributions include the formulation, as a product of the BUGGY project, of repair theory: a generative theory of bugs in procedural skills.

Preface

This book provides a synthesis of the field of Intelligent Tutoring Systems (ITSs). It is neither a collection of project reports nor a survey of ITS systems, but rather a coordinated set of essays, written at the general foundational level, each treating an integral aspect of the field. Each essay defines its topic, its relationship to other topics, the state-of-the art, basic research issues, and near-term applications projects.

The genesis of this book is a planning process set into motion by the Air Force Human Resources Laboratory (AFHRL) to develop its research agenda in ITSs. Acknowledged leaders in the field were contacted and agreed to participate in this process. The authors, editors, and sponsors held a meeting and agreed on the logical organization of the field reflected in the chapters and on the assignment of ITS topics to each chapter. Outlines for each chapter were developed and presented in a workshop held by the sponsor. Based on feedback from the workshop, the outlines were refined and draft papers were written. These were circulated among the authors, editors, and sponsors, critiqued and revised. The revised papers were presented at the AFHRL Research Planning Forum for Intelligent Tutoring Systems, held September 3-4, 1986 in San Antonio, Texas. The papers presented in San Antonio form the chapters of this book, augmented by the introductory and concluding chapters. The findings in this book regarding research and applications opportunities provided input to AFHRL's ITS research agenda.

We would like to acknowledge Lt. Col. Hugh L. Burns, Chief of the Intelligent Systems Branch, Training Systems Division, AFHRL for his role in sponsoring this work. Thanks to Lt. Charles G. Capps of AFHRL and to Dr. Matthew J. Wayner, Director of the Division of Life Sciences, University of Texas at San Antonio and his assistant, Janie Ramos, for their support in organizing the meetings, workshop, and conference associated with this work. Funding was through the

Air Force Office of Scientific Research, Grant Number AFSOR-86-0144, monitored by Dr. Alfred R. Fregley. Special acknowledgments are due to Janet L. Grassia, who did the copyediting for this book, and Marjorie J. DeFries, who edited and managed the copyright releases for the figures and to Tania M. Sizer, who prepared the glossary.

Martha C. Polson
Institute for Cognitive Science
University of Colorado

J. Jeffrey Richardson
Center for Applied Artificial Intelligence
Graudate School of Business Administration
University of Colorado

1

Foundations of Intelligent Tutoring Systems: An Introduction

Hugh L. Burns
Charles G. Capps
United States Air Force

Artificial intelligence in education comes of age in systems now called "intelligent tutors," a step beyond traditional computer-assisted instruction. Computer-assisted instruction evolves toward intelligent tutoring systems (ITSs) by passing three tests of intelligence. First, the subject matter, or domain, must be "known" to the computer system well enough for this embedded expert to draw inferences or solve problems in the domain. Second, the system must be able to deduce a learner's approximation of that knowledge. Third, the tutorial strategy or pedagogy must be intelligent in that the "instructor in the box" can implement strategies to reduce the difference between expert and student performance. At the foundation of ITSs, therefore, one finds three special kinds of knowledge and problem-solving expertise programmed in a sophisticated instructional environment. This book examines these knowledge foundations—expert knowledge, student diagnostic knowledge, and the instructional or curricular knowledge—in detail. This book also describes (a) how these kinds of knowledge are embodied in computer-assisted instructional environments; (b) how these systems accrue the advantages of advanced computer interface technologies; (c) how ITSs will emerge in the real world of complex problem solving; and finally (d) how researchers must learn to evaluate the effectiveness and overall quality of these dynamic systems in a world where one day machine tutoring will be taken for granted.

The purpose of this chapter is to introduce the major research issues and development themes that the primary authors—John Anderson, Kurt VanLehn, Henry Halff, Richard Burton, James Miller, William Johnson, David Littman, and Elliott Soloway—explore and amplify. At the core of this book is a simple notion that an ITS has an anatomy (see Figure 1.1), an anatomy that creates convenient classifications of the research and development dimensions.

The expert module contains the domain knowledge. The student diagnostic module diagnoses what the student knows. The instructor module identifies which deficiencies in knowledge to focus on and selects strategies to present that knowledge. The instructional environment and human-computer interface channel tutorial communication. In addition to these components, implementation and evaluation issues are most important. When, where, and how should these ITSs be used? How effective is the ITS and how is its quality understood? ITSs are hard to design and the field requires further study. Consequently, as the research community moves toward more and better ITSs, the need for integration of the "distinct" modules should be obvious. It should come as no surprise that in a complex, knowledge-based, problem-solving, computer-assisted tutoring system, the whole necessarily becomes more than the sum of its parts.

THE EXPERT MODULE

John Anderson, whose current research is in the architecture of cognition and in production systems capable of simulating intelligent human behavior, identifies the concepts and challenges of designing the expert module, that part of a tutor which provides the domain knowledge. The major lesson that the artificial intelligence community has learned from all of the research in expert systems is that any expert module must have an abundance of specific and detailed knowledge derived from people who have years of experience in a particular domain. Consequently, much effort is expended in discovering and codifying the domain knowledge, thus distilling years of experience into a knowledge representation. The enormous amount of knowledge in complex domains as well as the interrelationship of that knowledge means that designing and developing the expert module may be the most demanding chore in building an ITS. Authoring systems for intelligent tutors, alone, are unlikely to discover and codify all of the necessary domain knowledge. Thus, investigating how to encode knowledge and how to represent such expertise in an ITS remains the central focus of developing an expert module.

How does a research team explicitly go about encoding the knowledge in the ITS data structure? Three approaches are common, each moving toward a more cognitively faithful representation of the content expertise. The first is finding a way to encode the knowledge

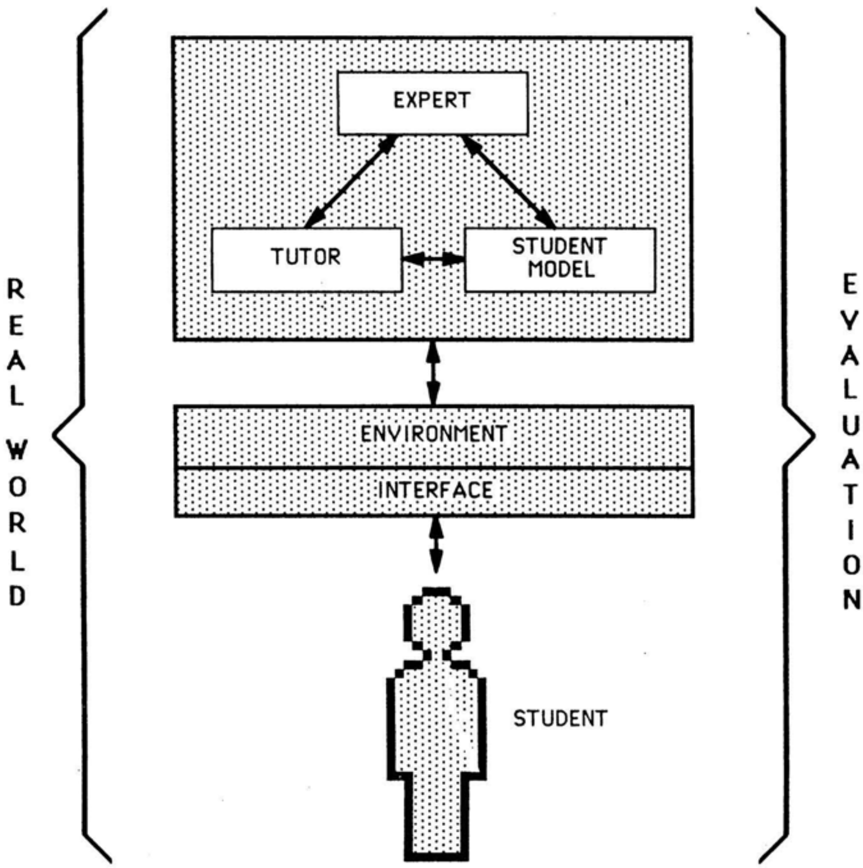

FIGURE 1.1

without actually codifying the underlying human intelligence. The literature often refers to these as "black box" expert systems. The simple input-output information available from a black box system is not suitable for instruction. One method of enhancing these models is to employ a methodology called "issue-based tutoring" (Burton & Brown, 1982). In other words, a programmer attaches instructions to specific issues observable in the behavior of both the expert and the student within the learning environment. Thus, when a student chooses (or fails to choose) a behavior, he or she may receive feedback about the particular behavior. The examples in the artificial intelligence canon include systems that use a mathematical equation-solving process in place of the symbolic human processing. SOPHIE (which stands for Sophisticated Instructional Environment) and

Steamer (Brown, Burton, & deKleer, 1982; Hollan, Hutchins, & Weitzman, 1984) perform their calculations through such techniques. Although the architectures of these systems have not represented human knowledge, they do produce outputs that are useful in recognizing differences between student and expert performance.

The second approach involves the building of a "glass box model" to influence the tutorial mechanisms of the system. To do this, a researcher must use knowledge-engineering techniques. A knowledge engineer interviews an expert and designs a computational representation for delivering the knowledge, usually a rule-based formalism. This implementation does not necessarily correspond to the way the human expert reasons, especially in novel, unfamiliar situations or when providing explanations. Thus, this glass box model allows only for explanations of the information process inherent in the rules of its knowledge base. The rules are typically more strategically aligned with performance rather than explanation, limiting their utility in an instructional setting. However, knowledge-engineering tools and techniques, that is, ways of extracting and codifying information, are becoming more and more useful for ITS development as attention is paid toward making representations more faithful to the breadth and depth of expert reasoning.

Nevertheless, because so much effort is expended in the knowledge acquisition process, turning preexisting expert systems into ITSs is a fond ambition. Clancey's (1982) GUIDON tried to implement MYCIN (Buchanan & Shortliffe, 1984)—an expert system for diagnosing bacterial infections—as the expert model within an ITS. MYCIN's representation of knowledge was highly "compiled." By analogy, a computer program's source code (high-level programming instructions written in languages such as FORTRAN, PASCAL, or COBOL) is compiled into object code (the primitive hardware instructions that the computer responds to). The source code is relatively easy to read, but not executable by a computer. The object code is "machine readable"—the computer can run it, but it is extremely difficult for people to understand. Extending the analogy even further, the "readability" of the source code itself depends on the extent to which the programmer followed structured programming practices. Similarly, the "readability," or utility in explanation, of a knowledge base depends on the principledness of the knowledge engineer's approach to representing the domain knowledge in the rule base (Clancey, 1981). The more principled and well-structured, the better the expert system serves for explanation and instruction. Clancey's research illustrates how limited expert systems can be in instructional settings.

The third approach to encoding the domain knowledge simulates not only the knowledge but also the way a human uses that knowledge. Here, in this area of cognitive modeling, the cognitive science community sees the greatest payoff for the design and development of ITSs. If the goal of cognitive modeling is to develop as realistic a simulation of human problem-solving processes as possible, many research questions must be answered. These questions include: (a) which psychological components are essential for tutoring, (b) at what level should they be represented, and (c) how should different types of knowledge be treated—procedural, declarative, and qualitative. One way of classifying the psychological components is according to problem-solving models as articulated by Newell and Simon (1972), among others. These problem-solving models should have the highest cognitive fidelity, that is, correspondence to an actual human through processes, as possible.

Because the three types of knowledge dictate the strategies of instruction, they need clear definition. Procedural knowledge is knowledge about how to perform a task and is well represented in the literature on expert systems as rule-based, production systems. Many artificial intelligence researchers believe that production rules with their recognize-act cycle capture the basic character of human cognition and, consequently, offer exciting possibilities for ITSs. Declarative knowledge contrasts with procedural knowledge in that it is fact-like, not specialized for a particular use. Finally, qualitative knowledge is the causal understanding that allows a human to reason about behavior using mental models of systems. One of the most challenging issues will be constructing a metatheory that unifies and shows the relationships between procedural, declarative, and qualitative knowledge. John Anderson concludes that research investigating expert modules for tutoring systems will be a unique test of the sufficiency of cognitive theories. Conversely, the design of an ITS will contribute to the discovery of more accurate theories of cognition.

THE STUDENT DIAGNOSIS MODULE

Kurt VanLehn describes the essential problems of student modeling in ITSs. Many ITSs infer a model of the student's current understanding of the subject matter and then use that understanding to adapt the instruction to the student's particular needs. The knowledge structure that depicts the student's current state is the student model, and the reasoning process to develop it is called "student diagnosis." Outputs from student diagnostic modules can be used for a variety of purposes, such as advancing through selected curriculums, coaching or offering unsolicited advice, generating new problems, and adapting sets of explanations. VanLehn describes the research issues in terms of three dimensions and discusses the need for: (a) improving the bandwidth of available knowledge about the student, (b) distinctly identifying types of knowledge to be learned, and (c) assessing differences between students and experts.

How much of the learner's activity is available to the diagnostic program? This is the bandwidth question, according to VanLehn. Most programs work on the low end of the information band where only the final state, that is, the student's answer to a question, is available to the system. Access to an intermediate state allows the diagnostic module to assess the observable physical activity, for example, key strokes or scratch work. The bandwidth of potentially the greatest value allows ITSs access to the learner's mental state, step by step as reasoning proceeds. Because the student diagnostic module needs reliable knowledge about the learner's mental state, bandwidth is critical in designing ITSs.

The second dimension of the student diagnostic module is the target knowledge type. VanLehn classifies knowledge types into two types of procedural knowledge, flat and hierarchical, and declarative knowledge. Specialized strategies for using (or interpreting) knowledge are paired with each type of knowledge. This interpretation process is more difficult to implement for declarative knowledge than for procedural. The interpreter for hierarchical procedural knowledge is more difficult than for flat procedural knowledge. Because the difficulty of the student diagnostic process is closely related to the difficulty of the interpretation process, a flat procedural knowledge base makes the student modeling process the easiest, whereas a declarative knowledge base presents the most difficult student modeling problem.

Assessing differences between students and experts is the third dimension VanLehn discusses. In programming a student diagnostic module, most ITS designers use the same knowledge representation scheme as was used in the expert module so that the expert and student modules actually share the same knowledge base. This is called the overlay method of student modeling, where the student's knowledge is represented as a subset of the expert's. Hence, missing conceptions are represented, but not misconceptions.

The next level of complexity in student modeling is to represent misconceptions, erroneous and incorrect knowledge, as opposed to simply incomplete knowledge. In this approach, the overlay model is augmented by a bug library. Bugs, that is, misconceptions or misunderstandings, must typically be collected empirically, but can be generated computationally from the target procedure, as is done in repair theory (Brown & VanLehn, 1980). To reduce the empirical work required for obtaining an exhaustive set of bugs, bugs are sometimes generated from bug part libraries, where bug parts, fragments of production rule clauses, are assembled into bugs. This represents the highest degree of sophistication in student modeling. Success in this is critically tied to the bandwidth issue.

Designing the student diagnostic module is a high-risk venture and, consequently, presents a wide range of issues to be investigated. How detailed do the descriptions of the student's knowledge have to be? What models of learning can be designed as a superstructure for the diagnostic algorithms? How much should the artificial intelligence research community push expert systems technology toward ITS technology? Of course, a variety of studies of bandwidth, knowledge type, and student-expert differences could be executed. This research promises many useful outcomes.

THE CURRICULUM AND INSTRUCTION MODULE

Henry Halff describes how the instructional module and curriculum issues give form and meaning to ITS research and development as instructional systems. An ITS should have three tutoring characteristics: (a) controls over the representation of the instructional knowledge for selecting and sequencing the subject matter; (b) capabilities for responding to students' questions about instructional goals and content; and (c) strategies for determining when students need help and for delivering the appropriate help. The goal of the instructional module is to circumscribe the nature of teaching and to implement teaching as a solution to the educational communication problem. Separating instructional and content expertise—or the "dancer from the dance," as William Butler Yeats once wrote—is the challenge in designing the instructional module. Obviously, the types of knowledge and the nature of the learning process interrelate with the teaching act. Less obvious is the interaction between content specifics and instructional strategy.

Specific knowledge necessary for learning but not necessary for proficient performance is called "propaedeutics," or enabling knowledge. Often this kind of knowledge is not represented in designing a tutor, when the focus is more on the knowledge in the expert module. In such cases, the required instructional background knowledge often comes about as an afterthought, once the building process has begun. This is the danger that lies in building an expert system first, then enhancing that expertise with explanations or instructional sequences designed to foster an effective learning experience for the ITS user.

Mitigating against this problem somewhat, Halff contends that there are families of instructional knowledges which could transfer from one tutor to another, for example, diagnostic tutorial routines and simulation tutorial routines. Instructional knowledge routines should allow a student to relate theory to practice, to propose solutions, to develop more effective problem-solving strategies. They should also minimize the load on the student's working memory while new concepts are being internalized.

Thus, the instructional module should and can be more than just a by-product of the expert and student modules, and some instructional principles should be robust and explicit enough to generalize across domains. The available literature on instructional theory provides instructional methodologies and can help designers decide such questions as what information to present in what sequence.

If a lesson can be found in curriculum design, it is simply that the overall goals of a tutor must be clear and well communicated. To that end, an ITS must appropriately manage the content and size of the content, conveying that structure to a learner and insuring that the instructional goals are within the learner's reach.

Presentation techniques all depend on the instructional objective. Elicitation and explanation help lead learners to an understanding of facts and concepts. Case presentations and simulated entrapment induce learners to formulate rules and to understand relations. Exercises, drills, and examples allow learners to generalize from subskills to the performance of the full task. Seeing the required skills prepares the student for the real-life situation. All of these strategies should be encompassed in an ITS design when the instructional module is laid out. The instructional engine that propels the presentation, Halff contends, must be investigated more fully.

Achieving any dynamic flexibility at the instructional level requires designing specific instructional states and means of transitioning from one state to another. Here is where artificial intelligence techniques may be the most useful in the instructional module. Meno-tutor (Woolf & McDonald, 1985) is one example of an attempt to achieve this flexibility by manipulating 27 interrelated instructional states. The ITS community is thus articulating needs for meta-rules to accommodate this dynamic reformulation of the tutor at the instructional level.

Another challenge to artificial intelligence involves understanding instructional discourse. Such understanding, for example, would include strategies appropriately intervening in the course of a student's problem-solving activity. Intervention, on the one hand, allows the

ITS control of the tutorial process, but it is also important in keeping a learner on the right track by preventing inappropriate or incorrect learning. Beyond intervention, that is, offering advice, hints or guidance, other strategies are needed for answering questions and providing explanations. These kinds of abilities must be incorporated in the design of an adequate instructional module and depend on further progress in the artificial intelligence field of natural language comprehension. Attempts to use templates (Carbonell, 1970) or semantic networks (Brown et al., 1982) have been tried; however, a comprehensive theory of explanation that would make automation possible has yet to be proposed.

As computer-assisted instruction becomes more intelligent in itself and more intelligently used in the classroom, educators will contribute models for properly shaping an automated instructional process. The fields of instruction and curriculum design can supply guidelines for the general support of ITS design and specifications for developing tool kits for certain educational applications. However, many of the tougher issues of ITS design are, so far, beyond the reach of these guidelines and tools. Still lacking, Halff points out, are (a) the design principles that determine whether a deductive or inductive approach is taken for the ITS instructional module, (b) precise theories that account for instructional effectiveness, and (c) explicit instructional principles in particular domains. Recognizing these deficiencies, however, is a sign of real progress. The effort to construct an instructional module—that explicit computational model of an instructor—ought to unravel some of the pedagogical paradoxes in the human tutoring process. Instructional knowledge acquisition promises to be a rich area for research and development—both for theory and for practice.

THE INSTRUCTIONAL ENVIRONMENT

An instructional environment consists of those elements of an ITS that support what the learner is doing: situations, activities, and tools provided by the system to facilitate learning. Richard Burton explores the issues pertaining to the instructional environment by establishing a pedagogical foundation, by carefully examining some of the more successful "microworlds," and by presenting near-term and long-term research agendas.

The activities and tools presented to the learner in an ITS always reflect an underlying educational philosophy. The trend, as computers get faster and as ITS researchers and educators become more creative and clever, is clearly to create a more open, more robust, more fulfilling, and more effective educational experience. Several principles for building instructional environments have emerged from this trend. An instructional environment should prove that there is more in an ITS than meets the eye. It should foster constructive learning through activities—tools, games, worlds—designed to use students' prior knowledge and to present students with new information and experiences from which they can construct new knowledge. The environment should emphasize conceptual understanding, not rote procedures. It should attempt to connect in-school and outside-school knowledge. It should be designed so that students feel self-monitored, allowing effective learners to assume responsibility for their own learning. The environment should also be developed in the premise that education is a life-long pursuit. From such principles, the educational technology community generally believes that computerized instructional environments become self-contained worlds that can enhance and motivate learning—even if the environments themselves are not intelligent.

Among research considerations pertaining to instructional environments are: (a) levels of abstraction, (b) fidelity, (c) sequences, and (d) help routines. Level of abstraction means what features of the real world are represented in the design of the environment. Fidelity means how closely the simulated environment matches the real world. Important here are considerations of the different types of fidelity; for example, physical fidelity, display fidelity, mechanical fidelity, and cognitive fidelity. Sequences refers to the framework a designer constructs for learning complex skills. A learner progresses through a sequence of increasingly complex microworlds, each providing new challenges and new sets of achievable goals. By means of help routines the designer takes into account additional information learners may need for operating the ITS. But there are different degrees of help. For example, help tells a learner what to do. Assistance or active help actually does the task for the user. In addition to help and such active assistance are empowering tools, reactive help systems, modeling, and—finally—tutoring itself.

The several instructional environments Burton examines share a sophistication in educational design. Burton's own research in sophisticated instructional environments is well known in the intelligent tutoring heritage. In the electronic troubleshooting environment of SOPHIE I, the learner must find a fault in a broken piece of equipment. The tools are the measurement devices, which receive their commands in English. The instructional environment of SOPHIE provides circuit simulation, a natural language program, and routines for judging the adequacy of student actions and for offering advice. Foundational research like Burton's research on SOPHIE opened many doors for ITS designers.

Research opportunities in instructional environments exist in the near term and long term. Generally, Burton sees near-term opportunities in taking advantage of new technologies; the long-term focus of research will be more on basic scientific issues concerning human conceptual problems. In the near future, studies that investigate the power of various simulation kits or ITS design tools should give the research community several environments to explore. The next generation of LISP processing machines should also spur development of ITSs as well as experimental testing of various intelligent tutors. Instructional environments will be enhanced to take advantage of innovations in computer hardware—graphics chips, for example. The ITS design community should also make advances based on new technologies such as read-write optical disks, speech processing input/output, and faster parallel machines.

The long-term issues center on scientific assessments of the ways *environments are conceptualized* by *experts, learners,* and *instructors.* For example, what tradeoffs must be made among the various environmental properties? What are the stages of conceptualization in a problem-solving environment? How can an ITS use information that the environment provides? Do we need color graphics? animation? natural language processing? speech synthesis? Instructional environments must also support the transformation from incorrect concepts to correct ones. How should that be accomplished?

Additionally, the research community should carry out studies to articulate appropriate fidelity requirements and to identify meta-skills useful in dynamic, instructional environments. It will also be necessary to study environments to support the teaching of social skills as well as intellectual skills. Simulation kits provide several exciting possibilities. Medium-scale testing of these in the classroom environment will also be necessary. Empowering tools that enable learners to design more explicit problem-solving settings for themselves should provide some exciting research.

Creativity and cleverness mark the design of the few environments that have been expressly designed for ITSs; creativity and cleverness will continue to be well exercised in the design and development of future intelligent tutors. But success in building instructional environments will largely depend on how well designed the ITS's human-computer interface is.

THE HUMAN-COMPUTER INTERFACE

When considering human-computer interactions in ITSs, James Miller emphasizes making appropriate tradeoffs in the design of ITS interfaces. The learner working with an ITS generally has two problems. First, the learner must learn some subject matter that he or she may not understand. Why else would an ITS be used? The other problem is that the learner must use the technology itself in order to learn and is very likely not an expert user. If the human-computer interaction is poorly designed, a training session will probably be ineffective. Simply put, if the learner has to spend significant intellectual energy working the computer, then the learner has less intellectual and emotional energy for learning what is supposedly being taught.

The goal of interface design, therefore, is to make the interface transparent. The research community is beginning to think of the human-computer interaction as a communication problem and to design this interaction as a system of semantic and contextual processes built on a solid conceptual model. The knowledge embedded in this component of an ITS thus evolves from knowledge of previous computer systems, from human interface research, from the real world objects that are being imitated in the computer system, and from knowledge of the entire range of the communication process—perceiving, understanding, and creating meaning.

The state of the art in interface research and development, Miller points out, allows for two basic styles of design. The first allows users to become direct participants in the domain; the second allows them to control the domain by instructing the system to carry out desired actions. First-person interfaces, or direct manipulation interfaces as they are sometimes called in the literature, are familiar as the icons in the images of the Apple Macintosh personal computer. The soul of these interfaces is the icon whose manipulation is intended to map directly to a desired outcome. The breakthrough for this kind of interaction has been large bit-mapped displays and the mouse, a pointing and selecting device. One of the advantages of iconic interaction is that learners do not have to remember names of documents, commands, and so forth, because all of this information is intrinsically part of the icon data structure. The strength of the first-person interface is its self-evident properties; its weakness is extensibility. In the second-person interface, an ITS user commands the system. Command languages are fairly well understood and can powerfully interact with a system. The general thrust of endeavor in the intelligent tutoring community, however, will be to minimize research on new command languages and concentrate on more direct manipulation and interaction in the actual delivery of the tutor.

Where are the promising research opportunities for the interface design team in an ITS project? First, the overall goal is to make the domain semantics visible. Studies that illustrate ways of constructing models of complex domains with special support for learners' acquisition of these representations and for special recognition of learners' corresponding conceptual models should be especially valuable. Investigation of the various graphical techniques for presenting models also offers a large payoff, especially if the graphical models are linked to various stages of the conceptualization. This direction points research of interfaces toward a few of the issues pertinent to the instructional environment, for example, level of abstraction and fidelity.

Another interesting research issue will be developing tool kits for interface development. Such kits would include direct manipulation techniques, natural language interfaces, speech processing, videodisks, touch screen technologies, and combinations of these. How these technologies will evolve for intelligent systems users is difficult to predict. When Miller speculates about the arrival of tomorrow's technology—three-dimensional graphics, continuous speech recognition, mammoth displays—he doubts that it will be immediately clear how to use this technology wisely. Finally, although many of the interface technologies could help integrate the separate ITS modules, developers must still suit the content to the interface and the interface to the content. If the interface is overdone and calls attention to itself, then the communication between the student and the instructional system will be impaired.

ITS PRAGMATICS

William Johnson reminds developers of intelligent tutors that the day will come when their systems—built in the laboratories—must make the transition to the real world. This generates a number of pragmatic considerations. The individual modules, the environments, and the interfaces must be integrated into a working entity. An ITS must be used in its educational, technical, or industrial setting. If the jury is still out on the success and promise of the ITS as a mode of instruction, then answers to many implementation questions have not been forthcoming either. Certainly it is true that nothing is as practical as good theory; nevertheless, there are pragmatic issues beyond good theory. Who are the users? What are the expectations? How can intelligent tutors be effectively implemented? Suffice it to say that a person does not simply decide in a vacuum that an ITS is the most appropriate means of instruction for a given domain—certain practical matters must be considered.

The willingness of the sponsors and the users to adapt this technology is an important practical consideration. Support must be generated across several levels of the affected organization. Initially, someone must have a desire to implement the ITS. Whether it is to introduce a new curriculum, to improve an old one, or perhaps to supplement existing courses, someone in a position of authority must see a need to institute intelligent tutoring. Then, that person must provide the necessary support both throughout the full research and development cycle and during the implementation of the system.

Five considerations are crucial to determining whether implementation of an ITS is currently feasible. First, ITSs are ready for trial application but admittedly are in their infancy at this time. Second, the programming tools developed in artificial intelligence for knowledge engineering and intelligent authoring as well as the necessary "field" hardware are not sufficient. Third, hardware and software are constantly changing—and with increasing speed. Fourth, the demand for various personnel resources within the development team is quite drastic—subject matter experts, students, instructors, computer engineers, computer scientists, managers of advanced technology programs are all needed. Fifth, evaluation of intelligent tutoring, or for that matter any evaluation of artificial intelligence systems, is expensive in terms of both money and time.

This initial picture may seem bleak but it is important to note these deficiencies. ITS developers must present their research fairly. Researchers in artificial intelligence have good reason to avoid publicity hype; it damages the credibility of the entire scientific community. Naturally, implementation issues will change rapidly during the next few years, but this state of flux justifies bringing the systems to the demonstration stage. Until the scientists are more aware of the real-world demands placed upon ITSs, many of the limitations cannot be reduced. Research must therefore be conducted in appropriate educational or training settings. The emphasis should be on evaluating some phases of these emerging tutors in the real world and on measuring their effect on their intended users.

The demands placed upon ITS developers are extensive. However, the science is indeed in its early years, and as more systems are built and implemented, the skills of the designers will improve drastically. Obviously, the technology will be ever changing and the development of ITSs will continue to require the effort of knowledge engineers, subject matter experts, computer programmers, and specialists in the science of human factors. These people ensure the efficacy of a necessarily complex system. Johnson offers a word of advice: "Keep the need for active involvement of the domain expert throughout the research and development cycle." Many times, the domain expert may not feel or understand the need for some alien, automated teaching machine; yet much of the success of an entire ITS project depends on the cooperation between the domain expert and the knowledge engineer.

Providing valuable information is only one of the contributions domain experts make to an ITS project. They can also identify criteria for selecting appropriate ITS instructional objectives. Johnson presents several characteristics that help identify candidates for applications areas: (a) high flow of students, (b) low availability of instructors, (c) expensive real equipment, (d) remote site training, (e) unavailable real equipment, (f) high public visibility, (g) unsafe real equipment, (h) high recurrent training volume.

These criteria for selecting appropriate applications for tutoring suggest domains that are complex and technical by nature. For example, maintenance of expensive, dangerous, and sensitive equipment is an excellent proving ground for ITSs. The intelligent components of the system allow learners to explore the environments and use the information conveyed by their instructors. Military technical schools are also prime candidates for ITSs. No other organization can furnish the sheer numbers of students who are available for relatively short periods of time. Not only are the requirements for the trainees quite rigorous, but also the obligations and course demands felt by the instructors are overwhelming—the task being to teach students from a myriad of backgrounds the competencies and skills for a particular occupation. Training in personnel, procurement, logistics, and space operations are all important domains for demonstrating that artificial intelligence approaches to training are both needed and effective. Finally, industrial settings also present an excellent opportunity for testing ITSs in the real world, and especially for testing computer-assisted, on-the-job tools in which the "intelligent coach" is an embedded feature, not an independent entity.

Johnson's bottom line is simply this: If ITSs are to reach their promise, then the laboratory systems must operate and survive in the real world.

ITS EVALUATION

David Littman and Elliot Soloway describe what has emerged as a serious gap in the ITS literature—evaluation methods and quality control. Obviously, an intelligent tutor must be evaluated so that one knows how good it truly is, and the evaluator must be able to articulate why such systems are good or bad. To date, designers and evaluators have yet to establish guidelines for use in judging a system's worth. In fact, evaluation is the aspect of the intelligent tutoring methodology least written about. Science calls for empirical testing of systems, theories, and models. Intelligent tutors have not, for the most part, met this requirement of the experimental method.

Both formative and summative evaluations are important in evaluating instructional products. Since the instructional impact of an ITS, that is, its summative evaluation, is critically dependent on how well it was designed and built, Littman and Soloway properly place primary emphasis on formative evaluation and a strategy for formative evaluation specifically suited for ITSs.

Formative evaluations take place during the development of a system. As data are collected and feedback received, scientists make changes. This ongoing process can involve any of the modules in an ITS. Advice may come from the knowledge engineers, the subject matter experts, and from early trials with potential users of the system. The system engineer can circumvent bugs that would have occurred and anticipate other undesirable behavior throughout the program.

Littman and Soloway emphasize the need for developing a systemic approach to formative evaluation of ITSs, and they outline a two-part methodology for performing formative assessments. The first part, external evaluation, focuses on the impact of the ITS on students' problem-solving processes and is based on explicit models of how students solve problems. Student modeling techniques are used to identify the kinds of problems students should find hard to solve and easy to solve. An ITS can then be evaluated according to how well it teaches students the specific skills they need to solve problems. The effectiveness of the system is determined through measurement of observable phenomena that occur during the learning process. Instructional experts should be able to recognize these overt signs and determine whether or not the intended outcome was achieved. External evaluation can make rigorous testing possible. Because student modeling techniques capture how students solve problems, those techniques can be used to predict the ease or difficulty of additional problems and the knowledge necessary to solve them. The performance of students actually solving these problems can be compared to the predictions.

The second formative evaluation method, internal evaluation, addresses the question of why the ITS behaves as it does. It involves analysis of the architectural components of the ITS and the way these components respond to input values. Littman and Soloway recommend that their internal evaluation answer three questions. First, what does the ITS know? Second, how does the ITS do what it does? Third, what should the ITS do?

Littman and Soloway discuss many of the lessons their research group learned in applying ideas for external and internal evaluation to an ITS called PROUST, a LISP program that finds the nonsyntactic bugs in PASCAL programs written by students, and then tutors the students about these errors. During the external evaluation, the PROUST research group used its cognitive model of novice programming to determine whether PROUST helped students acquire programming skills. One conclusion drawn from that evaluation effort was that simply counting the number of answers a student got right or wrong did not provide a useful measure of the effects of PROUST on novice programmers. It appeared that a more fine-grained analysis was necessary. Therefore the evaluation focused on PROUST's impact on specific "micro" problem-solving skills, such as students' ability to determine whether a computer program is protected against certain invalid input data.

The kind of evaluation Littman and Soloway propose has some intriguing implications. One is that evaluation can help designers identify the kinds of reasoning capacities their tutorial systems must have. For example, the PROUST evaluation uncovered a need for the system to reason about how students name variables in their programs. Without this reasoning capacity, the program was unable to completely understand programs that humans find very easy to understand. When this capability was added to PROUST, its tutorial performance was noticeably improved. Thus, evaluation can have a very real impact on the design of tutorial systems.

TOWARD KNOWLEDGE-BASED EDUCATIONAL SYSTEMS

With more than 20 ITSs scattered throughout the literature (see Table 1.1), a well-understood technology for ITS development cannot be expected. More experience is needed and more ITSs need to be built in exploration of the possibilities. However, the education and training communities can expect high payoffs only when an ITS technology does formally emerge. So, more ITSs need to be built not only for exploration, but for determining a generalizable body of knowledge about how to build ITSs. This development will not be a simple task. What is clearly understood, however, is that such systems will require seven kinds of expertise, at least. This expertise pertains to the components that must be integrated as the foundation for ITSs:

1. content expertise in the expert module,
2. diagnostic expertise (determining what learners know and need to learn) in the student diagnostic module,
3. instructional and curriculum expertise in the instructor module,
4. expertise in creating instructional environments,
5. human-computer interface expertise,
6. implementation expertise, and
7. evaluation expertise.

These components comprise the anatomy of ITSs and together provide the educational community with a basic conceptual model for designing, developing, deploying, and evaluating machine tutors.

It is not easy to integrate all of this knowledge in a single delivery system. The hope of achieving, through artificial intelligence, a rich, interactive, flexible, real-time capacity to support learning is the basic motivation for research and development in ITSs. ITSs promise not only to help people learn how to perform complex tasks better, but also to reveal how people learn. This collection of essays examines the ITS's anatomy and proposes two things: (a) achievable ITS capabilities for the near term and (b) fundamental research questions that must be answered along the way toward more robust and effective, knowledge-based educational systems.

REFERENCES

Brown, J. S., Burton, R. R., & deKleer, J. (1982). Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II, and III. In D. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 227–282). New York: Academic Press.

Brown, J. S., & VanLehn, K. (1980). Repair theory: A generative theory of bugs in procedural skills. *Cognitive Science, 4,* 389–426.

Buchanan, B. G., & Shortliffe, E. H. (1984). *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project* (p. 576). Reading, PA: Addison-Wesley.

Burton, R. R., & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 79–98). New York: Academic Press.

Carbonell, J. R. (1970). AI in CAI: An artificial intelligence approach to computer-assisted instruction. *IEEE Transactions on Man–Machine Systems, 11,* 19–202.

Clancey, W. J. (1981). *The epistemology of a rule-based expert system: A framework for explanation* (Report No. STAN-CS-81-896). Stanford, CA: Department of Computer Science, Stanford University.

Clancey, W. J. (1982). Tutoring rules for guiding a case method dialogue. In D. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 201–225). New York: Academic Press.

Hollan, J. H., Hutchins, E. L., & Weitzman, L. (1984). Steamer: An interactive inspectable simulation-based training system. *AI Magazine, 5,* 15–27.

Newell, A., & Simon, H. (1972). *Human problem solving.* Englewood Cliffs, NJ: Prentice-Hall.

Woolf, B., & McDonald, D. D. (1985). Building a computer tutor: Design issues. *AEDS Monitor, 23,* 10–18.

Foundations of Intelligent Tutoring Systems: An Introduction

Brown, J. S. , Burton, R. R. , & deKleer, J. (1982). Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II, and III. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 227-282). New York: Academic Press.

Brown, J. S. , & VanLehn, K. (1980). Repair theory: A generative theory of bugs in procedural skills. Cognitive Science, 4, 389-426.

Buchanan, B. G. , & Shortliffe, E. H. (1984). Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project (p. 576). Reading, PA: Addison-Wesley.

Burton, R. R. , & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 79-98). New York: Academic Press.

Carbonell, J. R. (1970). AI in CAI: An artificial intelligence approach to computer-assisted instruction. IEEE Transactions on Man-Machine Systems, 11, 19-202.

Clancey, W. J. (1981). The epistemology of a rule-based expert systemml: A framework for explanation (Report No. STAN-CS-81-896). Stanford, CA: Department of Computer Science, Stanford University.

Clancey, W. J. (1982). Tutoring rules for guiding a case method dialogue. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 201-225). New York: Academic Press.

Hollan, J. H. , Hutchins, E. L. , & Weitzman, L. (1984). Steamer: An interactive inspectable simulation-based training system. AI Magazine, 5, 15-27.

Newell, A. , & Simon, H. (1972). Human problem solving. Englewood Cliffs, NJ: Prentice-Hall.

Woolf, B. , & McDonald, D. D. (1985). Building a computer tutor: Design issues. AEDS Monitor, 23, 10-18.

The Expert Module

Anderson, J. R. (1976). Language, memory, and thought. Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R. (1983). The architecture of cognition. Cambridge, MA: Harvard University Press.

Anderson, J. R. Analysis of student performance with the LISP tutor. In N. Fredericksen, R. Glaser, A. Lesgold, & M. Shafto (Eds.), Diagnostic monitoring of skill and knowledge acquisition. Hillsdale, NJ: Lawrence Erlbaum Associates, in press.

Anderson, J. R. (in press). Methodologies for studying human knowledge. Behavioral and Brain Sciences.

Anderson, J. R. , Boyle, C. F. , & Yost, G. (1985). The geometry tutor. In A. Joshi (Ed.), Proceedings of the Ninth International Joint Conference on Artificial Intelligence (pp. 1–7). Los Altos, CA: Morgan Kaufmann.

Anderson, J. R. , Corbett, A. T. , & Reiser, B. J. (1986). Essential LISP. Reading, MA: Addison-Wesley.

Barr, A. , Beard, M. , & Atkinson, R. C. (1975). Information networks for CAI curriculum. In O. Lecareme & R. Lewis (Eds.), Computers in education (pp. 477–482). Amsterdam: North Holland.

Bobrow, D. G. , & Winograd, T. (1977). An overview of KRL: A knowledge representation language. Cognitive Science, I, 3–46.

Bobrow, S. , & Bower, G. H. (1969). Comprehension and recall of sentences. Journal of Experimental Psychology, 80, 455–461.

Brackman, R. J. (1978). A structural paradigm for representing knowledge (Tech. Rep. 3605). Cambridge, MA: Bolt Beranek & Newman Inc.

Brown, J. S. , (1983). Process versus product: A perspective on tools for communal and informal electronic learning (Tech. Rep.). In Report from the learning lab: Education in the electronic age. Educational Broadcasting Corporation.

Brown, J. S. , & Burton, R. R. (1975). Multiple representation of knowledge for tutorial reasoning. In D. Bobrow & A. Collins (Eds.), Representation and understanding: Studies in cognitive science (pp. 311–349). New York: Academic Press.

Brown, J. S. , Burton, R. R. , & deKleer, J. (1982). Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 227–282). New York: Academic Press.

Brown, J. S. , & VanLehn, K. (1980). Repair theory: A generative theory of bugs in procedural skills. Cognitive Science, 4, 379–426.

Buchanan, B. , & Shortliffe, E. (1984). Rule-Based Expert Systems. Reading, MA: Addison-Wesley.

Burton, R. R. (1982). Diagnosing bugs in a simple procedural skill. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 157–183). New York: Academic Press.

Burton, R. R. , & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 79–98). New York: Academic Press.

Carbonell, J. R. (1970). AI in CAI: An artificial intelligence approach to computer-aided instruction. IEEE Transactions on Man-Machine Systems, 11, 190–202.

Clancey, W. J. (1982). Tutoring rules for guiding a case method dialogue. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 201–225). New York: Academic Press.

Collins, A. M. (1976). Processes in acquiring knowledge. In R. C. Anderson , R. Spiro , & W. E. Montague (Eds.), Schooling and the acquisition of knowledge (pp. 339–363). Hillsdale, NJ: Lawrence Erlbaum Associates.

Collins, A. M. & Quillian, M. R. (1972). Experiments on semantic memory and language comprehension. In L. Gregg (Ed.), Cognition in learning and memory (pp. 117–137). New York: Wiley.

Collins, A. M. , Warnock, E. H. , Aiello. N., & Miller, M. L. (1975). Reasoning from incomplete knowledge. In D. G. Bobrow & A. M. Collins (Eds.), Representation and understanding (pp. 383–415). New York: Academic Press.

Collins, A. , Warnock, E. , & Passaliume, J. (1975). Analysis and synthesis of tutorial dialogues. In G. Bower (Ed.), The psychology of learning and motivation (Vol. 9, pp. 49–87). New York: Academic Press.

deKleer, J. , & Brown, J. S. (1984). A physics based on confluences. AI Journal, 24, 7–83.

Forbus, K. D. (1984). Qualitative process theory. Artificial Intelligence, 24, 85–168.

Forgy, C. L. (1982). Rete: A fast algorithm for the many pattern/many object pattern match problem. Artificial Intelligence, 19, 17–37.

Genesereth, M. R. (1982). The role of plans in intelligent teaching systems. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 137–155). New York: Academic Press.

Goldstein, I. (1982). The genetic graph: A representation for the evolution of procedural knowledge. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 51–77). New York: Academic Press.

Goldstein, I. P. , & Roberts, R. B. (1977). NUDGE, a knowledge-based scheduling program. Proceedings of the Fifth International Joint Conference of Artificial Intelligence (pp. 257–263).
Hayes, J. R. (1985). Three problems in teaching general skills. In S. Chipman , J. Segal , & R. Glaser (Eds.), Thinking and learning skills (pp. 391–406). Hillsdale, NJ: Lawrence Erlbaum Associates.
Hayes-Roth, F. , Waterman, D. A. , & Lenat, D. B. (1983). Building expert systems. Reading, MA: Addison-Wesley.
Hollan, J. D. , Hutchins, E. L. , & Weitzman, L. (1984). Steamer: An interactive inspectable simulation-based training system. AI Magazine, 5, 15–27.
Holland, J. H. , Holyoak, K. , Nisbett, R. E. , & Thagard, P. R. (1986). Induction: Processes of inference, learning, and discovery. Cambridge, MA: Massachusetts Institute of Technology Press.
Kimball, R. (1982). A self-improving tutor for symbolic integration. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 283–307). New York: Academic Press.
Kuipers , B. (1984). Commonsense reasoning about causality: Deriving behavior from structure. Artificial Intelligence, 24, 169–203.
Laird, J. E. , Rosenbloom, P. S. , & Newell, A. (1986). Chunking in SOAR: The anatomy of a general learning mechanism. Machine Learning I, 11–46.
Langley , P. (1985). Learning to search: From weak methods to domain-specific heuristics. Cognitive Science, 9, 217–260.
Miller, M. L. (1982). A structured planning and debugging environment for elementary programming. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 119–135). New York: Academic Press.
Minsky, M. (1975). A framework for representing knowledge. In P. H. Winston (E.d.), The psychology of computer vision (pp. 211–277). New York: McGraw-Hill.
Nagel, L. W. , & Pederson, D. O. (1973). Simulation program with integrated circuit emphasis. Proceedings of the Sixteenth Midwest Symposium on Circuit Theory.
Newell, A. (1973). Production systems: Models of control structures. In W. G. Chase (Ed.), Visual information processing (pp. 463–526). New York: Academic Press.
Newell, A. , & Simon, H. (1972). Human problem solving, Englewood Cliffs, NJ: Prentice-Hall.
Reiser, B. J. , Anderson, J. R. , & Farrell, R. B. (1985). Dynamic student modeling in an intelligent tutor for LISP programming. Proceedings of the International Joint Conference on Artificial Intelligence-85 (Vol. 1, pp. 8–14). Los Altos, CA: Morgan Kaufmann.
Rouse, W. B. , & Morris, N. M. (1985). On looking into the black box: Prospects and limits in the search for mental models (Tech. Rep. No. 85–92). Atlanta: Georgia Institute of Technology.
Schank, R. C. , & Abelson, R. P. (1977). Scripts, plans, goals, and understanding. Hillsdale, NJ: Lawrence Erlbaum Associates.
Shortliffe, E. H. (1976). Computer-based medical consultations: MYCIN. New York: American Elsevier.
Sleeman, D. (1982). Assessing aspects of competence in basic algebra. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 185–199). New York: Academic Press.
Stefik, M. (1980). Planning with constraints (Tech. Rep. No. 784). Palo Alto, CA: Stanford University.
Stevens, A. , & Collins, A. M. (1977). The goal structure of a Socratic tutor. (Tech. Rep. No. 3518). Cambridge, MA: Bolt, Beranek, and Newman, Inc.
Stevens, A. , Collins, A. M. , & Goldin, S. E. (1982). Misconceptions in students' understanding. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 13–24). New York: Academic Press.

VanLehn, K. (1983). Felicity conditions for human skill acquisition: Validating an AI-based theory (Tech. Rep. CIS-21). Palo Alto, CA: Xerox Parc.
Wenger, E. (1987). Artificial intelligence and tutoring systems: Computational approaches to the communication of knowledge. Los Altos, CA: Morgan Kaufmann.
Westcourt, K. , Beard, M. , & Gould, L. (1977). Knowledge-based adaptive curriculum sequencing for CAI: Application of a network representation. Proceedings of the Association for Computing Machinery Annual Conference. Association for Computing Machinery (ACM-77) (pp. 234–240).
While, B. Y. , & Fredericksen, J. R. (1986). Progressions of qualitative models as foundations for intelligent learning environments (BBN Report 6277). Cambridge, MA: Bolt, Beranek, and Newman, Inc.
Zhang, G. (1986). Learning to program in OPS5. Unpublished doctoral dissertation. Pittsburgh, PA: Carnegie-Mellon University.

Student Modeling

Anderson, J. R. , Boyle, C. , & Yost, G. (1985). The geometry tutor. Proceedings of Ninth International Joint Conference on Artificial Intelligence (pp. 1–7). Los Altos, CA: Morgan Kaufmann.
Brown, J. S. , & VanLehn, K. (1980). Repair Theory: A generative theory of bugs in procedural skills. Cognitive Science, 4, 379–426.
Brown, J. S. , & Burton, R. B. (1978). Diagnostic models for procedural bugs in basic mathematical skills. Cognitive Science, 2, 155–192.
Brown, J. S. , Burton, R. B. , & deKleer, J. (1982). Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 227–282). New York: Academic Press.
Burton, R. B. (1982). DEBUGGY: Diagnosis of errors in basic mathematical skills. In D. H. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 157–183). New York: Academic Press.
Burton, R. B. , & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 79–98). New York: Academic Press.
Buswell, G. T. (1926). Diagnostic studies in arithmetic. Chicago, IL: University of Chicago Press.
Carbonell , R. (1970). AI in CAI: An artificial intelligence approach to computer aided instruction. IEEE Transactions on Man-Machine Systems, 11, 190–202.
Carbonell, J. R. , & Collins, A. (1973). Natural semantics in artificial intelligence. Proceedings of the Third International Joint Conference on Artificial Intelligence (pp. 344–351). Los Altos, CA: Morgan Kaufmann.
Clancey, W. J. (1982). Tutoring rules for guiding a case method dialogue. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 201–225). London: Academic Press.
Genesereth, M. R. (1982). The role of plans in intelligent teaching systems. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 137–155). New York: Academic Press.
Gentner, D. , & Stevens, A. (1984). Mental models. Hillsdale, NJ: Lawrence Erlbaum Associates.
Goldstein, I. (1982). The genetic graph: A representation for the evolution of procedural knowledge. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 51–77). New York: Academic Press.

Goldstein, I. , & Carr, B. (1977). The computer as coach: An athletic paradigm for intellectual education. Proceedings of ACM77 (pp. 227–233).
Grignetti, M. , Hausman, C. , & Gould, I. . (1975). An intelligent on-line assistant and tutor: NLS-Scholar. Proceedings of the National Computer Conference (pp. 775–781).
Hollan, J. D. , Hutchins, E. L. , & Weitzman, L. (1984). Steamer: An interactive inspectable simulation-based training system. The AI Magazine, 5(2), 15–27.
Johnson, L. , & Soloway, E. (1984a). Intention-based diagnosis of programming errors. Proceedings of American Association of Artificial Intelligence Conference (pp. 162–168). Los Altos, CA: Morgan Kaufmann.
Johnson, L. , & Soloway, E. (1984b). Proust: Knowledge-based program debugging. Proceedings of the Seventh International Software Engineering Conference (pp. 369–380).
Kimball, R. (1982). A self-improving tutor for symbolic integration. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 283–308). New York: Academic Press.
Langley, P. , & Ohlsson, S. (1984). Automated cognitive modeling. Proceedings of American Association of Artificial Intelligence (pp. 193–197). Los Altos, CA: Morgan Kaufman.
London, B. , & Clancey, W. J. (1982). Plan recognition strategies in student modeling: Prediction and description. Proceedings of the American Association of Artificial Intelligence Conference (pp. 193–197). Los Altos, CA: Morgan Kaufmann.
Miller, M. L. (1982). A structured planning and debugging environment for elementary programming. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 119–135). New York: Academic Press.
Ohlsson, S. , & Langley, P. (1985). Identifying solution paths in cognitive diagnosis (Tech. Rep. CMU-RI-TR-84-7). Pittsburgh, PA: Carnegie-Mellon University, Robotic Institute.
Reiser, B. V. , Anderson, J. R. , & Farrell, R. G. (1985). Dynamic student modeling in an intelligent tutor for lisp programming. Proceedings of Ninth International Joint Conference on Artificial Intelligence (pp. 8–14). Los Altos, CA: Morgan Kaufman.
Sleeman, D. H. (1982). Assessing competence in basic algebra. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 186–199). New York: Academic Press.
Stevens, A. , Collins, A. , & Goldin, S. E. (1982). Misconceptions in students' understanding. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 13–24). New York: Academic Press.
VanLehn, K. (1982). Bugs are not enough: Empirical studies of bugs, impasses and repairs in procedural skills. The Journal of Mathematical Behavior, 3(2), 3–71.
VanLehn, K. (1983). Felicity conditions for human skill acquisition: Validating an AI-based theory (Tech. Rep. CIS-21). Palo Alto, CA: Xerox Palo Alto Research Center.
VanLehn, K. (1987). Learning one subprocedure per lesson. Artificial intelligence, 31(1), 1–40.
VanLehn, K. , & Garlick, S. (1987). Cirrus: An automated protocol analysis tool. In P. Langley (Ed.), Proceedings of the Fourth Machine Learning Workshop. Los Altos, CA: Morgan Kaufmann.

Curriculum and Instruction in Automated Tutors

Anderson, J. R. (1983). The architecture of cognition. Cambridge, MA: Harvard University Press.

Anderson, J. R. , Boyle, C. F. , Corbett, A. , & Lewis, M. (1986). Cognitive modeling and intelligent tutoring (Tech. Rep. No. ONR-86-1). Pittsburgh, PA: Carnegie-Mellon University, Psychology Department.

Anderson, J. R. , Boyle, C. F. , & Reiser, B. J. (1985). Intelligent tutoring systems. Science, 228, 456–462.

Anderson, J. R. , Boyle, C. F. , & Yost, G. (1985). The geometry tutor. In A. Joshi (Ed.), Proceedings of the Ninth International Joint Conference on Artificial Intelligence (Vol. 1, pp. 1–7). Los Altos, CA: Morgan Kaufmann.

Ausubel, D. P. , (1968). Educational psychology: A cognitive view. New York: Holt, Rinehart & Winston.

Bitter, G. G. , Greenes, C. W. , Sobel, M. A. , et al. (1981). Third-grade mathematics.New York: McGraw-Hill.

Brown, J. S. , Burton, R. R. , & Bell, A. G. (1974). Sophie: A step towards a reactive learning environment. International Journal of Man-Machine Studies, 7, 675–696.

Brown, J. S. , Burton, R. R. , & deKleer, J. (1982). Pedagogical, natural language, and knowledge engineering techniques in SOPHIE I, II, and III. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 227–282). New York: Academic Press.

Burton, R. R. , & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 79–98). New York: Academic Press.

Carbonell, J. R. (1970). AI in CAI: An artificial intelligence approach to computer-aided instruction. IEEE Transactions on Man-Machine Systems. MMS-11, 190–202.

Clancey, W. J. (1984). Methodology for building an intelligent tutoring system. In W. Kintsch , J. R. Miller , & P. G. Polson (Eds.), Method and tactics in cognitive science (pp. 51–83). Hillsdale, NJ: Lawrence Erlbaum Associates.

Clancey, W. J. , & Letsinger, R. (1984). NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching. In W. J. Clancey & E. H. Shortliffe (Eds.), Readings in Medical Artificial Intelligence: The First Decade (pp. 361–381). Reading, MA: Addison-Wesley.

Collins, A. , & Stevens, A. L. (1982). Goals and strategies of inquiry teachers. In R. Glaser (Ed.), Advances in instructional psychology (Vol. 2, pp. 65–119). Hillsdale, NJ: Lawrence Erlbaum Associates.

Collins, A. , Warnock, E. H. , & Passafiume, J. J. (1975). Analysis and synthesis of tutorial dialogues. In G. Bower (Ed.), The psychology of learning and motivation (Vol. 9, pp. 49–87). New York: Academic Press.

Crawford, A. M. , & Hollan, J. D. (1983). Development of a computer-based tactical training system (Spec. Rep. Nprdc-SR-83-13). San Diego, CA: Navy Personnel Research and Development Center.

deKleer, J. (1984). How circuits work. In D. G. Bobrow (Ed.), Qualitative reasoning about physical systems (pp. 205–280). Cambridge, MA: MIT Press.

deKleer, J. , & Brown, J. S. (1983). Assumptions and ambiguities in mechanistic mental models. In D. Gentner & A. L. Stevens (Eds.), Mental models (pp. 155–190). Hillsdale, NJ: Lawrence Erlbaum Associates.

Gagné, R. M. & Briggs, L. J. , (1979). Principles of instructional design. New York: Holt, Rinehart & Winston.

Gentner, D. R. (1979). Coach: A schema-based tutor (Rep. No. 7903). La Jolla: University of California, San Diego, Center for Human Information Processing.

Gentner, D. R. , & Norman, D. A. (1977). The FLOW tutor: Schemas for tutoring (Rep. No. 7702). La Jolla: University of California, San Diego, Center for Human

Information Processing.
Goldstein, I. P. (1982). The genetic graph: A representation for the evolution of procedural knowledge. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 51–77). New York: Academic Press.
Halff, H. M. , Hollan, J. D. , & Hutchins, E. L. (1986). Cognitive science and military training. American Psychologist, 41, 1131–1139.
Heller, J. I. , & Reif, F. (1984). Prescribing effective human problem-solving processes: Problem description in physics. Cognition and Instruction, 1, 177–216.
Hutchins, E. , & McCandless, T. P. (1982). MANBOARD: A graphic display program for training relative motion concepts (NPRDC Tech. Note 82–100). San Diego, CA: Navy Personnel Research and Development Center.
Hutchins, E. , McCandless, T. P. , Woodworth, G. , & Dutton, B. (1984). Maneuvering board training systemml: Analysis and redesign (NPRDC Tech. Rep. 84–119). San Diego, CA: Navy Personnel Research and Development Center.
Kimball, R. (1982). A self-improving tutor for symbolic integration. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 283–307). New York: Academic Press.
Mehan, H. (1979). Learning lessons: Social organization in the classroom. Cambridge, MA: Harvard University Press.
Merrill, D. M. (1983). Component display theory. In C. M. Reigeluth (Ed.), Instructional design theories and models: An overview of their current status (pp. 279–332). Hillsdale, NJ: Lawrence Erlbaum Associates.
Norman, D. A. (1973). Memory, knowledge, and the answering of questions. In R. L. Solso (Ed.), Contemporary issues in cognitive psychology: The Loyola Symposium (pp. 135–165). Washington, DC: V. H. Winston & Sons.
O'Shea, T. (1982). A self-improving quadratic tutor. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 309–336). New York: Academic Press.
Reif, F. , & Heller, J. I. (1982). Knowledge structure and problem solving in physics. Educational Psychologist, 17, 102–127.
Reigeluth, C. M. , & Merrill, D. M. (1978). A knowledge base for improving our methods of instruction. Educational Psychologist, 13, 57–70.
Reigeluth, C. M. , & Stein, F. S. (1983). The elaboration theory of instruction. In C. M. Reigeluth (Ed.), Instructional design theories and models: An overview of their current status (pp. 335–381). Hillsdale, NJ: Lawrence Erlbaum Associates.
Schneider , W. (1985). Training high-performance skills: Fallacies and guidelines. Human Factors, 27, 285–300.
Smith, R. L. , Walker, P. , & Spool, P. (1982). The recognition of instructional strategies in the modeling of student acquisition of problem-solving skills (Final Rep. 1). New Brunswick, N J: Rutgers-The State University, Laboratory for Computer Science Research.
Stevens, A. , & Collins, A. (1977, October). The goal structure of a Socratic tutor. In Proceedings of the 1977 Annual Conference, Association for Computing Machinery (pp. 256–263). New York: Association for Computing Machinery.
Stevens, A. L. , & Collins, A. (1980). Multiple conceptual models of a complex system. In R. Snow , P. A. Federico , & W. Montague (Eds.), Aptitude, learning, and instruction: Vol. 2. Cognitive process analyses of learning and problem solving (pp. 177–197). Hillsdale, NJ: Lawrence Erlbaum Associates.
VanLehn, K. (1983). Felicity conditions for human skill acquisition: Validating an AI theory (Report No. CSL-21). Palo Alto, CA: Xerox Palo Alto Research Center.
VanLehn, K. (1985). Acquiring procedural skills from lesson sequences (Rep. No. ISL-9). Palo Alto, CA: Xerox Palo Alto Research Center.
VanLehn, K. (1987). Learning one subprocedure per lesson. AI Journal, 32, 1–40.
Wescourt, K. , Beard, M. , & Gould, L. (1977). Knowledge-based adaptive curriculum sequencing for CAI: Application of a network representation. In Proceedings of the

1977 Annual Conference, Association for Computing Machinery (pp. 234–240). New York: Association for Computing Machinery.
Winston, P. H. (1984). Artificial intelligence (2nd ed.). Reading, MA: Addison-Wesley.
Woolf, B. , & McDonald, D. D. (1985). Building a computer tutor: Design issues. AEDS Monitor, 23(9–10), 10–18.

The Environment Module of Intelligent Tutoring Systems

Abelson, A. , & diSessa, A. A. (1980). Turtle geometry. Cambridge, MA: MIT Press.
Anderson, J. R. (1981). Tuning of search of the problem space for geometry proofs. In A. Drinan (Ed.), Proceedings of the International Joint Conference on Artificial Intelligence-81 (Vol. 1, pp. 165–170). Los Altos, CA: Morgan Kaufmann.
Anderson, J. R. , Boyle, C. F. , Corbett, A. , & Lewis, M. (1986). Cognitive modeling and intelligent tutoring (CMU Tech. Rep.). Pittsburgh, PA: Carnegie-Mellon University, Psychology Department.
Anderson, J. R. , Boyle, C. F. , & Yost, G. (1985). The geometry tutor. In A. Joshi (Ed.), Proceedings of the Ninth International Joint Conference on Artificial Intelligence (pp. 1–7). Los Altos, CA: Morgan Kaufmann.
Anderson, J. R. , & Skwarecki, E. (1986). The automated tutoring of introductory computer programming. Communications of the ACM, 29, 842–849.
Bonar, J. (1985, June). Bite-sized intelligent tutoring: Newsletter 85–3, 1–11. Pittsburgh, PA. University of Pittsburgh, Learning Research and Development Center, Intelligent Tutoring Systems Group.
Bonar, J. , Cunningham, R. , & Schultz, J. (1986). An object-oriented architecture for intelligent tutoring systems. In N. Meyrowitz (Ed.), Proceedings of the First Conference on Object Oriented Programming, Systems, Languages and Applications (pp. 269–276). New York: ACM.
Brown, A. L. & Palincsar, A. S. . Reciprocal teaching of comprehension strategies: A natural history of one program for enhancing learning. In J. B. Borkowski & J. D. Day (Eds.), Intelligence and cognition in special children: Comparative studies of giftedness, mental retardation, and learning disabilities. Norwood, NJ: Ablex.
Brown, J. S. (1983). Learning by doing revisited for electronic learning environments. In M. A. White (Ed.), The future of electronic learning (pp. 13–32). Hillsdale, NJ: Lawrence Erlbaum Associates.
Brown, J. S. (1985). Idea amplifiers: New kinds of electronic learning environments. Educational Horizons, 63, 108–112.
Brown, J. S. , & Burton, R. R. (1975). Multiple representations of knowledge for tutorial reasoning. In D. G. Bobrow & A. Collins (Eds.), Representation and understanding (pp. 311–349). New York: Academic Press.
Brown, J. S. , & Burton, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. Cognitive Science, 2, 155–192.
Brown, J. S. , & Burton, R. R. (1987). Reactive learning environments for teaching electronic troubleshooting. In W. B. Rouse (Ed.), Advances in man-machine systems research (Vol. 3). Greenwich, CT: JAI Press.
Brown, J. S. , Burton, R. R. , & deKleer, J. (1982). Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 227–282). London: Academic Press.
Brown, J. S. , Moran, T. P. , & Williams, M. D. (1982). The semantics of procedures: A cognitive basis for training procedural skills for complex system maintenance (CIS working paper). Palo Alto, CA: Xerox Palo Alto Research Center, Intelligent Systems

Laboratory.
Burton, R. R. , & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 79–98). London: Academic Press.
Burton, R. R. , Brown, J. S. , & Fischer, G. (1984). Skiing as a model of instruction. In B. Rogoff & J. Lave (Eds.), Everyday cognition: Its development in social context (pp. 139–150). Cambridge, MA: Harvard University Press.
Chi, M. T. H. , Feltovich, P. J. , & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. Cognitive Science, 5, 121–152.
Clancey, W. J. (1982). Tutoring rules for guiding a case method dialogue. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 201–226). London: Academic Press.
Clancey, W. J. (1986). From GUIDON to NEOMYCIN and HERACLES in twenty short lessons: ONR final report 1979–1985. AI Magazine, 7, 40–60.
Clement, J. (1983). A conceptual model discussed by Galileo and used intuitively by physics students. In D. Gentner & A. L. Stevens (Eds.), Mental models (pp. 325–340). Hillsdale, NJ: Lawrence Erlbaum Associates.
Collins, A., Brown, J. S., & Newman, S. E. (in press). The new apprenticeship: Teaching students the craft of reading, writing, and mathematics. In L. B. Resnick (Ed.), Cognition and instruction: Issues and agendas. Hillsdale, NJ: Lawerence Erlbaum Associates.
Copeland, W. D. (1984). Creating a historian's microworld. Classroom Computer Learning, 5, 48–53.
diSessa , A. (1982). Unlearning Aristotelian physics: A study of knowledge-based learning. Cognitive Science, 6, 37–75.
diSessa, A. (1983). Phenomenology and the evolution of intuition. In D. Gentner & A. L. Stevens (Eds.), Mental models (pp. 15–34). Hillsdale, NJ: Lawrence Erlbaum Associates.
Erlwanger, S. H. (1973). Benny's conception of rules and answers in IPI mathematics. Journal of Children's Mathematical Behavior, 1, 7–26.
Fischer, G. (in press). Enhancing incremental learning processes with knowledge-based systems. In H. Mandl & A. Lesgold (Eds.), Learning issues for intelligent tutoring systems. New York: Springer Verlag.
Fischer, G. , Brown, J. S. , & Burton, R. (1978). Aspects of a theory of simplification: Debugging and coaching. Proceedings of the Second National Conference of Canadian Society for Studies of Intelligence (pp. 139–145).
Foss, C. L. (1987). Learning from errors in AlbegraLand (IRL Tech. Rep. 3). Palo Alto, CA: Xerox Palo Alto Research Center, Institute for Research on Learning.
Genesereth, M. (1982). The role of plans in intelligent teaching systems. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 137–156). London: Academic Press.
Goldstein, I. P. (1982). The genetic graph: A representation for the evolution of procedural knowledge. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 51–78). London: Academic Press.
Haertel, H. (1987). A qualitative approach to electricity (IRL Tech. Rep.). Palo Alto, CA: Xerox Palo Alto Research Center, Institute for Research on Learning.
Hollan, J. D., Hutchins, E. L., McCandless, T. P., Rosenstein, M., & Weitzman, L. (in press). Graphical interfaces for simulation. In W. B. Rouse (Ed.), Advances in man-machine systems research (Vol. 3). Greenwich, CT: JAI Press.
Hollan, J. D. , Hutchins, E. L. , & Weitzman, L. (1984). Steamer: An interactive inspectable simulation-based training system. AI Magazine, 5, 15–28.
Hutchins, E. L. , Hollan, J. D. , & Norman, D. A. (1985). Direct manipulation interfaces. Human-Computer Interaction, 1, 311–338. Hillsdale, NJ: Lawrence

Erlbaum Associates.
Hutchins, E. , & McCandless, T. P. (1982). MANBOARD: A graphic display program for training relative motion concepts (NPRDC Tech. Note 82–110). San Diego: Navy Personnel Research and Development Center.
Johnson, W. B. (1987). Development and evaluation of simulation-oriented computer-based instruction for diagnostic training. In W. B. Rouse (Ed.), Advances in man-machine systems research (Vol. 3). Greenwich, CT: JAI Press.
Johnson, W. B. , Maddox, M. E. , Rouse, W. B. , & Kiel, G. C. (1985). Diagnostic training for nuclear plant personnel: Volume I. Courseware development (EPRI NP-3829). Palo Alto, CA: Electric Power Research Institute.
Larkin, J. H. (1983). The role of problem representation in physics. In D. Centner & A. L. Stevens (Eds.), Mental models (pp. 75–98). Hillsdale, NJ: Lawrence Erlbaum Associates.
Nickerson, R. (1986). Technology in education in 2020: Thinking about the not-distant future. Background paper for Harvard Center for School Technology Conference. Cambridge, MA: Bolt, Beranek, and Newman, Inc.
Orr, J. E. (1986). Narratives at work: Story telling as cooperative diagnostic activity. Proceedings of the Conference on Computer-Supported Cooperative Work (pp. 62–72).
O'Shea, T. , Bornat, R. , Du Boulay, B. , Eisenstadt, M. , & Page, I. (1983). Tools for designing intelligent computer tutors. In A. Elithorn & R. Banerjii (Eds.), Human and artificial intelligence (pp. 181–199). London: North-Holland.
Papert, S. (1980). Mindstorms. New York: Basic Books.
Papert, S . (1986, May 2). Rethinking mathematics learnability in a computer culture. The Karl de Leeuw Memorial Lecture. Stanford, CA: Stanford University.
Richer, M. , & Clancey, W. J. (1985). Guidon-Watch: A graphical interface for viewing a knowledge-based system. IEEE Computer Graphics and Applications, 11, 51–64.
Roschelle, J. (1986). The envisioning machine: Facilitating students' re-conceptualization of physics (ISL working paper). Palo Alto, CA: Xerox Palo Alto Research Center, Intelligent Systems Laboratory.
Rouse, W. B. (1979). Problem solving performance of maintenance trainees in a fault diagnosis task. Human Factors, 21, 195–203.
Rouse, W. B. , & Hunt, R. M. (1984). Human problem solving in fault diagnosis tasks. In W. B. Rouse (Ed.), Advances in man-machine systems research (Vol. 1). Greenwich, CT: JAI Press.
Russell, D. M., Moran, T. P., & Jordan, D. S. (in press). Instructional design environment. In J. Psotka, L. D. Massey, & S. A. Mutter (Eds.), Intelligent tutoring systems: Lessons learned. Hillsdale, NJ: Lawrence Erlbaum Associates.
Scardamalia, M. , Bereiter, C. , & Steinbach, R. (1984). Teachability of reflective processes in written composition. Cognitive Science, 8, 173–190.
Schoenfeld, A. H. (1985). Mathematical problem solving. New York: Academic Press.
Schwartz, J. L., & Yerushalmy, M. (in press). The geometry supposer: Using microcomputers to restore invention to the learning of mathematics. In D. N. Perkins, J. Lochhead, & J. Butler (Eds.), Thinking: The second international conference. Hillsdale, NJ: Lawrence Erlbaum Associates.
Sheil, B. (1983a). The artificial intelligence tool box. In W. Reitman (Ed.), Proceeding of the NYE Symposium on Artificial Intelligence and Business (pp. 287–295). Norwood, NJ: Ablex.
Sheil, B. (1983b). Power tools for programmers. Datamation, 29, 131–144.
Smith, R. B. (1986). The alternate reality kit: An animated environment for creating interactive simulations. Proceedings of the IEEE Computer Society Workshop on Visual Languages (pp. 99–106). Washington, DC: IEEE Computer Society Press.

Stansfield, J. , Carr, B. , & Goldstein, I. P. (1976). Wumpus Advisor I: A first implementation of a program that tutors logical and probabilistic reasoning skills (MIT AI Laboratory Memo No. 381). Cambridge, MA: Massachusetts Institute of Technology.
Towne, D. M. (in press). The generalized maintenance trainer: Evolution and revolution. In W. B. Rouse (Ed.), Advances in man-machine systems research (Vol. 3). Greenwich, CT: JAI Press.
VanLehn, K. , & Brown, J. S. (1980). Planning nets: A representation for formalizing analogies and semantic models for procedural skills. In R. W. Snow , P. A. Fredico , & W. E. Montague (Eds.), Aptitude learning and instruction: Vol. 2. Cognitive process analyses of learning and problem-solving (pp. 95–137). Hillsdale, NJ: Lawrence Erlbaum Associates.
Weinreb, D. , & Moon, D. (1981). Lisp Machine manual. Cambridge, MA: MIT Artificial Intelligence Laboratory.
Wenger, E. (1987). Artificial intelligence and tutoring systems: Computational approaches to the communication of knowledge. Los Altos, CA: Morgan Kaufmann.
White, B. Y. , & Frederiksen, J. R. (1986a). Intelligent tutoring systems based upon qualitative model evolutions. Proceeding of AAAI-86 (pp. 313–319). Los Altos, CA: Morgan Kaufmann.
White, B. Y. , & Frederiksen, J. R. (1986b). Progressions of qualitative models as foundations for intelligent learning environments (BBN Report 6277). Cambridge, MA: Bolt, Beranek and Newman, Inc.
Wilensky, R. , Arens, Y. , & Chin, D. (1984). Talking to UNIX in English: An overview of UC. Communications of the ACM, 27, 574–593.
Wiser, M. , & Carey, S. (1983). When heat and temperature were one. In D. Gentner & A. L. Stevens (Eds.), Mental models (pp. 267–298). Hillsdale, NJ: Lawrence Erlbaum Associates.
Xerox Special Information Systems . (1985). Notecards release 1.2i reference manual. Pasadena, CA: Author.

The Role of Human-Computer Interaction in Intelligent Tutoring Systems

Adams, D. A. , &. Bisiani, R. (1986). The Carnegie-Mellon University distributed speech recognition system. Speech Technology, 12, 14–23.
Allen, J. F. , & Perrault, C. R. (1980). Analyzing intention in utterances. Artificial Intelligence, 15, 143–178.
Anderson, J. R. , Boyle, C. F. , Farrell, R. , & Reiser, B. (1984). Cognitive principles in the design of computer tutors. Proceedings of the Sixth Cognitive Science Conference (pp. 2–9). Boulder, CO: University of Colorado.
Anderson, J. R. , Boyle, C. F. , & Yost, G. (1985). The geometry tutor. In A. Joshi (Ed.), Proceedings of the Ninth International Joint Conference on Artificial Intelligence (Vol. 1, pp. 1–7). Los Altos, CA: Morgan Kaufmann.
Anderson, J. R. , & Jeffries, R. (1985). Novice LISP errors: Undetected losses of information from working memory. Human-Computer Interaction, I, 107–132.
Anderson, J. R. , & Skwarecki, E. (1986). The automated tutoring of introductory computer programming. Communications of the ACM, 29, 842–849.
Ballard, B. W. , & Stumberger, D. E. (1986). Semantic acquisition in TELI: A transportable, user-customized natural language processor. Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics (pp. 20–29). New York: Association for Computing Machinery.

Bates, M. , & Bobrow, R. J. (1983). A transportable natural language interface. Proceedings of the 6th Annual International ACM SIGIR Conference on Research and Development in Informtion Retrieval. Bethesda, MI): Association for Computing Machinery.
Biermann, A. W. , Fineman, L. , & Gilbert, K. C. (1985). An imperative sentence processor for voice interactive office applications. ACM Transactions on Office Information Systems, 3, 321–346.
Bitzer, D. L. , & Easley, J. A., Jr . (1965). PLATO: A computer-controlled teaching system. In Sass & Wilkinson (Eds.), Computer augmentation of human reasoning. Washington, DC: Spartan Books.
Bloom, B. S. . (1984, June/July). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. Educational Researcher.
Bocker, H. , Fischer, G. , & Nieper, H. (1986). The enhancement of understanding through visual representations. In M. Mantei & P. Orbeton (Eds.), Proceedings of the CHI ’86 Conference on Human Factors in Computing Systems (pp. 44–50). Boston, MA: Association for Computing Machinery.
Bolt, R. A. (1980). Put That There: Voice and gesture at the graphics interface. In Computer Graphics: Proceedings of ACM SIGGRAPH ’80 (pp. 262–270). New York: Association for Computing Machinery.
Bonar, J . (1985, Newsletter 85–93). Bite-Sized Intelligent Tutoring. Pittsburgh, PA: University of Pittsburgh, Learning Research and Development Center.
Bonissone, P. P. , & Johnson, H. E., Jr . (1984). DELTA: An expert system for diesel locomotive repair. In Artificial intelligence in maintenance: Proceedings of the joint services workshop (pp. 181–195). Brooks Air Force Base, TX: Air Force Systems Command.
Brewer, B. (1986). Video on CD: The big picture. In S. Lambert & S. Ropiequet (Eds.), CD ROmml: The new papyrus (pp. 291–294). Redmond, WA: Microsoft Press.
Brown, J. S. , Burton, R. R. , & deKleer, J. (1982). Pedagogical natural language, and knowledge engineering techniques in SOPHIE I, II, and III. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 227–282). New York: Academic Press.
Carbonell, J. G. , & Collins, A. M. (1973). Natural semantics in artificial intelligence. Proceedings of the Third International Joint Conference on Artificial Intelligence (pp. 344–351). Stanford, CA: Stanford Research Institute.
Carbonell, J. G. , & Hayes, P. J. (1983). Recovery strategies for parsing extragrammatical language. American Journal of Computational Linguistics, 9, 123–146.
Cater, J. C. (1984). Electronically hearing: Computer speech recognition. Indianapolis, IN: Sams.
Chi, M. T. H. , Feltovich, P. J. , & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. Cognitive Science, 5, 121–152.
Christ, R. E. (1975). Review and analysis of color coding research for visual displays. Human Factors, 17, 542–570.
Clancey, W. J. (1982). Tutoring rules for guiding a case method dialogue. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 201–226). New York: Academic Press.
Douglas, S. A. , & Moran, T. P. (1983). Learning text editor semantics by analogy. In A. Janda (Ed.), Proceedings of the CHI ’83 Conference on Human Factors in Computing Systems (pp. 207–211). Boston, MA: Association for Computing Machinery.
Dreiman, G. H. J. (1962). Differences between written and spoken language: An exploratory study. Acta Psychologia, 10, 36–100.

Erman, L. D. , Hayes-Roth, F. , Lesser, V. R. , & Reddy, D. R. (1980). The Hearsay-II speech understanding system: Integrating knowledge to resolve uncertainty. ACM Computing Surveys, 12, 213–253.
Feldmann, R. (1985). Interfaces for molecular modeling systems. Paper presented at the MCC Symposium on Computer Graphics and Human Interfaces: MCC, Austin, TX.
Fischer, G. , Lemke, A. , & Schwab, T. (1985). Knowledge-based help systems. In L. Borman & B. Curtis (Eds.), Proceedings of the CHI ’85 Conference on Human Factors in Computing Systems (pp. 161–167). San Francisco, CA: Association for Computing Machinery.
Gentner, D. , & Gentner, D. R. (1983). Flowing water or teeming crowds: Mental models of electricity. In D. Gentner & A. L. Stevens (Eds.), Mental models (pp. 99–130). Hillsdale, NJ: Lawrence Erlbaum Associates.
Gentner, D. , & Stevens, A. L. . (Eds.). (1983). Mental models. Hillsdale, NJ: Lawrence Erlbaum Associates.
Gould, J. D. , Conti, J. , & Hovanyecz, T. (1981). Composing letters with a simulated listening typewriter. Proceedings of the First Conference on Human Factors in Computing Systems (pp. 367–370). Gaithersburg, MD: Association for Computing Machinery.
Granger, R. H. (1983). The NOMAD system: Expectation-based detection and correction of errors during understanding of syntactically and semantically ill-formed text. American Journal of Computational Linguistics, 9, 188–196.
Grosz, B. J. , Appelt, D. , Martin, P. , & Pereira, F. (1987). Team: An experiment in the design of transportable natural language interfaces. Artificial Intelligence, 32, 173–244.
Halasz, F. , & Moran, T. P. (1983). Analogy considered harmful. In A. Janda (Ed.), Proceedings of the CHI ’83 Conference on Human Factors in Computing Systems (pp. 45–49). Boston: Association for Computing Machinery.
Hayes, P. J. , Szekely, P. A. , & Lerner, R. A. (1985). Design alternatives for user interface management systems based on experiences with COUSIN. In L. Borman & B. Curtis (Eds.), Proceedings of the CHI ’85 Conference on Human Factors in Computing Systems (pp. 169–175). San Francisco: Association for Computing Machinery.
Henderson, D. A., Jr . (1986). The Trillium user interface design environment. In M. Mantei & P. Orbeton (Eds.), Proceedings of the CHI ’86 Conference on Human Factors in Computing Systems (pp. 221–227). Boston: Association for Computing Machinery.
Hendrix, G. G. (1979). The LIFER manual: A guide to building practical natural language interfaces (Tech. Note 138). Palo Alto: SRI International AI Center.
Hendrix, G. G. , & Lewis, W. H. (1981). Transportable natural-language interfaces to databases. In Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics (pp. 159–165). New York: Association for Computing Machinery.
Hertzog, B. (1985). Three-dimensional computer graphics for CAD/CAM. Paper presented at the MCC Symposium on Computer Graphics and Human Interfaces: MCC, Austin, TX.
Hollan, J. D. , Hutchins, E. L. , & Weitzman, L. (1984). Steamer: An interactive inspectable simulation-based training system. AI Magazine, 5, 15–27.
Horowitz, M. W. , & Newman, J. B. (1964). Spoken and written expression: An experimental analysis. Journal of Abnormal and Social Psychology, 68, 640–647.
Hutchins, E. L. , Hollan, J. D. , & Norman, D. A. (1986). Direct manipulation interfaces. In D. A. Norman & S. W. Draper (Eds.), User centered system design (pp. 87–124). Hillsdale, NJ: Lawrence Erlbaum Associates.

Johnson, W. B. (1987). Development and evaluation of simulation-oriented computer-based instruction for diagnostic training. In W. B. Rouse (Ed.), Advances in man-machine systems research (Vol. 3, pp. 99–127). Greenwich, CT: J AI Press.

Lambert, S. , & Ropiequet, S . (Eds.). (1986). CD ROmml: The new papyrus. Redmond, WA: Microsoft Press.

Laurel, B. K. (1986). Interface as mimesis. In D. A. Norman & S. W. Draper (Eds.), User centered system design (pp. 67–86). Hillsdale, NJ: Lawrence Erlbaum Associates.

Lewis, C. , & Mack. R. (1981). Learning to use a text processing systemml: Evidence from “thinking aloud” protocols. Proceedings of the First Conference on Human Factors in Computing Systems (pp. 387–392). Gaithersburg, MD: Association for Computing Machinery.

Littman, D. J. (1986). Linguistic coherence: A plan-based alternative. Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics (pp. 215–223). New York: Association for Computing Machinery.

Mackinlay, J. (1986). Automatic design of graphical presentations. Unpublished doctoral dissertation, Palo Alto, CA: Stanford University.

Malone, T. W. (1981). Towards a theory of intrinsically motivating instruction. Cognitive Science, 4, 333–370.

Marcus, A. (1985). Screen design for iconic interfaces. In L. Borman & B. Carter (Eds.), CHI ’85 Conference on Human Factors in Computing Systems. San Francisco: Association for Computing Machinery.

Martin, P. , Appelt, D. , & Pereira, F. (1983). Transportability and generality in a natural language interface system. Proceedings of the Eighth Intrnational Joint Conference on Artificial Intelligence (Vol. 1, pp. 574–581). Los Altos, CA: Morgan Kaufmann.

Miller, J. R. (1982). System-initiated user assistance: More than just a luxury (Tech. Rep.). Dallas, TX: Texas Instruments Artificial Intelligence Laboratory.

Miller, J. R. , & Blumenthal, B. (1985). An architecture for generalized intelligent user assistance systems (Tech. Rep. HI-085-85). Austin, TX: Microelectronics and Computer Technology Corporation.

Moran, T. P. (1983). Getting into a systemml: External-internal task mapping analysis. In A. Janda (Ed.), Proceedings of the CHI ’83 Conference on Human Factors in Computing Systems (pp. 45–49). Boston: Association for Computing Machinery.

Myers, B. A. , & Buxton, W. (1986). Creating highly interactive and graphical user interfaces by demonstration. Computer Graphics: Proceedings of ACM SIGGRAPH ’86 (pp. 249–258). Dallas: Association for Computing Machinery.

Nakatani, L. H. , Egan, D. E. , Ruedisueli, L. W. , Hawley, P. M. , & Lewart, D. K. (1986). TNT: A talking tutor ’n trainer for teaching the use of interactive computer systems. In M. Mantei & P. Orbeton (Eds.), Proceedings of the CHI ’86 Conference on Human Factors in Computing Systems (pp. 29–34). Boston, MA: Association for Computing Machinery.

Norman, D. A. (1981). The trouble with Unix. Datamation, 27, 139–150.

Norman, D. A. (1986). Cognitive engineering. In D. A. Norman & S. W. Draper (Eds.), User centered system design (pp. 31–61). Hillsdale, NJ: Lawrence Erlbaum Associates.

Payne, S. J. , & Green, T. G. R. (1986). Task-action grammars: A model of the mental representation of task languages. Human-Computer Interaction, 2, 93–134.

Polson, P. G. , & Kieras, D. E. (1985). A quantitative model of the learning and performance of text editing knowledge. In L. Borman & B. Curtis (Eds.), Proceedings of the CHI ’85 Conference on Human Factors in Computing Systems (pp. 207–212). San Francisco: Association for Computing Machinery.

Reiser, B. J. , Anderson, J. R. , & Farrell, R. G. (1985). Dynamic student modeling in an intelligent tutor for LISP programming. In A. Joshi (Ed.), Proceedings of the Ninth International Joint Conference on Artificial Intelligence (Vol. 1, pp. 8–13). Los Altos, CA: Morgan Kaufmann.
Richer, M. H. , & Clancey, W. J. (1985). Guidon-Watch: A graphic interface for viewing a knowledge-based system. IEEE Computer Graphics and Applications, 5, 51–64.
Savage, R. E. , Habinek, J. K. , & Barnhart, T. W. (1981). The design, simulation, and evaluation of a menu-driven interface. Proceedings of the First Conference on Human Factors in Computing Systems (pp. 36–40). Gaithersburg, MD: Association for Computing Machinery.
Schmandt, C. , & Hulteen, E. A. (1981). The intelligent voice-interactive interface. Proceedings of the First Conference on Human Factors in Computing Systems (pp. 363–366). Gaithersburg, MD: Association for Computing Machinery.
Schulert, A. J. , Rogers, G. T. , & Hamilton, J. A. (1985). ADmml: A dialog manager. In L. Borman & B. Curtis (Eds.), Proceedings of the CHI ’85 Conference on Human Factors in Computing Systems (pp. 177–184). San Francisco: Association for Computing Machinery.
Shneiderman, B. (1986). Designing the user interface. Reading, MA: Addison-Wesley.
Shrager, J. , & Finin, T. (1982). An expert system that volunteers advice. Proceedings of the National Conference on Artificial Intelligence (pp. 74–78). Los Altos, CA: Morgan Kaufmann.
Sibert, J. L. , Hurley, W. D. , & Bleser, T. W. (1986). An object-oriented user interface management system. Computer Graphics: Proceedings of ACM SIGGRAPH ’86 (pp. 259–268). Dallas, TX: Association for Computing Machinery.
Smith, D. C. , Irby, C. , Kimball, R. , Verplank, W. , & Harslem, E. (1982). Designing the Star user interface. Byte, 7, 242–282.
Soloway, E. , & VanLehn, K. (1985). Tutorial notes for AI applications for education. International Joint Conference on Artificial Intelligence, Los Angeles, CA.
Stevens, A. L. , & Collins, A. (1977). The goal structure of a socratic tutor (Tech. Rep. 3518). Cambridge, MA: Bolt Beranek and Newman.
Teitelman, W. (1984a). A display-oriented programmer’s assistant. In D. R. Barstow , H. E. Shrobe , & E. Sandewall (Eds.), Interactive programming environments (pp. 240–287). New York: McGraw-Hill.
Teitelman, W. (1984b). A tour through Cedar. Proceedings of the Seventh International Conference on Software Engineering (pp. 181–195). Orlando, FL: IEEE.
Tennant, H. (1979). Experience with the evaluation of natural language question answerers. Proceedings of the Sixth International Joint Conference on Artificial Intelligence (Vol. 1, pp. 8–13). Tokyo: Morgan Kaufmann.
Tennant, H. R. , Ross, K. M. , Saenz, R. M. , Thompson, C. W. , & Miller, J. R. (1983). Menu-based natural language understanding. Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics (pp. 151–158). New York: Association for Computing Machinery.
Thacker, C. , McCreight, E. , Lampson, B. , Sproull, R. , & Boggs, D. (1979). Alto: A personal computer (Tech. Rep. CSL-79-11). Palo Alto: Xerox Palo Alto Research Center.
Tufte, E. R. (1983). The visual display of quantitative information. Cheshire, CT: Graphics Press.
Weitzman, L. (1986). Designer: A knowledge-based graphic design assistant (Tech. Rep. 138). LaJolla, CA: University of California, San Diego, UCSD Institute for Cognitive Science.

Wong, P. C. S. , & Reid, E. R. (1982). Flair: User interface dialog design tool. Computer Graphics, 16, 87–98.
Zimmerman, T. G. , Lanier, J. , Blanchard, C. , Bryson, S. , & Harvill, Y. (1987). A hand gesture interface device. In J. M. Carroll & P. Tanner (Eds.), Proceedings of the CHI ’87 Conference on Human Factors in Computing Systems and Graphics Interface (pp. 189–192). Toronto: Association for Computing Machinery.
Zissis, A. Y. , & Witten, I. H. (1985). User modeling for a computer coach: A case study.International Journal of Man-Machine Studies, 23, 729–750.

Pragmatic Considerations in Research, Development, and Implementation of Intelligent Tutoring Systems 1

Clancey, W. J. (1982). Tutoring rules for guiding a case method dialog. In D. Sleeman and J. S. Brown (Eds.), Intelligent tutoring systems (pp. 201–225). New York: Academic Press.
Hollan, J. D. , Hutchins, E. L. , & Weitzman, L. (1984). Steamer: An interactive inspectable simulation-based training system. The AI Magazine, 5, 15–27.
Johnson, W. B. (1981). Computer simulations for fault diagnosis training: An empirical study of learning from simulation to live system performance. (Doctoral dissertation. University of Illinois, 1980). Dissertation Abstracts International, 41, 4625-A.
Johnson, W. B. (1987). Development and evaluation of simulation-oriented computer-based instruction for diagnostic training. In W. B. Rouse (Ed.), Advances in man-machine systems research (Vol. 3, pp. 99–127). Greenwich, CT: JAI Press.
Maddox, M. E. , Johnson, W. B. , & Frey, P. R. (1986). Diagnostic training for nuclear power plant personnel: Vol. 2. Implementation and evaluations (EPRI NP-3829). Palo Alto, CA: Electric Power Research Institute.
Munro, A. , Fehling, M. R. , & Towne, D. M. (1985). Instruction intrusiveness in dynamic simulation training. Journal of Computer-Based Instruction, 12, 50–53.
National Research Council . (1986). Isolation of faults in Air Force weapons and support systems. Washington, DC: National Academy Press.
Newell, A. , Shaw, J. C. , & Simon, H. A. (1960). A variety of intelligent learning in a general problem-solver. In M. C. Yovits & S. Cameron (Eds.), Self-organizing systems (pp. 153–189). New York: Pergamon.
Richardson, J. J. , & Jackson, T. E. (1986). Developing the technology for intelligent maintenance advisors. Journal of Computer-Based Instruction, 13, 47–51.
Richardson, J. J. , Maxion, R. A. , Polson, P. G. , & DeJong, K. A. (1985). Artificial intelligence in maintenance: Synthesis of technical issues (AFHRL-TR-85-7). Brooks Air Force Base, TX: Air Force Human Resources Laboratory.
Rouse, W. B. , & Hunt, R. M. (1984). Human problem solving in fault diagnosis tasks. In W. B. Rouse (Ed.), Advances in Man-Machine Systems Research (Vol. 1, pp. 195–222). Greenwich, CT: JAI Press.
Rouse, W. B. , Rouse, S. H. , & Pellegrino, S. J. (1980). A rule-based model of human problem solving performance in fault diagnosis tasks. IEEE Transactions on Systems, Man, and Cybernetics, Scm-10, 366–376.
Towne, D. M. (1987). The generalized maintenance trainer: Evolution and revolution. In W. B. Rouse (Ed.), Advances in Man-Machine Systems Research (Vol. 3, pp. 1–63). Greenwich, CT: JAI Press.207
White, G. B. , & Cross, S. E. (1986). TEMPO-AI: Using artificial intelligence concepts in war gaming and professional military education. Proceedings of the Air Force

Conference on Technology in Training and Education (pp. V-98-V-117). Montgomery, AL: Air University, U.S. Air Force Academy.

Evaluating ITSs: The Cognitive Science Perspective

Anderson, J. , Boyle, C. , Farrell, R. , & Reiser, B. (1984). Cognitive principles in the design of computer tutors. Proceedings of Sixth Cognitive Science Conference, (pp. 2–9). Boulder: University of Colorado.

Anderson, J. , Farrell, R. , & Sauers, R. (1984). Learning to program in LISP. Cognitive Science, 8, 87–129.

Baker, E. , Feifer, R. , Aschbacher, P. , Bradley, C. , & Herman, J. (1985). Intelligent computer-assisted instruction (ICAI) study. Los Angeles: University of California Center for the Study of Evaluation.

Brown, J. S. , & Burton, R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. Cognitive Science, 2, 155–192.

Brown, J. S. , & VanLehn, K. (1980). Repair theory: A generative theory of bugs in procedural skill. Cognitive Science, 4, 379–426.

Burton, R. , & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman and J. S. Brown (Eds.), Intelligent tutoring systems (pp. 79–98). New York: Academic Press.

Clancey , W. (1983). The epistemology of a rule-based expert system: A framework for explanation. Artificial Intelligence, 20, 215–251.

Goldstein, I. (1982). The genetic graph: A representation for the evolution of procedural knowledge. In D. Sleeman & J. S. Brown (Eds.), Intelligent tutoring systems (pp. 51–77). New York: Academic Press.

Hasling, D. , Clancey, W. , & Rennels, G. (1984). Strategic explanations for a diagnostic consultation system. International Journal of Man-Machine Studies, 20, 3–19.

Johnson, L. (1986). Intention-based diagnosis of errors in novice programs. Palo Alto, CA: Morgan Kaufmann.

Kulik, J. (1985, April). Consistencies in findings on computer-based instruction. Paper presented at the annual meeting of the American Educational Research Association, Chicago, IL.

Pirolli, P. (in press). A cognitive model and computer tutor for programming recursion. Human-Computer Interaction.

Sack, W. , Littman, D. , Spohrer, J. , Liles, A. , Fertig, S. , Hughes, L. , Johnson, L. , & Soloway, E. (1986). Empirical evaluation of the educational effectiveness of PROUST. (Working paper). Yale University, New Haven.

Directions for Research and Applications

Anderson, J. R. (1983). The architecture of cognition. Cambridge, MA: Harvard University Press.

Bobrow, D. B. . (Ed.). (1984). Qualitative reasoning about physical systems. Cambridge, MA: MIT Press.

Brown, J. S. , & VanLehn, K. (1980). Repair theory: A generative theory of bugs in procedural skills. Cognitive Science, 4, 379–426.

Brown, J. S. , Burton, R. R. , & deKleer, J. (1982). Pedagogical, natural language, and knowledge engineering techniques in SOPHIE I, II, and III. In D. Sleeman & J.

S. Brown (Eds.), Intelligent tutoring systems (pp. 227–282). London: Academic Press.
Card, S. K. , Moran, T. P. , & Newell, A. (1983). The psychology of human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum Associates.
Clancey, W. J. , & Letsinger, R. (1981). NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching. Proceedings of the Seventh International Joint Conference on Artificial Intelligence (Vol. II, pp. 829–836). Los Altos, CA: Morgan Kaufman.
Collins, A. M. (1976). Processes in acquiring knowledge. In R. C. Anderson , R. Spiro , & W. E. Montague (Eds.), Schooling and the acquisition of knowledge (pp. 339–363). Hillsdale, NJ: Lawrence Erlbaum Associates.
Gentner, D. , & Stevens, A. L. . (Eds.). (1983). Mental models. Hillsdale, NJ: Lawrence Erlbaum Associates.
Keller, R. A. (1985). Human troubleshooting in electronics: Implications for intelligent maintenance aids (AFHRL-TP-85-34). Brooks Air Force Base, TX: Air Force Human Resources Laboratory.
Klahr, D. , & Siegler, R. S. (1977). The representation of children’s knowledge. In H. Reese & L. P. Lipsitt (Eds.), Advances in child development (Vol. 12, pp. 61–116). New York: Academic Press.
Larkin, J. H. (1980). Expert and novice performance in solving physics problems. Science, 208, 1335–1342.
Woolf, B. , & McDonald, D. D. (1985). Building a computer tutor: Design issues. AEDS Monitor, 23, 10–18.