AD-A211 738

DTIC FILE COPY

(4)

Computer-Based Tutors for Explaining and Managing the Process of Diagnostic Reasoning

ONR Final Report

March 1985 - December 1988

DTIC
ELECTE
AUG 16 1989
D
D

William J. Clancey

Senior Research Scientist
Institute for Research on Learning

N00014-85-K-0305

Background

Artificial intelligence (AI) is a branch of computer science that develops programming techniques for modeling processes qualitatively, that is, in terms of explicit representations that describe spatial, temporal, and causal relations among objects, *as opposed to just numeric measures*. Two basic methods exist for representing processes: as a classification of types of processes (e.g., a taxonomy of recurring faults in equipment) and as a simulation, either as a causal state-transition network, or as a hierarchical, structural-functional model. While AI research originally emphasized the representation of problem-solving processes, application to scientific and engineering problems has shifted concern to modeling systems in the world ("Viewing knowledge bases as qualitative models" "Heuristic classification" -- see publications list for full citations).

AI-based instructional programs, often called intelligent tutoring systems (ITS), use these qualitative modeling techniques to represent: 1) processes in the subject domain (e.g., a steam propulsion plant, an electronic circuit), 2) problem-solving processes (e.g., diagnostic strategy, programming methods), and 3) communication processes (e.g., the Socratic method, case-method discourse, and rhetorical principles in explanation) ("Qualitative student models"). Typically, instructional programs may represent only one or two kinds of these processes. When a simulation model of problem-solving processes is incorporated in the program, a basis is provided for evaluating and assisting the student in a very general way. Such programs, which can solve the same problems given to a student, are called knowledge-based tutors (Knowledge-based Tutoring).

Research in the ITS field focuses on the three areas described above: qualitative modeling of system processes, problem-solving processes, and instructional discourse processes. Research involves simulating these processes, involving both empirical work to develop the content of models and

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution unlimited

AI programming to develop techniques for representing the models. Research generally proceeds incrementally, by cyclical critique and re-representation. Research also proceeds by building models on top of each other, in the way GUIDON was constructed on top of MYCIN. In this way, student modeling, explanation, and instructional discourse research has tended to follow advances in modeling problem-solving.

In the late 1980's, a major opportunity has opened for exploiting problem-solving research of the past decade. In particular, it is now clear that when problem-solving processes are factored from the model of the subject domain, a knowledge acquisition (learning) program is better able to identify the causes of problem-solving failures . Of considerable interest to the ITS community, these models of learning are based on a process of explaining problem-solving, using metaknowledge about strategy and knowledge organization.

Better representations and models of learning make this a particularly productive time for implementing knowledge acquisition (learning) programs as the basis of computer tutors. This proposal describes an approach that makes explicit the relation between: 1) the constraints a problem-solver is seeking to satisfy (the form of a good solution), 2) the language used for representing the real-world system being reasoned about, and 3) a psychological model of the problem-solving process. Furthermore, we leverage this additional knowledge about problem solving against the history of interactions. For example, we place the program's failure in the context of a library of previously solved cases or relate a student's inquiry to the earlier context of the interaction in which he had no questions.

Project accomplishments

Early in our research, we identified the importance of representing problem-solving processes in a well-structured procedural language. This enables the explanation and student modeling programs to reason about the problem-solving process, so it can be selectively articulated and identified in student behavior. In a sequence of programs, we demonstrated basic AI techniques for achieving the separation of domain facts from a diagnostic procedure (NEOMYCIN), and the advantages of this separation for explanation and student modeling (IMAGE, ODYSSEUS). A sequence of articles describe the NEOMYCIN program as a psychological model of diagnosis ("Acquiring, representing, and evaluating a competence model of diagnostic strategy"), an architecture for representing strategic knowledge ("Representing control knowledge as abstract tasks and metarules"), and as a general problem-solving method (called "heuristic classification").

The generalization of our work has had a significant impact on expert systems and tutoring research. The article on heuristic classification is now one of the most-cited in the field; the description of AI programming in terms of qualitative modeling has been well-received at many conferences and tutorials [most recently tutorials at AAAI-87 (Seattle), IJCAI-87 (Milan), ECAI-87 (Oslo) and Australia's AI-88 (Adelaide)]; and prominent articles and a book describing the original GUIDON program have highlighted the progress and importance of ONR-sponsored ITS research ("Intelligent tutoring systems: A

tutorial survey" "Qualitative student models" "From Guidon to Neomycin and Heracles in twenty short lessons: ONR final report, 1979-1985.")

In 1984 we proposed to develop a family of tutoring programs built upon NEOMYCIN, or more precisely, the general shell out of which it is constructed, HERACLES. Research programming accomplishments in the past three years include:

❑ Completion of GUIDON-WATCH, a program that integrates the HERACLES consultation program with a graphics browser and explanation program, refined through a series of student trials ("GUIDON-WATCH: A graphic interface for viewing a knowledge-based system"). The most important window is the "situation-specific model" which shows a diagnosis in the form of a proof graph, with abnormal findings explained by successively more specific causal and subtype descriptions of processes towards the top of the graph. Other key windows include: the line of reasoning displayed as a "task stack" and a summary of evidence displayed as a table.

❑ Re-implementation of the explanation program with improved bookkeeping of consultation reasoning, enabling "roll back" of GUIDON-WATCH's display to any moment during a previously run consultation (for student exploration) and explanation of reasoning at the level of individual metarule clauses (replacing hand-coded text strings associated with metarules). Menus created dynamically, in the context of particular student inquiries, bring together relevant related information, enabling a student to get detailed information without browsing through the extensive menu system.

❑ Implementation of a script-based graphics display program, GUIDON-TOURS, a facility for automating subject matter lectures or system documentation. An interactive program allows the script writer to design a sequence of window displays and text, which a student or programmer plays back, pausing and exploring the display at will.

❑ Completion of a tutoring system, called GUIDON-MANAGE, in which HERACLES carries out a student's diagnostic tasks (e.g., to test a particular hypothesis or determine the implications of a new piece of problem information). Research has focussed on interpreting the diagnostic process to carry out the tasks (trapping at intermediate levels of detail that we expect a student to detect and complete on his own) and to provide problem-solving assistance (simulating what the program would do next). Research has also considered, but not solved, the problem of providing appropriate feedback so the student understands what each command to the program accomplished.

❑ Extension of GUIDON-MANAGE to indicate changes to the situation-specific model resulting from each student command, and implementation of the explanation program within GUIDON-MANAGE to allow explanations of what a task did and why a task is being suggested (in response to a student request for help).

Student trials of GUIDON-MANAGE suggested a variety of possible extensions and improvements.

per ltr

A-1

❑ Demonstration of a prototype knowledge-acquisition program, GUIDON-DEBUG, that integrates the GUIDON-WATCH browsing capabilities with the ODYSSEUS modeling program and grapic editing ("GUIDON-DEBUG: The student as knowledge engineer"). By this design, an edited consultation typescript is analyzed by ODYSSEUS to determine how the knowledge base would have to be modified to produce the new sequence of data requests, consistent with the diagnostic procedure.

❑ Extension of ODYSSEUS to incorporate metarules in its model of student reasoning, rather than just sequences of diagnostic tasks. (Thus replacing hand-coded heuristics and enabling the program to read the metarules to make analyses required by GUIDON-DEBUG.)

❑ In GUIDON-DEBUG, representation of tasks to make explicit the constraint each seeks to accomplish in modifying the situation-specific diagnostic model. Proof of concept demonstration of detecting failure to satisfy a constraint and reading the metarules to conjecture missing domain knowledge (leading towards full implementation of the theory of learning in the GUIDON-DEBUG knowledge acquisition program, and providing the basis for proposed instructional research).

❑ Demonstration of HERACLES' generality through the development of a knowledge base for sandcasting diagnosis called CASTER (faults in iron cast in sand molds) ("A qualitative modeling shell for process diagnosis").

❑ Extension of the task/metarule (procedural) language, allowing the explanation program to be represented in the same language (so potentially it can reason about itself). Implementation of a compiler to combine already compiled metarules into Lisp functions for each task (enabling the explanation program to run more quickly and avoid stack overflow).

❑ Re-implementation of HERACLES and porting to Xerox-Common Lisp, repartitioning knowledge files so domain-general components are better separated from specific knowledge bases. This basic system maintenance will allow us to export the program to other sites, as well as to provide a foundation for re-using and extending the procedural language in future research (e.g., solving design problems or adapting the program to use an agenda).

Comparison of these accomplishments to our proposal reveals a mixture of about 40% unexpected new ideas and projects (e.g., the situation-specific model and capability to edit it, "roll back" display capability, dynamic explanation menus, GUIDON-MANAGE feedback and assistance by simulation of HERACLES), 30% re-implementation to provide a better foundation or to enhance the original design (e.g., completely recoded display system following object-oriented principles, completely new bookkeeping for explanation), and 30% implementation of proposed projects (e.g., new windows and trials of GUIDON-WATCH, clause-level explanation, and GUIDON-MANAGE). In retrospect, the three-year plan looks now like a decade of work, which isn't too surprising, considering that designs for six tutoring programs were described.

In summary, in developing AI-based instructional programs, we are developing principles for the design of expert system shells, cognitive modeling, and knowledge acquisition. We are exploiting graphics for constructing knowledge bases and browsing reasoning. We are developing

models of physical systems (e.g., physiology and sandcasting), called knowledge bases. We are developing a language in which the domain knowledge is representing declaratively in first-order predicate calculus and the inference procedure (so-called control knowledge or strategy) is representing procedurally (as tasks and metarules); this enables us to model the factual and strategic components of human reasoning. We are relating knowledge acquisition to student modeling ("An overview of the ODYSSEUS learning apprentice"), showing how explanation and debugging capabilities are integral parts of interpreting behavior and completing an interpretation (learning). A subproject focuses on text generation for lines of reasoning explanations and summaries; extending previous work in natural language generation.

Finally, we have articulated general theories to support or justify design considerations in our programs. These theories begin as abstractions (generalizations) of existing programs; observed patterns are then restated as principles or rationales that could generate them. For example, this is the approach we took in reformulating MYCIN's knowledge base into NEOMYCIN and then abstracting the method of heuristic classification from this. The general drift is to describe AI programming as a methodology for representing models of processes; in the extreme, we relate these models to what people know and how they learn by considering the nature of representation ("The knowledge engineer as student: Metacognitive bases for asking good questions").

With this vision in mind, our research in the last period focused more narrowly on learning and explanation, rooted in the specific knowledge acquisition program we will construct. GUIDON-DEBUG thus serves as a model of how good questions are generated by the failure to understand some process, both the abnormal real-world process being diagnosed and the program behavior the student is trying to understand. The relation between debugging and explanation is hardly more than a metaphor at this point; we are continuing to pursue the connection by showing the limitations and advantages of using GUIDON-DEBUG as an underlying model of learning.

Implications for AI. As always, our work has very general implications for AI research, going well beyond the specific emphasis on learning, explanation, and teaching. In particular, the research has significance for advancing the state of the art in representation, particularly of procedural knowledge. The following aspects of our work are relevant to the expert systems/robotics research program as a whole:

1) We have demonstrated that *knowledge engineering is a kind of engineering* by designing expert systems so they can detect the boundaries of their capabilities, specifically to detect limitations in their conclusions. (GUIDON-DEBUG work is formalizing a theory of diagnosis in the form of constraints a diagnostic model should satisfy.) This is particularly important for robots that can't communicate with humans.

2) We have *related classification models of abnormal processes to more complete structural/functional models of devices.* Our work shows that you need both for modeling open systems. GUIDON-DEBUG provides an explanation-based learning approach for integrating the two forms of knowledge. Upon detecting possible limitations (failed diagnostic constraints),

the program could test changes to its classification model by testing them against separate knowledge about how the device/system normally behaves, what it is designed to do, and domain-general knowledge about the kinds of processes involved (e.g, knowledge about flows, feedback). Many researchers working with electronic and mechanical systems do not realize that classification knowledge is essential; they view it as inferior. Also, many people (e.g., Davis at MIT) view classification knowledge as a fallback, which we believe is backwards.

3) We have analyzed how *NEOMYCIN's diagnostic procedure derives from a theory of diagnosis, knowledge about the world (types of cases), and the nature of communication with the world (data gathering, test methods, interaction with people).* NEOMYCIN/HERACLES remains the only AI project in the expert systems community, among programs using classification, that has an explicit, inspectable inference procedure. This well-structured procedural language allows an EBL program (GUIDON-DEBUG) to reason directly through the inference procedure used by the performance program. The resulting formal framework and methodology will be useful for programs that do control, design, interpretation, etc., not just diagnosis. Since a robot will face many similar problems (e.g., diagnosis in different "domains"), developing multiple knowledge bases will be greatly aided by the principles we establish here for the design of reusable inference procedures.

Oral Presentations 1985-88

Major Invited Talks and Symposium Presentations:

1985

International Professorship in Computer Science (Expert Systems), Universite de L'Etat, Belgian National Foundation for Scientific Research, Namur, Belgium, May 6-15, 1985, twenty hours of lectures.

Conference on Learning Issues for Intelligent Tutoring Systems. Tuebingen Institute for Foreign Studies, West Germany, June 1985.

Tutorial Speaker, Tutorial on Expert Systems, Parts I and II, IJCAI-85, Los Angeles, August 1985.

Conference on Artificial Intelligence in Medicine, Pavia, Italy, September 13-15, 1985.

Main tutorial speaker, Tutorial in Knowledge-Based Systems, Swiss Group for Artificial Intelligence and Cognitive Science, Lugano, Switzerland, September 30 - October 4, 1985, nine hours of lectures.

Future Medical Systems Forum, Monte Carlo, Monaco, November 4-6, 1985.

1986

Tutorial on Information Management in the Academic Medical Center: Strategic Options and Opportunities, Association of American Medical Colleges (AAMC), San Diego, January 28, 1986.

Committee on Cognition and Survey Research, Social Science Research Council, Stanford Center for Advanced Studies in the

Behavioral Sciences, March 1, 1986.
ICAI Research Workshop, Windermere, Cumbria, England, sponsored by the Alvey Directorate and AAAI, April 12-15, 1986.
Keynote speaker, Trends in Computer-Assisted Education, 6th Conference in the series "Computers in Higher Education," University of Lancaster, England, April 15, 1986.
Macy Foundation Symposium on Cognitive Science and Medicine, Montreal, September 11-12, 1986.
Program Organizer, Higher-Level Tools Workshop, Ohio, October 7-8, 1986.
Program organizer, Knowledge-Acquisition Workshop, Banff, Canada, November 2-7, 1986.

1987

IEEE/Applications invited talk, Orlando, February 1987.
Tutorial Speaker, Evaluation of Expert Systems, AAAI-87, July 1987.
Tutorial Speaker, Second Advanced Course on AI, Norway, August 1987, six hours of lectures.
Tutorial Speaker, Knowledge-Based Tutoring, IJCAI-87, Milan, August 1987.
Invited Speaker, AI-87, Osaka, Japan, October 1987.

1988

Main Tutorial Speaker, "A Perspective on Knowledge Engineering," Inter Access, The Hague, Netherlands, February 1988, eight hours of lectures.
Keynote address, "Frontiers in AI Research," Forest Simulation Conference, November 1988.
Tutorial Speaker, "Knowledge Engineering Methodology," AI-88, November 1988.
Keynote address, "Reconsidering the Foundations of Intelligence," AI-88, Adelaide, Australia, November 1988.
Keynote address, "Reconceiving the Role of AI in Education," ASCILITIE (Computers in Higher Education) Conference, Canberra, Australia, December 1988.

Conference presentations

"Expert systems and cognitive psychology: Implications for medicine," AERA Annual meeting, Chicago, April 1985.
"The Cognitive Sciences in Medicine: Nature, Acquisition, and Application of Medical Reasoning Skills," AERA Annual meeting, San Francisco, April 1986.
"Intelligent Tutoring Systems," Cognitive Science Society, Seattle, July 1987.
"What's wrong with ITS Research?" ITS-88, Montreal, June 1988.
"State of AI Research," Program Committee Panel, AAAI-88, August 1988.

Colloquia

USC-ISI, March 1985.
UC Berkeley, SUSE colloquium, October 1985.
CMU, Computer Science colloquium, November 1985.
SF State University, Computer Science colloquium, December 1985.
Stanford University, seminar "AI Applications in Design and Manufacturing," April 1986.

Signaal, Hengelo, Netherlands, CS Forum, April 1986.
Bolt, Beranek, and Newman, Cambridge, colloquium, July 1986.
Texaco, Houston, CS Forum, November 1984, October 1985, November 1986.
Proctor & Gamble, Cincinnati, CS Forum, October 1986.
New Mexico State University, Las Cruces, January 1987.
National Space Sciences Educational Foundation, Stanford, July 1987.
Fujitsu, Tokyo Japan, CS Forum, November 1987.
CSK/CRI, Tokyo Japan, CS Forum, November 1987.
Stanford University, three seminars on current research, Spring 1988.
University of Colorado, Cognitive Science Colloquium, April 1988

Publications 1985-88

Book

Clancey, W.J. (1987)
Knowledge-Based Tutoring: The GUIDON Program.
Cambridge: MIT Press.
Revision of Transfer of rule-based expertise through a tutorial dialogue.
Computer Science Doctoral Dissertation, Stanford University, STAN-CS-769, 1979.

Journal Articles

Richer, M. and Clancey, W.J. (1985)
GUIDON-WATCH: A graphic interface for viewing a knowledge-based system.
IEEE Computer Graphics and Applications, 5(11):51-64.
Also STAN-CS-85-1068, KSL 85-20.

Thompson, T. and Clancey, W.J. (1985)
A qualitative modeling shell for process diagnosis.
IEEE Software 3(2):6-15, March 1986.

Clancey, W.J. (1985)
Heuristic classification.
Artificial Intelligence, 27:289-350.
Also in Knowledge-based Problem Solving, J. S. Kowalik (editor).
Prentice-Hall, Inc., 1985.
Also STAN-CS-85-1066, KSL 85-5.

Clancey, W.J. (1986)
Viewing knowledge bases as qualitative models.
To appear in IEEE/Expert, 1989.
Also KSL-86-27.
[Presented at Symposium for AI and Simulation, Tucson, January 1987.]
[Invited talk presented at IEEE/Applications, February 1987.]

Book Chapters

Clancey, W.J. (1985)
Software tools for developing expert systems.
In Artificial Intelligence in Medicine, I. De Lotto and
M. Stefanelli (editors), Elsevier Science Publishers.

Wilkins, D.C., Clancey, W.J., Buchanan, B.G. (1986)
An overview of the ODYSSEUS learning apprentice.
In Machine Learning: A Guide to Current Research,
T.M. Mitchell, J.G., Carbonell, and R.S. Michalski (editors).
New York, Academic Press, pages 369-373.
Also KSL-85-26.

Clancey, W.J. (1986)
Qualitative student models.
Annual Review of Computer Science.
Palo Alto: Annual Reviews Inc., pages 381-450.
Also excerpted in Artificial Intelligence and Human Learning:
Intelligent Computer-Aided Instruction,
J. Self (editor), London: Chapman and Hall, 1987, pages 49-68.
Also KSL 86-16.

Clancey, W.J. (1987)
From Guidon to Neomycin and Heracles in twenty short lessons:
ONR final report, 1979-1985.
In Current Issues in Expert Systems,
A. vanLamsweerde (editor), London: Academic Press, Inc., pps. 79-123.
Also The AI Magazine, 7(3):40-60, Conference, 1986.
Also KSL 86-11.

Clancey, W.J. (1987)
Intelligent tutoring systems: A tutorial survey.
In Current Issues in Expert Systems,
A. vanLamsweerde (editor), London: Academic Press, Inc., pps. 39-78.
Also KSL 86-58.

Clancey, W.J. and Bock, Conrad. (1988)
Representing control knowledge as abstract tasks and metarules.
In Computer Expert Systems, L. Bolc and M.J. Coombs (editors).
Heidelberg: Springer-Verlag, pps. 1-77.
Also KSL 85-16.
To appear in revised form in Artificial Intelligence

Wilkins, D.C., Clancey, W.J., and Buchanan, B.G. (1988)
On Using and Evaluating Differential Modeling in Intelligent Tutoring and Apprentice Learning Systems.
In Intelligent Tutoring Systems: Lessons Learned,
J. Psotka, D. Massey, and S. Mutter (editors).
Lawrence Erlbaum Publishers.
Also KSL-86-62.

Clancey, W.J. (1988)
The knowledge engineer as student: Metacognitive bases for asking good questions.
In Learning Issues in Intelligent Tutoring Systems,
H. Mandl and A. Lesgold (editors), Springer-Verlag.
Also KSL 87-12.

Clancey, W.J. (1988)
An authoring shell for apprenticeship learning.
In (title tbd),
F. Reif (editor), (publisher tbd), in preparation.
Also Proceedings of ITS-88, Montreal, June.

Conference articles

Clancey, W.J. (1985)
Expert system design: Lessons from applications to education (abstract)
IEEE Spring Computer Conference, San Francisco, page 115.
[Co-recipient of "Best session of the day" award.]

Wilkins, D.C., Clancey, W.J., and Buchanan, B.G. (1985)
Odysseus: A Learning Apprentice.
Proceedings of the Third International Machine Learning Workshop
pages 221-223, June.

Clancey, W.J., Richer, M., Wilkins, D.C., Barnhouse, S., Kapsner, C.,
Leserman, D., Macias, J., Merchant, A., and Rodolitz, N. (1986)
GUIDON-DEBUG: The student as knowledge engineer.
KSL Working paper 86-34.

Clancey, W.J. (1986)
Know-how vs. knowledge representation (extended abstract)
Proceedings of the Workshop on Knowledge Compilation,
Oregon State Technical report. September 1986, pages 1-2.

Wilkins, D.C., Clancey, W.J., and Buchanan, B.G., (1987)
Knowledge Base Refinement Using Abstract Control Knowledge.
January, KSL-87-01.

Wilkins, D.C., Buchanan, B.G., and Clancey, W.J., (1987)
The Global Credit Assignment Problem and Apprenticeship Learning.
January, KSL-87-04.

Clancey, W.J. (1988)
Detecting and Coping with Failure (extended abstract).
Proceedings of the Symposium on Explanation-Based Learning,
AAAI Spring Symposium Series, March.

Book Reviews

Clancey, W.J. (1984)
Review of Weiss and Kulikowski's "A Practical Guide to Designing Expert Systems."
The AI Magazine, 5(4):84-86, Winter, 1985.

Clancey, W.J. (1985)
Review of Sowa's "Conceptual Structures:
Information Processing in Mind and Machine."
Artificial Intelligence 27(1):113-124.
Also STAN-CS-85-1065.

Clancey, W.J. (1987)
Review of Winograd and Flores's "Understanding Computers and Cognition"
A favorable interpretation.
Artificial Intelligence, 31(2), 232-250, February.
Also KSL-86-48.