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

Martha W. Evens
Ru-Charn Chang[‡]
Yoon Hee Lee[‡]
Leem Seop Shim[‡]
Chong Woo Woo[‡]
Yuemei Zhang
Department of CSAM
Illinois Institute of Technology
10 W. 31st Street 236–SB
Chicago, IL 60616
csevens@minna.iit.edu

Joel A. Michael
Allen A. Rovick
Department of Physiology
Rush Medical College
1750 West Harrison
Chicago, IL 60612
jmichael@steve.iit.edu
aar@rpslmc.edu

Abstract

CIRCSIM-Tutor version 2, a dialogue-based intelligent tutoring system (ITS), is nearly five years old. 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. It uses natural language for both input and output, and can handle a variety of syntactic constructions and lexical items, including sentence fragments and misspelled words.

Using CIRCSIM-Tutor

One of the important topics which beginning medical students must learn is how blood pressure is regulated in the human body. When something happens to change the blood pressure, such as a change in the volume of blood in the body, the body must compensate. The negative feedback loop which controls this process, known as the *baroreceptor reflex,* can be a difficult topic for students.

CIRCSIM-Tutor is based on a qualitative model involving seven core physiological parameters. The core parameters and the causal relationships between them are shown in the concept map in Figure 1. Figure 1 also shows the influence of the nervous system, which plays an essential role in blood pressure regulation. (In the diagram, Baro = baroreceptor pressure and NS = nervous system response.)

Students use CIRCSIM-Tutor to learn to solve problems like those taught in their physiology course. Students are asked to predict the value of the seven parameters at three points in time: the DR or *direct response* stage immediately after the precipitating event, the RR or *reflex response* stage after the nervous system responds, and the SS or new *steady state* stage.

When students start CIRCSIM-Tutor, they see the main user interface screen illustrated in Figure 2. The precipitating event (a broken pacemaker in this case) is shown at the top of the screen. One side of the screen contains room for students to enter their predictions. After the predictions are entered, the dialogue will unfold on the other side of the screen.

Instructional and Discourse Planning

The main components of CIRCSIM-Tutor v.2 are the input understander, the student modeler, the instructional planner, the discourse processor, the text generator, and the knowledge base/problem solver.

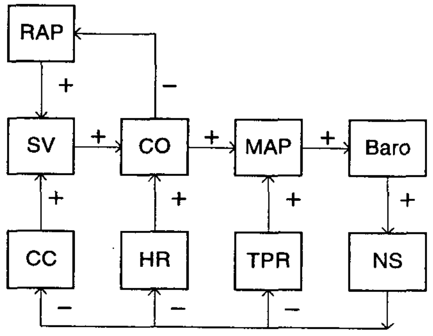

Figure 1

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

[‡]Ru-Charn Chang is now at Baxter Laboratories, North Chicago, IL. Yoon Hee Lee is currently Director of Training and Education, Institute of Defense Analysis, Seoul, Korea. Leem Seop Shim is currently at the Department of Information Science and Telecommunications, Hanshin University, Osan, Korea. Chong Woo Woo is currently Chair of the Department of Computer Science, Kookmin University, Seoul, Korea.

CIRCSIM-Tutor picks a problem for the student to solve and obtains the correct answers from the problem solver. It collects the student's predictions and calls the instructional planner to conduct a conversation.

The planner is given the student's predictions, plus a student model showing student errors and possible misconceptions. CIRCSIM-Tutor v. 2 uses an overlay model. For each causal relation in the knowledge base, the student model keeps a record of whether the student is correct or mistaken about that relationship. This record is built when the predictions are first entered and updated during the dialogue.

The planning cycle is implemented with an augmented finite state machine. The tutoring plans are kept on a stack. State transitions are chosen based on the student's current input, whether the student has attempted the question before, and domain knowledge. For example, the system only generates a hint when the student makes a mistake on the first try at a question. Otherwise it gives the student the right answer.

The planner calls the discourse processor with a list of discourse actions such as the following:

1. Elicit the determinants of the erroneous variable
2. Elicit the currently active determinant
3. Elicit the relationship between the active determinant and the erroneous variable
4. Elicit the correct value

Alternatives to *elicit* are to give a declarative explanation or a hint ("Remember that ...").

Parsing, Interpretation and and Spelling Correction

The input understanding component of CIRCSIM-Tutor v. 2 contains a bottom-up chart parser, producing first a phrase-structure parse and then a Lexical Functional Grammar f-structure. Matching the f-structure functional representation of the student's utterance to the logic form of the tutor's question is largely performed by *ad hoc* code. This task is simplified by the fact that ten predicates cover most student answers.

Problem: Pacemaker malfunctions				
	DR	RR	SS	Dialogue
RAP				
CC				
SV				
HR				
CO				
TPR				
MAP				

Figure 2

The CIRCSIM-Tutor v. 2 input lexicon is comprised of approximately 750 lemmata. Also in the lexicon are basic Lexical-Functional Grammar annotations to be used by the parser. Spelling correction is an important aspect of the input understander, as students frequently misspell words, abbreviate creatively, and make word boundary errors (two words joined together or a single word split in two). Spelling correction is based on 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, and similar errrors are counted and the most likely candidate is picked.

Text Generation

The text generator produces sentences from logic forms generated by the planner. A single tutor turn may involve several logic forms, such as an acknowledgment, an explanation expressed as a declarative statement, and a question. The logic forms are used to fill in abstract templates. Transformations such as passivization, insertion of auxiliary verbs and *wh*-movement are performed, and the final sentence is linearized with the help of an LFG grammar.

Example of Generated Dialogue

Here is an excerpt from a conversation generated from the tutoring tutoring tactic illustrated above:

T: Remember that the Direct Response occurs immediately and produces changes in the system before any reflex is activated. What controls HR?
S: Nervous system.
T: Correct. Heart rate is neurally controlled. What is the correct value of HR?
S: Unchanged.
T: Correct, the value of HR is unchanged ...

Acknowledgments

We thank Stefan Brandle, Reva Freedman and Michael Glass for continued enhancements to v. 2 as part o their research on v. 3, and for writing this document.

References

The CIRCSIM-Tutor project has published over 5(papers. For a complete bibliography, please see ou Web page `http://www.csam.iit.edu/~circsim`.