Abstract

We are developing an intelligent tutoring system (ITS) that will assist first year medical students to learn the causal relationships between the parameters of the circulatory system and to solve problems of disturbances to the system. The central component of the ITS, the instructional planner, is responsible for determining what to do next at each point during a tutoring session. The approach taken in this paper is to build the planner by combining the capabilities of lesson planning and discourse planning in order to provide globally coherent and adaptive instruction. The planner consists of two parts: a lesson planner and a discourse planner. A sample dialogue, extracted from the transcript of an actual human tutor-student interaction, is used to provide a framework for the development of the overall system, especially from the planner's point of view. The focus is on the capability of the planner to combine lesson planning and discourse planning in a sophisticated manner.

Introduction

We are building an ITS that teaches medical students about the behavior of cardiovascular reflex system that stabilizes blood pressure. Our tutoring system contains a cardiovascular knowledge base, a student modeler, a natural language input module, a text generator, and an instructional planner that acts as a control element of the system. The planner is responsible for determining what subject matter to focus on, how to present it to the student and when to interrupt the student's problem-solving activity [Dede, 1986]. The decision must be based on many different knowledge sources: domain knowledge, knowledge about the student, and pedagogic knowledge. Ohlsson [1987] divides the pedagogic knowledge into two parts: the content of the subject matter to be taught and the presentation formats in which it appears, in order to provide adaptive instruction for the student. This distinction leads the researchers to a development of two stage planning process: the lesson planning, what subject matter to focus on, and the discourse planning, how to present it to the student.

Ideally, the design of the planner should allow it to generate globally coherent instruction and also to utilize the local diagnostic information arising from the interaction with the student [Wenger, 1987]. However, most ITSs are designed using an opportunistic control approach where local diagnostic information is used to recognize opportunities for intervention in the student's problem-solving activities [Murray, 1990]. Lesson planning is not generally done in...
Track 3: Medical Imaging Systems/Session 4 227

this way, although the several recent systems have focused on sophisticated discourse planning. This lack of a lesson planning capability limits the ability of a system to generate adaptive instruction, to provide globally coherent and consistent instruction throughout the tutoring session. Thus there is a need to build a planner that integrates opportunistic control with a sophisticated planning method. For example, among a number of recent efforts, the TAPS Architecture [Derry et al., 1988] attempts to overcome this problem by combining curriculum planning and discourse planning. Our paper describes an approach to an instructional planner that combines lesson planning and discourse planning to overcome the problem.

The Background

Over a ten year period, Computer Aided Instruction (CAI) in cardiovascular (CV) physiology at Rush has evolved from HEARTSIM [Rovick and Brenner, 1983], to CIRCSIM [Rovick and Michael, 1986], to the CIRCSIM-TUTOR prototype [Kim et al., 1989] and finally to our current version (CIRCSIM-TUTOR). HEARTSIM was a Plato program and CIRCSIM is a stand-alone Basic program. The CIRCSIM-TUTOR prototype is a Prolog prototype of our ITS designed and implemented by Kim [1989]. It was designed in a modular fashion by adapting major ITS architecture. It analyzes the student's knowledge status using a simple overlay model. It solves problems using a hierarchical knowledge network, which contains explicit domain knowledge and problem-solving knowledge. However, the prototype system does not possess all of the capabilities needed for an ITS. It lacks natural language capabilities, it does not analyze the student's misconceptions, and the instructional planner is very primitive. Our new version of CIRCSIM-TUTOR uses the same architecture as Kim's prototype but includes complete student modelling, planning, and natural language understanding and generation facilities.

Subject Area

The subject area of our system is Cardiovascular Physiology. The Cardiovascular system consists of many mutually interacting components, and the student must understand the cause and effect relationships for each individual component of the system. Fig. 1 shows a causal model of our system, called the concept map [Michael et al., 1988]. Each box in the map represents a physiological variable, such as SV for Stroke Volume, RAP for Right Atrial Pressure. An arrow with "+", "-" sign between two boxes indicates whether the causal relationship between the variables is direct or inverse. There are three stages in the body's response to a perturbation in the system that controls blood pressure. The first stage is the Direct Response (DR) in which a perturbation in the system will directly affect many other components. The second stage is the Reflex Response (RR); other components are affected by the negative feedback mechanism to stabilize the blood pressure. The final stage is a Steady State (SS) that is achieved as a balance between the qualitative changes directly caused by the initial perturbation and the compensatory changes induced by negative feedback.

Organization

The system begins with a brief introductory message and then asks the student to choose any procedure from the curriculum list. The curriculum is stored as eight different experimental procedures designed by our expert human tutors (JAM and AAR). Each procedure begins by introducing a possible perturbation of the cardiovascular system, and asking the student to predict how the system variables will respond to the perturbation with qualitative answers in the Predictions Table (See Fig. 2): using a "+" sign to represent an increase, a "-" for a decrease, and "0" to indicate no change for that variable. The first column of the table is used to predict the Direct Response of each variable to the perturbation; the second is used for Reflex Responses after the negative feedback system is activated; and the third for the Steady State. There are some system variables that need to be described: the procedure variable is the variable changed by the perturbation; the primary variable is the first variable in the Predictions Table affected by the procedure variable; the neural variables are the variables under neural control. The student may
not predict the variables in any arbitrary order, since there are constraints that he or she must follow:

Constraint 1: The student must predict the primary variable first, and the value must be correct. 
Constraint 2: The student must predict in the correct sequence. 
Constraint 3: The student may predict neural variables in any order. 

The student will receive a canned error message, when either of the first two constraints is violated, and will be told what to do next. When the student finishes predicting all seven parameters in the table for one stage, for example DR stage, the student’s answers will be compared with the correct answers. This part of the system tests the student’s knowledge by requiring multiple simultaneous inputs before tutoring begins. Then the second part, the mixed-initiative tutoring session, begins to correct the student’s misconceptions based on the errors from the Predictions Table. This process will be discussed in the next section in detail.

A Sample Tutoring Session

After empirical studies of face to face and keyboard to keyboard tutorial sessions, we extracted some possible tutorial strategies and tactics that provided us with a frame work for building the instructional planner and the overall system. It is assumed that students already have learned the domain knowledge, hence the system will mainly assist the students to correct their misconceptions and to solve problems. We are currently building a simple version of the system that can handle dialogues like the following.

Tutor: What are the determinants of SV? 
Student: SV is determined by RAP and CO. 
Tutor: RAP is correct, but CO is not a determinant of SV. Remember, SV is the amount of blood pumped per beat. What is the other determinant of SV?

Assume that the current goal is to tutor the causal relationships between two parameters, RAP and SV. This goal gets refined into a set of subgoals by using plan generation rules. The generated subgoals, such as "determinant, actual determinant, relation, and value," are kept in a stack, which is used by the discourse planner to pick the next topic. The following scenario describes what each component does, what kind of information it needs, and what is the result after each step. The steps are numbered to show the execution sequence. The discourse planner begins with the first topic in the stack, "determinant," and when it gets done, continues with the next one, "actual determinant," and so on.

1. Planner: Picks the current topic from the stack, selects the discourse tactic, and passes it to the text generator as internal logical form.
   
   \[
   \text{current topic: (determinant SV), discourse tactic: question.}
   \text{logic form: (question (determinant SV))}
   \]

2. Text-Generator: Generates a sentence: "What are the determinants of SV?"

3. Screen-Manager: Displays the sentence in the Tutor window.

4. STUDENT: Answers "SV is determined by RAP and CO." in the Student window.

5. Input-Understander: Parses the student’s answer, and passes the answer to the planner.
   "student answer: (RAP, CO)"
6. Planner: Passes the current topic and student answer to the student modeller in logic form.

   current topic:  (determinant SV).
   student answer: (RAP, CO)
   logic form:     ((determinant SV) (RAP, CO))

7. Student Modeller: Calls the problem solver, gets the correct answer: (RAP, CC), compares the correct answer with the student answer, and updates the student model.

The dialogue generated by steps one through four is thus:

   Tutor:  What are the determinants of SV?
   Student: SV is determined by RAP and CO.

During steps five through seven, the system analyzes the student's answer, and records the result in the student model. Now it is time for the planner to check the student model, and then to decide what to do next. Since one of the student's answers is wrong, the planner first gives an acknowledgement of the correct answer, and gives a negative acknowledgement for the incorrect one. A possible response from the Tutor would be:

   Tutor:  RAP is correct, but CO is not a determinant of SV.

Then the planner consults the tutoring rules and decides to "give a hint" since this is the first trial, and then asks a question to complete the previous answer.

   Tutor:  Remember, SV is the amount of blood pumped per beat.
   What is the other determinant of SV?

A different tutoring rule will be applied if the student again makes an error after receiving a hint; the student will be given a direct answer for the second question. Our current tutoring rules vary according to the topic and the student's responses. (i.e., the tutor gives different responses in the different situations). The question may be about neural variables or causal relationships, in each case the tutoring rules are different.

**Organization of the Instructional Planner**

The instructional planner consists of two parts (Fig. 3); the lesson planner and the discourse planner. The lesson planner constructs a sequence of instructions to achieve a goal. This is accomplished in two phases: goal generation, and the plan generation. The planner begins by generating lesson goals using goal generation rules, and then refines each goal into subgoals using plan generation rules. The discourse planner presents the instruction to the student by making decisions about whether ask a question, give a hint, give a direct answer, etc. Finally, the planner monitors the execution of the plan and generates a new plan if the current plan is inappropriate.

**Lesson Planning**

The generation of tutoring goals, based on the student's prediction table errors, is guided by a set of explicit domain-dependent heuristics, such that the most serious misconception is selected and tutored first. Fig. 4 shows two examples of goal generation rules and Fig. 5 shows the content of the goal stack after generating all the goals. The lesson planner picks the first goal in the goal stack and expands it into a set of subgoals (i.e., topics) by applying two sets of rules, rules for selecting tutorial strategies to achieve the goal and rules for selecting pedagogic tactics to execute those strategies. For instance, if the goal is "teach the causal relationship between the two parameters," then the fired strategy rule is "tutor the prerequisite," and successively, the fired pedagogic tactical rule is "ask about: determinants, actual determinant, relationship, and
correct value." The result is a hierarchical plan tree (Fig. 6). The subgoals generated at the tactical level are kept in a subgoal stack, which is used by the discourse planner to pick the next topic. The tutoring goals remain in force until they are changed by the planner dynamically (because of changes in the student model). This process provides globally coherent, consistent tutoring goals.

Discourse Planning

The discourse planner picks the topic from the subgoal stack and presents it to the student according to its discourse tactics, such as direct question, hint, explanation, reminder, etc. The overall structure of the discourse planner is organized as a discourse management network (Fig. 7), as in MENO-TUTOR [Woolf, 1984]. The network consists of two levels: the top level specifies pedagogic decisions (whether to "introduce" a new topic or "tutor" a misconception); the lower level consists of a set of discourse tactical states, the execution of which causes text generation, student model updates, and moves to other states. The advantage of using this network is that it provides a means of specifying instructional states, and it can transit from one state to another dynamically. By combining the capability of lesson planning with discourse planning, our system has overcome the one disadvantage of the DMN in MENO-TUTOR; lack of a global lesson planning capability [Murray, 1988].

Conclusion

This paper describes an implementation of an instructional planner for a Physiology ITS, which combines globally coherent lesson plans with flexible local discourse plans. The planner consists of a lesson planner and a discourse planner. The discourse planner is designed to use a discourse management network like the one in Woolf's [1984] MENO-TUTOR, but with extended capabilities because it is combined with the lesson planner. The planner supports a dynamic planning capability that can generate a plan, monitor the execution of the plan, and replan when the current plan is not appropriate. A short scenario was discussed to explain what each component of the system does, and what the planner has to do to make pedagogic decisions dynamically. Our future research will focus on responding to student initiatives in a more sophisticated manner, an aspect of ITS design that needs further exploration.

![Figure 1. The Concept Map](image-url)
**Parameters**

<table>
<thead>
<tr>
<th></th>
<th>DR</th>
<th>RR</th>
<th>SS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cardiac Contractility</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Atrial Pressure</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stroke Volume</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heart Rate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cardiac Output</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Peripheral Resistance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean Arterial Pressure</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2. The Predictions Table

---

**INSTRUCTIONAL PLANNER**

Goal Generation Rules

Plan Generation Rules

Discourse Meta Rules

Discourse Default Rules

LESSON PLANNER

Lesson Plan

DISCOURSE PLANNER

Plan Monitor

Figure 3. Instructional Planner

If current Primary Variable is CC and Student does not the value of TPR
Then build Lesson Goal (NEURAL-CONTROL (TPR))

If current Primary Variable is RAP and Student does not know CAUSAL RELATIONSHIP between RAP and SV
Then build Lesson Goal (CAUSAL-RELATION (RAP,SV))

Figure 4. The Goal Generation Rules
Lesson Goals

<table>
<thead>
<tr>
<th>Order</th>
<th>Lesson Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>NEURAL-CONTROL (HR)</td>
</tr>
<tr>
<td>2.</td>
<td>CAUSAL-RELATION (RAP, SV)</td>
</tr>
<tr>
<td>3.</td>
<td>CAUSAL-RELATION (SV, CO)</td>
</tr>
</tbody>
</table>

Figure 5. Generated Lesson Goals in goalstack

Figure 6. Hierarchical Plan Tree for "causal_relation (RAP,SV)"

Figure 7. Discourse Network
References


