In Michael Gasser, ed., *Online Proceedings of the 1996 Midwest Artificial Intelligence and Cognitive Science Conference*, Bloomington, IN, 1996. URL http://www.cs.indiana.edu/event/maics96/Proceedings/Freedman/freedman.html.

Using a Text Planner to Model the Behavior of Human Tutors in an ITS

Reva Freedman
Department of EECS
Northwestern University
Evanston, IL 60208

freedman@delta.eecs.nwu.edu

Abstract

CIRCSIM-Tutor v. 3 is a natural-language based intelligent tutoring system (ITS) for cardiac physiology currently under development. In this paper we describe how the CIRCSIM-Tutor text planner uses a small set of primitives to generate a large and varied set of dialogues which respond to the student's concerns and display detailed knowledge of the subject matter. We start by describing our main conceptual data structure, the *dialogue schema*, and the data structure used to implement it, the *dialogue schema section*. We explain how the semantic forms which constitute the dialogue schemata are transformed into lower-level primitives which can be realized as text for the student. We illustrate how a single dialogue schema can be used to generate major discourse phenomena of interest to tutoring researchers, such as *hint*, *explanation*, and *interactive explanation*. Finally, we demonstrate how the planner interpolates responses to the student into the conversation without losing coherence.

1. Introduction

CIRCSIM-Tutor v. 3 is a natural-language based intelligent tutoring system (ITS) for cardiac physiology currently under development. In this paper we outline the functioning of TIPS, the text planning engine in CIRCSIM-Tutor. TIPS stands for "Text generation Interactively, a Planning System." This acronym was chosen for the pun involved, as one of the system's primary pedagogical goals is to generate hints for the student, i.e. tips. TIPS carries out a goal-oriented tutoring plan while maintaining a coherent conversation with the student and using the same discourse structures as human tutors.

Most text-based ITSs view text generation as a back end to pedagogical planning and implement it in a very simple fashion. In contrast, we view the generation of a response as primarily a text planning problem since all decisions, even pedagogical ones, must eventually be expressed as text. We believe that this approach will produce more varied and higher-quality language, and improve student understanding and retention as a result.

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.

Our main conceptual data structure is the *dialogue schema*. From studying transcripts of human tutors, we have isolated a small group of dialogue schemata which suffice to teach the concepts in the CIRCSIM-Tutor domain model. As these essential dialogue schemata can be implemented with a small number of semantic primitives, the use of dialogue schemata, along with a backtracking planner and a sufficiently rich lexicon, allows us to generate a large and varied set of dialogues at moderate cost.

2. Dialogue schemata and their implementation

2.1. Definition and use of dialogue schemata

In their beginning physiology course, medical students are given problems to solve based on a simplified qualitative model of the heart. In each problem, something happens to change the processing of the heart. The student is then asked to predict the direction of change of seven core variables in each of three resulting physiological stages. After predictions are made for each stage, CIRCSIM-Tutor conducts a dialogue with the student to discuss incorrect answers. Different from many ITSs, CIRCSIM-Tutor uses natural language for both input and output.

Over the last four years, the CIRCSIM-Tutor project has collected over 5000 turns of keyboard-to-keyboard tutoring sessions using similar problems and live tutors in order to model pedagogical and linguistic strategies. Within each physiological stage, we have observed that the tutorial dialogue is divided into segments, one for each incorrect core variable. Within a segment, each attempt to teach the value of a variable ends with the tutor requesting the correct value. If the student gives the correct answer, the segment ends. Otherwise, the attempt fails, causing the remaining goals associated with it to be removed from the agenda, although turns which have already been uttered remain part of the conversation. If an attempt fails, the tutor can make another attempt or give the student the answer.

Within an attempt, the tutor can employ any one of a number of correction mechanisms in a nested fashion in an attempt to get the student to give the correct answer. The choice of mechanisms cannot be expressed as a simple algorithmic process but is implemented using a classical planner called the *tutorial planner*. Each correction mechanism is expressed as a *dialogue schema* which contains the raw material to enable the tutor to achieve a communicative goal. A dialogue schema contains one or more discourse goals which must be satisfied in the coming turn or turns. The applicability condition for the first turn tells the tutorial planner when the schema applies. Applicability conditions for later turns tell the planner which responses on the part of the student are sufficient to continue the schema. If the student gives a wrong or unexpected response, the planner will search for another schema, such as an error recovery schema, which can be used instead.

For implementation purposes, an offline process breaks each dialogue schema into *dialogue schema sections*, or plan operators for individual turns. Each dialogue section, whether derived from the expected path of the schema or from an error recovery schema, is implemented as a plan operator. Plan operators are also known as *semantic forms*.

2.2. *Interface to the text realization component*

The output of the tutorial planner is a series of *semantic primitives*, or discourse goals which cannot be further decomposed. Since semantic primitives do not map neatly onto sentences, utterances must be planned a turn at a time. Therefore the tutorial planner accumulates discourse goals until a turn is complete, i.e. until a goal requires a response from the student. At that point the set of accumulated semantic primitives is passed to the *turn planner*, which organizes the discourse goals into one or more sentences. The turn planner is also responsible for choosing lexical items for the concepts involved and the final realization as surface text. Although turns must be cohesively linked to earlier turns in the conversation, they are essentially separate paragraphs, so any of a number of existing paragraph planners could be used as the basis for the turn planner.

2.3. *Some basic semantic forms*

For CIRCSIM-Tutor v. 3 the choice of which semantic forms to leave as primitive is more a practical than a theoretical one. If the tutorial planner does not need to make a distinction between two concepts, then we can use the same semantic primitive to represent both. In this section, we define the semantic forms which are used in the examples in this paper. In these definitions P(x) represents a proposition about an item in the domain knowledge base.

1) *S-knows(P(x))*
   This form is used when we want to make sure that P(x) has been included in the discourse at some point. In a philosophical sense, we cannot know what the student knows, so we use the operational definition of asking whether one of the speakers has stated the concept.

2) *T-teaches(P(x))*
   This form is used when we want the proposition P(x) to be taught at this point even if it has been stated earlier in the conversation. This form occurs frequently in our dialogue schemata because many argumentative structures require information to be stated explicitly. (For example, there is a big difference between "I'm not going" and "I'm not going because it's raining," even if everyone knows that it's raining.)

3) *T-conveys(P(x))*
   This form is used to state the proposition P(x).

4) *T-elicits(x, P(x))*
   This form is used when the tutor wants to obtain some information from the student. The additional argument is used to identify the desired information.

These semantic forms are related as follows:

S-knows(P(x))	*if*	P(x) is already known *or* *T-teaches(P(x))*
T-teaches(P(x))	*if*	*T-elicits(x, P(x))* *or* *T-conveys(P(x))*

The choice of semantic primitive does not constrain the form of the eventual surface structure. For example, *T-elicits(x, P(x))* does not require that the resulting text have the surface form of a question, but could use any of the major sentence structures:

Interrogative:	*(direct)* How is TPR controlled?
	(indirect) Could you tell me how TPR is controlled?
Imperative:	Please tell me how TPR is controlled.
Declarative:	I'd like to know how TPR is controlled.

Similarly, *T-conveys(P(x))* need not be realized only by a declarative sentence:

Interrogative:	Did you forget that TPR is neural?
Imperative:	Remember that TPR is neural.
Declarative:	TPR is neural.

Although many semantic primitives represent actions, not all of them do. In particular, some semantic forms represent discourse-based concepts such as "therefore."

3. Using dialogue schemata to implement tutoring phenomena

The variables TPR (total peripheral response), HR (heart rate) and CC (cardiac contractility) are called *neural variables* because they are controlled by the nervous system. Students often forget that neural variables do not change during the first or *direct response* (DR) stage. The following syllogism is the most common way to remind the student of this fact.

Variable V is controlled by the nervous system.
The nervous system has no effect in DR.
Therefore the value of V does not change in DR.

There are many ways to teach this syllogism. Because we want to use discourse structures similar to those used by our expert tutors, we represent the desired discourse structure as a schema.[1]

Correct-neural (V): *(PQ: V is neural and non-primary)*
    S-knows(V is neural)
    S-knows(Current stage is DR)
    S-knows(Correct value of V is no-change)

From the rules defined in the previous section, we can see that *S-knows(P(x))* can be expressed in three ways:

- *T-conveys(P(x))*
- *T-elicits(x, P(x))*
- Say nothing

In other words, each form in the dialogue schema can be implemented by giving the student some information, asking the student for some information, or going on to the next form. The following instantiation for *correct-neural* occurs the most frequently in our transcripts:

[1] For simplicity, the inner logic forms are expressed in English here.

T-elicits:	Ask student for the control mechanism for TPR
T-conveys:	Inform student that the nervous system hasn't kicked in yet
T-elicits:	Ask student for the correct value of TPR

After lexical and syntactic decisions are made, this option would generate text such as the following, assuming that the student gives the correct answer to the initial question:[2]

(1) T: How is TPR controlled?
S: Nervous system.
T: Right. And we're talking about what happens before there are any neural changes. Now what do you say about TPR?

This text is an example of an *interactive explanation*, a generalization of the phenomenon which Sanders (1995) calls a "directed line of reasoning" (DLR). If the student finds interactive reasoning too difficult, the tutorial planner may try an *explanation* which is not interactive. The following common pattern generates an explanation with a followup question.

T-conveys:	Inform student that TPR is neurally controlled
T-conveys:	Inform student that nervous system hasn't kicked in yet
T-elicits:	Ask student for the correct value of TPR

(2) T: TPR is a neurally controlled variable…Then what value would you assign to TPR in DR?

Although the human tutors usually prefer to use explicit followup questions, occasionally they terminate an explanation by giving the student the answer. We can use the *correct-neural* schema to generate this option also, as in the following example:

T-conveys:	Inform student that TPR is neurally controlled
T-conveys:	Inform student that the nervous system hasn't kicked in yet
T-conveys:	Inform student of the correct value of TPR

(3) T: TPR is controlled by the nervous system, and we're talking about what happens before there are any neural changes. So TPR doesn't change.

Notice that in (2) the second and third semantic forms have been combined into one sentence, while in (3) the first two forms have been combined.

A final option is to instantiate one of the first two semantic forms as *nil*, giving rise to a form such as the following:

T-conveys:	Inform student that the nervous system hasn't kicked in yet
T-elicits:	Ask student for the correct value of TPR

This option is used to generate a *hint* for the student. In particular, this option generates a CI-hint in the terminology of Hume et al. (1993) (CI = 'convey information'):

[2] The generation of the acknowledgment ("Right."), will be described in the following section.

(4) T: Remember that we're talking about what happens before there are any neural changes. Now what do you say about TPR?

Since each instance of *T-conveys* and *T-elicits* can be implemented using any of the syntactic forms suggested at the end of the previous section, a large number of dialogues can be generated. Lexical variation adds to the count. Still other options can be generated by starting with a different discourse schema for teaching this topic. In particular, a different schema is required to generate a PT-hint such as "Think about what controls TPR" (PT = 'point to').

4. Creating real dialogues

In the previous section we demonstrated how a single dialogue schema can be used to generate several tutorial dialogue patterns. But the dialogue pattern only provides a base for the dialogue. Since the tutor cannot predict the student's responses, TIPS provides the opportunity to update the plan at every turn. In this section we give brief illustrations of the two most common types of update: adding a verbal response before continuing with the plan, and changing the plan.

4.1. Verbal responses to the student's utterance

In the typical dialogue pattern, predicted by the Conversation Analysis school (Sinclair & Coulthard, 1975) and observed in our transcripts, every turn has the following basic structure:

Turn:
- Response to student's previous statement
  - Acknowledgment of student's statement *(optional)*
  - Content-oriented reply *(optional)*
- New material *(optional)*
- Question or request for the student

Although human tutors don't need to, the mechanized tutor always ends a turn with an explicit question so that the student knows when to respond. Turn-taking rules work in person-to-person conversation because we are socialized to understand and use them (Sacks, Schegloff & Jefferson, 1974), but people have different expectations from a computer (Dahlbäck & Jönsson, 1991). The question may belong to either the content-oriented section of the response or the new material. Since each part of the turn must be contiguous, the question can be part of the response only when the turn contains no new material.

An acknowledgment may take one of two forms, or a combination of the two:

- A particle such as *yes* or *no*
- A more extensive statement such as *you're right*

If the student gives a correct answer, even if the language used is not precisely what the tutor would like, the content-oriented reply, if provided, usually takes one of the following forms:

- A restatement of the student's statement, possibly in more precise language
- A statement supporting the student's statement

If the student's reply is wrong, the tutor will probably issue one of these common forms for negative content-oriented replies:

- A statement denying the student's statement or a related fact
- A rebuttal of the student's statement or a related fact
- A statement supporting the opposite of the student's statement

After replying to the student's statement, the tutor returns to the tutorial plan and generates the next turn's worth of semantic forms.

In the following example, the student does not give the right answer to the initial question. The tutor gives a negative acknowledgment ("not quite"), then responds directly to the student's error before going on to ask the question again. The second time the student gives the correct response ("by neural control"), permitting the tutor to continue with the *correct-neural* schema.

(5) T: In what way is CC controlled?
S: It's controlled by the volume of blood in the compartment and affected by inotropic changes. *(wrong)*
→ T: Not quite. Changing the volume changes the performance of the muscle via the length/tension relationship, i.e. Starling's Law. Changing the inotropic state of the myocardium is what we mean when we refer to CC. By what mechanism is CC controlled, then?
S: By neural control?
T: So how will CC be affected in DR?

Since utterances once spoken cannot be retracted, the text generated by the initial instantiation of *T-elicits* remains part of the conversation. Thus the surface form of a conversation does not necessarily resemble the underlying schema or schemata.

4.2. Changing the plan

If the student gives a wrong answer, the tutor may choose to change schemata, either in addition to or instead of adding a verbal response. The following example starts with the interactive explanation pattern illustrated in (1). After the student gives the wrong answer to the final question, the tutor retries the *convey-neural* pattern, but uses the hint instantiation illustrated in (4). There is some evidence that the first instantiation of the schema is more likely to convey the concept that we are in the DR stage, while the second defines it.

(6) T: I need to remind you. Things work according to the way that they are controlled. How is HR controlled?
S: Autonomic nervous system.
T: This is DR. How will HR change?
S: MAP [mean arterial pressure] changing affects baroreceptor reflex changing, affecting HR. *(wrong)*
→ T: In DR no reflex changes have occurred yet.
S: So HR will not change.
T: Correct…

In addition to the case of wrong answers, the tutor often changes schemata when the student says something which is on the path to a correct answer but which needs further dialogue. In that case the new schema usually contains a more detailed discussion of the physiology involved. The following excerpt starts out with the interactive explanation pattern illustrated in (1). The student's answer is correct but incomplete. To help the student trace the response back to a core variable, the tutor uses an interactive explanation pattern which is a variant of the one in (1).

(7) T: What is the primary mechanism of control of TPR?
S: Radius of arterioles.
→ T: Yes. And what is the primary mechanism by which arteriolar radius is controlled?
S: Sympathetic nervous system.
T: Yes. And we're dealing with the period before any change in nervous activity occurs. So what do you think about TPR now?

5. Conclusions

In this paper, we have demonstrated how a set of dialogue schemata, along with a mechanism for updating a plan during a conversation, can be used to generate a large number of natural-sounding, useful and varied dialogues. In particular, we have shown that major tutorial discourse phenomena, such as hints, explanations and interactive explanations, can all be generated by the same process.

6. Acknowledgments

This work could not have been completed without the help of principal investigator Professor Martha W. Evens of the Illinois Institute of Technology; my adviser, Professor Gilbert K. Krulee of Northwestern University; and co-principal investigators Professors Allen A. Rovick and Joel A. Michael of Rush Medical College, who provided extensive help on both pedagogical and domain issues.

7. References

Dahlbäck, N. & Jönsson, A. (1989) Empirical studies of discourse representations for natural language interfaces. *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester (pp. 291–298).

Hume, G. D., Michael, J. A., Rovick, A. A. and Evens, M. W. (1993). Use of hints as a tutorial tactic. In *Proceedings of the 15th Annual Conference of the Cognitive Science Society*, Boulder. Hillsdale, NJ: Lawrence Erlbaum.

Sacks, H., Schegloff, E. A. and Jefferson, G. (1974). A simplest systematics for the organization of turn-taking in conversation. *Language*, 50(4), 696–735.

Sanders, G. A. (1995). Generation of Explanations and Multi-turn Discourse Structures in Tutorial Dialogue, Based on Transcript Analysis. Doctoral dissertation. Chicago: Illinois Institute of Technology, Department of Computer Science.

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