1993-01-16

Accounting Tutor: Intelligent Computer-Aided Instruction

Rayman D. Meservy
Brigham Young University

Accounting Tutor:
Intelligent Computer-Aided Instruction

by
Rayman D. Meservy
School of Accountancy and Information Systems
Brigham Young University

January 16, 1993

Initial draft prepared for the SOAIS workshop, January 21, 1993. This paper represents the beginnings of a new stream of research that I have been interested in for some time.

Please do not quote.

Abstract

This study investigates several issues central to the successful development of intelligent tutoring systems: (1) representing domain knowledge, (2) representing instructional knowledge, (3) modeling student learning, and (4) developing user-friendly interfaces. The focus of domain learning tasks includes accounting basics, often covered in the first few chapters of an accounting principles course.

Introduction

Educators have long sought methods of helping students learn more effectively and efficiently, in and out of the classroom. Students often seek the help of student tutors. Tutors interact with the students on a one-to-one basis, explaining basic concepts, showing students how to solve problems, watching and correcting students as they solve problems, and evaluating progress. Research by educational psychologists indicates that private tutoring, both student and computer, is even more effective than conventional classroom instruction in which a student listens to lectures, reads tests and solves homework problems. [Delaney, 1985] An example is the LISP tutor at Carnegie Mellon University.

Field studies at Carnegie Mellon University, showed LISP tutor helped college students learn programming faster and more effectively than other methods. In one comparison, students were tutored by graduate and undergraduate Carnegie Mellon students with considerable experience in both LISP and tutoring. Another group of students read the instructional material and solved the same sequence of problems on their own, with help from a teacher only when they were stuck. John Anderson noted that "the basic finding was that the human tutor is best, the computer is not that far behind, and the traditional on-your-own practice conditions were much worse." The advantages of private tutoring by human or computer, relative to the on-your-own condition, increased with the difficulty of the material. The student subjects took an average of 11.4 hours with the human tutors, 15 hours with the computer tutor, and 26.5 hours with the on-your-own condition. Students in traditional classroom settings indicated that they took over 40 hours to cover the material. In another study, students who had used the LISP tutor did 43 percent better on a final examination than those who had conventional instruction. [Delaney, 1985]

Some students struggle to learn accounting. A course in accounting principles is required in most business schools. Unfortunately, many students have painful memories of learning accounting. Too often in the fundamental courses, classes are large and sufficient tutors are unavailable. It could be effective to leverage individualized instruction, even augmenting conventional classroom instruction with intelligent computer-aided instruction

(ICAI). Creating an intelligent computer requires knowledge that human tutors must have, including domain problem-solving knowledge, instructional techniques, and recognition of difficulties a typical student may encounter. Furthermore, a good user-machine interface is important.

To effectively observe student progress, discover errors and offer instruction, a tutor must be able to solve the problems a student is working on in a manner similar to how an ideal student would solve problems. This "ideal model" represents the accounting knowledge we want the student to acquire. The accounting tutor follows along as the student solves the problem, symbol by symbol, trying to figure out what correct or faulty rule would have led to that input. If it appears that the student is solving the problem correctly, the tutor only observes and offers encouragement, but if an error is identified, the tutor may decide to interrupt with appropriate remedial instruction. Concurrently, the computer is building an internal model of the student's progress, skills and difficulties, and how the student prefers to learn. The Accounting Tutor can thus tailor it's instructions for that particular student.

Finally, the student is provided with a user-friendly windows environment, with tools which facilitates accounting problem-solving. The student has considerable control over the environment, selecting the degree of computer-aided instruction, amount of help to be provided, and what objects will be seen and how they will be arranged on the screen. In addition, to do this in accounting requires a workspace similar to accounting workpapers or a computerized accounting package.

Additional Motivation

More than just building another potentially useful program, Anderson [1984] noted that building an intelligent tutoring system is a valuable research tool for the study of mental processes that are involved in learning. Thus, modelling the student and investigating tutorial strategies provides insights into interesting psychological questions as well. Some of these questions are: How can an individual student be helped to learn effectively and efficiently? What exactly is 'learning'? What styles of teaching interactions are most effective and when? ICAI is important to education because it tries to answer these questions. [Self, 1988] It has also been argued that the intelligent models may be considered as theories which can be tested. [Meservy et. al., 1985]

Background

Many educational researchers consider computers to be the most important technological development for education since the invention of the printing press or writing itself. They believe that computers will one day transform education, leaving their mark on education as writing and books have done. Even though computer-aided instruction (CAI) has been used for many years, results at the college level have been mixed, although generally moderately positive, reducing instructional time while increasing learning effectiveness. [Kulik, 1990]

Early versions... (to be inserted)

Recent research...(to be inserted)

Accounting related ICAI...(to be inserted)

The following explains how the rest of the paper is organized. The paper first describes the various components of the Accounting Tutor system and how they interact. The paper then examines some of the strategies that were used to address various issues and problems encountered. A classroom experiment is then conducted along with other tests examining various aspects of the validity of the model. Results of study are reported, and conclusions, including future research, are suggested.

The Model

The model for the Accounting Tutor is analogous to several experts sitting around a blackboard and computer terminal in a small room. As they watch the input on the computer screen from a student solving problems, the experts write various notes on the blackboard as they also solve the problem. These notes include not only correct solutions, but also problems and/or successes the student seems to be having, the students individualized learning approach, remedial and expansive instructional material, tailored learning strategies, and additional accounting problems for the student to work. Deciding as a group how they can best help the student, they then place the appropriate messages on the computer. At times, these messages may be simple positive or negative feedback. At other times, the messages may include more detailed definitions, rules, or hints. The system may decide to restart the problem and demonstrate the correct solution, and then provide the student with additional problems to solve. The system will also respond to a student's request for various levels of help.

The experts, described below, consist of a Student Modeler, a Problem Solver, a Diagnoser, and a Tutor. Modeling the system as a number of individual experts allows departmentalization of knowledge, decoupling a number of interactions leading to complexity. It also allows easier modification or addition of experts in the system. The first expert in the system models the student, keeping track of the students knowledge, problems, and learning strategies.

The Student Modeler

Intelligent tutoring systems generally include some simulation of current student knowledge. A student model represents the knowledge a student has learned and the difficulties he has had in learning. Student models differ in whether they explicitly model the individual learning history or not. Shank considers knowledge to be non-formal and primarily case-based, which may be generalized upon demand. Building on this foundation, the student model will be episodic in nature, incorporating the case-exercises the student has encountered, previous instructions and helps received, and problems identified. This differs from more static models (e.g. differential learner models). A case-based student model allows the tutor system to offer examples based on reminders of earlier problems the student has successfully solved. Generalization of these episodes may also reflect the student's learning process. [Weber et. al., 1988]

Some student models are based only on the work accomplished in a given session, on the basis of rules that detect bugs and corrects solutions. [Anderson and Reiser, 1985] Other models summarize what concepts were learned and how effectively they were applied. [Burton and Brown, 1982] Our student model will incorporate both rules and a case-based model proposed by Weber et. al. [1988] in which knowledge is considered non-formal and dynamic. Expert knowledge is sometimes represented as a formal collection of rules. In contrast, knowledge can also be represented as a set of examples or cases based on previous experience from which an expert generalizes to construct rules as needed. Keeping track of both the student's history and current skills will allow a wide range of tutorial responses built on both generalization and specific behavior. It will also allow the tutor to remind the student of specific examples, which the student has already completed, in order to assist the student in the generalization process. Summarization for research and grading purposes is also possible.

The Problem Solver

The problem solver solves the accounting problem at the same time the student is solving the problem on the screen. Initially, the problem solver may use pre-solved (template) solutions to the problems, however, additional knowledge about the accounting domain and accounting rules will continue to be added, so that the problem solver can examine most common accounting problems, applying the appropriate rules to solve the problem. Adding such knowledge will greatly expand the versatility of the Accounting Tutor.

The Diagnoser

Diagnosis plays a distinctive role in instructional programs. The diagnoser works hand-in-hand with the student model and problem solver. The diagnoser watches the student's actions, looking for discrepancies from the ideal solutions produced by the problem solver. Potential problems and difficulties are recognized. It then tracks these discrepancies back to faults in the student's presumed world model or inference procedure. The diagnoser interprets the performance of the student, making judgements relating to the student's strengths and weaknesses. The diagnoser then tries to 'repair' the student by passing this information on to the tutor. In cognitive systems, the process of causally tracking backwards from the discrepant reasoning behavior to the hidden faults is called *diagnostic modelling*. [Clancey, 1986] The diagnoser also is programmed to recognize various student learning strategies. Long-term results are stored with the student model.

The Tutor

Results of performance interpretation are transferred to the tutor if need is inferred from student's performance, otherwise only system state and user models are updated. The recognized needs and resulting functions of the tutor are similar to those found in EUROHELP tutor. [see Breuker, 1988] When the student exhibits specific needs, the following tutor functions are available:

Need	Function
Lack of basic concepts	Initial instruction
Lack of potentially relevant concepts	Expansion
Lack of Information of correct action	State feedback
Error/misconception	Remedial

The *initial instruction* functions set forth the basic concepts and principles, and may be presented by the tutor, or requested by the user. *Expansion* is aimed at instructing new concepts that may augment the student's understanding, usually through additional problem solving and comment. Besides the basic practice of learning-by-doing, expansion helps students refine their expertise. Expansion differs from the initial instruction by proceeding in small steps, with an acquired understanding of what the student knows. *State feedback* provides information to the student about where they are in the problem and suggests goals of what needs to be done next. As with other tutorial systems, identified errors and misconceptions result in *remedial* learning action.

The Structure of Goals in Accounting Tutor

The tutor's structure consists of three layers: *didactic* (educational) *goals, strategic decisions,* and *tactical methods*. [Woolf and McDonald, 1984] These three layers provide control over tutorial actions taken. Although help requested is based on local problems and apparently simple questions, what will be taught is not only dependent on current needs and skills, but also on long-term didactic goals. These goals lead the computer to decide whether simple, straight-forward advice will be given and/or new concepts will be taught. [Breuker, 1988] The *didactic goals* are represented in the form of semantic nets. The semantic net includes both (1) the order in which concepts are to be learned and (2) reference to the expansion of knowledge required.

At the *strategic level*, decisions include whether or not to interrupt the user, and what sequence of tactics should be applied. This decision is not only dependent on current performance, but interruptions also may be set by the user. For example, one user may want to be interrupted only for the most serious errors, while another may desire feedback on each step taken. A user may always override the advice, for instance when they already realize what they did wrong. Interrupt levels represent a style of tutoring, which along with levels of specificity can be set by the user.

Tactics are represented by structural text frames, where slots may be inserted with the value of objects based on the current problem. This is similar to the definition of tactics used in Geometry and LISP tutors by Anderson et. al. [1985]. To the user, tactics appear to be similar to context-sensitive help windows.

The Accounting Platform

The accounting platform provides much of the computer screen medium on which the student will solve numerous accounting problems. The platform contains not only what is normally considered a full accounting system with the required financial statements, ledgers, and journals, but also allows a considerable amount of input flexibility and user-friendly interfaces. The platform allows for small end-of-chapter textbook problems or large case studies, with the ability to save intermediate results. Besides automatically calculating all math and ratios as needed, the system provides the flexibility of pointing to amounts in the problem case window and automatically picking up these amounts with a mouse, dragging, and then dropping them onto the appropriate journal, ledger, or financial statement. The updated results in all related financial accounts are immediate, allowing the student to see the flow-through to the financial statements. If appropriate permission is set, amounts may be dropped directly on the financial statement line-items (accounts), updating concurrently the journals and ledgers. This is possible because financial statements, journals, and ledgers only represent views of the underlying data entries. The platform is implemented with an object-oriented design, allowing considerable domain knowledge to be stored and inherited in the underlying classes. It is hoped that such a platform will facilitate the learning process by eliminating some of the mundane tasks often associated with bookkeeping.

The User-Interface

The user-friendly interface is achieved by using a windows environment, with numerous buttons for setting up the student's own personalized desktop on screen. A large variety of windows can brought-up, sized, and placed on the screen where the student desires. Most of these windows also contain attributes that can be set, such as color. In addition to various accounting platform windows, other windows available include the problem, current problem goals and status, student work summaries, various help and hint windows, tutorial windows, student notes and scratchpad windows. Other buttons suggest to the tutor the amount of help the student desires.

Special Problems and Strategies (to be inserted)

Tuning and refining.

Each learning concept will be systematically documented to determine accuracy of student learning models, problem areas, and effective tutorial strategies. Observations of student interactions with the Accounting Tutor will also be made.

The Experiment (to be inserted)

Results (to be inserted)

Conclusion (to be inserted)

Future Plans

Future plans include the investigation of "failure-driven learning." Erroneous strategies, missing concepts, and faulty generalizations result in failures which guide the reorganization of structures in episodic memory. Sometimes using analogies of the student's own learning history should be offered by the tutor in order to help the student learn from those mistakes. [Weber et. al., 1988]

Bibliography

Anderson, J.R., "Cognitive Psychology and Intelligent Tutoring," Sixth Annual Conference of the Cognitive Science Society, Boulder, Colorado, 1984.

Anderson, J.R., C.F. Boyle, and B.J. Reiser, "Intelligent Tutoring Systems," Science, pp. 228, 456-462, 1985.

Anderson, J.R., and B.J. Reiser, "The LISP tutor," Byte, 10, 159-75, 1985.

Breuker, J., "Coaching in Help Systems," in Artificial Intelligence and Human Learning, edited by John Self, Chapman and Hall, 1988.

Burton, R.R., and J.S. Brown, "An Investigation of Computer Coaching for Informal Learning Activities," Intelligent Tutoring Systems, eds. D.H. Sleeman and J.S. Brown, Academic Press, New York, 1982.

Clancey, W.J., "The Role of Qualitative Models in Instruction," Annual Review Computer Science, 1, Annual Review Inc., 1986.

Delaney, E., "Computer Tutors and the Ideal of Individual Instruction for Everyone," Carnegie Mellon Magazine, Vol. 4, No. 2, 1985.

Kulik, C.C. and J.A. Kulik, "Effectiveness of computer-Based Education in Colleges," in Research in Instructional Effectiveness, edited by E.K. St.Pierre, M.P. Riordan, and D.A. Riordan, published by Center for Research in Accounting Education, James Madison University, Harrisonburg, VA., 1990.

Meservy, R.D., A.D. Bailey, and P.E. Johnson, "Internal Control Evaluation: A Computational Model of the Review Process," Auditing: A Journal of Practice and Theory, Vol. 6, No. 1, Fall 1986.

Self, J., preface to Artificial Intelligence and Human Learning, Chapman and Hall, 1988.

Weber, G., G. Waloszek, and K.F. Wender, "The Role of Episodic Memory in an Intelligent Tutoring System," in Artificial Intelligence and Human Learning, edited by John Self, Chapman and Hall, 1988.

Woolf, B. and D.D. McDonald, "Design Issues in Building a Computer Tutor," IEEE Computer, September, pp. 61-73, 1984.