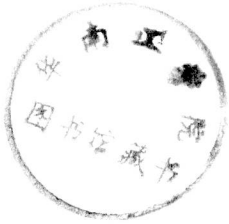
# Knowledge-Based Tutoring
## The GUIDON Program

William J. Clancey




The MIT Press
Cambridge, Massachusetts
London, England

PUBLISHER'S NOTE

This format is intended to reduce the cost of publishing certain works in book form and to shorten the gap between editorial preparation and final publication. Detailed editing and composition have been avoided by photographing the text of this book directly from the author's prepared copy.



This book was printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Clancey, William J.
Knowledge-based tutoring.

(The MIT series in artificial intelligence)
Revision of the author's thesis.
Bibliography: p.
Includes index.
1. Expert systems (Computer science) 2. Computer-assisted instruction—Computer programs. 3. GUIDON (Computer program) I. Title. II. Series.
QA76.76.E95C57 1987 371.3'9445 87-3449
ISBN 0-262-03123-X

# Knowledge-Based Tutoring

**The MIT Press Series in Artificial Intelligence**
Edited by Patrick Henry Winston and Michael Brady

*Artificial Intelligence: An MIT Perspective, Volume I: Expert Problem Solving, Natural Language Understanding, Intelligent Computer Coaches, Representation and Learning* edited by Patrick Henry Winston and Richard Henry Brown, 1979

*Artificial Intelligence: An MIT Perspective, Volume II: Understanding Vision, Manipulation, Computer Design, Symbol Manipulation* edited by Patrick Henry Winston and Richard Henry Brown, 1979

*NETL: A System for Representing and Using Real-World Knowledge* by Scott Fahlman, 1979

*The Interpretation of Visual Motion* by Shimon Ullman, 1979

*A Theory of Syntactic Recognition for Natural Language* by Mitchell P. Marcus, 1980

*Turtle Geometry: The Computer as a Medium for Exploring Mathematics* by Harold Abelson and Andrea diSessa, 1981

*From Images to Surfaces: A Computational Study of the Human Early Visual System* by William Eric Leifur Grimson, 1981

*Robot Manipulators: Mathematics: Programming and Control* by Richard P. Paul, 1981

*Computational Models of Discourse* edited by Michael Brady and Robert C. Berwick, 1982

*Robot Motion: Planning and Control* by Michael Brady, John M. Hollerbach, Timothy Johnson, Tomas Lozano-Perez, and Matthew Mason, 1982

*In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension* by Michael G. Dyer, 1983

*Robotics Research: The First International Symposium* edited by Michael Brady and Richard Paul, 1984

*Robotics Research: The Second International Symposium* edited by Hideo Hanafusa and Hirochika Inoue, 1985

*Robot Hands and the Mechanics of Manipulation* by Matthew T. Mason and J. Kenneth Salisbury, Jr., 1985

*Legged Robots that Balance* by Marc Raibert, 1985

*The Acquisition of Syntactic Knowledge* by Robert C. Berwick, 1985

*The Connection Machine* by W. Daniel Hillis, 1985

*Machine Interpretation of Line Drawings* by Kokichi Sugihara, 1986

*ACTORS: A Model of Concurrent Computation in Distributed Systems* by Gul A. Agha, 1986

*Knowledge-Based Tutoring: The GUIDON Program* by William J. Clancey, 1987

*Visual Reconstruction* by Andrew Blake and Andrew Zisserman, 1987

# Series Foreword

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

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

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

There are side effects that deserve attention, too. Any program that will successfully model even a small part of intelligence will be inherently massive and complex. Consequently, artificial intelligence continually confronts the limits of computer science technology. The problems encountered have been hard enough and interesting enough to seduce artificial intelligence people into working on them with enthusiasm. It is

natural, then, that there has been a steady flow of ideas from artificial intelligence to computer science, and the flow shows no signs of abating.

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

Patrick Henry Winston
Michael Brady

For my parents and brothers,
Kevin, Bryan, and Brendan

# Preface

In this decade we have witnessed a phenomenal growth of interest in expert systems—computer programs that codify the knowledge of experts in diverse areas of science, engineering, medicine, and business. These programs use *qualitative modeling* techniques, developed in the subfield of computer science called *artificial intelligence* (AI). Routine expert practice is thus codified, allowing knowledge to be distributed, accumulated, and conserved in what is called a *knowledge base.* The question naturally arises: Can we use a knowledge base in a teaching program? With hundreds of knowledge bases already in existence, many university and industrial researchers ask, "How can I adapt my existing expert system for use in teaching?" Others ask, "I am planning to build an expert system; how should it be designed to maximize its potential for teaching?" This book is intended for such people, especially for those without formal training in artificial intelligence, who wish to learn what simple methods will allow.

While the problem-solving capabilities of today's expert systems are limited, and perhaps always bounded on philosophic grounds, even simple knowledge-bases can have practical value for education. AI programming methods allow uncertain, heuristic knowledge to be efficiently represented so it can be easily modified and used for multiple purposes. The first generation of knowledge-based teaching programs exploit this methodology, automating instruction in a manner that is easy to understand and obviously improves upon traditional computer-aided instruction (CAI) programming.

Many good CAI programs provide a student with numerical simulations of physical and biological processes, which he can examine and experiment with. Using the qualitative modeling methods of ar-

tificial intelligence, we can now simulate *problem-solving processes* as well. In this way, we can provide the student with a model of a problem solver, which he might emulate in certain respects. For the teaching program, this model, often augmented by descriptions of non-expert methods, serves as a basis for understanding what the student is doing as he solves a problem, enabling the program to evaluate partial solutions and to offer assistance when the student is uncertain what to do next.

Of course, it has been possible to use conventional programming techniques to achieve these same capabilities to a certain extent. However, such programming is tedious. The solution for each problem, for example, a patient to be diagnosed, must be hand-coded, and the "course author" must anticipate and code responses for all possible student solution paths.

AI programming techniques allow a remarkable degree of generality. First, a given knowledge base can be used to solve many different problems, so each solution needn't be individually coded. A knowledge base can be used to teach an entire library of example problems. Second, knowledge about how to teach—how to interpret what a student is doing and how to respond to his needs—is encoded as a *second knowledge base.* Since this teaching knowledge is separate from the subject material, it can be reused for teaching problems in *different problem areas.* As a software engineering approach, it is obvious that knowledge-based tutoring, which allows one program to teach multiple problems in multiple domains, is the wave of the future. This book describes an architecture for implementing such a program, made concrete by many examples of teaching rules for directing an instructional dialogue.

The program described here, named GUIDON (pronounced "GUIDE-on"), represents the first attempt to adapt a pre-existing expert system for use in teaching. The underlying expert system is no ordinary program; it is MYCIN, the first and most well-known rule-based expert system. In this setting, MYCIN is a simulation program, serving as a partial model of how a student should diagnose a patient. MYCIN's knowledge base is interpreted by GUIDON to provide feedback as the student gathers information about a patient and makes a diagnosis. This book describes what GUIDON does, how it is constructed, and the benefits and limitations of its design.

While this book focuses on the rule-based formalism for encoding knowledge, the opportunities and limitations are applicable to other expert system representations now in common use, such as frames and semantic networks. The penultimate chapter analyzes MYCIN's knowledge in detail and makes these general lessons clear.

GUIDON was developed at Stanford University in the late 1970s, as a PhD programming project under the direction of Bruce G. Buchanan. We received much advice from researchers at MIT and Bolt, Beranek, and Newman (BBN), who were developing their own AI-based teaching programs at the time. This small community of researchers quickly shared ideas and built on each other's work. GUIDON's design was most strongly influenced by the electronic diagnosis program, SOPHIE (the effort of John Seeley Brown and Richard Burton) and by the Socratic-style geography tutor, SCHOLAR (the effort of the late Jaime R. Carbonnel and Alan Collins). In particular, the form of GUIDON's interaction with a student, in which he gathers problem data and makes hypotheses, comes from SOPHIE. The idea of encoding teaching knowledge as "dialogue management" rules comes from SCHOLAR and its follow-on, WHY. The idea of modeling a student in terms of an underlying "expert" simulation program was developed in WEST by Richard Burton. Finally, the program modules of a knowledge-based tutor were first specified in the form used here by Goldstein in his coach for the WUMPUS board game, a contemporary program whose design parallels GUIDON in several ways.

The final chapter compares GUIDON to other AI-based instructional programs. Knowledge-based problem solving in medicine is contrasted with problem solving in formal domains such as mathematics, board games, introductory computer programming, and electronics diagnosis. Problem solving in domains such as medicine and business, lacking an axiomatic or formal model of how things in the world work, places demands that restrict what can be accomplished with expert systems technology today. These limitations are especially important for people who seek to apply expert systems technology to teach decision-making. Much of practical value can be accomplished, but the limitations must be understood for the methods to be used responsibly and the greatest value to be realized.

In the years since GUIDON was constructed, an extensive effort

has been underway to reconfigure MYCIN to provide a more effective basis for a teaching program. This program, called NEOMYCIN, is briefly described in Chapter 8, following the description of MYCIN's limitations. While GUIDON is limited by MYCIN's design, there are many lasting innovations that can be applied for constructing knowledge-based tutors today. These tutors may perhaps use a very different knowledge base representation, but much of the teaching logic developed in the context of the simple, rule-based form of MYCIN will apply.

- The architecture of GUIDON provides a good example of how a complex knowledge-based tutoring program is constructed (Chapter 2). This book describes specifically what *knowledge relations* are used by the tutor (Chapter 3), the *communication model* which keeps track of what has been discussed and helps the tutor focus the dialogue (Chapter 5), and the *student model* which relates student behavior to the knowledge base (Chapter 6).

- In GUIDON teaching knowledge is viewed as a separate, domain-general form of expertise that can be codified in a knowledge base in its own right (Chapter 4). This teaching knowledge base is accumulated incrementally by experimentation, just as a domain knowledge base is developed by testing an expert system on a library of cases. The basic idea is that there are *recurrent conversations* in teaching, allowing knowledge about how to interpret student behavior and how to respond to be formalized to a useful degree. These "discourse patterns" are stated and refined in a way analogous to the "disease and therapy patterns" captured in MYCIN. Thus, this book describes practical *programming methods* for capturing teaching knowledge and refining it according to the knowledge-based paradigm (Appendix D). These methods organize the teaching knowledge so that it is is easy to change and provide a simple explanation capability by which GUIDON can describe how it is managing the dialogue.

- The *content* of GUIDON's teaching knowledge, the set of 200 teaching rules, is a model that can be reused and adapted (Ap-

pendix E). Twenty-six dialogue situations are codified, with a dozen auxiliary sets of rules for selecting alternative dialogue sequences. More than twenty formats for transforming rules into context-dependent quizzes are also described (Appendix B). Stating the teaching knowledge separately, in a reasonably disciplined way, facilitates its later study and abstraction, thus providing a basis for sharing and improving what we have written down.

- Special attention is paid to the *design of an interface language* that enables a student to express many different kinds of initiative (Chapter 5). Even with a simple rule-based system as a foundation, the space of possible student initiatives is impressively broad. As examples of what is possible, this book describes domain-general procedures for responding to over two dozen forms of student initiative, ranging from requests for explanation details, to statements about what part of the problem is solved. While no program can possibly anticipate or understand everything a student might intend, GUIDON's architecture allows us to develop a language of requests and statements that recur in teaching interactions.

In summary, this book attempts to provide a balanced retrospective of the GUIDON project. There is a place for both enthusiasm and caution in evaluating this work. As indicated by the above list, there are many design innovations of direct application for constructing any knowledge-based tutor. Perhaps of most general interest are the limitations of today's expert systems that are revealed by examining what GUIDON cannot say and student behaviors it cannot understand. We are exploiting these lessons in our continuing research. This research and related AI efforts in tutoring, discourse, and knowledge representation are briefly surveyed in the final chapter.

# Acknowledgments

We sometimes joke about the futility of progress in a field in which scientists stand on each other's toes, rather than building upon past work. In developing GUIDON, I was privileged in two ways: First, I was able to build my work upon the MYCIN consultation program, to which more than 20 people contributed their programming and medical expertise over more than 5 years (1972-1979); second, I was advised by some of the most capable researchers in my field.

I thank Bruce Buchanan, my principal dissertation adviser, whose poise and sense of proportion taught me more than just how to do AI research. In reviewing my doctorate research, his efforts to simplify, "reduce to known terms," and "bring the main ideas forward" set standards that I found to be very satisfying.

The medical expertise in MYCIN was provided (chronologically) by Ted Shortliffe (whose thesis suggested the tutorial project to me), Stan Axline, Stan Cohen, Frank Rhame, Rudy Chavez-Pardo, John Foy, Victor Yu, and Bob Blum. A large part of my research involved making an archeological study of the 200 meningitis-oriented rules that Victor wrote (with the help of Larry Fagan, Carli Scott, and Shari Wraith). While I sometimes complain in this book about inconsistencies in these rules, I want to make clear that without Victor's painstaking effort in researching and formalizing the meningitis rule set, GUIDON would not have been possible. After moving away, Victor answered my questions by mail, helping me to make the study that constitutes Chapter 8.

MYCIN programmers and fellow students provided well-written code, which I used throughout GUIDON. They also provided a friendly, if somewhat noisy, office. Randy Davis's technical contributions laid the foundation for much of my work. The sense of analytical style that he conveyed to me was no less important. Jim Bennett and Avron Barr also encouraged me to look beyond my program to the larger issues of transfer of expertise. Carli Scott taught me MYCIN's design and helped to work out some computationally difficult extensions to the code. Bill van Melle served as a programmer's assistant. Over a period of two years, he meticulously answered my almost daily messages about the INTERLISP file package, I/O, and hash-

files. His package for refiling modified code, assembling programs, and documenting changes was essential for developing GUIDON.

John Brown, Adele Goldberg and Terry Winograd served on my PhD oral and reading committees. John started working with me in late 1977, patiently helping me to focus on instructional research problems. His interest and investment of time was at first puzzling, but ultimately taught me something about the importance of the community in science. Adele provided stylistic help and broadened my perspective on previous work in CAI. Terry's interest in stepping back to understand AI's methods and limitations has always been in the back of my mind, but his influence has perhaps now surfaced most clearly in this revision of my original dissertation. I would also like to thank Ira Goldstein for the afternoon at MIT when he sketched out the modules for a knowledge-based tutor, which I adopted for the design of GUIDON.

Finally, I thank my other friends at Stanford during the development of GUIDON, particularly Thomas, Greg, and Jan.

The GUIDON project was supported in part by the Office of Naval Research (ONR) in conjunction with the Advanced Research Projects Agency (ARPA, 1979-1981) and the Army Research Institute (ARI, 1981-1985) under ONR contract N00014-79C-0302. Computational resources have been provided by the Sumex-Aim National Resource (NIH grant RR00785). Continuing research is supported in part by ONR contract N00014-85-K-0305 and by a grant from the Josiah Macy, Jr. Foundation.

Ted Crovello, Elliot Soloway, Peter Szolovits, and Beverly Woolf provided helpful suggestions for revising the original dissertation; Barbara Tall made the initial conversion from PUB format. I am deeply indebted to my TEXnician, Susan King, who has struggled with LATEX for many months, persistently attending to the many details necessary in producing elegant, camera-ready copy. Susan now probably dreams at night of squeezing her entire life into a 4.5 inch line.

William J. Clancey
Palo Alto, California
November, 1986

# Table of Contents