12th Midwest AI and Cognitive Science Conference, Oxford OH, 2001, pp. 16–23

CIRCSIM-Tutor: An Intelligent Tutoring System Using Natural Language Dialogue*

Martha W. Evens
Computer Science Department, IIT, Chicago, IL 60616, evens@iit.edu
Stefan Brandle
Taylor University, Upland, IN
Ru-Charn Chang
Baxter Laboratories, North Chicago, IL
Reva Freedman
Northern Illinois University, DeKalb, IL
Michael Glass
University of Illinois, Chicago, IL
Yoon Hee Lee
Institute for Defense Analysis, Seoul, Korea
Leem Seop Shim
Trinity University, Palos Heights, IL
Chong Woo Woo
Kookmin University, Seoul, Korea
Yuemei Zhang
Little Canada, MN
Yujian Zhou
Webex.com, San Jose, CA
Joel A. Michael and Allen A. Rovick
Rush Medical College, Chicago, IL 60612

March, 2001

*This work was supported by the Cognitive Science Program, Office of Naval Research under Grant No. N00014-94-1-0338. The content does not reflect the position or policy of the government and no official endorsement should be inferred.

Abstract

CIRCSIM-Tutor uses natural language for both input and output. It can handle a variety of syntactic constructions and lexical items, including sentence fragments and misspelled words. It conducts a conversation with a student to help the student learn to solve a class of problems in cardiovascular physiology dealing with the regulation of blood pressure. We describe the lessons learned about knowledge representation, input understanding and text generation and how they influence the design of the next generation of our text-based ITS, CIRCSIM-Tutor Version 3, now under construction.

1 Introduction

CIRCSIM-Tutor is an intelligent tutoring system for the domain of cardiovascular physiology, which carries out a natural language dialogue with the user, using a set of tutoring tactics that mimic those employed by two expert human tutors. CIRCSIM-Tutor has been used extensively by students at Rush Medical College, and the learning outcomes of a one hour interaction with the program have been demonstrated. Student response was positive. Here we describe what the system does and how it does it.

In November, 1998, fifty medical students tried the system in a two hour laboratory with pre-tests and post-tests and questionnaires. The students showed marked improvement from pre-test to post-test. What is more, the improvement appeared not just on CIRCSIM-like problems but carried over to different kinds of reasoning problems. Twenty-four students used the system in pairs and twenty-six were alone at a machine. The majority of the students completed four procedures. The students expressed a great deal of enthusiasm in a survey we administered immediately after their session; they also made a number of very useful suggestions.

What is more, the system did not fail under the onslaught and the new input understander was able to respond to over 95% of the inputs. When the system did not understand the student, it described the kind of input it was expecting in terms the student could understand.

2 The Tutoring Domain – Cardiovascular Physiology

One of the important topics that beginning medical students must learn is how blood pressure is regulated in the human body. Human life is only compatible with a specific range of blood pressures. When something happens to change the blood pressure, such as increasing or decreasing the volume of blood in the body, the body must compensate. The negative feedback loop that controls this process, known as the *baroreceptor reflex*, can be a difficult topic for students.

CIRCSIM-Tutor is based on a qualitative model of blood pressure regulation involving seven key physiological parameters: Central Venous Pressure (CVP), Stroke Volume (SV), Cardiac Output (CO), Heart Rate (HR), Inotropic State (IS), and Total Peripheral Resistance (TPR). The nervous system plays an essential role in blood pressure through its control of the three variables IS, HR, and TPR, which are called *neural*

variables. The fact that neural and non-neural variables behave differently is a key fact in understanding blood pressure regulation.

Students use CIRCSIM-Tutor to learn to solve problems like those presented in their first year physiology course. In these problems, students are asked to predict the value of the seven core parameters at three points in time: the 'Direct Response' (DR) stage immediately after the precipitating event, the 'Reflex Response' (RR) stage after the nervous system has responded, and after the establishment of a new 'Steady State' (SS).

When students start CIRCSIM-Tutor, they are presented with a perturbation to the cardiovascular system such as a hemorrhage or a broken pacemaker. Then they are asked to make qualitative predictions about the direct response of the seven core variables - tell whether each will increase, decrease, or stay the same. The system assesses those predictions and starts a tutoring dialogue on the screen to remedy any errors found. Once that dialogue is complete, the system asks for predictions for the RR phase and carries out another dialogue. Finally, the process is repeated for the SS stage.

The qualitative model and the tabular format for predictions are derived from CIRCSIM, an earlier CAI system which analyzed the student's predictions and used canned text to explain the student's conceptual errors. The goal of the CIRCSIM-Tutor project is to investigate how the use of an interactive format and the use of natural language as part of that format can enhance student learning.

3 Obtaining Human Tutoring Data

One goal of the CIRCSIM-Tutor project was to generate text that is both pedagogically and linguistically realistic. To that end, an experiment was devised to capture tutoring discourse using human-to-human tutoring sessions. Our colleagues (JAM and AAR), professors of Physiology at Rush Medical College, served as tutors; some of their students volunteered to serve as subjects. To evoke language that more closely resembles what students would type into a computer (and to avoid the transcription problem), the sessions were conducted in a keyboard-to-keyboard format.

These transcripts were initially used as a source of information about lexical, syntactic requirements and pedagogical requirements for the machine tutor. We have used them to study pedagogical issues, such as the best protocol for interleaving predictions and dialogue, and discourse phenomena, such as the use of long-distance anaphora in conversation. Recently, we have conducted some sessions where the tutors were novices (i.e., other medical students who have already completed the first-year physiology course) to attempt to identify the differences between expert tutors and novices.

4 System Architecture

The main components of CIRCSIM-Tutor are the planner, the text generator, the input understander, the student model, the knowledge base, the problem solver and the screen manager (Woo et al., 1991; Woo, 1992). The main loop in CIRCSIM-Tutor asks the

student to pick a problem to solve (a perturbation). For each of the three stages of the problem it first calls the problem solver to obtain a set of correct answers, then it collects the student's predictions and sets up a tutoring dialogue for that stage.

The planner is given the student's predictions, plus a student model showing errors that the student has made. It sets up a series of tutoring goals to remedy those errors in a logical sequence. The planner is a hierarchical one and so once it has set up a list of tutoring goals, it calls itself to make a discourse plan for tutoring the top goal on the list, by choosing one of a small number of tutoring strategies. The most common discourse plan for tutoring a single erroneous variable looks like this:

1. Elicit the determinants of the erroneous variable
2. Elicit the actual determinant which is operating now
3. Elicit the relationship between the actual determinant and the erroneous variable
4. Elicit the correct value

The planner sequences through these steps, invoking the text generator to ask the question, invoking the input understander to fetch the answer, and invoking the student modeler to verify the answer and update the student model.

The system constructed by Woo (Woo et al., 1991; Woo, 1992) was tested extensively by our colleagues at Rush Medical College, Professor of Physiology and expert tutors, Joel Michael and Allen Rovick, and by their students. They complained that the system gave terrible hints, that the tutoring strategies were very limited and, worst of all, that the system reused failed tutoring stratgies with the same student. This led us into a series of studies of hinting (Hume et al., 1996a,b). Under the leadership of Freedman (1996) we lloked for new and better ways to study tutoring dialogues, to describe tutoring strategies, and to generate tutoring language.

As we began to implement these ideas it became clear that we needed to improve our language understanding modules as well, since students were not comfortable carrying on a dialogue with a program that does not seem to understand them. Major changes in the system over time included a new interface (Brandle and Evens, 1997), a new input understander (Glass, 1999), better analysis of student answers (Zhou et al., 1999a), a new student modeler (Zhou and Evens, 1999) to support our discoveries about hints (Zhou et al., 1999b), and new knowledge to support a wider variety of procedures. As we describe each of the major components of the system, we will try to explain briefly the differences between the old components and the new ones, and what motivated the changes.

5 Lexicon and Spelling Correction

The current input vocablary was derived from the transcripts of tutoring sessions. Yoon Hee Lee (Lee and Evens, 1998), who developed the original parser and lexicon, looked at the sentences the student typed and was horrified by the number of errors. He realized immediately that spelling correction was an essential function if we hoped to make the program understand student input. The students used a lot of medical abbreviations,

which Lee added to the lexicon, along with error forms too short to recognize by standard correction algorithms (like "teh" and "hte" and "fo"). The students also invented spontaneous abbreviations quite often by stopping typing part of the way through a word. Lee handled this by reducing error cost for missing letters as the system got closer to the end of a word. We also eliminated our morphology routine and included all forms in order to speed up the spelling correction process. The current lexicon is comprised of approximately 3000 words, including all forms of about 750 different lemmata.

The original spelling correction algorithm used a three-way match algorithm, which slides a small window simultaneously across both the unknown input word and a candidate word from the lexicon. Transpositions, elisions, substitutions, etc., are simply counted, and the most likely candidate is picked. Elmi turned the three-way match algorithm into a four-way match algorithm, added a weighting routine that recognizes that some errors are more likely than others, and stored all the unique strings in a hash table to increase the speed still further (Elmi and Evens, 1998).

6 Parsing and Interpretation

The original language understanding program for CIRCSIM-Tutor was a simple bottom-up chart parser written by Yoon Hee Lee (Lee and Evens, 1998) on a Xerox Lisp machine. Over time we found that this program was coming up with the default output "I am sorry I do not understand you" too often. So Michael Glass (1999) developed a new understander, modeled on information extraction technology.

The central mechanism is a cascade of finite state transducers. Each machine produces an output, which is usually some modification of the input. The new module has a number of special purpose finite state machines. One FSM copes with copula deletion, removing finite forms of the verb "to be," but leaving the abbreviation "is" for inotropic state. Another looks for names of parameters and their abbreviations, and also verbs of change. Another looks for negations and combines them with verbs of change, so that "doesn't change" is tranformed into "**neg + change.**" Another looks for words and phrases that indicate proportionality.

This new module performed exceptionally well in the experiment in November, 1998. Out of 1801 student turns, only 24 were not understood. Ten of these were so garbled or ambiguous that humans could not understand what the student meant either. Another nineteen made sense but were not recognized in any useful manner. Six of these nineteen involved spelling errors that the system could not correct (but it did correct thirty such errors appropriately). In seven cases the system failed because of a missing or incomplete lexical entry (two of these involved abbreviations). In two cases the student asked for help but the system did not understand. Two more turns included unprintable expressions of frustration. Finally, two involved domain concepts beyond Circsim-Tutor's knowledge.

The input understander can also produce representations for a handful of student initiatives, e.g. "I don't understand inotropic state." The discourse processor is prepared to accept these kinds of initiatives in most contexts. To "I don't understand inotropic state?" it responds by defining the term.

7 Domain Knowledge Base and Problem Solving

The current knowledge base (Zhang, 1991) is used in two different ways. It is used by the problem-solver to produce a solution to the problem presented to the student, and it is used by the generation program to discuss the relevant domain knowledge and problem-solving algorithms with the student.

The information in the knowledge base in organized around variables and causal-relations. Some typical variables are Ra (radius of the arterioles) and TPR (total resistance of all peripheral blood vessels), with a causal-relation representing the fact that when Ra changes in one direction then TPR changes in the opposite direction. Causal relationships are divided into various categories, reflecting that some are mediated by the nervous system (and therefore don't always operate), some reflect invariant equations, and so on. The main problem solver starts with the disturbance to one of the cardiovascular parameters, which was specified in the problem picked by the student. It finds this parameter in the knowledge base and follows the causal links until it has values for all seven variables that the student must predict. It returns these values, as well as the order in which they were found. This trace of the solution of the problem is used to drive both the collection of the student's predictions and tutoring of the student's errors. Similarly, the tutorial planner tries to fix erroneous predictions in the same causal order.

The knowledge base is also accessed from the student modeler, which uses it to evaluate the truth of student input. The discourse processor uses the knowledge base to issue hints. If the student types "I don't understand" in response to a question, the discourse planner uses the knowledge base to find a variable immediately affecting the variable under discussion, then mentions this variable to the student as a hint, then reissues the question. (Unfortunately this is sometimes not a very good hint.) Much of the additional knowledge that we would like to include in future versions of the knowledge base relates to this issue of discussing the problem (as opposed to simply solving the problem). We are now working on some enhancements to the CIRCSIM-Tutor knowledge base proposed by Freedman (1996):

- We observe in transcripts that students and tutors often refer to more detailed knowledge.
- In a similar vein, we also observe references to anatomical explanations for physiological phenomena, and for the values of our cardiovascular parameters.
- The ability to describe the problem-solving process, i.e., to *say* something like "X and Y affect Z, but X hasn't been predicted yet, so we consider only the effect of Y" or to point out contradictions implied by the student's answer.
- The ability to have multiple linguistic descriptions of a single object, again so that the tutor can provide alternate explanations.
- More tutorial knowledge, separate from the domain knowledge, so we can represent the rhetorical patterns of tutoring in a domain-independent fashion.

One interesting observation is that for purposes of dialogue generation, it is not necessary that causal links in the knowledge base join together to provide a complete solution for all variables. It suffices to be able to argue for a correct solution to any particular cardiovascular variable of interest. If there is an anatomical argument for a certain pressure or volume, for example an argument involving veins and the parts of the heart, then the anatomical reasoning for the correct value of that variable should be known to the tutor. You cannot necessarily solve the entire problem for all variables using this kind of anatomical reasoning, nor would we want to (CIRCSIM-Tutor is teaching physiology, not anatomy). But often it is useful for the tutor to be able to state several different explanations as to why one particular variable has the value it has, in order to have several different ways to try to convince the student of the truth of the answer. Our current knowledge base has only *one* causal explanation for the change in each variable, derived from the causal links it used to solve the problem.

8 Student Modeling

The version of CIRCSIM-Tutor constructed by Woo (1992) used an overlay model. For all of the causal-relationships in the knowledge base, the student model kept a record of whether the student is correct or mistaken about that relationship. This record was built when the predictions were first entered and updated as the student answered questions and changed predictions. There was no history in this student model – only the student's most recent predictions and answers were reflected. This overlay model drives the tutorial planner. The planner picks which variables to tutor based on the solution trace combined with which relationships have been predicted incorrectly. Shim (1991) designed a more detailed model but it was not implemented as part of the system.

This model did not contain enough information to create sophisticated hints or to tailor response strategies to different kinds of student errors. Yujian Zhou (Zhou and Evens, 1999) has recently built a much more detailed student modeler. Zhou's model contains four components, designed to provide input for four different levels of planning: the global assessment (an overall assessment of the student's performance), the procedure-level assessment (an assessment of how the student is performing on this procedure so far), the stage assessment (one for each stage, DR, RR, and SS), and the local assessment (measured for each variable that has been tutored in this stage).

The global assessment combines an assessment of how well the student is performing in making initial predictions in the prediction table, how well the student responds to hints, and how well the student is doing in the tutoring dialogue. The procedure assessment contains these same variables looked at only in the current procedure. Each answer is categorized as correct, partially correct, a near miss, an "I don't know" answer, or totally wrong, and this answer category is recorded in the tutoring history and weighted to produce a performance score (Zhou and Evens, 1999).

The student model is still not storing any measure of student unease. We are convinced that, when the student makes angry remarks or indicates uncertainty, the system should notice this and try to relieve the student's frustration.

9 Planning the Response

After the student's input has been received and evaluated, the system must plan a response. The tutor's first step is to identify the concepts that the student has missed. CIRCSIM-Tutor can tutor two basic solution strategies: the rules for assigning a value to a neural variables and the rules for propagating values along a link of the concept map. These two strategies will cause all the incorrect variables to be corrected, one at a time.

The current version of CIRCSIM-Tutor contains a large decision table to determine the order for tutoring the selected concepts. However, when doing research for the new version Freedman (1996) discovered that the sequence could be generated algorithmically. Essentially, concepts are tutored in the order they would be encountered while solving the problem and in the new version now under construction, a small set of rules is used to generate the sequence. After the sequence is determined, a plan is chosen for tutoring each concept. A plan consists of a single *tutoring tactic* or a series of tutoring tactics.

There are four major types of tutoring tactics, one for each major category of plan:

Plan	Tactic
Tutor	Ask a series of questions
Give answer	Explain (give a declarative sentence)
Hint	Remind ("Remember that. . . ")
Acknowledge	4 possible forms (see below)

The four potential forms of acknowledgments evolved from the possibility that a question demanded several answers, e.g. "What are the determinants of X?" Whether the answer was partially correct (one correct determinant and one incorrect), or maybe even contains two wrong answers (requiring that *both* be negatively acknowledged), adds several categories. The resulting tutoring dialog consists of a repeated cycle:

1. Ask the next question.
2. Evaluate the student's response
3. Acknowledge the student's answer.
4. If answer is wrong, either give a hint or the right answer.

If a hint is given, the next question will in fact be the same question restated. If the student's answer is correct, or the tutor gives the student the answer, the tutor goes on to the next question in the tutorial plan. When the plan for each missed concept has been completed, the tutor goes on to the next stage of the problem. As a result of the above cycle, note that the text of each turn, or tutor's response, can be described as follows:

- Acknowledge the student's answer.

- If the answer is wrong, give either a hint or the right answer.
- Ask the next question.

Thus each turn has a similar structure. In fact, this structure is consistent with findings of the Conversation Analysis school (Sinclair and Coulthard, 1975). Although the turn structure for the new version is essentially similar, it is more flexible. For example, the tutor can omit the acknowledgment when it is redundant. In the current version, each tutoring tactic is realized as a separate sentence, with the exception that an acknowledgment may be followed by a comma ("Right, ..."). In the new version, the semantic forms generated for a turn will be buffered and realized together, so that ideas may be combined to produce a greater variety of surface structures. Generating a turn as a whole will also allow us to deal explicitly with the issue of intra-turn coherence.

In the current version, the basic planning cycle is implemented with an augmented finite state machine. The tutoring plans are kept on a stack. The main factor governing state transitions is which type of student input is received. Other factors used are the type of concept being tutored (value of neural variable or causal link) and whether this is the student's first or second try at answering a question. Hints are generated for the first or second try. After two failures the system will tell the student the right answer.

One result of this architecture is that the tutor cannot change its mind in the middle of a tutoring plan. It can retry a goal exactly once, and it cannot abandon a goal. As the examples below show, this can lead to extremely repetitive text. The new version uses an agenda instead of a stack. If the student consistently gives wrong answers, the tutor can change to a more effective plan. Conversely, if the student catches on quickly, the tutor has the option of dropping unnecessary repetitions. Another consequence of the augmented finite state machine architecture is that the tutor cannot generate nested structures unless they are hard-coded in the states of the machine. The new version is designed to permit arbitrary nested structures. It also permits arbitrary forms with full unification, so that ideas can be represented in a more general fashion.

10 Text Generation

The current text generator (Chang, 1992) accepts logic forms one at a time from the discourse processor and produces sentences. A single tutor turn may involve several logic forms, for instance a negative acknowledgment, followed by a statement, followed by a question. In the current version these are not generally combined, meaning that three separate sentences are emitted, except that the system can tack a negative acknowledgment onto the next sentence.

Logic forms contain one discourse goal, such as negative-ack or give-answer, and one content goal, such as affected-by, value or mechanism. Thus (negative-ack (value-dr *co* +)) produces "Wrong, CO did not increase in DR." Certain transformations are performed if needed on the filled-in trees (e.g. passivization, insertion of auxiliary verbs and complementizers, some movement for questions), and the final sentence is linearized with the help of an LFG grammar.

Prompted by student observations that the dialogue from CIRCSIM-Tutor can be repetitive and boring, we have experimented with a certain amount of lexical varia-

tion. The results are mixed. It seems helpful to vary acknowledgments, even if only by picking randomly between alternatives such as "you're right" and "correct." However certain other variations caused a decrease in the perceived coherence of the generated text, for example where a variable might be abbreviated in one sentence and then spelled out in the next. These experiments were an important factor in the decision to add a turn planning module to the new version to produce coherent turns in the dialogue generation process.

11 Example of Generated Dialogue

Here we show examples of frequently used tutoring strategies, If a student makes a bad prediction for one of the neural variables in the reflex response stage, then a dialogue like this will ensue:

T: Remember that the reflex changes the value of all neurally controlled variables first, in order to correct the change in arterial pressure. Which variables are changed by the reflex?
S: tpr.
T: Right. Total Peripheral Resistance is one. Inotropic State and Heart Rate and Total Peripheral Resistance are directly affected by the reflex. What is the correct value of Total Peripheral Resistance?
S: +
T: Correct, the value of Total Peripheral Resistance is increased. Please change of the value of TPR in the predictions table.
S: *(changes prediction table)*

The system then goes on to correct an error in CVP, using the plan described earlier.

T: What is the determinant of Central Venous Pressure?
S: co
T: Correct, Central Venous Pressure is determined by Cardiac Output.
T: Is the relationship from Cardiac Output to Central Venous Pressure direct or inverse?
S: inverse
T: Good, you got the correct answer.
T: What is the corrrect value of Central Venous Pressure?
S: -
T: Correct, the value of Central Venous Pressure is decreased.
T: Please change the value of CVP in the predictions table.
S: *(changes prediction table)*

Clearly, the text is repetitive. For one thing, it would read more smoothly if some of the explicit acknowledgments were omitted. We are now implementing Brandle's rules to give more appropriate acknowledgments in our new version (Brandle and Evens, 1997).

Here is another example in which the system is tutoring the same relationship. Here the student gives an answer that is correct but not the one that the system expected. Our

improved ability to recognize these answers and categorize them effectively allows the system to respond appropriately.

T: What is the determinant of Cenral Venous Pressure?
S: cbv
T: And what parameter in the predctions table determines Central blood Volume?
S: cardiac output
T: Right, CO determines CBV, therefore determines CVP.

In the following example from another session with CIRCSIM-Tutor the student gives an answer that the system does not understand, but it manages to respond appropriately anyway.

T: By what mechanism is Total Peripheral Resistance controlled?
S: Metabolic factors
T: Is the mechanism of control neural or physical?
S: Neural

Default strategies like this, which keep the dialogue moving whatever the student response, are important in making the system comfortable to work with.

12 Conclusions

In this paper, we described the architecture of the current version of CIRCSIM-Tutor a dialogue-based interactive tutoring system in the domain of cardiovascular physiology. We discussed the input it can handle, the knowledge representation and student model, and the planning and text generation phases. We have described our experiences with this system and what we have learned about the design of natural-language based ITSs. We have described the changes in progress for our new version of the system. One lesson that we have learned is the importance of planning in tutoring sessions and the complex nature of the plans that are produced. That is why we have chosen to base our new version of CIRCSIM-Tutor on Freedman's (2000a,b) Atlas Planning Environment.

13 References

Brandle, Stefan and Martha W. Evens. (1997). Acknowledgments in Tutorial Dialogue. *Proceedings of the Midwest Artificial Intelligence and Cognitive Science Conference,* Dayton, Ohio, June 2-3. 13-18.

Chang, Ru-Charn. (1992). *Surface Level Generation of Tutorial Dialogue Using a Specially Developed Lexical Functional Grammar and Lexicon.* Doctoral dissertation, Illinois Institute of Technology, Chicago, Illinois.

Elmi, Mohammad A. and Martha W. Evens. (1998). Spelling Correction Using Context. *Proceedings of COLING 98,* Montreal, Canada. 360-364.

Freedman, Reva K. (1996). *Interaction of Discourse Planning, Instructional Planning and Dialogue Management in an Interactive Tutoring System.* Doctoral dissertation, Northwestern University, Evanston, Illinois.

Freedman, Reva K. (2000a). Plan-Based Dialogue Management in a Physics Tutor. *Proceedings of the Sixth Applied Natural Language Processing Conference*, Seattle, WA. 52–59.

Freedman, Reva K. 2000b. Using a Reactive Planner as the Basis for a Dialogue Agent. *Proceedings of FLAIRS 2000*, Orlando, FL. 203–208.

Glass, Michael. (1999). *Broadening Input Understanding in a Language-Based Intelligent Tutoring System.* Doctoral dissertation, Illinois Institute of Technology, Chicago, Illinois.

Hume, Gregory D., Joel A. Michael, Allen A. Rovick and Martha W. Evens. (1996a). Hinting as a Tactic in One-on-One Tutoring. *Journal of the Learning Sciences*, 5(1): 23–47.

Hume, Gregory D., Joel A. Michael, Allen A. Rovick and Martha W. Evens. (1996b). Responses and Follow up Tutorial Tactics in an ITS. *Proceedings of the 9th Florida Artificial Intelligence Research Symposium*, Key West, Florida, May 20–22. 168–172.

Lee, Yoon Hee and Martha W. Evens. (1998). Natural Language Interface for an Expert System. *Expert Systems (International Journal of Knowledge Engineering)*, 15(4): 233–239.

Shim, Leem Seop. (1991). *Student Modeling for an Intelligent Tutoring System: Based on the Analysis of Human Tutoring Sessions*. Doctoral dissertation, Illinois Institute of Technology, Chicago, Illinois.

Sinclair, John M., and Richard M. Coulthard. (1975). *Towards an Analysis of Discourse: The English Used by Teachers and Pupils.* Oxford University Press, Oxford.

Woo, Chong Woo, Martha W. Evens, Joel A. Michael, and Allen A. Rovick. (1991). Dynamic Planning in an Intelligent Cardiovascular Tutoring System. *Proceedings of the Fourth Annual IEEE Symposium on Computer-Based Medical Systems*, Baltimore, MD. 226–233.

Woo, Chong Woo. (1992). *A Multi-Level Dynamic Instructional Planner for an Intelligent Physiology Tutoring System.* ONR Technical Report, available from Dept. of Computer Science, Illinois Institute of Technology.

Zhang, Yuemei. (1991). *Knowledge-Based Discourse Generation for an Intelligent Tutoring System.* Doctoral dissertation, Illinois Institute of Technology, Chicago, Illinois

Zhou, Yujian, Reva K. Freedman, Michael Glass, Joel A. Michael, Allen A. Rovick, and Martha W. Evens. (1999a). What Should the Tutor Do When the Student Cannot Answer a Question? *Proceedings of the 12th Florida Artificial Intelligence Symposium (FLAIRS-99)*, Orlando, FL, 187–191.

Zhou, Yujian, Reva K. Freedman, Michael Glass, Joel A. Michael, Allen A. Rovick, and Martha W. Evens. (1999b). Delivering Hints in a Dialogue-Based ITS. *Proceedings of the AAAI*, Orlando, FL, 128–134.

Zhou, Yujian and Martha W. Evens. (1999). A Practical Student Model in an Intelligent Tutoring System. *Proceedings 11th IEEE International Conference on Tools with Artificial Intelligence*, Evanston, Illinois, 13–18.