DYNAMIC PLANNING MODELS TO SUPPORT CURRICULUM PLANNING AND MULTIPLE TUTORING PROTOCOLS IN INTELLIGENT TUTORING SYSTEMS

BY

BYUNG-IN CHO

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science
in the Graduate College of the
Illinois Institute of Technology

Approved ______________________________
Advisor

Chicago, Illinois
July 2000

ACKNOWLEDGMENT

I am very grateful to my parents for their endless love and encouragement. I am deeply grateful to my wife for her love, understanding and patience. I love you, my children, JaeHyung and YounKyung.

I have received a great deal of encouragement, advice and help from people associated with the CIRCSIM-Tutor research group. I am particularly deeply grateful to my advisor, Dr. Martha Evens for her intellectual advice, encouragement, understanding and a lot of support. I feel extremely fortunate to have her as my advisor. I am also grateful to the other members of my committee, Ali Cinar, Tzilla Elrad, and David Grossman. I owe great thanks to the members of the CIRCSIM-Tutor research group: JungHee Kim, Michael Glass, Reva Freedman, Stefan Brandle, Yujian Zhou, Bruce Mills, Feng-Jen Yang, and Norman Jiang. I also appreciate Allen Rovick and Joel Michael of Rush Medical College for their advice and assistance with my work.

I deeply appreciate my senior Dr. YoonHee Lee for his encouragement. He initiated me into the CIRCSIM-Tutor research. I also acknowledge my colleagues in the institute for various kinds of support.

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

Byung-In Cho

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF ABBREVIATIONS

Abbreviation	Definition
BR	Baroreceptor
CF	Confidence Factor
CNS	Central Nervous System
CO	Cardiac Output
CST	CIRCSIM-Tutor
CVP	Central Venous Pressure
DLR	Directed Line of Reasoning
DR	Direct Response
GA	Global Assessment
HR	Heart Rate
IS	Inotropic State
ITS	Intelligent Tutoring System
LA	Local Assessment
MAP	Mean Arterial Pressure
RR	Reflex Response
SS	Steady State
SV	Stroke Volume
TPR	Total Peripheral Resistance

ABSTRACT

We are building an Intelligent Tutoring System called CIRCSIM-Tutor designed to help medical students learn to solve problems in cardiovascular physiology. This dissertation describes two planning models intended to improve the adaptive dynamic planning capabilities of CIRCSIM-Tutor. These models have been implemented using Reva Freedman's new Atlas Planning Engine.

The first new planning model is a Curriculum Planning Model. This work endows our system with the ability to build a different individualized curriculum for each student. To make curriculum planning more effective, we also developed a new student assessment method for the curriculum planning model. This assessment is used in determining which set of problems should be presented to the student at each point. The new curriculum planning rules have been shown to be complete and consistent.

The other model extends our Discourse Planning Model to provide capabilities for Multiple Tutoring Protocols. A tutoring protocol defines the overall communication between the tutor and the student. The implementation of multiple tutoring protocols allows us to compare the effect of different protocols in teaching causal reasoning. We have used machine learning methods to analyze a set of human tutoring transcripts to discover how and when human tutors switch protocols. Although they originally planned to use a protocol that collects a whole series of predictions from students before tutoring them, our human tutors switched protocols when students showed serious difficulties in solving a problem, in order to give the students more immediate feedback.

We are especially interested in this issue of immediate feedback, so we carried out an experiment to find out how students feel about immediate feedback, in which we analyzed the students' performance using CIRCSIM-Tutor and their responses to a questionnaire. Results showed that the weaker students are more likely to prefer immediate feedback.

CHAPTER I

INTRODUCTION

1.1 Problem Statement

An Intelligent Tutoring System (ITS) is a computer program that uses AI techniques for representing knowledge and carrying on an interaction with a student [Clancey, 1987; VanLehn, 1988]. An ITS should have three capabilities important in tutoring [Burns and Capps, 1988; Halff, 1988]: dynamic planning, mixed initiative interaction, and hinting. The first and most important capability is dynamic planning. The planner must be able to decide what, when and how to teach next. It must have a dynamic planning capability; it must be able to generate plans, monitor the execution of the plans, and generate new plans. It must be able to replan when necessary [Woo, 1991]. The planner must be adaptive. It must customize tutoring plans for each student [Wilensky et al., 1989; Woo, 1991; Katz et al., 1992]. A second important capability is Mixed Initiative Interaction, which allows the tutor to share control over the session with the student. With this capability, an ITS can respond to questions from the student about instructional goals and content [Shah, 1997]. A third important capability is hinting or providing the student with a piece of information that the tutor hopes will stimulate recall of the facts needed to answer the question. A hint often helps the student make an inference needed to arrive at an answer to a question or to make a correct prediction of system behavior [Hume, 1995, p.4; Hume et al., 1996].

Planners select and sequence the subject matter. Curriculum Planning is concerned with selecting the next problem [Cho et al., 1999]. Discourse Planning selects and sequences the material in the problem to be tutored and controls the actual presentation of the material to the students [Freedman, 1999]. The first step of my research on planning starts with the Curriculum Planner, because the Curriculum Planning is the first, most global planning task.

The goals of Curriculum Planning are to motivate the students and make sure that they have the ability to solve all problems in the domain. But we do not want to bore the students or waste their time by making them solve all the procedures or solve the same procedure repeatedly. So the Curriculum Planner should assist the student to choose an appropriate procedure. Selecting the proper problems, at the appropriate difficulty level for the student, is very important in an ITS. The selected problem must be challenging, but not frustrating. The problems should be varied to maintain the student's interest and to ensure coverage of important material.

We believe that the study of human tutoring transcripts is the best possible way to figure out how to imitate human tutors. We have accumulated 75 transcripts of human tutoring sessions. Our domain experts Joel Michael (JAM) and Allen Rovick (AAR) have carried out keyboard-to-keyboard tutoring using a program, CDS [Li et al., 1992], that establishes communications between two computers using modems. Identifiers are produced by running a numbering program on the transcripts of human tutoring sessions. An identifier such as “K38-st-35-1” means that this sentence comes from the thirty-eighth keyboard-to-keyboard session, the student’s turn, the thirty-fifth turn in the session, and the first sentence in that turn.

Our research on multiple tutoring protocols was based on the study of human tutoring sessions with the goal of understanding human tutoring. The tutors had decided to use a protocol they had designed for our intelligent tutoring system in their human tutoring sessions. This protocol specifies that the tutor collect predictions first and tutor afterwards, in order to provide us with examples of the kind of tutoring they wanted the system to produce. I discovered in the analysis described here, however, that the tutors did not always follow the protocol. Sometimes the tutor did not wait until the student finished the predictions. If the student started out with poor predictions then the tutor immediately began to guide the student in the right track with hints or explanations. So, in fact, they changed the tutoring protocol to best fit the student's needs at the time.

Different approaches may be appropriate to different types of ITS. Beck et al. [1998] classified ITSs along two dimensions: abstraction of the learning environment and the knowledge type of the instruction. The simulation-based ITS is one of the abstraction types in the learning environment dimension. These simulation-based systems attempt to provide instruction by simulating a realistic working environment to reduce both the cost and the risks of training. The other type is the exact opposite of the simulation-based ITS. These systems provide problems for the student to solve without trying to connect those problems to real world situations.

In order to make development more manageable, ITS tend to concentrate on teaching one type of knowledge such as procedural skills, including the method of performing a particular task. As analyzing the domain knowledge is the key point of the system, cognitive psychology questions about human skill acquisition are the most important research topics in this area. The ITSs based on this principle, such as CIRCSIM-Tutor and Sherlock II [Katz et al., 1992], are cognitive tutors. The other ITS type is the knowledge-based tutor, which is based on the mental model concept. This type of ITS requires a large domain knowledge base. Because of a lack of skill acquisition or expert performance, these systems are forced to use general teaching strategies [Beck et al., 1998]. As CIRCSIM-Tutor is cognitive tutoring system, our analyses are based on human tutoring transcripts to obtain both domain knowledge and pedagogical knowledge.

1.2 Goals

The goal of my research is to improve two adaptive dynamic planning capabilities in ITSs. My first contribution is a Curriculum Planning Model. This research will endow ITSs with an individualized Curriculum Planning capability. The Curriculum Planner can evaluate the student's partially correct answers. It provides a more appropriate coarse-grained assessment method for measuring the student's performance.

My other contribution model is a new Discourse Planning Model. This research has succeeded in producing a Discourse Planning Model that can support Multiple Tutoring Protocols. The tutoring protocol controls the interaction between the tutor and the student in a tutoring session. Providing multiple tutoring protocols in Discourse Planning is really a dynamic planning problem, because the protocols differ mainly in when the tutoring should take place. The goal of Multiple Tutoring Protocols is to understand human tutoring so that we can make CIRCSIM-Tutor emulate it better. We also want to discover which tutoring protocol gives the best results in teaching causal reasoning. I used C5.0 to analyze a set of human tutoring transcripts to discover how and when human tutors switch protocols [Cho et al., 2000]. I also wanted to find out how the students feel about the issue of immediate feedback. To do so I analyzed the students' performance using CIRCSIM-Tutor and their responses to a questionnaire about their view of CIRCSIM-Tutor.

1.3 Organization of the Thesis

The remaining chapters are organized as follows. Chapter 2 introduces the ITS CIRCSIM-Tutor. Chapter 3 reviews previous work in the area of planning and in ITS. A coarse-grained assessment method is suggested in Chapter 4 in order to support the curriculum planner. My analysis of the CIRCSIM-Tutor Version 2.6 tutoring transcripts appears in Chapter 5. Next, my Curriculum Planning model is described in Chapter 6. Chapter 7 discusses the multiple tutoring protocols in discourse planning models. And Chapter 8 presents my conclusions.

CHAPTER II

CIRCSIM-TUTOR

2.1 Domain

The domain of CIRCSIM-Tutor is cardiovascular physiology. CIRCSIM-Tutor assists students to reason about the qualitative, causal responses of the human circulatory system when the blood pressure is perturbed. The system asks the student to enter predictions in the Prediction Table [Rovick and Michael, 1992] indicating how the perturbation affects seven important physiological variables in the first of three different stages of the response, and then it initiates a tutorial dialogue to remedy any errors. Then the system asks the student to predict how the variables change in the second stage and again carries out a remedial dialogue. Finally it repeats this process with the third stage. Table 2.1 shows a prediction table for the perturbation “Increase Venous Resistance (RV) to 200% of normal.”

Table 2.1 The Prediction Table of the Procedure “Increase RV to 200% of Normal”

Physiological Variable	DR	RR	SS
Inotropic State (IS)	0	+	+
Central Venous Pressure (CVP)	–	–	–
Stroke Volume (SV)	–	–	–
Heart Rate (HR)	0	+	+
Cardiac Output (CO)	–	+	–
Total Peripheral Resistance (TPR)	0	+	+
Mean Arterial Pressure (MAP)	–	+	–

The three stages are the Direct Response (DR): the immediate change in the variables induced by the perturbation; the Reflex Response (RR): the change induced by the response of the central nervous system to the change in blood pressure; and the Steady State (SS): the long term balance between the effects of the perturbation and the effects of the negative feedback. The system asks the student to predict the qualitative change from the values before the perturbation to the new steady state: + means Increased; – means Decreased; and 0 means unchanged. The primary variable is the first variable in the DR column of the Prediction Table that is affected by the current perturbation. Therefore the student should identify and predict the primary variable first.

The seven variables in the Prediction Table are as follows.

- Inotropic State (IS): the ion state (ion density) in the heart muscle. It determines the heart muscle contraction force.
- Central Venous Pressure (CVP): the pressure of the blood returning to the heart.
- Stroke Volume (SV): the volume of blood pumped out of the heart per stroke.
- Heart Rate (HR): the number of heart beats (strokes) per minute.
- Cardiac Output (CO): the volume of blood pumped out of the heart per minute.
- Total Peripheral Resistance (TPR): a measure of resistance to blood flow.
- Mean Arterial Pressure (MAP): blood pressure in the arteries.

Figure 2.1 illustrates the causal model underlying CIRCSIM-Tutor. The arrowheads indicate the direction of causal effects. The plus and minus signs indicate whether the causal relationship between the connected variable is direct or inverse. The Baroreceptor (BR) measures changes in the blood pressure, then the Central Nervous System (CNS) plays the role of a negative feedback controller that regulates the blood pressure in the system.

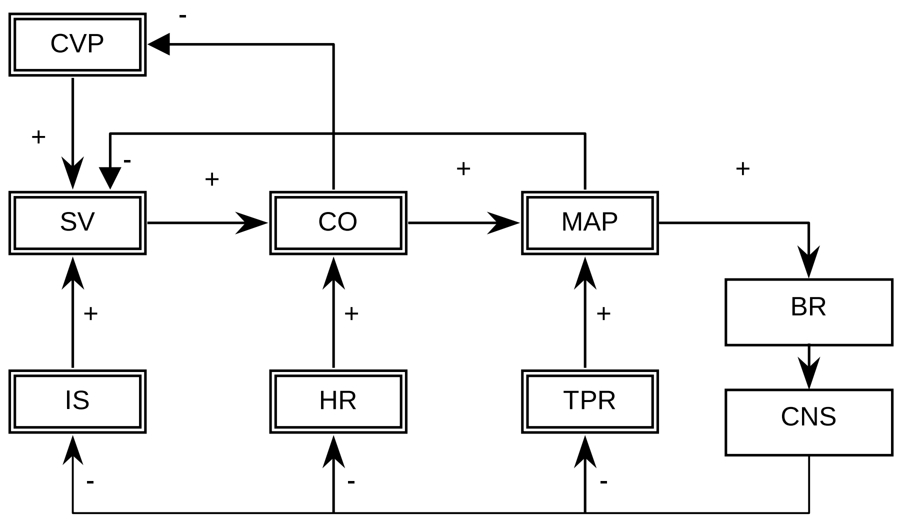

Figure 2.1 The Concept Map

2.2 Structure

2.2.1 ITS Structure. Burns and Capps [1988] argued that an ITS should have four interconnected software modules as shown in Figure 2.2: an Expert Module, a Student Module, a Tutor Module, and an Interface Module.

2.2.1.1 Expert Module. The expert module contains information about the domain knowledge, which are the facts and concepts to be taught. It provides a source for the knowledge needed in problem solving and explanations of concepts and responses to students.

2.2.1.2 Student Module. The student module maintains the student model, which contains the current status of the student's domain knowledge. It makes inferences about the student's current status by interpreting the student's actions. The result of inferencing is updates to the student model. Because of a narrow communication channel, only the student's actions such as predictions and answers to questions are available. Thus, it is difficult to determine the student's knowledge status accurately. Therefore, the student model may not be perfect. However, a student model is necessary in order to customize instruction for each individual student [Beck et al., 1998]. The student model may include the student's preferences. The tutoring history of the student increases the performance and the individualization of the tutoring.

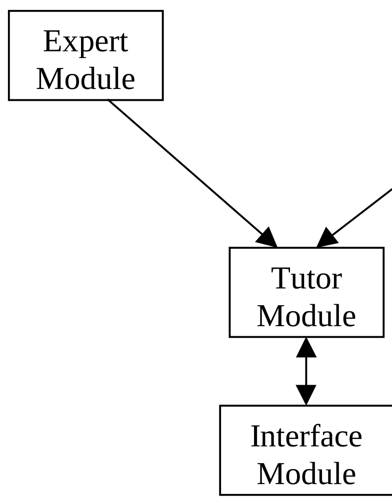

Figure 2.2 Structure of ITS

2.2.1.3 Tutor Module. The tutor module contains a planner with rules. The planner generates the plans for teaching each item of subject matter, which needs pedagogical decisions. An ITS always makes pedagogical decisions. The tutor module monitors the student's actions and adapts the system responses to the student.

2.2.1.4 Interface Module. The interface module manages the flow of communication between the student and the ITS. As media technology improves, an ITS can have a rich communication bandwidth. The interface must be clear. An ambiguous presentation style may lead to the misinterpretation of information. The interface module should make it easy for the student to understand what to do. Otherwise, the student may get tired soon, which will reduce the enthusiasm.

2.2.2 CIRCSIM-Tutor Structure. The Planners determine what, when, and how to teach next. Figure 2.3 illustrates the three planners in CIRCSIM-Tutor Version 3.0. First, the Curriculum Planner selects the next problem (procedure). Second, the Discourse Planner selects domain topics and tutoring strategies while the student is solving the procedure and plans the forthcoming dialogue.

The CIRCSIM-Tutor Knowledge Base contains the over all tutoring knowledge, which is needed for teaching the Cardiovascular System. It includes Procedure lists for the Curriculum Planner, a Tutoring History that keeps track of “How and what has been taught”, a Dialogue History that stores the current dialogue, a Domain Knowledge Base that stores knowledge about “What to teach”, a Pedagogical Knowledge Base, that stores knowledge about “How to teach”, and Error Patterns, which are possible causes of errors. The Input Understander module parses student input messages and produces a logic form. It uses a judger function to determine the correctness of each student response.

The Screen Manager in CIRCSIM-Tutor is the interface to the student. The Student Modeller constructs a student model that contains an evaluation of the Student's Domain Knowledge and maintains it during the tutoring sessions. The Surface Sentence Generation module generates utterances from the discourse plans.

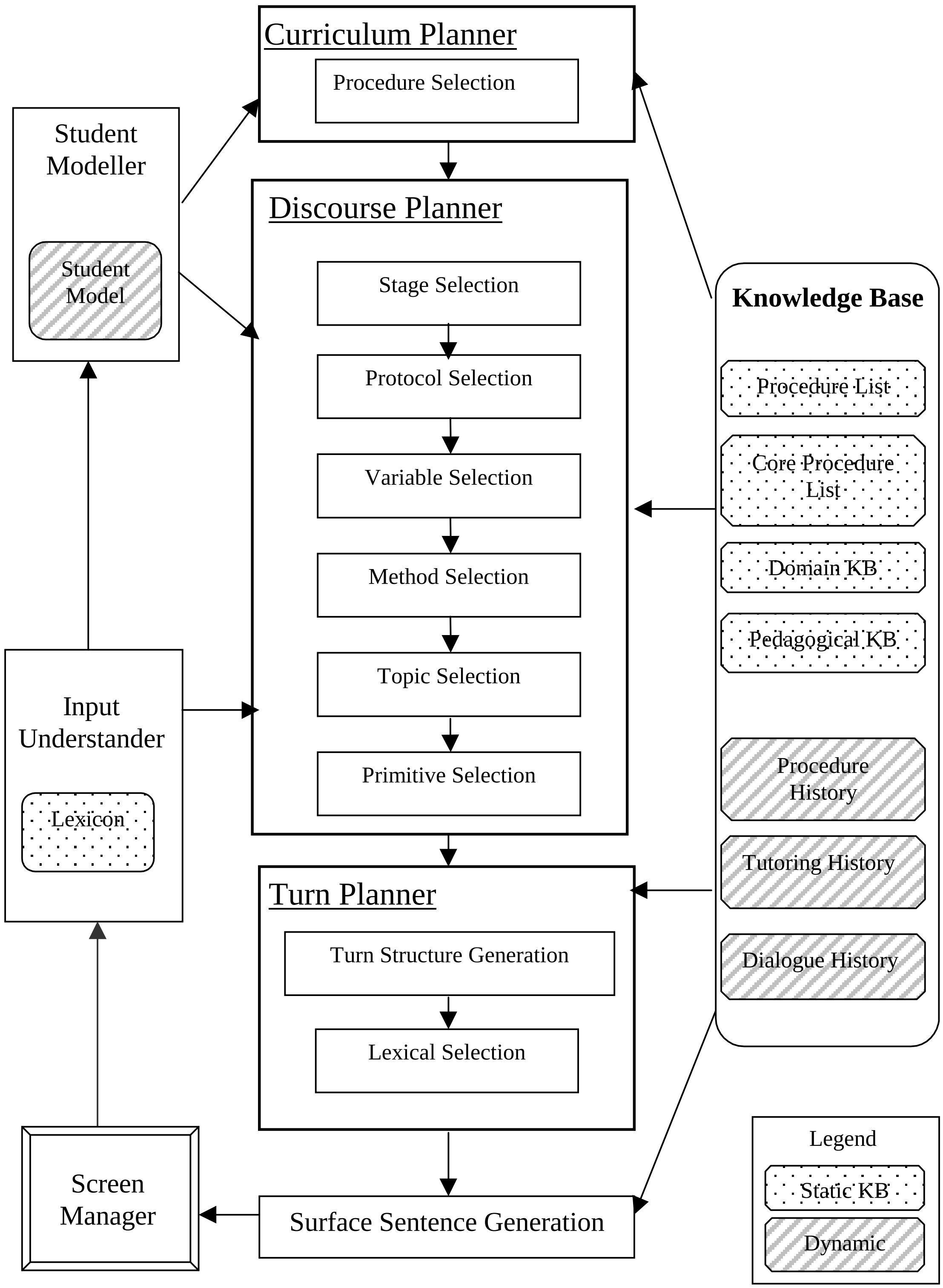

Figure 2.3 Planners in CIRCSIM-Tutor

2.3 Related Work on CIRCSIM-Tutor

Hume [1995] analyzed the behavior of the human tutors and suggested the relationship between student modelling and hinting in CIRCSIM-Tutor. Brandle [1998] proposed an acknowledgement model. I analyzed and implemented an Adaptive Dynamic Curriculum Planner [Cho et al., 1999]. In Discourse Planning, Gregory Hume [1995] analyzed the hinting feature now implemented by Yujian Zhou [Zhou et al., 1999b]. Stefan Brandle [1998] analyzed acknowledgements. Freedman et al. [1998] analyzed and integrated many discourse features. Freedman also produced the rules needed for a new Discourse Planner. Ramzan Ali Khuwaja [1994] defined three tutoring protocols and described the necessity for a multi-turn planning feature in tutorial planner. Gregory Sanders [1995] introduced a multi-turn planning feature in the discourse planner. JungHee Kim and Reva Freedman analyzed and defined tutoring methods in discourse planner. Several researchers have investigated mixed initiative discourse planning capability [Sanders, 1995; Freedman, 1997a; Shah et al., 2000].

2.3.1 Discourse Planner. Woo [1991] designed and implemented the instructional planner of CIRCSIM-Tutor Version 2.x. The instructional planner, which is replaced by the discourse planner in Version 3, consists of two kinds of planning mechanisms, which are controlled by the plan controller. The lesson planner determines the content and sequence of the subject matter to be taught in a single lesson. Several levels of planning then determine the details of the discourse. The planning sequence is first goal generation, then strategy planning, and finally tactical planning. These plans are placed in a goal stack.

Freedman [1996b] described the distinction between two different domains of planning Socratic dialogue. The discourse planner devoted to satisfying the goals of the tutorial dialogue. The turn planner is devoted to linguistic goals not dependent on tutorial structure only. A tutorial goal may be realized by several dialogue turns or a fragment of a dialogue turn. The architecture of the discourse planner described in this section started with Freedman's [1996a] description of tutorial goals and discourse structure.

The discourse planner is responsible for controlling interactions between the tutor and the student. It decides how the tutor should respond to a student. The discourse planner produces tutoring primitives based on the tutoring schemata (see figure 2.4).

The CIRCSIM-Tutor group has been using an SGML format to make up and analyze text. In SGML the opening delimiter of a piece of text is an identifier in angle brackets, and the closing delimiter is identical except for an initial slash [Freedman et al., 1998].

For example,

```
<T-does-neural-DLR>
   <T-tutor-mechanism>
       Tutoring Turns
   </T-tutor-mechanism>
</T-does-neural-DLR>
```

2.3.2 Turn Planner. In the computer tutor, the ultimate result of planning is a sequence of informing and eliciting dialogue acts. The turn planner processes this dialogue sequence. It collects all the tutorial primitives within a turn from the discourse planner and selects the related lexical items. It then passes these features to the surface sentence generator for sentence generation. This process is illustrated in Figure 2.5 [Yang et al., 2000b].

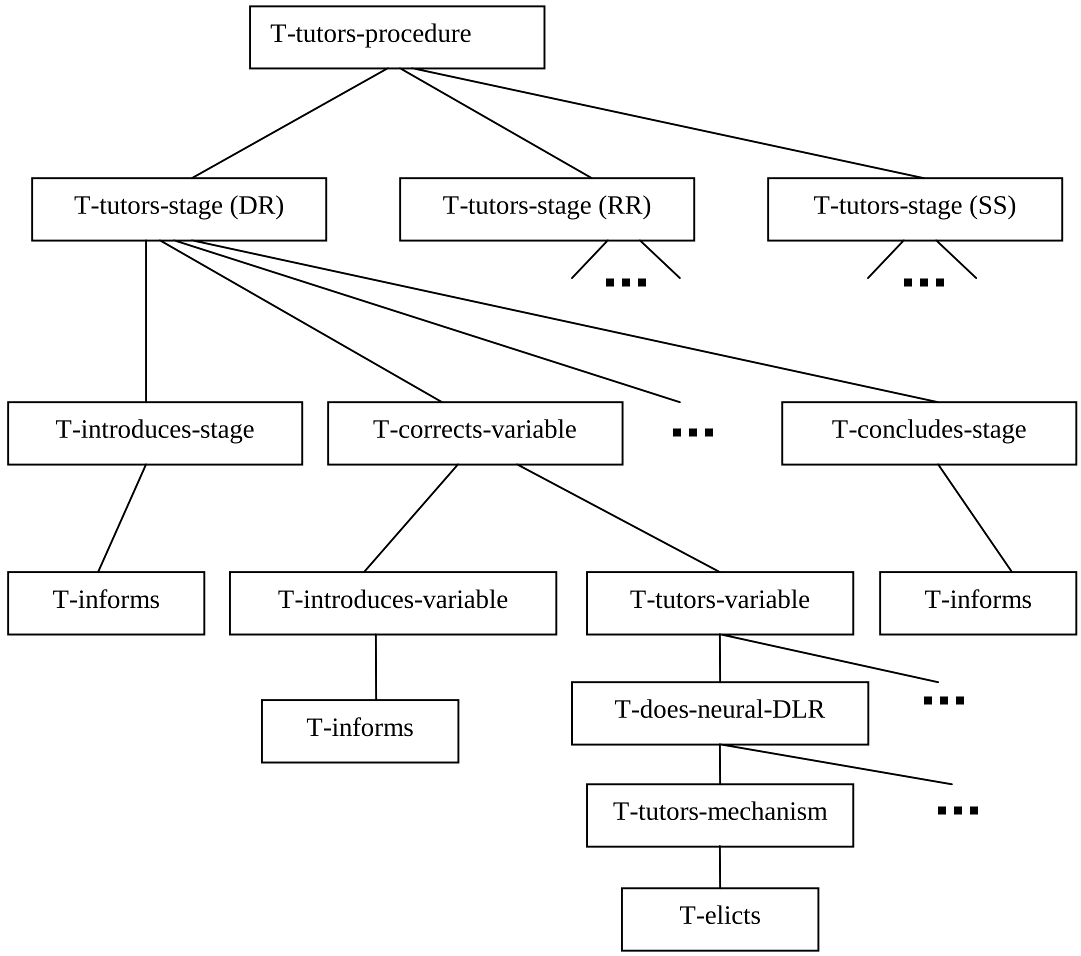

Figure 2.4 Tutoring Schemata of the Discourse Planner

The sentence generator processes the output of the turn planner. Based on an analysis of selected sentences in the transcripts [Kim et al., 1998], sentences can be described by a small number of features, for example:

<dm> <soft> what is the value of <var> <stage>?

where the following slots are determined by features:

- <dm> stands for an optional discourse marker
- <soft> stands for an optional politeness idiom or softner, e.g. “can you tell me”

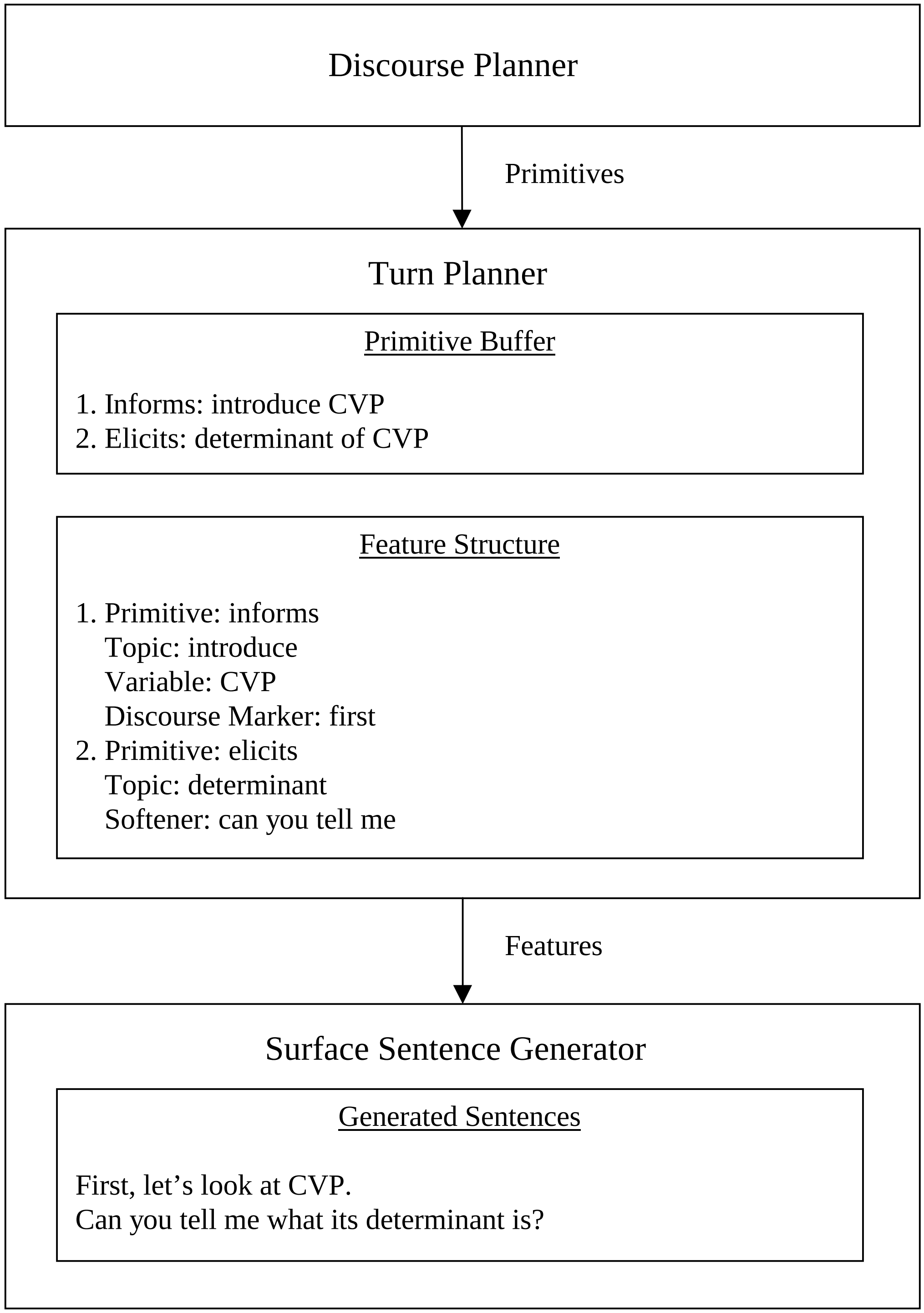

Figure 2.5 Turn Planner and Sentence Generator

- <var> stands for a variable name or abbreviation, or a pronoun
- <stage> stands for an optional prepositional phrase denoting the stage of the problem

A final realization might be “Now can you tell me what the value of TPR is in RR?”

2.3.3 ATLAS. Andes is an intelligent tutor that provides the student with a learning environment in which to solve physics problems in Newtonian physics [Schulze et al., 2000]. Andes is a student-led system because the tutor is always responding to the student’s action [Freedman, 1999]. CIRCSIM-Tutor is a tutor-led system in which the tutor first identifies the student’s error then the tutor helps the student learn how to solve the problem through hints and questions.

Atlas is another physics tutor developed by the same group. It conducts a mixed initiative natural language conversation between tutor and student. Atlas uses a Hierarchical Task Network style dynamic planner. This means that a goal is decomposed into a set of ordered subgoals [Freedman, 1999].

The ATLAS Planning Engine (APE) is an integrated planning and execution module [Freedman et al., 1998]. APE was designed to be an UCPOP-style (partial-order) planner [Barrett et al., 1995]. The core of the planning in CIRCSIM-Tutor Version 3 is also APE. This means that our planning is operator-based. Therefore every action is represented as operator. An ATLAS planning operator has the following form:

```
(def-operator  operator-name
 :goal ()
  :filter (()()...)
  :precond (()()...)
  :recipe (()()...)
  :temp (()()...)
```

Here is an example of an operator of the curriculum planner:

```
(def-operator obtain-problem-id-expert-cp
 :goal (did-obtain problem-id-cp)
 :filter ()
 :precond ((not (e-novice-user-is)))
 :recipe ((goal (did-obtain st-adj-req))
          (goal (made-a-pdr-set))
          (goal (did-obtain choose-procedure)))
 :temp ())
```

An operator has a name and the following fields

2.3.3.1 Goal. The goal is the intention of performing this operation. After successful completion of the operation, the goal will be achieved.

```
:goal (did-obtain problem-id-cp)
```

2.3.3.2 Precond. The precond is a list of the Well Formed Formulas (WFFs) that must be in the database in order to run the operator. The result of the precond list is dynamic and can be changed as planning goes on. Multiple conditions can be used. They are connected with the and-operator. The not function is also available.

```
:precond ((not (e-novice-user-is)))
```

This precond example shows that the planner checks whether the student is a novice user. If the result is true then the planner executes the recipe steps.

2.3.3.3 Recipe. The recipe contains the steps to be taken to achieve the goal. Eight types of steps can be included.

- Primitive: items/tasks which are not decomposable; in the text planner “say something” is a primitive

```
(primitive (say '(" Well, You just finished the most difficult procedure")))
```

- Interactive: primitive in which there is some sort of user interaction. In a dialog planner, “say” would often be followed by “ask”

    (prim-interactive (ask '("Please select a procedure number")))

- Goal: a goal to be met before continuing with rest of the plan

    (goal (did-obtain choose-procedure))

- Fact: A kind of test. fact(x) means if x is true then continue, otherwise cancel the rest of the recipe. This is a way to accomplish “if..then” reasoning. Fact can also contain preconditions and filters.

    (fact (e-equals ?result-set nil))

- Prune: It takes zero or more steps from a plan and replaces them with zero or more steps. The meaning of this step is don’t do X, do Y instead.

    (prune-replace ((w-level-is prompt)
    (goal (did-tutor-deep mechanism ?partial ?neural post-error))))

- Retry: It takes zero or more steps from the plan and tries to replan the current goal at the top of stack. The meaning of this step is don’t do X, go back to what the current goal is and find a different way to do it. It is possible that prune and retry could be used to accomplish the same thing.

    (retry-at (w-on-stack made-a-pdr-set))

- Retract: Retract a fact from the database.

    (retract (core-perturbation-level-is ?x))

- Assert: Assert a fact to be added to the database.

    (assert (core-perturbation-level-is 3))

2.3.3.4 Filter. The filter is a list of the WFFs that must be in the database in order to consider running the operator. The filter slot is used for static properties, such as properties of domain objects, while preconditions are used for characteristics that can change as the dialogue progresses, e.g., the number of times a particular construct has been used.

2.3.3.5 Temp. A temp is a WFF that is true while the operator is being run. It may be used instead of the assert/retract steps.

The system stores goals in an agenda, which is implemented as a stack. To initiate a planning session, the user invokes the planner with a goal. The system stores the initial goal on the stack then searches the operator library to find all operators whose goal field matches the goal on top of the stack and whose filter conditions and preconditions are satisfied. If more than one (operator, binding list) match is found, the last one found is used.

CHAPTER III

PLANNING ISSUES

3.1 Planning Environment

Russell and Norvig [1995] discussed five environmental categories in an ITS based on planning capabilities.

3.1.1 Accessible vs. Inaccessible. If sensors of an agent access the complete state of the environment, then the environment is accessible to that agent. An environment is effectively accessible if the sensors detect all aspects that are relevant to the choice of action. An accessible environment is convenient because the agent need not maintain any internal state to keep track of the world.

3.1.2 Deterministic vs. Non-deterministic. If the next state of the environment is completely determined by the current state and the actions selected by the agents, then the environment is deterministic.

3.1.3 Episodic vs. Non-episodic. In episodic planning each episode consists of the agent perceiving and then acting. The quality of its action depends just on the episode itself, not on previous episodes. It is simpler because the agent does not need to think ahead.

3.1.4 Static vs. Dynamic. If the environment can change while an agent is deliberating, then the environment is dynamic.

3.1.5 Discrete vs. Continuous. If there are a limited number of distinctions, clearly defined percepts and actions, the environment is discrete. For example, chess is discrete. There are a fixed number of possible moves on each turn.

The environment of a Medical Diagnosis System is Inaccessible, Non-deterministic, Non-episodic, Dynamic, Continuous. So it is the most challenging environment. The environment of CIRCSIM-Tutor, as well as the Interactive English Tutor [Russell and Norvig, 1995], is Inaccessible, Non-deterministic, Non-episodic, Dynamic, and Discrete.

3.2 Planning Approach

Planners can be classified into linear or non-linear planners in terms of goal dependency. The term non-linear means the sub-goals are partially ordered and interleaved. Linear planning assumes totally ordered sub-goals. Another criterion is the level of abstraction. Abstraction is the process of taking real-world domain knowledge and filtering it into a format and quantity that is manageable for a planner to handle and use. Hierarchical planners generate goals at multiple layers of abstraction. Case-based, non-hierarchical planners use only one abstraction level and sometimes have difficulty in reaching the main goal [Grama and Gonzalez, 1998; Harris and Cook, 1998]. CIRCSIM-Tutor has adopted hierarchical and linear planning approaches.

3.2.1 Hierarchical Planning. Hierarchical planning has been used to reduce the computational cost of planning [Yang, 1997, p.163]. The idea of hierarchical planning is to distinguish between goals and actions of different degrees of importance and solve the most important problem first.

Hierarchical planners divide a domain into abstraction levels. Hierarchical planners first find an “abstract” solution for part of the planning problem, then climb down to lower abstraction levels to refine this solution by incorporating additional details.

Yang suggested two ways in which details can be inserted in a plan. The first, precondition-elimination abstraction, mimics the human way of exploring and solving subgoals with the priority of importance. The second method is hierarchical task-network planning. Planning problems and operators are organized into a set of tasks. A high-level task can be reduced to a set of ordered low-level tasks [Yang, 1997, pp. 163 -170].

Harris and Cook designed the Hierarchical-Analogical planner. The plan generation portion of the planner would have the same search characteristics as conventional hierarchical planning, except that each abstraction level would have a proposed solution ready for replay or revision [Harris and Cook, 1998] .

3.2.2 Case-Based Planning

3.2.2.1 Case-Based Planning Concepts. Aamodt and Plaza [1993, p.2] define case-based reasoning: “To solve a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation”. Case-Based Planning is planning from experience. It has two characteristics different from other major AI approaches [Aamodt and Plaza, 1993]. First, a case-based planner uses libraries of goals and plans. It relies on an episodic memory of past planning experience instead of rules that are based on problem domain knowledge. A second difference is that a case-based planner uses an adaptation and learning approach. Memory organization, indexing, plan modification, and learning are important parts of case-based planning. In order to reuse the plans, the case-based planner records whether the plans succeed or not. Instead of constructing new plans, a case-based planner recalls and adapts past plans. The general rule-based adaptation is using a fixed set of adaptation rules. However, Leake [1995] claims that the specific rules are easy to apply and reliable, but only apply to a narrow range of adaptation problems. He presents a new method in which a case-based reasoning system can learn adaptation knowledge from experience. As new adaptation problems are solved then the adaptation component stores the successful adaptation episodes in an adaptation case library for future use. When no relevant cases are available, rule-based adaptation is used.

We can easily imagine case-based reasoning application areas in the real world. For example, after examining a particular patient, a physician remembers a patient that he treated two weeks ago. The reminding process was caused by a similarity of important symptoms. The physician uses the diagnosis and treatment of the previous patient to determine the disease. Given a difficult credit decision task, a financial consultant remembers a previous case and then decides whether to approve a loan application or not.

3.2.2.2 Case-Based Planning Processes. Case-Based Reasoning is described by the following processes.

- Retrieve the most similar cases
- Reuse the information and knowledge for solving the problem
- Revise the proposed solution
- Retain this experience to be used in future problem solving

Figure 3.1 illustrates these processes. A problem description defines a new case. The new case is used to retrieve a case from the collection of previous cases. The retrieved case is combined with the new case into a solved case through reuse. The revised process tests the solution for success; if it fails, it repairs it. During the retain phase, useful experience is stored for future reuse. The case base is updated by a new learned case.

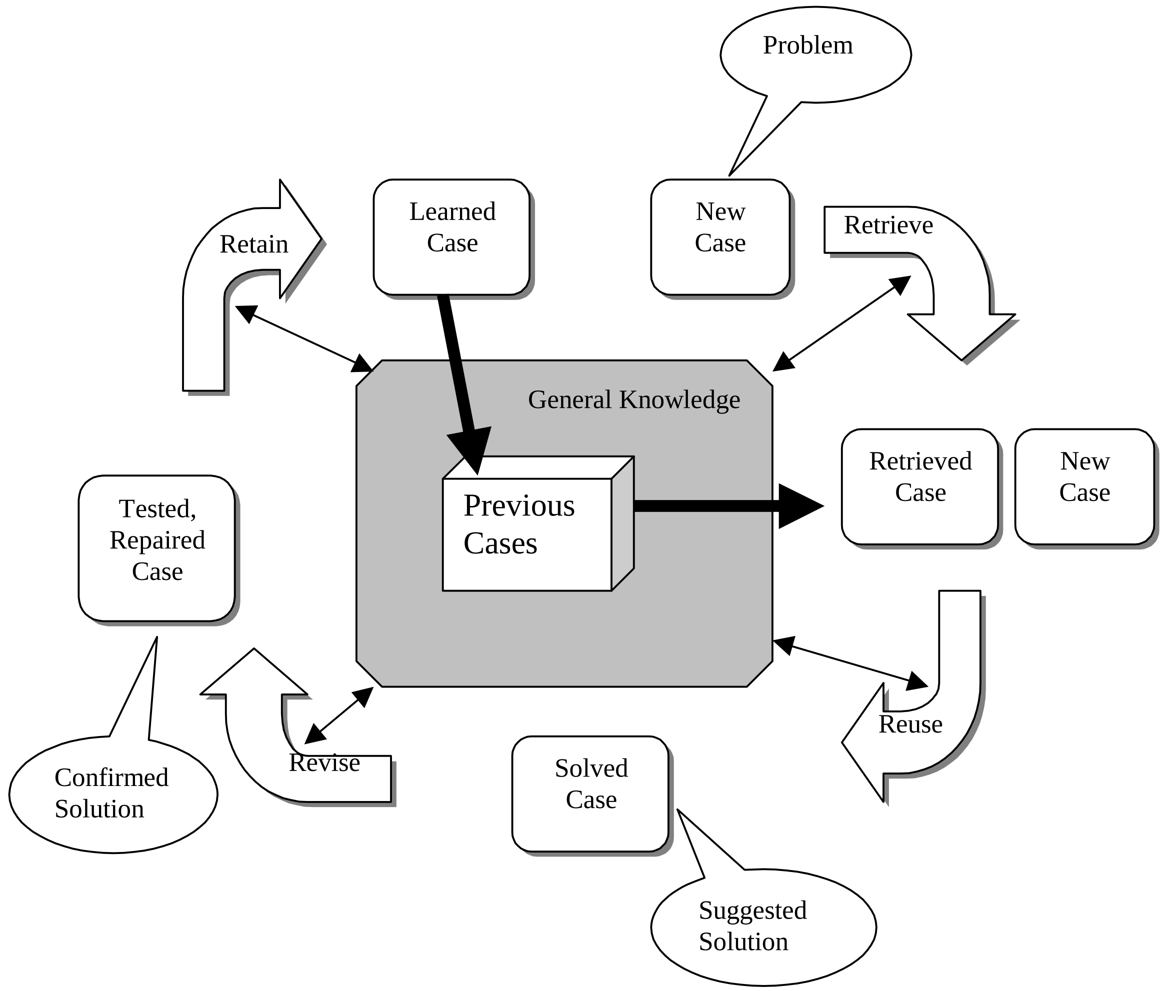

Figure 3.1 The Case-Based Reasoning Cycle [Aamodt and Plaza, 1993]

3.2.2.3 Case-Based Reasoning Methods. The typical case-based reasoning method has some characteristics. A typical case has rich information in a feature vector. This method can adapt a retrieved solution when applied in a different problem. General Case-Based Reasoning can be classified into four types by their reasoning characteristics [Aamodt and Plaza, 1993].

- Exemplar–based reasoning: sometimes solving a problem is a classification task. This means finding the right class for the unclassified exemplar. The class of the most similar past case becomes the solution of the classification problem. The set of classes consists of the set of possible solutions. However, the modification of a solution found is not within the scope of this method.

- Instance-based reasoning: this is a specialized kind of exemplar-based reasoning used in a highly syntactic Case-Based Reasoning approach. Because of the lack of the guidance from background knowledge, a large number of instances, with simple stored information, are needed for reasoning.

- Memory-based reasoning: this approach emphasizes a collection of cases as a large library, and reasoning as a process of accessing and searching this library. Memory organization and access are foci of this method. Often an important characteristic of this method is the utilization of parallel processing techniques.

- Analogy-based reasoning: this method sometimes used as a synonym of the typical case-based reasoning. However, the main difference is that this method solves new problems based on past cases from a different domain [Alterman, 1990], while the typical case-based reasoning method uses cases from the same domain.

3.2.2.4 Case-Based Planners. PERSUADER [Sycara, 1988] is a planner that acts as a labor mediator. It resolves conflicting goals of a union and company by finding compromises acceptable to them both. The input of the PERSUADER is the set of conflicting goals (e.g., wages, holidays, and pensions) of the company and its local union, and the context of the dispute (e.g., economic conditions in the industry and information about the disputants). The output is either a single plan in the form of an agreed upon compromise, or an indication of failure if did not reach agreement within a particular number of proposals.

The domain of CHEF, a case-based planner, is Szechwan cooking [Hammond, 1986]. Its task is to build new recipes on the basis of the user requests. CHEF’s input is a set of goals for different tastes, textures, ingredients, and types of dishes. And the output is a single plan in the form of a recipe that satisfies the user’s goals.

3.3 Adaptive Dynamic Planning

3.3.1 Student Assessment. The component of an ITS that represents the student's current status of knowledge is the student model [VanLehn, 1988]. Inferring a student model is called diagnosis or student modeling. The student model is used for several purposes. The ITS asks the student model for the level of mastery on the current topic. Then the ITS decides whether to advance to the next procedure or topic and its difficulty level. Furthermore, the ITS can use the student model to determine the appropriate level of detail in explanations and hints to the student.

3.3.1.1 Considerations. The student model attempts to represent the student's behavior and knowledge. However, this is not a simple task because there are limitations on the bandwidth of the communication channel between the student and the ITS.

A perfect student model may not be necessary for planning. Self [1990] claimed that detailed user models do not necessarily enhance the capability of an ITS. Good teaching can be done without a detailed user model, because in good teaching serious misconceptions are avoided, and errors are repaired on the spot. He pointed out some problems with student modeling. It is hard to identify the error and to determine the appropriate granularity of detail. However, defining, representing, and recognizing the students' misconceptions is more difficult than identifying the error. In order to make an accurate model; the student modeler needs to record the student's prior knowledge, immediate learning context, and personal learning preferences.

Self also suggested some principles to bypass the student modeling problem as follows [Self, 1990 pp. 110 - 120].

1) Avoid guessing - get the student to tell you what you need to know

2) Don't diagnose what you can't treat

3) Empathize with the student's beliefs and don't label them as bugs

4) Don't feign omniscience - adopt a "fallible collaborator" role

I agree CIRCSIM-Tutor has definitely obeyed principles 1) and 2). In order to assess the current knowledge of the student, CIRCSIM-Tutor only uses the prediction results and answers in the tutoring sessions. However the other two principles may not appropriate for our particular ITS. Slogan 3) argues that "the tutor knows best" is an arrogant style in an ITS. An ITS that has a discrete environment can determine whether the student answer is right or not. CIRCSIM-Tutor has a discrete environment, and the domain experts have defined the errors. Though describing a correct knowledge base is possible, adequate analysis of student misconceptions is not impossible, as mentioned in slogan 4). An ITS should guide students to give up their misconceptions with hinting and a Socratic tutoring strategy.

3.3.1.2 Assess the Student's Beliefs. Student Models should attempt to describe not only what the students know but also what they believe [Self, 1990]. Therefore it is important for an ITS to recognize the student's conceptions in the domain knowledge. Knowledge of a student's conceptions can be used to give hints or correct errors in the tutoring session.

Most ITS use an expert model as well as a student model. The expert model provides the correct answer for each procedure. There are two types of differences between the student model and the expert model [VanLehn, 1988 p.62]. A missing conception is an item of knowledge that the expert has and the student does not. A misconception is an item of knowledge that the student has and the expert does not. Conceptually, an ITS should have a knowledge base for the expert model and a different knowledge base for the student. Most ITS merge these two models. So the student model is based on an expert model and a collection of differences.

Some ITS have a student model that only represent the missing conceptions. Such student models are overlay models. This model regards the student model as a subset of the expert model. Other ITS can represent both missing conceptions and misconceptions. Such a model employs a library of predefined misconceptions, and an overlay for missing conceptions. The members of the library of predefined misconceptions are often called bugs. The student model consists of an expert model and bug library. An ITS that uses the buggy model approach maintains a library of plans for solving a problem and a bug catalogue of common incorrect or buggy variants of these plans.

Koehn and Greer [1994] claim that, given a specific problem and a fairly complete knowledge base of possible plans for solving the problems, instructional systems based on bug catalogues will frequently diagnose student behavior and misconceptions.

Another mechanism for measuring the student knowledge is Bayesian networks [Martin and VanLehn, 1995]. These networks reason probabilistically about the student's knowledge status. Each node in the network has a probability, which indicates the likelihood of the student's status with respect to each piece of knowledge.

Shim suggested that CIRCSIM-Tutor should have both Bug Library Modeling and Overlay Modeling [1991]. Overlay Modeling assumes that the student's knowledge is a subset of the expert's knowledge. Bug Library Modeling assumes that the student's current skill reflects a composite of correct and incorrect elementary sub-skills. Often an Overlay Model is used to assess the student's declarative knowledge, while a Bug Library model is used to assess the student's procedural knowledge. Though CIRCSIM-Tutor has a perfect expert model that was defined by domain experts, it is very expensive and difficult to make a perfect environment. Therefore we need to choose appropriate assessment granularity.

3.3.2 Adaptive Planning

3.3.2.1 Incomplete Information. In real world domains, planners have to deal with both incomplete and incorrect information. The planners should have a dynamic planning capability for an inaccessible, non-deterministic, non-episodic, and dynamic environment. There are two ways to deal with this problem.

Conditional (Contingency) Planning: A conditional plan tests the environment to determine whether the prepared plans are appropriate or not. Rather than replanning at run time, the planner develops a set of plans for every expected contingency. Conditional planning is necessary when the environment is incomplete. A conditional plan has sensing actions to test for the appropriate conditions. For example, the shopping agent includes a sensing action in its shopping plan to check the price of some object in case it is too expensive [Peot and Smith, 1992].

Execution Monitoring: Alternatively, the planner can monitor what is happening while it executes the plan. It can then do replanning to achieve its goals in the new situation. For example, if the agent discovers that it does not have enough money to pay for all the items it has picked up, it returns some and replaces them with cheaper versions.

If conditional planning makes almost perfect plans, then it guarantees a higher probability of success but it is very expensive. However, conditional planning requires that all possible conditions must be planned for to increase the probability of success.

3.3.2.2 Case Studies. The Berkeley UNIX Consultant (UC) [Wilensky et al., 1989] is an ITS that has a natural-language interface that helps naive users learn about the UNIX operating system through an English dialogue. UC has a Tutorial Planner that consists of two planning modules: The Agent (UCEgo) plans what to do in response to the user request. The Domain Planner (KIP) plans how to accomplish the goal. UC has an adaptive dynamic planning capability. The status of the user can be categorized into four levels according to his/her expertise in UNIX concepts, which have been categorized into four difficulty levels. It answers and provides examples that are adapted to the status of the student's level of understanding. However, UC does not have a Curriculum Planner. The discourse planner (UCExpress) and the text generator (UCGen) comprise the natural language interface of UC. But, it does not have a mixed initiative capability because UC always answers the student's questions.

Meno-Tutor [Woolf, 1984] is an ITS designed to teach causal reasoning. It has been applied to reasoning about rainfall and about looping constructs in the programming language Pascal. Meno-Tutor uses knowledge about tutoring strategies, complex communication skills, and student’s knowledge status to plan a reasonable tutoring discourse. The teaching component (planning module) adapts its response to the student’s level of knowledge and discourse history. And the teaching component can change its tutoring strategy if the student is not progressing well.

ELM Adaptive Remote Tutor II (ELM-ART II) [Weber and Specht, 1997] is a WWW-based interactive tutoring system to support learning programming in LISP. It supports adaptive navigation with an individualized user model. ELM-ART II plans and guides the next step just by sequencing the next teaching topic. And it finds the most relevant example from its previous individual learning history.

Sherlock II [Katz et al., 1992] was developed to train avionics technicians to troubleshoot faults in a complex electronic testing system. One of the main purposes of Curriculum Planning in Sherlock II is reducing dependence on their student model, which is not accurate and complete. The sorted problem set is divided evenly into four difficulty levels. The basic scheme is that if the student's performance is above a certain threshold, Sherlock will move the student up. If the student's score is below a threshold, then Sherlock will move the student down. After finishing a problem, Sherlock readjusts the selection point to choose a select set (the set of candidate next problems) with student input. For example, if the student asks for a harder problem and has done well in the current performance, then Sherlock adjusts the selection point to the student's selection with the equation.

$$S_{i+1} = (2S'_{i+1} + \delta) / 2$$

S'_{i+1}: Sherlock's initial placement of the selection point
δ: difference between Sherlock's and student's relocation of it
S_{i+1}: corrected selection point

This scheme seems to work well in the Sherlock II domain, which is quite different from ours. The main difference is the variety of problem levels. Sherlock II has 4 Difficulty Levels, but cardiovascular procedures have 5 procedure difficulty levels and 4 procedure description levels. The final difference is the assessment method. Sherlock II uses a comparative assessment, which simulates expert performance on the same problems and sets thresholds to adjust the selection point.

3.3.2.3 Multi-turn Planning. To do more effective tutoring we need to plan the dialogue multiple turns in advance. Multi-turn planning is particularly complex, because the student's response or initiative may require the tutorial planner to replan the dialogue.

Sanders [1995] demonstrated the necessity of the multi-turn planning capability. By analyzing the keyboard-to-keyboard transcripts, he discovered some multi-turn discourse structures such as Directed Line of Reasoning (DLR) exchanges, when a tutor leads a student step by step through a line of reasoning that consists of coherent bite-size questions. Replanning may be required during a DLR if the student introduces a wrong answer or moves to take the initiative.

Sanders pointed out that our experts, Joel Michael and Allen Rovick, sometime deliver multi-turn summaries, especially when the exchange has been long and complex. Because the students requested a correct solution of the Direct Response phase, before the student moves to the Reflex Response phase, we have added to CIRCSIM-Tutor version 2 a canned summary.

DLRs are used to evoke cause and effect reasoning based on information the student already knows [Khuwaja, 1994; Sanders, 1995]. DLRs may be used as hints, as summaries, and also to remedy misconceptions. They allow the tutor to verify that the student really knows the material and allow the student to review the previous steps. A DLR often begins and ends with an explanation, and continues with a discourse marker such as “And” [Sanders, 1995]. Here is an example of a DLR (adapted from session K12, beginning with turn 65).

Tu: Since we are now in the Reflex Response period, the variables that change first are the ones that are neurally controlled. Which of these variables would be affected first?

St: CC [Cardiac Contractility]

Tu: Of course! And in what direction?

St: Decrease

Tu: Right again. And how would that affect SV? [Stroke Volume]

St: Decrease

Tu: Sure. And what effect would that have?

St: Decrease CO [Cardiac Output]

Tu: Yes again. Then what?

St: MAP d [Mean Arterial Pressure decrease]

Tu: Yes again. And it is MAP that is regulated by the Baroceptor reflex, which is why it is called that.

[Sanders, 1995 p. 94]

During a DLR the tutorial planner uses a cause-and-effect chain that is normally at a given level of the knowledge base and that is a series of steps involving variables in the prediction table [p. 95]. These choices are based on the student model also. What the student knows and how well is an important factor of the construction of the DLR content. We hope to add a DLR facility at a later phase of the development of Version 3.

3.4 Mixed Initiative Planning

3.4.1 Basic Concepts

3.4.1.1 Definition. Mixed initiative means all participants have the ability to take the initiative in a dialogue. It means that either participant can change the direction or topic of the dialogue, or take the lead in discussing the current topic. Only one participant can have the initiative at a given time. Furthermore, all participants must be able to take control of the flow of a mixed initiative conversation. The participant who initiates a new direction in the problem solving process retains the initiative, unless a competing solution is proposed. Typical ITS have system initiative. Mixed initiative in an ITS means that the ITS allows the student to take the initiative and it responds to student requests. Shah et al. [2000, p3] defined a student initiative as "any student contribution to a dialogue that is not an answer to a question asked by the tutor."

As Cohen et al. [1998, p 3] quoted from a Burstein and McDermott paper, the object of research on Mixed Initiative Planning is "to explore productive syntheses of the complementary strengths of both humans and machines to build effective plans more quickly and with greater reliability."

3.4.1.2 Dialogue Flow Control. Control is the management of the direction of the dialogue flow. Cohen et al. [1998] classified five types of flow control in mixed initiative dialogues.

- Go forward: continue along the plan towards a goal. This happens when a participant issues a simple prompt (e.g. "yes", "no", "ok") or supplies responses. This does not include a shift in initiative.

- Change direction: discard or temporarily suspend the current plan and change to a new plan or topic (e.g. "But…", "What if…"). The participant who proposed the new direction has the initiative.

- Stop or pause: temporarily or permanently discontinue the current conversation and plan (e.g. "Let me think", "Wait a sec"). There is no change in possession of the initiative. If the pause becomes lengthy, the next speaker with a contribution to the conversation takes the initiative.

- Close or repeat: refine the conversational details before continuing, with a repetition or a summary. If the speaker has nothing new to say, then the listener may have the initiative.

- Interruption: the listener, the interrupter, takes the initiative.

In these types of flow control, the most important way to shift the initiative is interruption. Cohen et al. [1998] also claimed that the interruptions happened due to one of the following reasons.

- Listener believes assertion P is relevant and either believes that the speaker does not believe assertion P or believes that the speaker does not know assertion P.

- Listener believes that the speaker's assertion about P is relevant but ambiguous.

- Listener believes assertion P and either believes that assertion P presents an obstacle to the proposed plan or believes that assertion P has already been satisfied.

- Listener believes that an assertion about the proposed plan is ambiguous.

- Listener knows or tries to introduce the meaning of the speaker before the speaker finishes.

- Listener believes that there is another more important goal that must be satisfied before the current goal.

- Listener is no longer interested in the current plan.

3.4.1.3 Interactive Conversation. When a change in initiative occurs, the one who took the initiative also has the control of the conversation. An interactive conversation is one in which participants are controlling the dialogue by directing the conversation towards a particular goal.

A	B	A	B	A	B	A
↑		↑				↑

Figure 3.2 Not Interactive Conversation (only A has the Initiative)

A	B	A	B	A	B	A
↑	↑		↑			↑

Figure 3.3 Interactive Conversation (both A and B have the Initiative)

A or B in a box represents an interval of time that is controlled by one participant. ↑ represents the occurrence of an initiative. In Figure 3.2, the initiative always occurred when A controls the conversation. Therefore the Figure 3.2 is not an interactive conversation. Figure 3.3 illustrates an interactive conversation because both A and B take the initiative.

3.4.1.3 Process. A process is a finite set of turns in an interaction [Cohen et al., 1998]. A participant has the initiative in an interaction if the participant takes the first turn in a process. There is only one initiator per process, who is defined as the speaker. A sequence of turns is composed of utterances. Processes are not defined in terms of topics and goals alone, since goals and subgoals are related hierarchically and it would then be difficult to determine which level of goals defines a process. A process can have more than one turn. If one participant asks a question and another participant simply answers that question, then these turns will be grouped as one process. On the other hand, one turn can be categorized into more than one process.

Figure 3.4 illustrates two process cases. In both cases, P3 is initiated by interrupt P2. After P3, the process labeled P4 started. If P4 is a continuation of P2 then these procedures are called closed processes. Otherwise, the processes are called open processes, because not all processes are closed, because after interruption the participants often forget the previous process or start another process P4 which is relevent to P3 (e.g., argument).

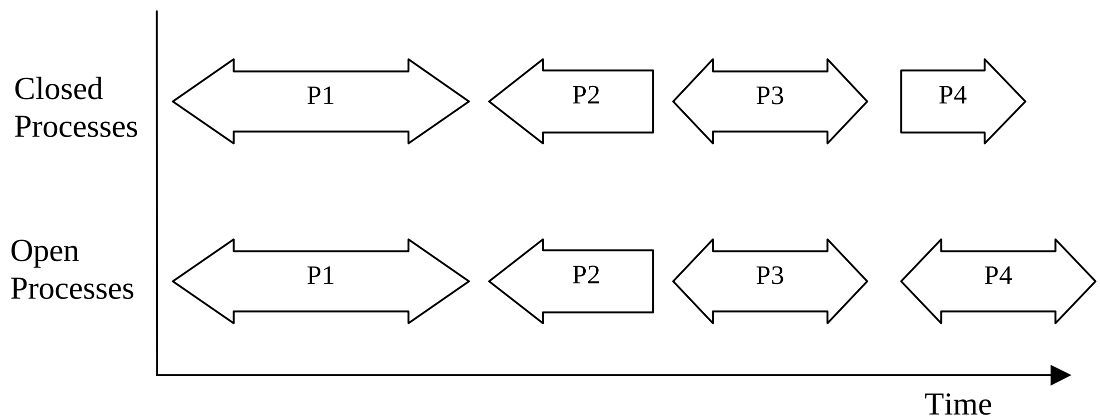

Figure 3.4 Closed Processes and Open Processes

3.4.2 Related Studies. Cohen et al. [1998] tried to separate the initiative from control of the turn. They introduced four theories and described how to analyze the initiative with each theory. According to theory #1, the flow of conversation represents the initiative. Any participant who interrupts the flow is allowed to have the initiative. The result of analysis by theory #4 is very similar to that of theory #1. However, in theory #4, if a participant does not propose new information, then the participant can not have the initiative.

Theories #2 and #3 are appropriate for focused task-oriented domains, which have more than two participants collaborating with each other. The basic concept of these domains is that the dialogue initiative should always pass immediately to the participant who has the best ability to handle the current task [Guinn, 1996]. The reasons for a shift in initiative in task-oriented domains are a little different from those in an ITS. First, the initiative taker can no longer proceed in attempting to solve the problem. Second, another participant detects invalidity and proposes a correction. Finally, another participant suggests an alternative plan, which must be considered with respect to the current proposal step. However, the initiative may or may not shift based on the merit of the new proposal. According to theory #2, if a participant proposes a solution that is accepted, then the participant has the initiative. According to theory #3, if a participant proposes a new goal or new subgoal, then the participant has the initiative.

Tutoring dialogues are different from task-oriented dialogues. The initiative in general tutoring dialogues is defined by the flow of conversation rather than by the party proposing a goal or a solution.

Cohen et al. [1998], though they could not find the exact threshold, introduced the concept of the degree of initiative, for example, a turn that results in a topic shift that gives information that was not requested, or in which the speaker asks for suggestions has strong initiative.

Lochbaum [1998] asserted that if an agent has the ability to understand a participant's intention, it has the ability to relate that utterance to the proceeding discourses. A discourse is composed of discourse segments. The agent must determine whether the utterance begins a new segment of the discourse, completes the current segment, or contributes to it. In order to do that, the agent must analyze the other participant's intention.

Figure 3.5 represents a sample dialogue structure of a task-oriented domain. An agent must recognize both the purpose of an embedded subdialogue and the relationship of that purpose to the purposes associated with the preceding discourse.

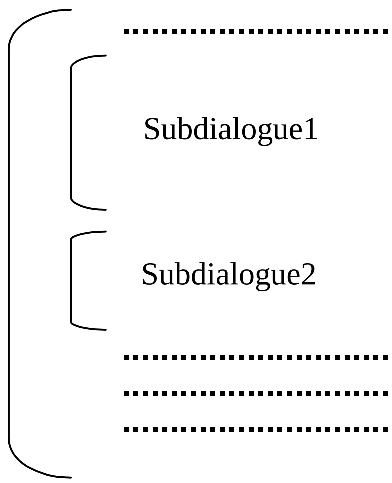

Figure 3.5 Subdialogues in a Longer Dialogue

Lochbaum introduced three example cases of recognizing the relationship of imbedded subdialogues. In the first case, a participant initiates a subtask subdialogue to support the subtask. The expert should decide whether the participant's belief in subdialogue1 is correct or not. In the second case, a participant initiates a correction subdialogue when he/she/it requires help to solve the current task. Finally, when a participant thinks he/she/it needs the preconditions to solve the task, a participant starts a subdialogue.

Smith [1997a] described three difficulties in modeling mixed-initiative interaction: maintaining coherence during an initiative change; choosing an appropriate linguistic form for a response as a function of initiative; and processing silence as a legitimate input and output in spoken natural language dialogues. An initiative change is characterized by switching the system to a new dialogue mode. How to ensure the coherence of the dialogues is a big issue at the time of changes in mode. When the other participant gains control, there is the risk that coherence may be lost. Smith tried but could not find algorithms for determining the relationship between the system's goal and the student's previous goal [Smith, 1997a].

In order to select the response strategy, the system needs three types of information: the current system goal, the current student focus, and the dialogue mode. A dialogue mode describes the level of initiative of the system [Smith, 1997b].

- Directive: the system has complete dialogue control. No interruptions to other sub-dialogues are allowed.
- Suggestive: the system still has dialogue control, but it is not as strong. Minor interruptions for closely related sub-dialogues are allowed.
- Declarative: the student has dialogue control and can interrupt for any desired sub-dialogues at any time. The system is free to mention relevant facts as a response to the student's statements.
- Passive: the student has complete dialogue control. The system will passively respond to the student's question.

A different user focus triggers an interruption. In this case the dialogue mode affects the system's response. In the directive mode, the system goal is selected without any regard for the student's focus. If there is a common relationship between the current system topic and the student's interrupt topic then select it, otherwise keep going on the system's goal in the suggestive mode. However, in the declarative mode, select a fact relevant to the student's focus. A common relationship is found when the student focus and the system's goal share a sufficiently close common ancestor in the knowledge hierarchy. In the passive mode, the system just acknowledges the student's last utterance.

The TRAINS project is an attempt to build a system that can interact and collaborate with humans in problem solving [Allen et al., 1995a]. The system understands and speaks natural language dialogue, which has a mixed initiative capability for transportation scheduling. The system imitates a manager who is responsible for planning emergency relief supplies to handle natural disasters. The project team collected about eight hours of human-human dialogue in the TRAINS domain. They analyze the dialogue to learn how this sublanguage is actually used and how humans collaborate to form plans [Allen et al., 1995b].

In order to provide a mixed initiative natural language interface, the TRAINS project needs to solve the following problems [Allen et al., 1995a].

- Incremental development of goals and solutions: instead of prespecified tasks that the system performs, the system can identify what goals the human has and how the human is planning to accomplish these goals.

- Using accumulated context to interpret the current situation: one of the most important properties, which make dialogue efficient for humans, is accumulated context. This allows complex scenarios and tasks to be described in an incremental fashion. This problem is concerned with context, with knowing what topic is being considered.

- Using effective acknowledgment and confirmation strategies: the system should provide some mechanism to store the understood utterances. Confirmation and acknowledgment in human dialogue frequently appear as explicit acknowledgments ("OK") or as appropriate responses.

- Supporting clarification, explanation and correction subdialogues: the system should have the ability to enter into subdialogues when the current conversations are not understood.

- Identifying intent: it should be an essential capability of a natural language interface system. Frequently human dialogues may have ambiguity with regard to the speaker's intent.

- Going on no matter what: the dialogue system should not give up. All problems should be resolved within the dialogue itself. If the situation is hopelessly confused then the user and system may restart the communication from some agreed topic or point.

3.4.3 Mixed Initiative in CIRCSIM-Tutor. Mixed Initiative Interaction is a big, complex issue in any ITS. Previous research in CIRCSIM-Tutor has involved the analysis of tutorial discourse in human tutoring sessions, but much more research is required into mixed initiative discourse.

One of the hot research issues in the mixed initiative dialogue is how to determine which participant has the initiative. It is especially important to categorize changes in initiative and analyze how they occur [Smith, 1997a; Cohen et al., 1998]. Sanders [1995, p. 65] defined eight classes of student initiatives as follows.

- Class 1: the student asks a question about the subject matter.

- Class 2: the student is having trouble "seeing" something or another. The student in not mainly requesting repair.

- Class 3: the student requests repair because the student did not understand the tutor.

- Class 4: do repair because the tutor did not understand the student.

- Class 5: hedging by the student.

- Class 6: explicit backward reference to some earlier topic, event, time.

- Class 7: initiatives specific to the keyboard-to-keyboard environment used in these sessions.
- Class 8: administrivia

The agreement of the analysis result between the raters is only 70 to 80 percent. Especially, the raters found it difficult to distinguish class 1 from classes 2, 3, and 5 in certain contexts. Also, more than a third of the initiatives were not classified [p. 66].

Sanders [1995, pp. 66-67] classified these types of tutor response to student initiatives.

- Explain or state some material in focus.
- Defer handling the initiative (perhaps modifying the student model or brushing off the initiative)
- Do repair, starting some material, where the student did not understand the tutor.
- Request repair (the tutor does not understand what the student means).
- Ask the student if stuck.
- Acknowledge whether or not the student's understanding is correct.
- Replan part or all of the remaining session.
- Give a hint or perhaps remind the student of material already covered in the session.
- Ask the student a question (Socratic tutoring).
- State, "You are confusing X with Y." (Declare a diagnosis)
- Invite the student to review his/her thinking with the tutor.

Based on transcript analysis, Sanders found certain aspects that appear to affect the content, style, and length of the tutor's responses. These are tactical uses of hints and the tutor's concern with the student's use of correct physiological language [p. 71]. Furthermore, the tutors have a preference for encouraging the students to solve the problems for themselves and a preference for hinting rather than simply answering questions [p. 81]. Because Sanders's classification failed to classify a third of the examples, Shah and Evens [1997] analyzed the keyboard-to-keyboard dialogues to describe both initiatives and responses in terms of the interactions between them.

Freedman [1997a] proposed potential plans for some mixed initiative interaction in CIRCSIM-Tutor. Any student statement that adds new content to the dialogue is considered as a student initiative. For example, the student adds new information such as an explanation, or the student changes the topic. In order to produce a tutor turn, CIRCSIM-Tutor parses the student's input utterance, derives an abstract representation of the input, plans a tutoring intervention, and finally generates a response [Freedman, 1997b].

Freedman pointed out some problems with unrestricted student initiatives. First, the student's utterance can be too difficult to understand or can be understood at a literal level. Second, even if CIRCSIM-Tutor understands the student's utterance it may not have a constructive response available. Finally, if a constructive response is available, responding to the student initiative may not help the tutor achieve its agenda [Freedman, 1997a, p. 46]. In order to reduce these unwanted student initiatives CIRCSIM-Tutor adopts two strategies for appropriate mixed initiative interaction: asking short-answer questions instead of open-ended questions and making each turn end with an explicit request. Our hope is that, with these strategies, the student will answer the question rather than change the topic. CIRCSIM-Tutor was designed as a tutor-led tutoring system that cannot handle a true cooperative conversation. These restrictions give fast response time and more content coverage [Freedman, 1997a].

Freedman also suggested ways to respond to simple student initiatives: respond and return to the plan, switch to a new plan instead of the current plan, put the student request elsewhere on the agenda, acknowledge the student input without responding, or ignore it [Freedman, 1997a, p. 48].

Shah [Shah and Evens, 1997] classified student initiatives and tutor responses in human tutoring sessions. Shah et al. [2000, pp.10-19] classified the student initiatives into four dimensions as follows.

1) Communicative Goal/Intention

This dimension represents what the student wants to do

- Request for Confirmation: the student generates an explanation and asks for confirmation of this theory.
- Request for Information: the student requests information about the topic in focus.
- Challenge: the student reflects some sort of disagreement with what the tutor has said.
- Refusal to Answer: the student does not respond at all. This usually triggers help from the tutor.
- Conversational Repair: the student requests clarification or repair of the previous discourse from the tutor.

2) Focus of Attention or Content

This dimension explains what the student initiative contains. “Initiatives are not fully understood until their focus has been determined” [p.13].

- Language Issues: the initiative is concerned with language issues such as correct terminology and its appropriate usage.
- Causal Reasoning: the initiative is concerned about causal relationships or equations.

- Problem-Solving Algorithm: the initiative is concerned with ways to attack problems.
- Rules of the Game: the initiative is concerned about how to use CIRCSIM-Tutor.

3) Surface Form

This dimension explains what type of sentence or fragment encodes the initiative.

- Declarative
- Interrogative
- Silence/Pause
- Fragment
- Imperative

4) Degree of Certainty

This dimension describes whether the student initiative is hedged or not.

Shah et al. [2000, pp.20-29] also classified the tutor's response into three dimensions as follows.

1) Communicative Goal

This dimension describes the tutor's communicative intention.

- Explanation
- Acknowledgment
- Conversational Repair
- Instruction in the "Rules of the Game"
- Teaching the Problem Solving Algorithm
- Probing the Student's Inference Process: the tutor asks with intent to teach through the student's self-explanation.
- Extending Help in Response to Pause
- Brushing Off: the tutor intends to stop the discussion and bring up higher priority issues.
- Teaching the Sublanguage: the tutor is concerned about teaching the correct usage of the language of physiology.

2) Delivery Modes

This dimension records the tutor's response style.

- Explanation

- Hinting
- Directed Line of Reasoning
- Tutor Monologue: the tutor gives a long and detailed explanation when the tutor recognizes that the student is confused about what is going on.
- Rephrasing
- Analogy: the tutor sometimes uses an analogy in order to catch the student's attention and interest or help the student's understanding.

3) Surface Form

The dimension explains the grammatical encoding of communicative goals in tutor's response.

- Declarative
- Interrogative
- Fragment
- Imperative

Figure 3.6 illustrates the mixed initiative interaction process. The Input Understander analyzes the student's sentence "I don't understand about SV" and recognizes that the student is asking for information about SV. In order to do that, the input understander needs to know what the expected answer is. This information is passed to the tutorial planner using the logic form (Question (Explain SV)).

Using the logic form, which contains the student's intention, the discourse planner decides on the response strategy. The possible strategies are respond to the initiative now, ignore it, or defer it. According to the strategy, the discourse planner needs to replan the goal stack. The student request (Explain SV) is placed the stack, on top of T-tutors-via-Determinants.

Since the appropriate response style here is explain by definition, then (Define SV) will be passed to the sentence generation module. Then the sentence generation module generates “Stroke Volume means the volume of blood pumped out…” to the student.

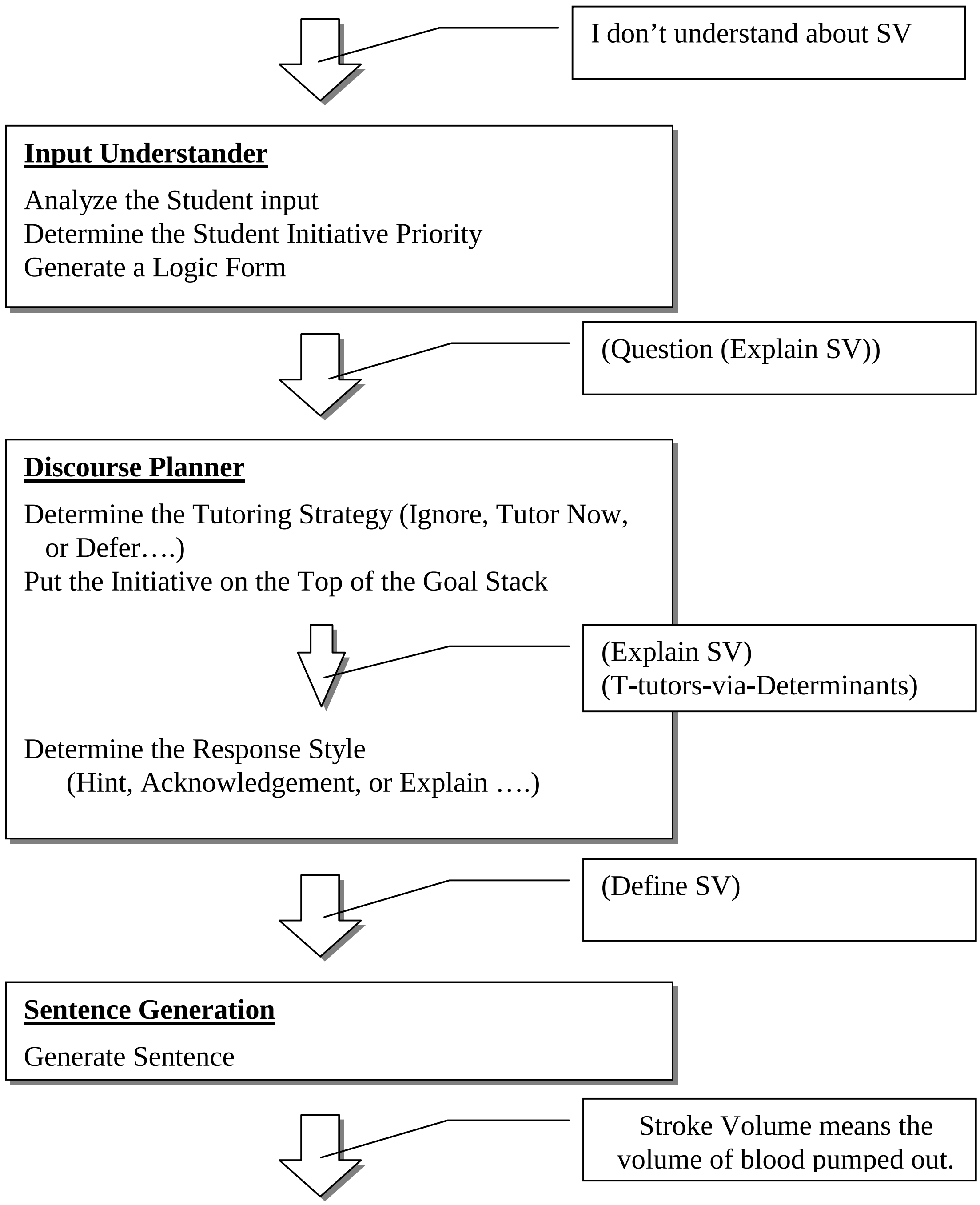

Figure 3.6 Mixed Initiative Service Step

CHAPTER IV

ASSESSMENT METHODS

4.1 Shim's Assessment for Student Modelling

Shim [1991] explored the use of Confidence Factors (CFs) for CIRCSIM-Tutor version 2. The purpose of CFs is to estimate the likelihood that a student understands a concept. The response history is a chronologically ordered (the rightmost response is the most recent) list of "C"s and "W"s representing correct and wrong. To evaluate the CFs, Shim suggested a Time-Dependent Function that produces real numbers from 0 to 1 as an assessment value.

$$CF(R_1,\ldots,R_n) = \frac{R_{n-k+1}W_{n-k+1} + \ldots + R_{n-1}W_{n-1} + R_nW_n}{W_{n-k+1} + \ldots + W_{n-1} + W_n}$$

where, $CF(R_1,\ldots,R_n)$: Confidence Factor after n responses

W_i: weight for ith response

$= 2(k + i - n) - 1 \qquad (n-k+1 \le i \le n)$

R_n: nth response; 1.0 for correct "C", 0.0 for wrong "W"

If $n-k+1 < 1$, then $R_{n-k+1} = 0.5$ for unknown value

The first assessment model in CIRCSIM-Tutor, Shim's model has some weak points. First, if an answer contains more than one variable, Shim's model cannot evaluate partially correct answers. In the recent version, many questions ask for more than one item in an answer (e.g., Determinants of CO are SV, HR). In order to solve this problem, Shim treated SV determines CO and HR determines CO as two separate pieces of information. Second, domain experts use a coarse-grained assessment. Shim used the three most recent responses in his actual computations [Brandle, 1998].

4.2 Hume Modified Shim's Assessment

Hume [1995] modified Shim's CF model. The response history column contains no more than the three most recent values. Hume's rules for evaluating the CF are very simple. Every time "C" is added, the CF goes up (except when there were already two "C"s) and every time a "W" is added, the CF goes down (except when there were already two "W"s) for every error pattern. The error patterns represent physiological concepts. The tutor can determine if student predictions violate any of these concepts.

Table 4.1 The Prediction Table

Response History(⇒ recent)	Confidence Factor
No history	No value
(C)	0
(W)	-1
(C C)	1
(W C)	0
(C W)	-1
(W W)	-2
(C C C)	2
(W C C)	1
(C W C)	0
(W W C)	0
(C C W)	0
(W C W)	-1
(C W W)	-2
(W W W)	-2

Table 4.1 shows Hume's CF values on each response history. In the response history, the rightmost value is the most recent response. And "no value (null)" means this error pattern has not been taught yet. Hume suggested that we could choose hints better with a local assessment and a global assessment. The local assessment estimates the current knowledge status of the student on the basis of the most recent responses whether

"C" or "W". The global assessment is based on responses, predictions and a variety of tutoring history information.

One of the drawbacks of Hume's model is complexity. Hume suggested too much history information for each error pattern. And, his global assessment is calculated by a complex equation. Second, the local assessment model has a skew distribution and too many response history types have the same value. Hume uses the three most recent responses in the evaluation of local assessments without weighting. Finally, Hume's model cannot evaluate partially correct answers either.

4.3 New Assessment Methods

4.3.1 Confidence Factor. The Confidence Factor (CF) is an estimation of the likelihood that a student understands a concept. Each error type corresponds to an important topic that we want students to learn. So each error type has its own Confidence Factor.

Table 4.2 Hume's CF Distribution

CF	-2	-1	0	1	2
Response History	(WW) (CWW) (WWW)	(W) (CW) (WCW)	(C) (WC) (CWC) (WWC) (CCW)	(CC) (WCC)	(CCC)

Table 4.2 shows Hume's CF distribution. This table illustrates that Hume's CF evaluation method has a skewed distribution and too many response history types have the same value. Hume uses the three most recent responses in the evaluation of local

assessments without weighting. But we believe that the most recent response is more important than previous ones as Shim asserted [1991]. A response can be a partially correct value. For example:

N14-tu-38-2: What determines SV?

N14-st-39-1: SV is determined by IS and CVP.

The previous version of the CF evaluation method cannot calculate a value for a partially correct answer. It is very hard to evaluate the CF if the answer to N14-tu-38-2 is "TPR, IS" (N14-st-39-1 is the correct answer). Furthermore a student may give answers like "CO, MAP and IS".

My new CF evaluation method can evaluate partially correct answers.

$$\text{Response } (R_n) = \frac{O^2}{N * D} * 2 - 1$$

O: Correct Items in an Answer
N: Number of Items in an Answer
D: Number of Items in desired Answer

If the n^{th} answer is perfectly correct "C" then R_n is 1 and if it is perfectly wrong "W" then R_n is -1. The response is normalized between -1 to 1 by "* 2 -1".

$$CF = \frac{R_{n-2}W_{n-2} + R_{n-1}W_{n-1} + R_nW_n}{W_{n-2} + W_{n-1} + W_n} = (R_{n-2} + 2*R_{n-1} + 3*R_n) / 6$$

This CF evaluation equation is similar to Shim's [1991] equation except the **k** is fixed on 3 (three most recent responses). I assume the weights are 1, 2, and 3. This means the most recent response R_n is three times and R_{n-1} is twice as important as R_{n-2}. Table 4.3 shows the CF distribution that is calculated by the new equation. The CF distribution is better balanced than Hume's distribution. The comparison graph in Figure 4.1 makes this clear.

4.3.2 Local Assessment. The current status of the domain knowledge can be evaluated by several assessment methods. The Local Assessment (LA) is the student's performance score on the most recent (or current) procedure.

Table 4.3 Cho's CF Distribution

CF	-1	-0.83	-0.66	-0.5	-0.33	-0.17	0
Rn History	(WWW)	(WW)	(CWW)	(W)	(WCW)	(CW)	(WWC) (CCW)

CF	0.17	0.33	0.5	0.66	0.83	1	
Rn History	(WC)	(CWC)	(C)	(WCC)	(CC)	(CCC)	

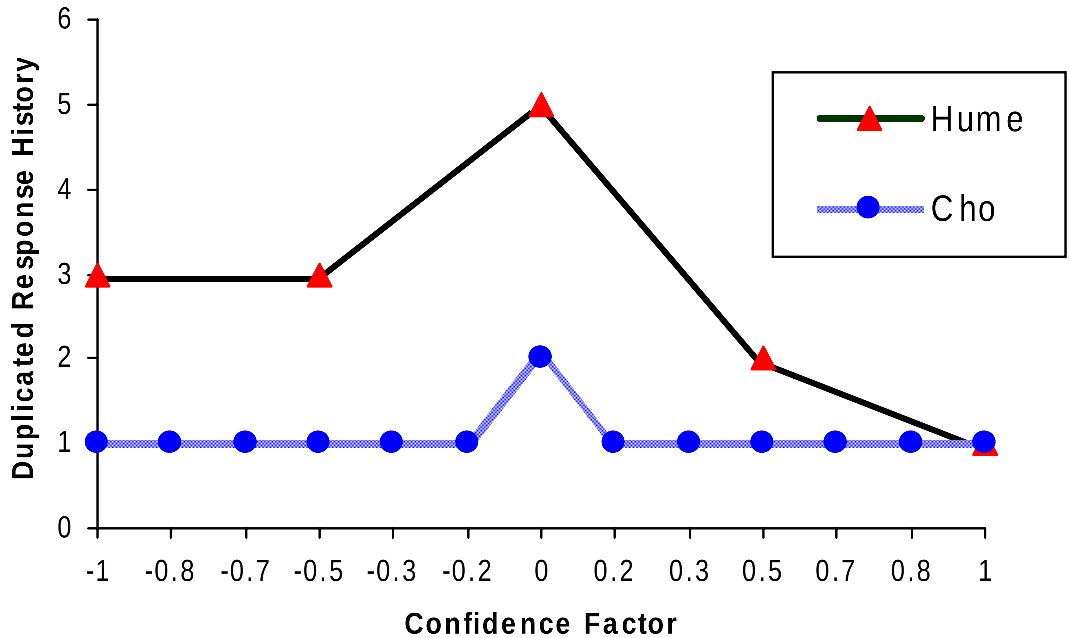

Figure 4.1 CF Distribution Comparison Chart

As a general rule of thumb, fine-grained assessments are useful for short-term pedagogical decisions, such as deciding how to phrase a hint or choosing the next discourse move, while coarse-grained assessments are useful for long-term pedagogical decisions, such as the next procedure for this student [Martin and VanLehn, 1995].

Hume claimed that the student modeller should keep a lot of detailed history data in each error pattern for the tutorial planner. Fine-grained assessments are difficult and expensive to make. It is hard to determine the right error pattern and the difficulty that is the cause of the error pattern. And fine-grained assessments may yield a wrong estimate of the student's knowledge status because of the uncertainty of the student modeller. On the other hand, fine-grained assessments may be useful for planning individual discourse moves like hinting. In order to determine the next procedure, CIRCSIM-Tutor does not need such a fine-grained assessment but needs coarse-grained assessments with error types. Hume [1995, pp. 43 - 45] defined error patterns for Tutorial and Discourse Planning. For example, Hume defined three error patterns for Neural Variables in each phase (DR, RR, and SS). The Curriculum Planner needs error types that are generalized error patterns. In this case the error type is "Neural Variable," which handles all responses involving Neural Variables. The coarse-grained assessment does not mean a rough evaluation but means evaluation with coarse-grained (generalized) items and perfect calculation, if possible.

One of the important factors in local assessment is a Prediction Table Score (PTS), which is related to the number of correct predictions in a procedure.

$$PTS = \sum_{i=1}^{s} \sum_{j=1}^{v} (PT_{ij}) / 21$$

$$LA_n = PTS + \sum_{i=1}^{e} (CF_i) / 2e$$

where s = 3, since there are three stages (DR, RR, and SS), v = 7 since there are seven variables in the prediction table, and e is the number of error types. A prediction result PT_{ij} is 1 if the prediction is correct, and -1 if the prediction is wrong. The sum of CFs for each error type and PTS are combined to produce the current local assessment value. The score of PTS is twice as important as the score of an error type. If the student predicts a value we assume the student knows the causal relationship and the value of the variable. Normalizing these results gives a number in the range from -1 to 1 for the current local assessment.

4.3.3 Global Assessment. The Global Assessment (GA) is the student's cumulative score on past procedures. The global assessment ensures that we are representing a student's current knowledge more perfectly than with the local assessment. For example, suppose a student earned bad scores in the past three procedures and suddenly makes an excellent score in the current procedure, should we believe that the last procedure score is the status of the student?

The current local assessment and the cumulated past local assessments (past global assessment) make the current global assessment. For the first calculation (n = 1), the global assessment is the current local assessment ($GA_1 = LA_1$).

$$GA_n = LA_n / 2 + \sum_{p=1}^{n-1} (LA_p) \;/\; 2(n-1)$$

Here, n is the number of solved procedures. This equation implies that the current local assessment makes up one-half of the current global assessment (GA_n) and the past global assessment makes up the other half.

The global assessment measures whether the student has the capability to solve a procedure, which has the expected procedure difficulty level. With the global assessment, the Curriculum Planner can determine whether the recent procedure level was suitable to the student's knowledge status and adjust the next procedure level.

CHAPTER V

ANALYSIS OF TRANSCRIPTS

5.1 The Test Data

I tested the student modeling strategies from the previous chapter on CIRCSIM-Tutor Version 2.6 tutoring session's transcripts with first year medical students from April 29, 1998. The language in these transcripts was analyzed immediately so that we could make improvements but no high level analysis was done, until I decided to try out some student modeling ideas on this data. I analyzed the transcripts from ten of the fourteen students who participated, modeling all who completed at least two procedures. The students selected procedures from the procedure selection menu using Procedure Names (see Figure 2). The questions consist of three types for each variable: Prediction Table (PT), Causal Relation, and Value of the Variable. Answers should be evaluated for each variable separately. We also considered questions about "the Primary Variable", "the Neural Variable", and "does the reflex compensate for the change in MAP in DR?" When a Causal Error Type is triggered, the system responds with a maximum of three questions that are T (deTerminant), M (doMinant), and R (diRection). The sequence of the questions is the Prediction Table followed by Causal questions followed by a Value question.

5.2 Evaluation Rules

I applied my new assessment methods in the form of the evaluation equations in Figure 5.1. The evaluation results are illustrated in Table 5.1.

$$\text{Response } (R_n) = \frac{O^2}{N * D} * 2 - 1$$

O: Correct Items in an Answer
N: Number of Items in an Answer
D: Number of Items in desired Answer

$$CF = (R_{n-2} + 2*R_{n-1} + 3*R_n) / 6$$

$$PTS = \sum_{i=1}^{s} \sum_{j=1}^{v} (PT_{ij}) / 21$$

s: 3 (three stages)
v: 7 (variables in Prediction Table)

$$LA_n = PTS + \sum_{i=1}^{e} (CF_i) / 2e$$

e: Number of Error Types

$$GA_n = LA_n / 2 + \sum_{p=1}^{n-1} (LA_p) / 2(n-1)$$

n: Number of Solved Procedures.

Figure 5.1 The Evaluation Equations in the Analysis

A correct answer in the Prediction Table has a score of two points. I assume that if the student predicts the correct value of a variable in the Prediction Table, then the student knows the causal relation and the value of the variable. Inevitable correct answers are not counted as correct. For example, the value of a variable should be one of three values "+", "0", or "– ". In the transcript in procedure 2 by "bycx", the student predicted the value of CVP in the RR stage as "+.""Tu" is the discourse of the tutor, and "St" is the discourse of the student.

Tu: What is the correct value of Central Venous Pressure?

St: 0

Tu: Nope, the value of Central Venous Pressure is changed.

Tu: Consider the value of Cardiac Output.

Tu: What is the correct value of Central Venous Pressure?

St: –

Tu: Correct, the value of Central Venous Pressure is decreased.

In this case, the student already answered “+” and “0” for CVP value. Finally, the student should choose the “– “ value. When the student changes a prediction value, only the final values are considered except for the Primary Variable. If the Primary Variable was chosen but the value was wrong in the Prediction Table we marked “W” for the Prediction Table score and marked “C” in the causal score.

Table 5.1 Evaluation Results Table

Student Name	Procedure Number	PT Score	Local Assessment	Global Assessment
bycx	2	0.142857	0.12374	0.12374
	3	0.333333	0.43186	0.2778
ev	2	0.809524	0.83894	0.83894
	3	0.904762	0.94398	0.89146
fugt	2	0.333333	0.41176	0.41176
	3	0.714286	0.81232	0.61204
	4	0.904762	0.91947	0.76576
irufgt	2	0.809524	0.86345	0.86345
	3	0.904762	0.94888	0.90616
	4	0.714286	0.81723	0.86169
jqxcwd	2	0.809524	0.87325	0.87325
	3	0.904762	0.91506	0.89415
	4	0.809524	0.86835	0.88125
kp	3	1	1	1
	4	0.904762	0.91947	0.95973
olaz	2	0.714286	0.79272	0.79272
	3	0.904762	0.94888	0.8708
	4	0.714286	0.7584	0.8146
pknm	2	0.857143	0.87185	0.87185
	1	0.904762	0.91947	0.89566
	3	1	1	0.94783
	4	0.714286	0.78782	0.85913
rishqj	2	0.619048	0.62395	0.62395
	3	0.904762	0.94888	0.78641
	4	0.619048	0.72689	0.75665
vehs	2	0.428571	0.53151	0.53151
	3	0.619048	0.66317	0.59734
	4	0.238095	0.38025	0.4888

5.3 Analysis Results

The analysis showed us three important results. First, a student who gets the Primary Variable prediction wrong the first time will probably make bad Prediction Table prediction results in the DR phase. Second, the procedure name may give a strong hint about the primary variable. Finally, the variables that are predicted wrong most frequently are CVP and SV in the RR phase.

Table 5.2 illustrates global assessments calculated from transcripts, which show the results from a total of 27 procedures performed by ten students. The sign “ – ” in the table means the student did not solve the procedure.

Table 5.2 The Global Assessment Result

Student Code	Procedure 2	Procedure 3	Procedure 4
bycx	0.12374	0.2778	-
ev	0.83894	0.89146	-
fugt	0.41176	0.61204	0.76576
irufgt	0.86345	0.90616	0.86169
jqxcwd	0.87325	0.89415	0.88125
kp	-	1	0.95973
olaz	0.79272	0.8708	0.8146
pknm	0.87185	0.94783	0.85913
rishqj	0.62395	0.78641	0.75665
vehs	0.53151	0.59734	0.4888

The global assessment is the student's cumulative score on past procedures in the real number range from –1 (worst) to 1 (best), so we used it to represent the student's current knowledge status. For long-term pedagogical decisions, the curriculum planner uses coarse-grained assessment methods [Martin and VanLehn, 1995]. With the global assessment, the Curriculum Planner can determine whether the recent Procedure Difficulty Level was suitable for the student's knowledge status and adjust the Difficulty Level for the next Procedure.

The results in Table 5.2 were a complete surprise to us. All the students improved from Procedure 2 to Procedure 3 as we expected. But then almost all of them performed worse on Procedure 4 than on Procedure 3 (see Figure 5.2). We decided that further analysis was necessary to explain what went wrong.

The analysis showed that sometimes the students can infer the primary variable from the procedure name, and that this fact really affects the results. For example, Procedure 3's procedure name (Decrease IS to 50% of normal) gives a strong hint that IS is the Primary Variable and its value has gone down. Table 6 shows the number of wrong predictions at each procedure. Almost all students (except "fugt") gave the correct Primary Variable right away. The student must get the Primary Variable correct and then CIRCSIM-Tutor allows the student to predict the status of the rest of the variables. Referring to Table 5.2, a student who gets the Primary Variable prediction wrong the first time will probably make a bad score throughout the procedure.

As a result of this analysis we have decided to present procedures to the student in Procedure Description format (shown in Figure 5.4) rather than Procedure Name format (shown in Figure 5.3).

Table 5.3 Wrong Prediction of Primary Variable

Student Code	Procedure 2	Procedure 3	Procedure 4
bycx	4	0	-
ev	0	0	-
fugt	1	1	0
irufgt	1	0	1
jqxcwd	0	0	0
kp	-	0	0
olaz	0	0	1
pknm	1	0	1
rishqj	2	0	0
vehs	1	0	1

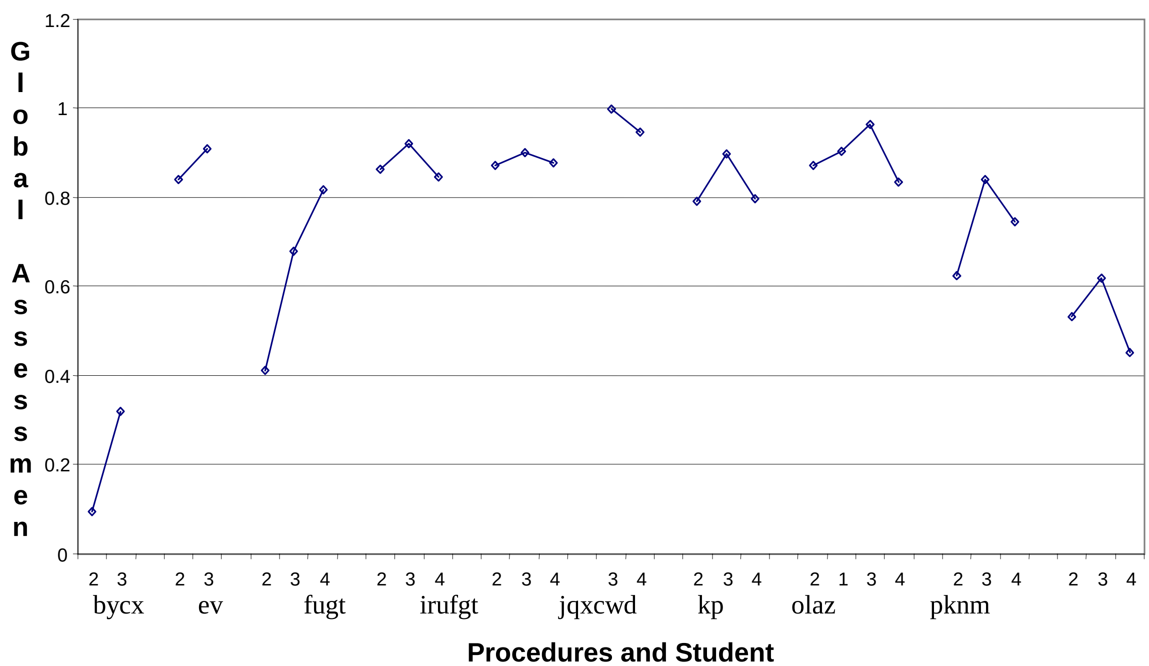

Figure 5.2 Global Assessment

1. Hemorrhage-Remove 1.0 Liter
2. Reduce Arterial Resistance (RA) to 50% of normal
3. Decrease Inotropic State (IS) to 50% of normal
4. Increase Venous Resistance (RV) to 200% of normal
5. Quit CIRCSIM - Tutor

Figure 5.3 Procedure Selection Menu with Procedure Names in CST V 2.6

1. A medical student donated 1 liter of blood to a patient about to undergo surgery. Predict the effects of the blood donation on the student.
2. Predict the effects of simultaneously increasing both heart rate and cardiac contractility (cardiac inotropic state) using the maintained infusion of a drug.
3. What would be the effects of continually infusing an individual with a potent, long-acting cholinergic muscarinic antagonist (blocking agent)?
4. A group of teenagers were experimenting with drugs. One of them swallowed some pills that contained a specific arteriolar smooth muscle relaxant.
5. A parent was preparing for her 5-year-olds birthday by blowing up balloons. One very large balloon was particularly stiff. What would be the cardiovascular effect of her effort to inflate this balloon? Assume that she tried to blow it up in very long, sustained expiratory effort
6. Quit CIRCSIM - Tutor

Figure 5.4 Procedure Selection Menu with Procedure Descriptions in CST V 3.0

CHAPTER VI

THE CURRICULUM PLANNING MODEL

This chapter illustrates a new Curriculum Planning model in CIRCSIM-Tutor version 3.0. Curriculum Planning is based on the student's knowledge of the cardiovascular system. So the Curriculum Planner needs to estimate the student's knowledge from various responses.

6.1 Khuwaja Started to Design a Curriculum Planner

Khuwaja [1994] recognized the need for Curriculum Planning and outlined a possible planner. He even provided some sample rules but they are neither complete nor consistent. He introduced procedure difficulty levels, which were defined by the domain experts. The domain experts created five Procedure Difficulty Levels and four Procedure Description Levels for the seven cardiovascular variables. A student can choose a procedure from a total of 14 procedures. His planner does not use the Procedure Description Level for classifying the procedures.

Khuwaja's Curriculum Planning model has some drawbacks. First, he tried to use student input for selecting a new procedure in restricted cases. For example, the students can select a Procedure/Description difficulty level only when they finished the current procedure well. But there are no proper rules on which to base a choice. So the student may select a very difficult procedure (challenging) after an easy (simple) procedure and vice versa. Second, Khuwaja's model has inconsistent procedure selection rules. There is a rule: "If the student's performance is good, and this is the third procedure, then introduce category 5". However there are no rules to support many other cases. For example, there is no rule that covers the situation when the student's performance is good on the 4th procedure, or when the 3rd procedure performance was not good.

Finally, in Khuwaja's system after all the procedures in a minimal set are solved, the Curriculum Planner organizes a new minimal set considering the status of the student. In this case, there are five procedures in the minimal set that the student has to solve before the procedure difficulty level is adjusted. So the student may not be able to move on to procedures with an appropriate difficulty level quickly enough.

6.2 Procedures

CIRCSIM-Tutor helps students reason about the qualitative causal effects on the human circular system when normal blood pressure is perturbed. CIRCSIM-Tutor version 3.0 has fourteen different types of perturbations of the cardiovascular system (see Table 6.1).

The Primary Variable is the first variable in the Prediction Table that is affected by the current perturbation. The Procedure Variable is the first variable in the concept map affected by the current perturbation. The set of Procedure Variables is a superset of the set of Primary Variables. First, all perturbations are classified into five categories by four primary variables (CVP, IS, HR, and TPR) and a procedure variable BRP, as shown in Table 6.2.

Another classification of the perturbations is based on their level of difficulty (PD). This classification divides perturbations into five levels. Basic procedures may be simple (s), moderate (m), difficult (d), or challenging (c); combinations are even more challenging.

Table 6.1 Procedures

Abb.	Procedure name	PD	DD
IRV	Increase Venous Resistance(RV) to 200% of Normal	s	1,2,3
DBV	Hemorrhage-Remove 1.0 L (Blood Volume = 4.0 L)	s	3
PIT	Increase Intrathoracic Pressure(PIT) from -2 to 0 mm Hg	c	3,4
DIS	Decrease Inotropic State(IS) to 50% of Normal	m	1,2,4
BAA	Administer a Beta-adrenergic Agonist	d	1,2,3
BAB	Administer a Beta-adrenergic Antagonist (blocker)	d	1,2,3
IHR	Install artificial pacemaker. Increase Heart Rate(HR) from 72 to 120	s	1
DHR	Install artificial pacemaker. Decrease Heart Rate(HR) from 72 to 50	s	1
CHA	Administer a Cholinergic Agonist	m	1,2,3
CHB	Administer a Cholinergic Antagonist (muscarinic)	m	1,2,3
AAA	Administer an Alpha-adrenergic Agonist	m	1,2,3
AAB	Administer an Alpha-adrenergic Antagonist (blocker)	m	1,2,3
DRA	Reduce Arterial Resistance(RA) to 50% of Normal	s	1,2,3,4
DBR	Denervate the Baroreceptors	d	1,4

The final classification is based on procedure descriptions. Each procedure description describes the initial effect of the perturbation on the cardiovascular system. A procedure description can explicitly or implicitly describe the effect of this action on the primary variables or the procedure variables. This classification divides the 83 procedures into four Procedure Description levels (DD). Level 1 has a Direct Definition of the Primary Variable in a procedure description. Level 2 has an Indirect Definition of the Primary Variable in a procedure description. Level 3 has a Direct Definition of the

Procedure Variable in a procedure description. Level 4 has an Indirect Definition of the Procedure Variable in a procedure description. The actual descriptions were written by Allen Rovick.

Table 6.2 The Procedure List

Category	Procedures
Central Venous Pressure (CVP)	Basic: IRVs1, IRVs2, DBVs3, IRVs3, PITc3, PITc4 Combination: IRV, DBV, PIT after (BAB, CHB, AAB, IHR, DHR)
Inotropic State (IS)	Basic: DISm1, DISm2, DISm4, BAAd1, BABd1, BAAd2, BABd2, BAAd3, BABd3 Combination: PIS after (CHB, AAB, IHR, DHR, DBR) BAA after (CHB, AAB, IHR, DHR) BAB after (DHR, DBR)
Heart Rate (HR)	Basic: IHRs1, DHRs1, BAAm1, BABm1, CHAm1, CHBm1, BAAm2, BABm2, CHAm2, CHBm2, BAAm3, BABm3, CHAm3, CHBm3 Combination: IHR, DHR after (BAB, CHB, AAB, DBR) CHA after (BAB, AAB)
Total Peripheral Resistance (TPR)	Basic: DRAs1, DRAs2, DRAs3, DRAs4, AAAm1, AABm1, AAAm2, AABm2, AAAm3, AABm3 Combination: DRA after (BAB, CHB, IHR, DHR, DBR) AAA after (BAB, CHB, IHR, DHR) AAB after DHR
Baro-Receptor Pressure (BRP)	Basic: DBRd1, DBRd4

Table 6.2 illustrates the Procedure List, which has procedures divided into these categories. For example, the procedure “IRVs1” in category CVP (see Table 6.1 and 6.2) means that the perturbation type is “Increase Venous Resistance (IRV) to 200% of normal”, the procedure difficulty level is “simple”, and the Procedure Description level is

"1." Figure 6.1 illustrates three Procedure Descriptions of the IRV perturbation. The number in the parentheses indicates the Procedure Description level.

In the Procedure List, procedures in a category are ordered by Procedure Difficulty Level, Procedure Description Level, and Procedure name. The ordering is based on the importance of classification. For example, Procedure Difficulty Levels are more important than Procedure Description Levels. With this ordering, the Curriculum Planner can find the next most difficult procedure easily.

(1) Predict the effects of increasing venous resistance. Assume that no change in venous capacitance or venous compliance occurs

(2) A patient was admitted to the hospital after experiencing a fainting spell. After a series of tests her problem was determined to be an abdominal tumor that was compressing her vena cava, reducing her venous return

(3) Certain agents are known to cause veno-constriction, without affecting venous compliance or capacitance. What would be the effect of administering this agent to a patient?

Figure 6.1 Procedure Descriptions of the Same Procedure (IRV)

6.3 Student Input

Individualized instruction is the main goal of an ITS. To achieve this, an ITS maintains a student model, which models the student's understanding of domain concepts. The Curriculum Planner module can use this model to choose the next problem for the student. Student modeling is fraught with uncertainty because of ambiguity [Katz et al., 1992]. It is hard to interpret student responses, distinguish misunderstandings from careless errors, and decide what is correct. So the Curriculum Planner uses both the student model and the student input to determine the next procedure set.

A merit of the Curriculum Planner in CIRCSIM-Tutor is a proper use of the student input. CIRCSIM-Tutor asks the student, “Do you want the next procedures to be easier, harder, or about the same?” Sometimes students may know more than the system does about their ability. For example, if the student has studied the cardiovascular system hard for the last few days, the student's knowledge may have increased significantly. On the other hand, perhaps the student used CIRCSIM-Tutor a couple of months ago and has forgotten much since.

In CIRCSIM-Tutor Version 3.0, a novice student must solve the “Reduce Arterial Resistance” procedure first, because this situation is intuitive for the students. And every student must solve some important (core) procedures before they do other procedures. But skilled students may not want to solve these procedures again. What is more, the student input improves motivation and enthusiasm.

6.4 The Procedure Selection Scheme

6.4.1 Defining a Procedure Set. The most important part of Curriculum Planning is defining a procedure set. The Curriculum Planner in CIRCSIM-Tutor recommends some procedures in a procedure set for the student's selection. A procedure set is a set of procedures that are displayed for student selection at a given point. Whenever the student finishes a procedure the Curriculum Planner constructs a new procedure set on the basis of the student input and the student's current assessment. Then the Screen Manager displays the procedure set on the screen so that the user can make a choice. An appropriate procedure is just a little beyond the student’s current capability.

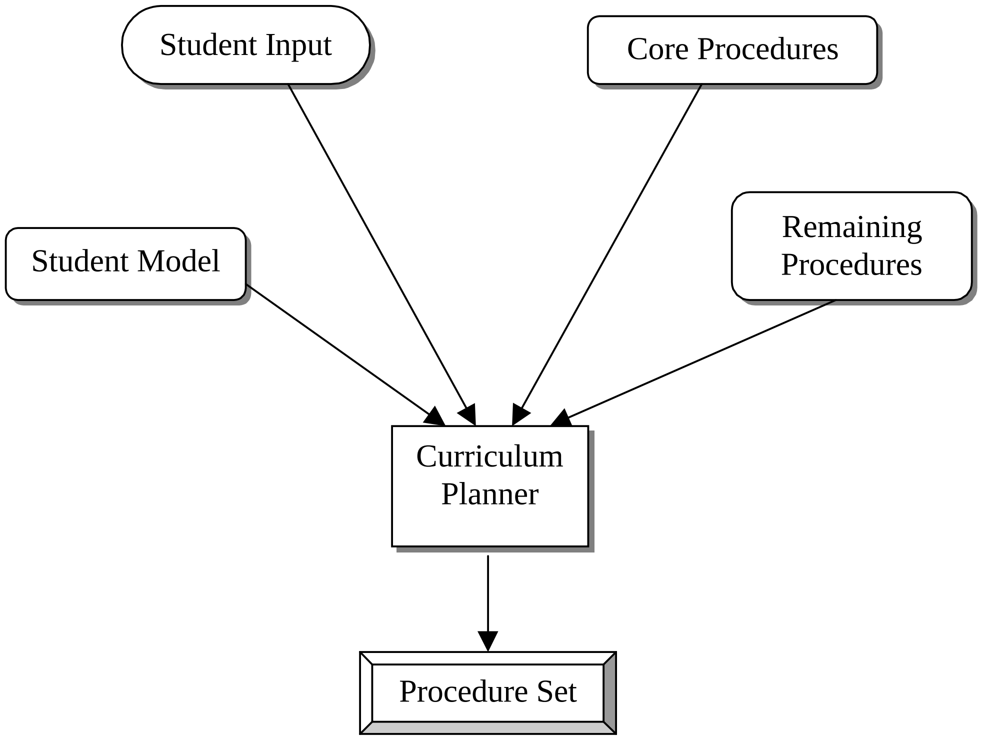

Figure 6.2 Structure of the Curriculum Planner

Figure 6.2 illustrates the structure of the Curriculum Planner. The Curriculum Planner chooses procedures from the Core Procedures or Remaining Procedures. The Core Procedures are so important that every student must solve some of them. The Remaining Procedures contains unsolved procedures. The Curriculum Planner decides the procedure difficulty level and the procedure description level of the next procedure set using information from the student model. The levels of difficulty of the descriptions in the next procedure set levels can be adjusted by student input. And then the curriculum planner chooses procedures for the appropriate procedure set from the core procedures or the remaining procedures. Our domain experts want to teach the core procedures first, then other procedures (see Table 6.3).

Table 6.3 The Core Procedures

(1) Reduce Arterial Resistance (RA) to 50% of Normal
(2) Decrease Inotropic State (IS) to 50% of Normal
(3) Increase Venous Resistance (RV) to 200% of Normal
(4) Denervate the Baroreceptors
(5) Reduce Arterial Resistance (RA) to 50% of Normal after Denervating the Baroreceptors
(6) Hemorrhage-Remove 0.5 L
(7) Hemorrhage-Remove 1.0 L

The core procedures can be categorized into three subgroups. The first core procedure group contains (1) the novice procedure. The second core procedure group consists of (2) (3) and (4). The third core procedure group consists of (5) (6) and (7). The strategies of selecting a core procedure set are as follows.

- The first procedure should be the novice procedure: procedure (1) - Reduce Arterial Resistance (RA) to 50% of Normal.
- The second procedure set should be selected from the second core procedure group.
- If the global assessment is high,

    then the third procedure set should be selected from the third core procedure group

    else the third procedure set should be selected from the second core procedure group.
- If the student selects the procedure (6) then the system should suggest procedure (7) as the next step.

- After the third core procedure is successfully completed the core procedure restriction disappears.

Figure 6.3 depicts the flow of Curriculum Planning. The basic assumptions and the procedure selection rules in the next section represent this Curriculum Planning scheme. The Curriculum Planner should organize the procedure set to fit the student at planning time. The main strategy for organizing a procedure set is based on the following [Cho et al., 1999].

- If the status of the global assessment and the student input are opposite in direction
  then the Procedure Difficulty Level does not change
  else the direction of the global assessment determines the Procedure Difficulty
  Level
- The direction of the student input moves the Procedure Description Level.

The global assessment value is categorized into three status levels (see Table 6.4). High status means that the global assessment is good and the value exceeds the upper threshold, so the student is ready to solve problems at a higher Procedure Difficulty Level. Medium status means that the global assessment is moderate, but not enough to change the procedure difficulty level. This means that the value is between the lower and the upper threshold. And low status means that the global assessment is poor, the value is under the lower threshold, so the student's next Procedure Difficulty Level should be lower. To determine an effective adjustment, the Curriculum Planner needs appropriate thresholds for deciding on the next Procedure Difficulty Level.

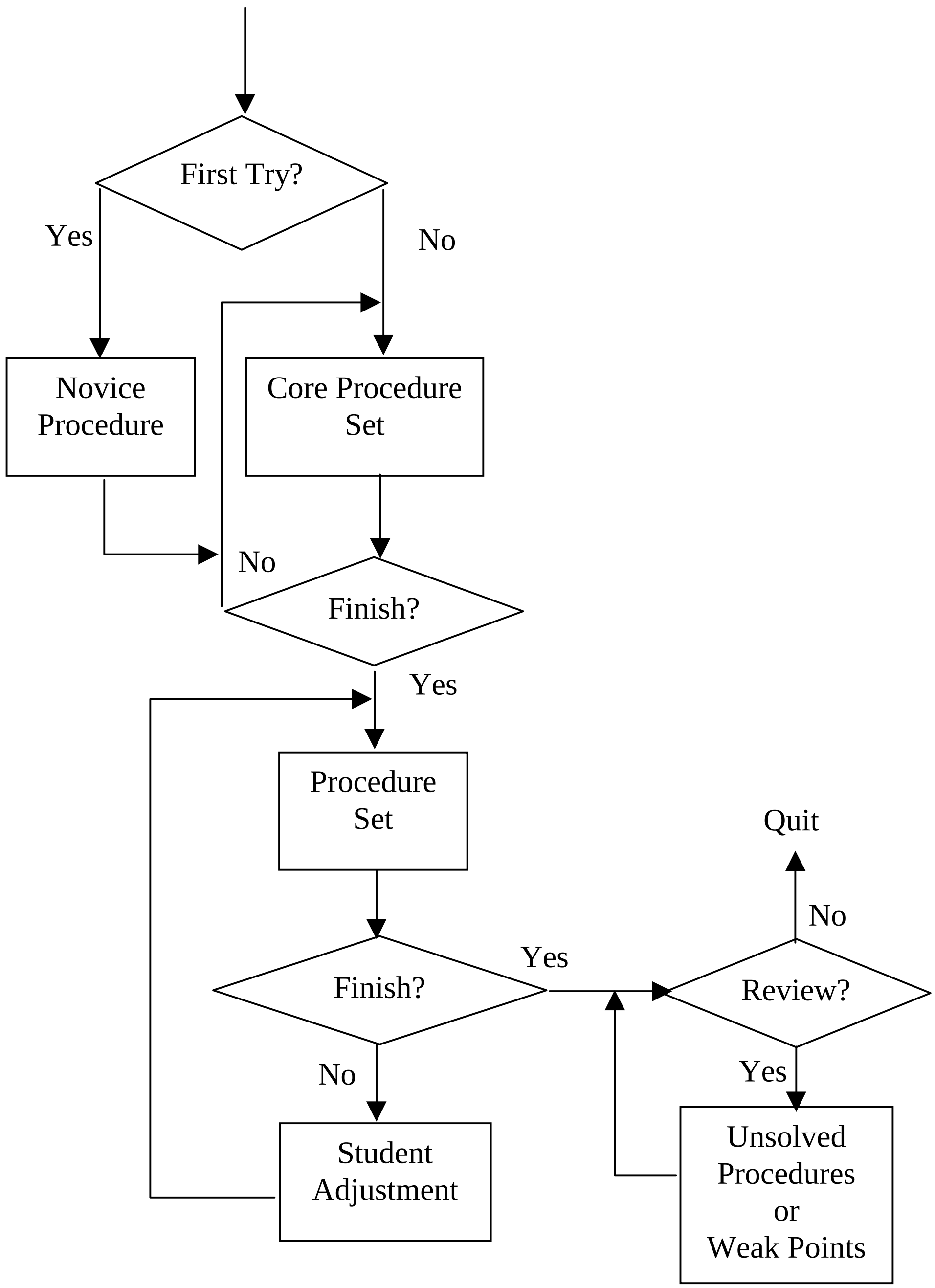

Figure 6.3 Flowchart for Curriculum Planning

Table 6.4 Computing Strategy

Global Assessment	Student Input	Procedure Difficulty Level	Procedure Description Level
High	Harder	⇧	⇧
Medium	Harder	•	⇧
Low	Harder	•	⇧
High	Same	⇧	•
Medium	Same	•	•
Low	Same	⇩	•
High	Easier	•	⇩
Medium	Easier	•	⇩
Low	Easier	⇩	⇩

Procedure/Description Level: ⇧ (Move up), ⇩(Move down), • (Stay in the same level)

The movement of the Procedure Description Level reflects the student requirements. This strategy makes the student aware of the movement of the difficulty level, in the way the student asked, with the Procedure Description Level in the procedure set. The movement of the Procedure Difficulty Level reflects the global assessment of the past procedure covertly. The student may not feel the difference at the next procedure selection time, but he/she must solve a more difficult procedure where the predictions are more difficult if the Procedure Difficulty Level increases.

We can imagine that in the worst case, when the student does not input any preference, the student might finish the entire category with the lowest Procedure Description Level. To prevent this, if the student performs two procedures well (or poorly) and inputs nothing twice in a row, then the Curriculum Planner increases (or decreases) one Procedure Description Level.

After determining the next Procedure Difficulty Level and Procedure Description Level, the Curriculum Planner finds a candidate procedure (a procedure with the calculated procedure and procedure description level) from each category in Table 6.2.

For each category, the Curriculum Planner tries to move a procedure from the procedure list into the procedure set. If the candidate procedure is in the procedure list then we put this procedure into the procedure set. Otherwise we find the procedure with the most similar difficulty in the procedure list. If the Curriculum Planner can not find a proper procedure we call this category "exhausted" and drop it from consideration even though some procedures too easy for the student remain in this category. So a procedure set may contain fewer than five procedures later in the session.

After all categories are exhausted, the student can choose one of three alternatives. First, if the student's selection is "Solve Skipped Procedures" then the Curriculum Planner rebuilds a procedure set with the Procedures remaining in each Category of the Procedure List. Though the student's knowledge is beyond the difficulty of these procedures, the student may want more practice with unsolved procedures. Second, if the student's selection is "Review my weak points" then CIRCSIM-Tutor constructs a procedure set that emphasizes procedures in categories where the student has performed less well. The Tutorial Planner in CIRCSIM-Tutor evaluates the weak points with the Student Model. Finally, if the student's selection is "Quit CIRCSIM-Tutor" then exit CIRCSIM-Tutor. Though some unsolved procedures remain, the student can exit CIRCSIM-Tutor by selecting the "Quit CIRCSIM-Tutor" menu item.

6.4.2 Design Principles and Procedure Selection Rules. This section describes some background assumptions and basic principles for designing the curriculum planning model. The central part, the procedure selection rules, in the curriculum planning model is designed based on these principles. The procedure selection rule includes two additional selection rules. The first one presents how to find the procedure with difficulty level closest to that of a given procedure. The other rules are related to the core perturbation.

6.4.2.1 Background Assumptions

BA1) The Curriculum Planner uses the knowledge base list of all procedures each with a procedure category with up to five procedure difficulty levels and up to four description levels.

BA2) The procedures are grouped into categories (CVP, IS, HR, TPR, BRP) on the basis of the primary variable.

BA3) Each category consists of basic and combination procedures, which are ordered by category, procedure difficulty level, procedure description level, and procedure name.

BA4) The system will display Procedure Descriptions instead of Procedure Names because too many of the Procedure Names pinpoint the Primary Variables.

BA5) The Curriculum Planner keeps students' previous Solved Procedures List and Procedure Performance History (global assessment).

BA6) The Procedure List contains all procedures that also includes Core Procedures. The remaining procedure list contains the unsolved procedures.

6.4.2.2 Principles

P1) Different procedure sets for different students.

P2) The maximum number of procedures in the procedure set is 5 (number of categories). The number of procedures can be reduced whenever a category is closed.

P3) The student must solve the Core Perturbation procedures first.

P4) The student does not have to solve all procedures and does not solve the same procedure more than once.

P5) The Curriculum Planner allows a student to adjust just one difficulty level.

P6) The Curriculum Planner computes the difficulty level of the procedure set from the global assessment and the student input.

P7) If the student is already at the highest level then an increase instruction is understood as “stay at the same level.” Similarly, a decrease in the difficulty level is interpreted as “stay put” if the student is already at the bottom.

6.4.2.3 Procedure Selection Rules

PSR1) If the student has not finished the Core Perturbation Procedures, then use the Core Perturbation Procedure Selection Rules

PSR2) If the global assessment is high and the student input is up, then increase one Procedure Difficulty Level and one Procedure Description Level.

PSR3) If the global assessment is low and the student input is up, then increase one Procedure Description Level.

PSR4) If the global assessment is high and the former global assessment was not high and there is no student input, then increase one Procedure Difficulty Level.

PSR4-1) If the global assessment is high and the former global assessment was high and the student does not input twice, then increase one Procedure Difficulty Level and one Procedure Description Level.

PSR5) If the global assessment is low and the former global assessment was not low and there is no student input, then decrease one Procedure Difficulty Level.

PSR5-1) If the global assessment is low and the former global assessment was low and the student does not input twice, then decrease one Procedure Difficulty Level and one Procedure Description Level.

PSR6) If the global assessment is high and the student input is down, then decrease one Procedure Description Level.

PSR7) If the global assessment is low and the student input is down, then decrease one Procedure Difficulty Level and one Procedure Description Level.

PSR8) If the global assessment is medium and the student input is up, then increase one Procedure Description Level.

PSR9) If the global assessment is medium and no student input, then change the Procedure Name.

PSR10) If the global assessment is medium and the student input is down, then decrease one Procedure Description Level.

PSR11) If the global assessment is changed, then change all the procedures in the procedure set. Else replace the recent procedure which was solved a moment ago with a new procedure.

PSR12) To change the procedure set, use rules PSR13 to PSR15.

PSR13) If the candidate procedure is in the procedure list, then put this procedure into the procedure set.

PSR14) If the candidate procedure is not in the procedure list, then find the procedure with the most similar difficulty level with "Rules for Finding the Procedure with the Most Similar Difficulty Level ".

PSR15) If the Curriculum Planner can not find a proper procedure, then drop this category.

PSR16) If the student selects a procedure in the procedure set, then move the procedure into the solved procedure list.

PSR17) If all categories are closed, then ask the student "Want to QUIT?"

PSR18) If the student wants to quit, then Stop CIRCSIM-Tutor.

PSR19) If all categories are closed and the student's selection is "Solve Skipped Procedures", then rebuild the procedure set with the remaining procedures in each category.

PSR20) If all categories are closed and the student's selection is "Review my weak points", then rebuild the procedure set with procedures in the student’s weakest category procedures.

6.4.2.4 Rules for Finding the Procedure with the Most Similar Difficulty Level

NPR1) If there is a harder procedure at the same Procedure Difficulty Level, then pick it out.

NPR2) If the current Procedure Difficulty Level is 4 (the highest level), then return with "no procedure to solve: exhausted".

NPR3) If there is an easier procedure in the same Procedure Difficulty Level, then pick it out. Else increase the Procedure Difficulty Level by one and find the procedure with the most similar difficulty level recursively.

6.4.2.5 Solve Skipped Procedure Rules

SSR1) If the student wants to solve a skipped procedure, then construct a procedure set with all unsolved procedures in the procedure list.

SSR2) If the student wants to solve a weak category procedure, then construct a procedure set with the weakest category procedures.

6.4.2.6 Core Perturbation Procedure Selection Rules

CPR1) If it is the first procedure, then the problem is “Reduce Arterial Resistance (RA) to 50% of Normal.” Else retrieve the previous global assessment of the student.

CPR2) If it is the second procedure, then the second procedure set should be selected from the second core perturbation group.

CPR3) If the global assessment is high, then the third procedure set should be selected from the third core perturbation group. Else the third procedure set should be selected from the second core perturbation group.

CPR4) If the student selects the procedure from “Hemorrhage-Remove 0.5 L” perturbation, then the system should suggest “Hemorrhage-Remove 1.0 L” perturbation as the next step.

CPR5) If the student completes the third core perturbation successfully, then the core perturbation restriction disappears so do PSR 1. Else do from PSR 2 to PSR 16 with the core perturbation list instead of the procedure list.

CHAPTER VII

MULTIPLE TUTORING PROTOCOLS

Determining effective tutoring strategies may be the most important and hardest issue in intelligent tutoring systems. The tutoring protocol controls the interaction between the tutor and the student in a tutoring session. Moore [1989] identified three types of interaction: student-content, student-teacher, and student-student. Much work has focused on student-teacher interaction and how tutoring strategies affect the student's learning [Sandoval et al., 1999; Tabak and Reiser 1999]. In traditional classroom teaching interaction between student and teacher is normally immediate. Much educational research supports the belief that immediate feedback increases the sense of excitement and spontaneity [Moore and Kearsley, 1996; Cuffman and MacRae, 1996; Travers and Decker, 1999]. Our colleagues, Joel Michael (JAM) and Allen Rovick (AAR) believe, however, that immediate feedback is not always the best choice. They feel that they can do a better job of tutoring if they ask the student to make predictions first, because the improved student model allows them to plan a tutorial strategy that targets the student's misconceptions [Michael et al., 1992].

7.1 Tutoring Protocols

Tutorial planning determines the content and sequence of the subject matter to be taught in a single procedure. One of the important features of the tutorial planning process is the tutoring protocol. The tutoring protocol defines the overall communication between the tutor and the student. We want to be able to compare the effects of different protocols or to change the protocol during a session.

Khuwaja described three tutoring protocols that he found used in human tutoring sessions [Khuwaja et al., 1994]. In Tutoring Protocol 1 (see Figure 7.1) the tutor ignores the sequence of the student's predictions and explores the student's response at each point in problem solving. Here the tutor provides immediate feedback for each student's prediction and response.

- Tutor (problem)
  - Tutor (DR)
    - Prediction & Tutoring (Primary Variable)
      - Prediction (Primary Variable)
      - Tutoring (Primary Variable)
    - Prediction & Tutoring (Rest of the prediction table variables)
      - Prediction (Variable X)
      - Tutoring (Variable X)
  - Tutor (RR)
    - Prediction & Tutoring (Prediction table variables)
      - Prediction (Variable X)
      - Tutoring (Variable X)
  - Tutor (SS)
    - Prediction & Tutoring (Prediction table variables)
      - Prediction (Variable X)
      - Tutoring (Variable X)

Figure 7.1 Tutoring Protocol 1 [Khuwaja, 1994. p. 128]

In Tutoring Protocol 2 (see Figure 7.2) the tutor insists that the student follow the preferred prediction sequence but does not correct the values of the variables until all predictions have been made.

In Tutoring Protocol 3 (see Figure 7.3) the tutor makes sure that the student chooses the primary variable (DR) first and predicts its change correctly before asking the student to predict the remaining variables in any order. In RR and SS the students are free to start with any variable and to make predictions in whatever sequence they choose.

```
Tutor (problem)
 Tutor (DR)
      Prediction & Tutoring (Primary Variable)
           Prediction (Primary Variable)
           Tutoring (Primary Variable)
      Prediction (Rest of the prediction table variables)
           Prediction (Variable X)
                If "sequence violation"
                Then give a generic hint (but do not tutor)
      Tutoring (Rest of the prediction table variables)
Tutor (RR)
      Prediction (Prediction table variables)
           Prediction (Variable X)
                If "sequence violation"
                Then give a generic hint (but do not tutor)
      Tutoring (Prediction table variables)
Tutor (SS)
      Prediction (Prediction table variables)
      Tutoring (Prediction table variables)
```

Figure 7.2 Tutoring Protocol 2 [Khuwaja, 1994. p. 129]

```
Tutor (Problem)
      Tutor (DR)
           Prediction & Tutoring (Primary Variable)
                Prediction (Primary Variable)
                Tutoring (Primary Variable)
           Prediction (Rest of the prediction table variables)
           Tutoring (Rest of the prediction table variables)
      Tutor (RR)
           Prediction (Prediction table variables)
           Tutoring (Prediction table variables)
      Tutor (SS)
           Prediction (Prediction table variables)
           Tutoring (Prediction table variables)
```

Figure 7.3 Tutoring Protocol 3 [Khuwaja, 1994. p. 131]

Each tutoring session in our transcripts can be divided into prediction phases and tutoring phases. The structure of the Prediction Table divides the problem into three stages - DR, RR, and SS. In each stage the tutor performs two common operations. During the first operation “Prediction”, the student predicts whether a physiology variable will increase (go up, +, up,…), decrease (go down, –, down, …), or stay the same (unchanged, 0, stay, …). During the second operation “Tutor”, the tutor starts a dialogue to remedy any prediction errors.

7.2 A New Tutoring Protocol

In this study we made a detailed analysis of a set of tutoring sessions to see how the tutors used the tutoring protocols. Most sessions contain only one procedure and are one hour long. We chose to study the nine sessions that involved two procedures and lasted up to two hours, so we could observe changes in behavior over time. In four sessions the tutor started with the Centrifuge Procedure in which the primary variable is CVP. In five sessions the tutor started with the Alpha-Adrenergic Procedure in which the primary variable is TPR.

7.2.1 Protocol Switching Examples. The results of the analysis of protocol use in the tutoring sessions were a complete surprise to us. The tutors had decided to use Protocol 3 in all the tutoring sessions. This means that the tutor analyzes the student’s prediction result and then plans the tutoring strategy based on these results. Sometimes, however, the tutor does not wait until the student finishes the predictions. If the student starts with poor predictions then the tutor starts to guide the student in the right track with hints or explanations. Figure 7.4 shows an example of a protocol switch from Protocol 3 to Immediate Feedback in the DR stage of a TPR procedure. I added little comments between "/*" and "*/". The transcript in Figure 7.4 begins with a discussion of the basic concepts. The discussion is finished in an unsatisfactory manner. And then the student makes a correct prediction of the first variable from K38-st-39-1 to K38-st-41-1. When the student gives the wrong prediction in K38-st-43-1, however, the tutor starts the tutoring feedback immediately. The tutor does not wait until the student predicts all six remaining variables.

On the other hand, the tutors go back to Protocol 3 from immediate feedback when the students are doing well. Figure 7.5 and Figure 7.6 show an example of the protocol switching from immediate feedback to Protocol 3 on a CVP procedure. After finishing the TPR procedure the tutor continued using immediate feedback at the beginning of the CVP procedure. Figure 7.5 shows that the DR stage starts with immediate feedback. The student answers the primary variable correctly and then makes correct predictions of five of the six variables.

As the results of the previous stage (DR) were good, the tutor decides to use protocol 3 again at the beginning of the RR stage as shown in Figure 7.6. The tutor allows the student to predict all seven variables in any order. The tutor starts the tutoring session at K38-tu-188-1, though the student missed the three of the seven predictions during the prediction session (from K38-st-167-1 to K38-st-187-1).

/* K38 TPR; DR stage; Protocol 3 */
K38-st-35-1: THEY PROBABLY ARE ALPHA-ADRENERGIC RECEPTORS.
K38-tu-36-1: No. /* Wrong Answer */
K38-tu-36-2: The heart contains largely beta receptors, while the blood vessels have predominately alpha receptors.
K38-tu-36-3: So, to get back to our problem.
K38-tu-36-4: If an alpha agonist is administered, which variable will be changed first?
K38-st-37-1: TPR WILL INCR. /* Correct Answer */
K38-tu-38-1: Good.
K38-tu-38-2: Which variable do you want to predict next?
/* End Discussion with unsatisfactory */
K38-st-39-1: MAP
K38-tu-40-1: Good, and how will it change?
K38-st-41-1: IT WILL INCR. /* Correct Answer */
K38-tu-42-1: Great.
K38-tu-42-2: What do you want to predict next?
K38-st-43-1: RAP /* Wrong Answer */
/* Immediate Feedback starts here */
K38-tu-44-1: Do you know which parameter in the prediction table determines rap?
K38-st-45-1: SV AND CO?
K38-tu-46-1: CO is the right answer. /* Partially Correct Answer */
K38-tu-46-2: But in any event how can you predict a change in rap without knowing how co will change?
K38-st-47-1: TRUE.
K38-st-47-2: CO MUST INCREASE. /* Wrong Answer */
K38-tu-48-1: It’s important that you try to think in terms of the causal relationships between the variables.
K38-tu-48-2: What determines CO?
K38-st-49-1: SV X HR /* Correct Answer */
K38-tu-50-1: Have you predicted either of those yet?
K38-st-51-1: NOT REALLY.
K38-tu-52-1: Let me give you two hints that might help you.
K38-tu-52-2: First, think about the definition of DR.
K38-tu-52-3: Second, think about after load.

Figure 7.4 A Protocol Switch from Protocol 3 to Immediate Feedback

```
/* K38 CVP; DR Stage; Immediate Feedback */
K38-tu-150-1: Right.
K38-tu-150-2: What next?
K38-st-151-1: CO
K38-tu-152-1: Good.
K38-tu-152-2: How will it change?
K38-st-153-1: IT WILL DECR BEACUSE SV DECR    /* Correct Answer */
K38-tu-154-1: What next?
K38-st-155-1: MAP WILL DECR BECAUSE CO IS NOW DECR
                                              /* Correct Answer */
K38-tu-156-1: Super!
K38-tu-156-2: What next?
K38-st-157-1: CC WILL NOT BE AFFECTED         /* Correct Answer */
K38-tu-158-1: Good.
K38-tu-158-2: What next?
K38-st-159-1: TPR WILL BE INCR BEACUSE MORE BLOOD IS IN THE
    PERIPHERY                                 /* Wrong Answer */
K38-tu-160-1: How would that cause incr tpr?
K38-st-161-1: IT WOULDN'T.
K38-st-161-2: TPR WOULD PROBABLY BE UNAFFECTED.
                                              /* Correct Answer */
K38-tu-162-1: Exactly!.
K38-tu-162-2: Finally, what happens to hr?
K38-st-163-1: HR WILL BE UNAFFECTED BECAUSE IT IS UNDER
    NEURAL CONTROL.                           /* Correct Answer */
K38-tu-164-1: Good work!
K38-tu-164-2: You now have a completely correct set of predictions for dr.
K38-tu-164-3: Read page 10.
K38-st-165-1: OK
```

Figure 7.5. A Protocol Switch from Immediate Feedback to Protocol 3 (DR Stage)

According to our analysis of the transcripts, when the tutor used immediate feedback they also pursued a particular tutoring sequence. After changing the protocol to immediate feedback the tutor enforces the sequence. In Protocol 3 the tutor allows an arbitrary prediction sequence. However after changing the protocol, the tutor does not allow sequence errors. The student must follow the solution order that the tutor expects.

```
/* K38 CVP; RR Stage; Start with Protocol 3 */
K38-tu-166-1: What do you want to predict first?
K38-st-167-1: CC
K38-tu-168-1: How will it change?
K38-st-169-1: IT WILL INCR                    /* Correct Answer */
K38-tu-170-1: Next?
K38-st-171-1: HR
K38-tu-172-1: Change?
K38-st-173-1: INCR                            /* Correct Answer */
K38-tu-174-1: Next?
K38-st-175-1: CO WILL INCR                    /* Correct Answer */
K38-tu-176-1: Next?
K38-st-177-1: RAP
K38-tu-178-1: Change?
K38-st-179-1: INCR                            /* Wrong Answer */
K38-tu-180-1: Next?
K38-st-181-1: SV
K38-tu-182-1: Change?
K38-st-183-1: INCR                            /* Wrong Answer */
K38-tu-184-1: Next?
K38-st-185-1: MAP WILL INCR                   /* Correct Answer */
K38-tu-186-1: And tpr?
K38-st-187-1: WILL DECR                       /* Wrong Answer */
/* Start Tutoring Session */
K38-tu-188-1: Why?
K38-st-189-1: TO ALLOW FOR THE INCR SV AND CO
K38-tu-190-1: I don't understand what you just said.
K38-st-191-1: BY DECR TPR, IT WILL BE EASIER TO PUSH THE BLOOD
    OUT OF THE LET VENTRICLE AND IT WILL BE EASIER TO
    ACCOMODATE THE INCR SV AND HR
K38-tu-192-1: I understand what you are saying, but let's think about the situation
    this way.
K38-tu-192-2: What is the physiological input that determines tpr?
```

Figure 7.6 A Protocol Switch from Immediate Feedback to Protocol 3 (RR Stage)

Figure 7.7 shows an example of a sequence error during an immediate feedback tutoring session in the RR stage of a TPR procedure. The student gives correct answers about the CO variable at K38-st-107-1 and K38-st-109-1. Then the student tries to predict the Stroke Volume at K38-st-111-1. However the tutor does not allow this variable sequence. The tutor insists that the student predict Right Atrial Pressure before Stroke Volume (The variable Right Atrial Pressure is replaced by CVP in the later transcripts.). The tutor then allows the student to predict the SV at K38-tu-120-1.

/* K38 TPR; RR Stage; Immediate Feedback */
K38-tu-106-1: Good.
K38-tu-106-2: What next?
K38-st-107-1: CO
K38-tu-108-1: How will it change?
K38-st-109-1: IT WILL DECR DUE TO DECR HR (O)
K38-tu-110-1: Good thinking!
K38-tu-110-2: What next?
K38-st-111-1: SV
/* Sequence Error */
K38-tu-112-1: What determines sv?
K38-st-113-1: END-SYSTOLIC VOL, HR, AND CO
K38-tu-114-1: No.
K38-tu-114-2: SV is determined by the extent of filling (which is determined by rap) and by cardiac contractility.
K38-tu-114-3: Rap is the more potent of the two and you haven't predicted it yet.
K38-tu-114-4: What do you want to predict now?
K38-st-115-1: RAP
K38-tu-116-1: And how does it change?
K38-st-117-1: IT WILL DECR
K38-tu-118-1: No, there is an inverse relationship between co and rap and co decreased so rap must increase.
K38-tu-118-2: OK?
K38-st-119-1: I UNDERSTAND
K38-tu-120-1: Now you can predict sv.
K38-st-121-1: SV WILL INCR DUE TO MORE BLOOD BEING PRESENT AT EJECTION
K38-tu-122-1: Well, more blood filling the heart (end diastolic volume is increased).
K38-tu-122-2: So, let's review.
K38-tu-122-3: Because map increased in dr the reflex attempted to decr map.
K38-tu-122-4: Cc and hr were decreased.
K38-tu-122-5: The decrease in hr caused co to decrease.
K38-tu-122-6: This caused rap to increase and in turn this caused sv to incr.
K38-tu-122-7: OK?

Figure 7.7 An Example of a Sequence Error

Figure 7.8 shows prediction sequences in the RR stage of the Alpha-Adrenergic Procedure. Since Protocol 3 is used in K33 and K37, the prediction sequences are not followed any particular order. However the prediction sequences are restricted to follow the solution order that the tutor expects in K31 and K38.

Alpha-adrenergic Procedure (RR stage)

[Protocol 3]
(K33) IS → SV → HR → CO → TPR → CVP → MAP
(K37) HR → CO → CVP → SV → IS → TPR → MAP
[Immediate Feedback]
(K31) TPR → MAP → HR → IS → CO → CVP → SV
(K38) TPR → MAP → HR → IS → CO → CVP → SV

Figure 7.8 Some Example Prediction Sequences

7.2.2 Protocol 4. We named this new protocol, Protocol 4. In Protocol 4 (see Figure 7.9) the tutor considers the student's prediction sequences. The tutor explores the student's response at each point in the problem solving process. Therefore the tutor provides immediate feedback for each student prediction and response. In Protocol 4, like other protocols, the primary variable is predicted and taught first. The tutor insists that the rest of the variables be predicted and taught in the sequence defined by the problem. If the student does not follow the sequence, the tutor gives a sequence hint about the prediction order based on the causal reasoning to be followed. Otherwise, the tutor gives instant feedback for the predicted variable [Cho et al., 2000].

```
Tutor (Problem)
    Tutor (DR)
        Prediction & Tutoring (Primary Variable)
            Prediction (Primary Variable)
            Tutoring (Primary Variable)
        Prediction & Tutoring (Rest of the prediction table variables)
            If not "sequence violation "
                then Prediction (Variable X)
                     Tutoring (Variable X)
                Else give a sequence hint
    Tutor (RR)
        Prediction & Tutoring (Prediction table variables)
            If not "sequence violation"
                then Prediction (Variable X)
                     Tutoring (Variable X)
                Else give a sequence hint
    Tutor (SS)
        Prediction & Tutoring (Prediction table variables)
            Prediction (Variable X)
        Tutoring (Variable X)
```

Figure 7.9 Tutoring Protocol 4

7.3 Analysis of Human Tutoring Transcripts

We used C5.0 [RuleQuest, 1999], which is an upgraded version of the decision tree induction program C4.5 [Quinlan, 1993], in a series of machine learning experiments to discover the rules that describe when our domain experts switch tutoring protocols. In this experiment we had 44 cases (the number of tutoring phases recorded in the 18 procedures studied), each with 11 attributes.

The first three attributes in Table 7.1 are related to the discussion about the basic concept. The basic concept involves the effects of the centrifuge in the Centrifuge procedure, or the Alpha-adrenergic Receptors in the Alpha-adrenergic procedure. The students often had difficulty in determining the primary variable. Therefore tutoring frequently began with a discussion of the relationship between the basic concept and the primary variable. Thirteen of the eighteen procedures started with a discussion of the basic concepts. The Discussion Type (DT) in Table 7.1 is T if the tutor started the discussion to remedy a wrong primary variable prediction. It is S if the student began the discussion with a request for an explanation. The Discussion Success (DS) indicates whether the discussion was successful or not, that is, whether the tutor is satisfied with the student's responses at least 50% of the time. The Discussion Length (DL) indicates the number of turns in the discussion counting from the start to the turn in which the student gave the right prediction for the primary variable.

Table 7.1 Attributes for Rule Extraction

attribute	value	remark
Discussion Type	T, S	Tutor-Primary / Student-Explanation
Discussion Success	S, U	Satisfactory / Unsatisfactory
Discussion Length	continuous	How many turns in the discussion
Primary Error	continuous	Wrong answers for a primary variable
Prediction Score	continuous	(right - wrong) prediction
Remediation	continuous	Correct answers in the total answers
Sequence Error	continuous	Sequence error in Protocol 4
Pre-Prediction Score	continuous	Previous stage, (right - wrong) prediction
Pre-Remediation	continuous	Previous stage, correct answers in the total answers
Pre-Sequence error	continuous	Previous stage, sequence error in Protocol 4
Current Stage	dr, rr, ss	Current stage

The Primary Error (PE) is a count of the number of wrong answers entered for the primary variable. The number of primary errors reflects the comprehension of the procedure. The Prediction Score (PS) indicates how many wrong or right answers were made in the prediction phase. The score we used was the number of right answers minus the number of wrong answers. The value of the Remediation (RM) attribute is the percentage of correct answers among the total answers given by the student in that stage. The sequence error (SE) attribute represents the number of sequence errors during Protocol 4. The Pre-Prediction Score, Pre-Remediation, and Pre-Sequence Error: These attributes represent the Prediction Score, Remediation, and Sequence Error from the previous stage. The Current Stage (CS) indicates the stage on which the student is now working.

Table 7.2 The Summary of Human Tutoring Sessions K30 - K38

Tutor	Session	DR							RR			SS		
		DT	DS	DL	PE	PS	RM	SE	PS	RM	SE	PS	RM	SE
	K30 CVP	-	-	-	0	1	0	-	3	0.22	-	1	0.25	-
	K30 TPR	T	U	19	3	3	0.44	-	5	0	-	3	0	-
	K31 TPR	S	U	28	5	-1	0.3	-	-3	-	-		=	
	K31 CVP	-	-	-	2	-	0.2	0		=			=	
	K32 TPR	S	S	4	1	5	0.28	-	3	0.25	-	5	0	-
AAR	K32 CVP	T	S	13	2	7	0	-	7	0	-	5	0.4	-
	K33 CVP	-	-	-	0	7	0	-	1	0	-	5	0.5	-
	K33 TPR	T	S	13	1	1	0.5	-	3	0.5	-	7	0	-
	K34 CVP	T	U	5	1	1	0.125	-	-1	-	-		=	
	K34 TPR	S	U	4	1	-	0.45	0		=			=	
	K35 CVP	-	-	-	1	1	0	-	3	0.4	-	1	-	-
	K35 TPR	S	U	14	1	-	0.17	0	-	0.25	0	-	0	0
	K36 TPR	S	U	14	1	1	-	-		=			=	
	K36 CVP	T	S	7	1	7	0	0	1	0.33	-		=	
JAM	K37 CVP	-	-	-	0	-1	0	-	7	1	-	3	0	-
	K37 TPR	S	S	10	0	3	0.1	-	5	0	-	7	0	-
	K38 TPR	T	U	17	1	-2	-	-	-	0.25	2		=	
	K38 CVP	T	S	11	1	-	0.17	1	1	0.75	-	3	0.33	-

Table 7.2 summarizes sessions K30 - K38, which are the input to the rule induction program. CVP indicates the Centrifuge Procedure (for which CVP is the primary variable) and TPR indicates the Alpha-Adrenergic Procedure (for which TPR is the primary variable). “=” means the transcript does not have the stage data. “-” means that the data is not available. For example, the “K30 CVP” procedure did not include a

discussion about the basic concepts. A white cell indicates that Protocol 3 was in use and a shaded cell indicates that Protocol 4 was in use in that stage.

The target feature (switch) has four possible outcomes. 3f: remain with Protocol 3; 3t: switch from Protocol 3 to Protocol 4; 4f: remain with Protocol 4; 4t: switch from Protocol 4 to Protocol 3. The rules extracted by C5.0 do not classify all cases correctly; there is an error rate of 9.1 % in the decision tree (see Figure 7.10). A cryptic (n) or (n/m) follows every leaf of the tree. For example, tutoring > 0.5: 4t (1.5/0.4), for which n is 1.5 and m is 0.4. The value of the number n is the number of cases in the data file that mapped to this leaf, and m (if it appears) is the number of them that were classified incorrectly into this leaf. A class size may be shown with a decimal fraction, because, when the value of an attribute in the tree is not known, C5.0 splits the case and sends a fraction down each branch.

The switching rules are:

```
If Discussion Success = S
  If Remediation <= 0.5
    If Current Stage = ss
      If Prediction Score <= 1
      then switch from Protocol 3 to Protocol 4

If Discussion Success = U
  If Primary Error > 2
    If Prediction Score <= -2
    then switch from Protocol 3 to Protocol 4

If Discussion Success = U
  If Primary Error <= 2
    If Pre-Remediation > 0.35
      If Current Stage <> rr
      then switch from Protocol 3 to Protocol 4
```

```
If Discussion Success = U
  If Primary Error <= 2
    If Pre-Remediation <= 0.35
      If Discussion Type = T
        If Current Stage = rr
        then switch from Protocol 3 to Protocol 4

If Discussion Success = S
  If Remediation > 0.5
  then switch from Protocol 4 to Protocol 3

If Discussion Success = S
  If Remediation <= 0.5
    If Current Stage = dr
      If Pre-Sequence Error <= 1
        If Pre-Prediction Score <= 1
        then switch from Protocol 4 to Protocol 3
```

Examing these rules we see that two important factors determined whether the tutor switches the protocol from Protocol 3 to Protocol 4. The occurrence of a discussion about the basic concepts of the procedure at the very beginning of a procedure is an important factor in protocol switches. For example, the protocol switch is likely if the student asked for some explanation before the prediction and did not understand that explanation right away, or if the tutor asked some question about the basic concepts to remedy the student's wrong primary variable prediction, but the student replied with unsatisfactory answers. The other important factor that makes the tutor switch the protocol from Protocol 3 to Protocol 4 is the student's performance scores. The scores are the prediction score, the tutoring scores, and the number of primary variable errors. However, if the student performed well in the previous stage then the tutor did not switch but gave some hints about the primary variable.

```
Discussion Success = S:
:...Remediation > 0.5: 4t (1.5/0.4)
:   Remediation <= 0.5:
:   :...Current Stage = rr: 3f (4.3)
:       Current Stage = ss:
:       :...Prediction Score <= 1: 3t (1.9/0.2)
:       :   Prediction Score > 1: 3f (2.5)
:       Current Stage = dr:
:       :...Pre-Sequence Error <= 1:
:           :...Pre-Prediction Score <= 1: 4t (1.6/0.4)
:           :   Pre-Prediction Score > 1: 3f (1.4)
:           Pre-Sequence Error > 1:
:           :...Pre-Remediation <= 0.125: 3f (1.4)
:               Pre-Remediation > 0.125: 4f (3/1.2)
Discussion Success = U:
:...Primary Error > 2:
    :...Prediction Score <= -2: 3t (1.1)
    :   Prediction Score > -2: 3f (4.4)
    Primary Error <= 2:
    :...Pre-Remediation > 0.35:
        :...Current Stage = dr: 3t (1.5/0.7)
        :   Current Stage = rr: 4f (1.1)
        :   Current Stage = ss: 3t (3.3/0.5)
        Pre-Remediation <= 0.35:
        :...Discussion Type = S: 4f (11.7/3.6)
            Discussion Type = T:
            :...Current Stage = dr: 3f (1.8/0.8)
                Current Stage = rr: 3t (1.5/0.3)
                Current Stage = ss: 3f (0.1)
```

Figure 7.10 The Decision Tree

On the other hand, the tutor went back from Protocol 4 to Protocol 3 under the opposite conditions. If the student performed well in the previous stage and there was a satisfying discussion about the basic procedure concepts then the tutor switched back to Protocol 3.

7.4 Some Other Important Characteristics

We found some by-product rules during the analysis of human transcripts. We wondered how long the tutor keeps a given protocol. The tutor always starts a procedure with Protocol 3. However, after switching the protocol, the tutor sticks with the Protocol 4 to the next stage or next procedure.

Student initiatives also affect protocol switches. A student initiative means any student contribution to the dialogue that is not an answer to a question asked by the tutor [Sanders, 1995; Shah and Evens, 1997; Freedman, 1997a]. In Protocol 3, sometimes our tutor met with a simple student initiative that requires only a short response from the tutor. In the prediction phase, for example, the student may ask a simple question [Freedman, 1997a] or the student could not answer in more than one minute. In these cases, the tutor gives a simple hint [Zhou et al., 1999a] and sticks with the current protocol. In particular, if the student performed well in the previous stage, but starts with poor predictions, then the tutor does not switch protocols but gives hints about the current stage.

7.5 Which Students Prefer Immediate Feedback?

Who prefers immediate feedback? Which students feel a lack of instant feedback during the prediction phase in Protocol 3? We can imagine the situation intuitively. Sometimes the student may want to know whether the current prediction is correct or not. The variables in the prediction table have causal relations among them. Therefore if the student is not sure of one variable then the uncertainty may affect the following variables. In order to discover how the students feel about the protocol issue we compared the students' performance using CIRCSIM-Tutor with their answers to a questionnaire asking about their view of CIRCSIM-Tutor.

This data came from an experiment that was performed by forty-eight first year students at Rush Medical College, in November 1998. The system presented four cardiovascular procedures to be solved. The system was designed to use Protocol 3, which means no immediate feedback and discard the prediction sequence except for the primary variable.

YOUR VIEWS ON CIRCSIM-TUTOR
1= definitely YES …2…3…4…5=definitely NO

1. The print in the display was readable	1 2 3 4 5
2. The screen layout was helpful	1 2 3 4 5
3. The sequence of displays was appropriate	1 2 3 4 5
4. The system was easy to use	1 2 3 4 5
5. The introductory screens were helpful	1 2 3 4 5
6. The system's dialogue seemed varied and interesting	1 2 3 4 5
7. The tutor's hints and explanations were informative	1 2 3 4 5
8. **I would prefer that the system always tell me about my mistakes immediately**	1 2 3 4 5
9. CIRCSIM-Tutor helped me to understand the behavior of the baroreceptor reflex	1 2 3 4 5
10. CIRCSIM-Tutor improved my ability to predict the cardiovascular responses to disturbance in blood pressure	1 2 3 4 5

Figure 7.11 The Questionnaire

After using CIRCSIM-Tutor the students answered a questionnaire (see Figure 7.11) that asks the students' view of the system. The questionnaire had ten questions and employed a five point Likert scale. Question 8 "I would prefer that the system always tell me about my mistakes immediately" asked the student's opinion about the tutoring protocol.

Seven students answered that they would prefer immediate feedback and six of them (86%) made poor predictions while using the program. We defined a poor/good prediction result to mean that the student's prediction score on four procedures was under/over the average prediction result (34.7) of all students. Which students do not ask for immediate feedback? Twelve students answered that they prefer the current protocol, Protocol 3, which does not give feedback immediately, in the questionnaire and eight of them (67%) made good predictions when using the program. The analysis results suggest that the students who made poor predictions are eager to know their mistakes immediately.

Table 7.3 illustrates both the student's questionnaire data and procedure performance score. ST means student number. The Total Prediction Score (PSTOT) is an average value of the student's performances on the four procedures. P is the number of primary variable prediction errors in a procedure. The Prediction Score (PS) and PSTOT are evaluated by following equations. The number followed by PS indicates the procedure number (ex. PS4: Prediction Score of the procedure 4).

$$PS = 2^* (D + R + S) - P$$

$$PSTOT = (PS4 + PS1 + PS5 + PS6) / 4$$

D, R, and S are the number of correct predictions in a DD, RR, and SS stage.

The students are sorted by PSTOT. The color of each row, white, lightly shadowed, and dark shadowed can categorize the students. A lightly shadowed row means that the student selected 1 or 2 in Question 8. It implies that the student prefers immediate feedback as in protocol 1 or 4. On the other hand, a dark shadowed row means that the student selects 4 or 5 in Question 8. It implies that the student prefers the current protocol (protocol 3.) A white row means that the student chooses three in Question 8 or the student's questionnaire data was accidentally deleted.

Table 7.3 The Questionnaire and the Prediction Results

ST	PSTOT	P	D4	R4	S4	PS4	P1	D1	R1	S1	PS1	P5	D5	R5	S5	PS5	P	D6	R6	S6	PS6
11	23	1	5	4	1	19	2	5	2	3	18	3	4	6	5	27	2	4	4	7	28
1	29.8	1	7	4	2	25	1	5	4	6	29	0	7	5	5	34	3	4	7	6	31
17	30.5	2	4	5	1	18	3	5	5	7	31	0	7	5	7	38	1	7	4	7	35
38	30.5	1	7	5	2	27	0	0	5	5	20	0	7	7	6	40	1	5	7	6	35
26	31.3	1	6	4	5	29	1	5	7	6	35	0	7	3	4	28	1	6	6	5	33
22	33	1	6	5	7	35	2	5	4	6	28	0	7	6	7	40	1	4	4	7	29
23	33.8	1	6	5	7	35	0	5	5	6	32	0	6	4	7	34	0	6	4	7	34
8	34	1	6	5	6	33	0	6	5	6	34	0	7	6	4	34	1	7	5	6	35
10	34.3	0	7	5	4	32	0	4	6	6	32	0	5	4	7	32	1	7	7	7	41
15	34.3	0	7	7	5	38	0	4	6	6	32	0	6	5	7	36	1	4	5	7	31
35	34.3	1	7	4	6	33	3	6	6	7	35	0	5	5	7	34	1	5	6	7	35
6	34.5	2	7	6	7	38	0	5	3	7	30	0	6	6	5	34	4	7	7	6	36
13	34.5	1	6	5	7	35	0	4	3	6	26	0	6	6	7	38	1	6	7	7	39
32	34.5	1	6	5	7	35	0	5	5	6	32	0	6	6	7	38	3	5	6	7	33
12	34.8	1	4	6	7	33	0	4	5	7	32	1	7	7	7	41	1	5	5	7	33
2	35	0	7	5	7	38	2	4	6	6	30	0	7	6	7	40	4	5	6	7	32
4	35	2	5	5	7	32	0	5	5	7	34	1	6	6	6	35	1	6	7	7	39
7	35.5	1	6	5	7	35	0	6	5	6	34	0	7	6	7	40	1	5	5	7	33
36	35.5	2	7	4	7	34	1	5	6	6	33	0	7	6	7	40	1	5	6	7	35
33	36.8	1	7	5	7	37	0	4	7	6	34	0	6	7	7	40	4	6	7	7	36
21	37	1	4	7	6	33	0	4	7	6	34	0	7	7	6	40	1	7	7	7	41
24	37	3	7	5	7	35	0	4	7	6	34	0	7	6	7	40	1	7	6	7	39
31	37	1	6	7	5	35	0	4	6	6	32	0	7	6	7	40	1	7	7	7	41
9	37.8	2	7	5	7	36	0	5	7	6	36	0	7	6	7	40	1	6	7	7	39
34	38.3	0	7	7	7	42	0	6	1	7	28	0	7	7	7	42	1	7	7	7	41
16	38.5	1	7	5	7	37	0	6	7	6	38	0	6	6	7	38	1	7	7	7	41
14	38.8	1	7	6	7	39	0	4	6	7	34	0	7	6	7	40	0	7	7	7	42
5	39	1	7	6	7	39	1	4	7	7	35	0	7	7	7	42	2	7	7	7	40
3	39	2	7	5	7	36	1	7	7	7	41	1	7	6	7	39	0	6	7	7	40

7.6 Summary

In this study we carried out a detailed analysis of tutoring sessions to see how the tutors used the tutoring protocols. Two important factors that determined whether the tutor switches the protocol are the discussion about the basic concepts of the procedure and the student's performance scores. If the student did not perform well then the tutor responded immediately. The tutor did not wait until the student finished the predictions but started to tutor the student in the right track. However, when the tutor met a simple student initiative the tutor did not change the protocol but gave some hints. If the student performed well in the previous stage, giving good predictions and a satisfactory discussion about the basic procedure concepts, then the tutor switched back to Protocol 3. In order to find out which protocol the students prefer we also analyzed the result of the CIRCSIM-Tutor experiment and the questionnaire. The analysis results say that the students' protocol preference is closely related to their performance using CIRCSIM-Tutor. In general, students who are doing well are comfortable waiting for feedback while most students who are doing badly want immediate feedback.

The Multiple Tutoring Protocols model can switch the tutoring protocols so that we can emulate the human tutors better and to discover which tutoring protocol gives the best results in teaching causal reasoning.

CHAPTER VIII

CONCLUSIONS

8.1 Summary

I have presented a study of the dynamic planning capability for an intelligent tutoring system. This included the following: 1) the development of a curriculum planning model that constructs an individualized problem set for the student, 2) an approach to student assessment at the global level to support curriculum planning, 3) an analysis of how tutors employ tutoring protocols to respond to their students' needs and the extraction of protocol switching rules, and 4) an implementation of these models in CIRCSIM-Tutor.

Most existing ITSs do not carry out extensive curriculum planning because they have only a small number of problems or their problems are not classified in terms of the levels of difficulty. The development of a curriculum planning model makes the planning in CIRCSIM-Tutor more dynamic. My Curriculum Planning Model has the following important capabilities. First of all the model provides a dynamic planning capability. Therefore it plans an individualized curriculum for each student. The Curriculum Planning Model presents Procedure Descriptions instead of Procedure Names, so that the students must reason from the Procedure Variable to the Primary Variable. It makes use of student input. Student input helps avoid boredom for a skilled user and increases the motivation. The Curriculum Planning Model is complete and consistent unlike the previous one. It covers the whole session. The new Curriculum Planning Model allows the student to explore a wide range of procedures. Also, if the student wants to exercise

CIRCSIM-Tutor more, then this model suggests appropriate procedures to remedy any weak points.

The research of multiple tutoring protocols was based on the study of human tutoring sessions to understand human tutoring so that we can emulate it better and to discover which tutoring protocol gives the best results in teaching causal reasoning. I used machine learning method to analyze a set of human tutoring transcripts to discover how and when human tutors switch protocols. I analyzed the students' performance using CIRCSIM-Tutor and their responses to a questionnaire about their view of CIRCSIM-Tutor. The analysis results say that the students' protocol preference is closely related to their performance using CIRCSIM-Tutor whether students ask for more immediate feedback.

8.2 Significance of Research

CIRCSIM-Tutor System Version 2.8 has only eight procedures with no difference in difficulty level. So these procedures can be chosen in any order. However CIRCSIM-Tutor version 3.0 has 83 procedures that are classified into five Procedure Difficulty levels and four Procedure Description levels. This variety requires Curriculum Planning. The Curriculum Planning is based on the student input and assessments of the student's current knowledge. The goal of the Curriculum Planning is to use assessments and student input to make CIRCSIM-Tutor construct an individualized problem set for the student.

My personal work on multiple tutoring protocols is significant because the protocols differ mainly in when the tutoring should take place and whether feedback is immediate or not. My improved discourse planning model, which has multiple tutoring protocols, can emulate the human tutoring better so that it will give the best results in teaching causal reasoning.

8.3 Future Work

8.2.1 Experimental Analysis on the Multiple Tutoring Protocols. I have enabled CIRCSIM-Tutor to switch protocols. This will make it possible for the team to carry out a controlled experiment to find out which protocol improves the student's learning outcomes. If these research results suggest that different protocols suit different students better then CIRCSIM-Tutor will have a rich planning capability for adapting the protocol to the student's status.

8.3.2 Further Machine Learning Tests. My analysis using machine learning techniques provided some promising results, but much more complete data must be gathered and prepared for use in machine learning. For instance, we could test more human tutoring transcripts to determine when the tutors switch from one tutoring protocol to another.

8.3.3 Improvement of Planning Capability. In the real world classroom any student can potentially take control of the flow of a conversation. Mixed initiative means all participants have an ability to take the initiative in a dialogue. However, most typical ITS allow only system initiative or student initiative. The current CIRCSIM-Tutor can handle only a few simple student initiatives. Mixed initiative in an ITS means that the ITS allows the student to take the initiative. This is necessary if the system is to be able to respond to student requests. Mixed Initiative Interaction is a big, complex issue in any ITS. One of the hot issues in research on the mixed initiative dialogues is how to recognize when the student is taking the initiative as opposed to answering a question.

8.3.4 Accurate Assessment of Student Knowledge Status. Dynamic planning needs not only good planning rules but also accurate student assessment methods. We need to research student assessments based on various types of student assessment models.

BIBLIOGRAPHY

Aamodt, Agnar, and Plaza, Enric. 1993. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, (download:1999.Feb.3.) http://online.loyno.edu/cisa494/papers/Aamodt.html.

Allen, James, Ferguson, George, Miller, Bradford, and Ringger, Eric. 1995a. Spoken Dialogue and Interactive Planning, Proceedings of the ARPA Spoken Language Technology Workshop, Austin, TX. Jan. 1995.

Allen, James, Schubert, Lenhart, Ferguson, George, Heeman, Peter, Hwang, Chung Hee, Kato, Tsuneaki, Light, Marc, Martin, Nathaniel, Miller, Bradford, Poesio, Massimo, and Traum, David. 1995b. The TRAINS Project: A Case Study in Defining a Conversational Planning Agent. Journal of Experimental and Theoretical AI, 7(1995), pp. 7-48.

Alterman, Richard. 1990. An Adaptive Planner. In James Allen, James Hendler, and Austin Tate, (eds)., Readings in Planning, Morgan Kaufman Publishers Inc., San Mateo, CA, pp. 660-664.

Barrett, Anthony, Christianson, Dave, Friedman, Marc, Kwok, Chung, Golden, Keith, Penberthy, Scott, Sun, Ying, and Weld, Daniel. 1995. UCPOP User's Manual (Version 4.0). Technical Report 93-09-06d, November 6.

Beck, Joseph, Stern, Mia, and Haugsjaa, Erick. 1998. Applications of AI in Education. Crossroads. The ACM's First Electronic Publication. Last Modified: 1998. Oct. 3. http://www.acm.org/crossroads/xrds3-1/aied.html

Brandle, Stefan. 1998. Using Joint Actions to Explain Acknowledgements in Tutorial Discourse: Application to Intelligent Tutoring Systems, Ph. D. Dissertation. Chicago, IL: Illinois Institute of Technology.

Burns, Hugh, and Capps, Charles. 1988. Foundations of Intelligent Tutoring Systems: An Introduction, Foundations of Intelligent Tutoring Systems. Hillsdale, NJ: Lawrence Erlbaum Associates. pp. 1-19.

Cho, Byung-In, Michael, Joel, Rovick, Allen, and Evens, Martha. 1999. A Curriculum Planning Model for an Intelligent Tutoring System. Proceedings of 12th International Florida Artificial Intelligence Research Symposium. Orlando, FL. pp. 197–201.

Cho, Byung-In, Michael, Joel, Rovick, Allen, and Evens, Martha. 2000. An Analysis of Multiple Tutoring Protocols. Fifth International Conference on Intelligent Tutoring Systems (ITS '2000). Montreal, Canada. to appear.

Clancey, William. 1987. Methodology for Building Intelligent Tutoring Systems. In Greg P. Kearsley, (ed.), Artificial Intelligence & Instruction Applications and Methods, Reading, MA: Addison-Wesley. pp. 193-227.

Cohen, Robin, Allaby, Coralee, Cumbaa, Christian, Fitzgerald, Mark, Ho, Kinson, Hui, Bowen, Latulipe, Celine, Lu, Fletcher, Moussa, Nancy, Pooley, David, Qian, Alex, and Siddiqi, Saheem. 1998. What is Initiative? http://www.lpaig.uwaterloo.ca/~b2hui/mypapers.html

Cuffman, D. and MacRae, N. 1996. Faculty Development Programs in Interactive Television. Proceedings of the 1996 Mid-South Instructional Technology Conference. Track 2: Distance Learning. (download:1999.Dec.30.) http://www.mtsu.edu/~itconf/proceed96.html

Freedman, Reva. 1996a, Interaction of Discourse Planning, Instructional Planning and Dialogue Management in an Interactive Tutoring System, Ph.D. Dissertation, Computer Science Department, Evanston, IL: Northwestern University.

Freedman, Reva. 1996b, Using a Text Planner to Model the Behavior of Human Tutors in an ITS, Proceedings of the Seventh Midwest AI and Cognitive Science Conference, np, Bloomington, IN.

Freedman, Reva. 1997a. Degrees of Mixed-Initiative Interaction in an Intelligent Tutoring System. In Haller, S. and McRoy, S. (eds.), Working Notes of AAAI97 Spring Symposium on Mixed-Initiative Interaction, Stanford, CA. pp. 44-49. http://www.csam.iit.edu/~circsim/

Freedman, Reva. 1997b. Representing Communicative Action in a Dialogue-Based Intelligent Tutoring System. AAAI97 Fall Symposium on Communicative Action in Human and Machines, http://www.csam.iit.edu/~circsim/

Freedman, Reva, Zhou, Yujian, Glass, Michael, Kim, Jung Hee, and Evens, Martha. 1998. Using Rule Induction to Assist in Rule Construction for a Natural-Language Based Intelligent Tutoring System, Proceedings of the Twentieth Annual Conference of the Cognitive Science Society, Madison, WI, pp. 362-367, http://www.csam.iit.edu/~circsim/

Freedman, Reva. 1999. Atlas: A Plan Manager for Mixed-Initiative, Multimodal Dialogue. AAAI '99 Workshop on Mixed-Initiative Intelligence, Orlando. pp. 107-114.

Grama, Carmen, and Gonzalez, Avelino. 1998. Automated Generation of Plans through the Use of Context-Based Reasoning, Proceedings of the Eleventh International Florida Artificial Intelligence Research Symposium (FLAIRS-98), Sanibel Island, FL, pp. 7-11.

Guinn, Curry. 1996. Mechanisms for Mixed-Initiative Human–Computer Collaborative Discourse. Proceedings of ACL96. Santa Cruz, CA. http://www.cs.duke.edu/~cig/HomePage.html

Halff, Henry. 1988. Curriculum and Instruction in Automated Tutors, Foundations of Intelligent Tutoring Systems. Hillsdale, NJ: Lawrence Erlbaum Associates. pp 79-108.

Hammond, Kristian. 1986. CHEF: A Model of Case-Based Planning. AAAI 1986 Proceedings, pp. 261-271.

Harris, Billy, and Cook, Diane. 1998. Integrating Hierarchical and Analogical Planning, Proceedings of the Eleventh International Florida Artificial Intelligence Research Symposium (FLAIRS-98), Sanibel Island, FL, pp. 126-130.

Hume, Gregory. 1995. Using Student Modelling to Determine When and How to Hint in an Intelligent Tutoring System, Ph. D. Dissertation, Illinois Institute of Technology.

Hume, Gregory, Michael, Joel, Rovick, Allen, and Evens, Martha. 1996. Hinting as a Tactic in One-On-One Tutoring, Journal of the Learning Sciences, Vol. 5, No. 1, pp. 23-47.

Katz, Sandra, Lesgold, Alan, Eggan, Gary, Girdin, Maria, and Greenberg, Linda. 1992. Self-adjusting Curriculum Planning in Sherlock II, Lecture Notes in Computer Science: Proceedings of the Fourth International Conference on Computers in Learning (ICCAL '92). Berlin: Springer Verlag. pp. 343-355.

Khuwaja, Ramzan. 1994. A Model of Tutoring: Facilitating Knowledge Integration Using Multiple Models of the Domain, Ph. D. Dissertation, Illinois Institute of Technology.

Khuwaja, Ramzan, Rovick, Allen, Michael, Joel, and Evens, Martha. 1994. A Tale of Three Tutoring Protocols: The Implications for Intelligent Tutoring Systems. Proceedings of Golden West, Las Vegas, Nevada, June 9-12. pp. 109-118.

Kim, JungHee, Freedman, Reva, and Evens, Martha. 1998. Relationship Between Tutorial Goals and Sentence Structure in a Corpus of Tutoring Transcripts, Proceedings of the Ninth Midwest AI and Cognitive Science Conference, 124-131, Dayton OH.

Koehn, Gina, and Greer, Jim. 1994. Recognizing Plans in Instructional Systems Using Granularity. Proceedings of the International Conference on User Modelling, Hyannis, MA, August, 1994.

Leake, David. 1995. Combining Rules and Cases to Learn Case Adaptation (download:1999.Feb.3.) http://online.loyno.edu/cisa494/papers/Leake.html, Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society

Li, Jun, Seu, Jai, Evens, Martha, Michael, Joel, and Rovick Allen. 1992. Computer dialogue system (CDS): A System for Capturing Computer-Mediated Dialogue. Behavior Research Methods, Instruments, & Computers, 24(4): pp. 535-540.

Lochbaum, Karen. 1998. A Collaborative Planning Model of Intentional Structure. Computational Linguistics, Vol 24, No. 4. pp. 525-572

Martin, Joel, and VanLehn, Kurt. 1995. A Bayesian Approach to Cognitive Assessment, (download:1998.Aug.2.) http://www.pitt.edu/~vanlehn/distrib/Chipman95a-abstract.html

Michael, Joel, Rovick Allen, Evens, Martha, Shim, Leemseop, Woo, Chong, and Kim, Nakhoon. 1992. The Uses of Multiple Student Inputs in Modeling and Lesson Planning in CAI and ICAI Programs. In I. Tomek (editor), Computer Assisted Learning. Berlin: Springer Verlag. pp 441-452

Moore, M. 1989. Three Types of Interaction. American Journal of Distance Education. Volume 3, Number 2, pp.1-6.

Moore, M., and Kearsley, G. 1996. Distance Education: A Systems Perspective. Belmont, CA: Wadsworth.

Peot, Mark, and Smith, David. 1992. Conditional Nonlinear Planning. Proceedings of the First International Conference on Artificial Intelligence Planning Systems, College Park, MD.

Quinlan, J. 1993. C4.5: Programs for Machine Learning. Los Altos, CA: Morgan Kaufmann.

Rovick, Allen, and Michael, Joel. 1992. The Predictions Table: A Tool for Assessing Students' Knowledge. American Journal of Physiology 263 (Advances in Physiology Education 8): S33-S36.

RuleQuest. 1999. RuleQuest Research Pty Ltd. (download:1999.Sept.8.) http://www.rulequest.com/download.html

Russell, Stuart, and Norvig, Peter. 1995. Artificial Intelligence: A Modern Approach, Upper Saddle River, NJ: Prentice Hall

Sanders, Gregory. 1995. Generation of Explanations and Multi-Turn Discourse Structures in Tutorial Dialogue, Based on Transcript Analysis. Ph. D. Dissertation, Illinois Institute of Technology.

Sandoval, William, Daniszewski, Kenneth, Spillane, James, and Reiser, Brian. 1999. Teachers' Discourse Strategies for Supporting Learning Through Inquiry. Annual Meeting of the American Educational Research Association (AERA 1999). Montreal, Canada April 19-23, (download:2000.Apr.5.) http://www.ls.sesp.nwu.edu/bguile/papers.html

Schulze, K.G., Shelby, R.N., Treacy, D.J., and Wintersgill, M.C. 2000. Andes: A Coached Learning Environment for Classical Newtonian Physics. To appear in Proceedings of the 11th International Conference on College Teaching and Learning. Jacksonville, FL, April, 2000.

Self, John. 1990. Bypassing the Intractable Problem of Student Modelling. In Claude Frasson and Gilles Gauthier, (eds.), Intelligent Tutoring Systems: at the Crossroad of Artificial Intelligence and Education, Norwood, NJ:Ablex Publishing Corp., pp. 107-123.

Shah, Farhana. 1997. Recognizing and Responding to Student Plans in an Intelligent Tutoring System: CIRCSIM-Tutor, Ph. D. Dissertation, Illinois Institute of Technology.

Shah, Farhana, and Evens, Martha. 1997. Student Initiatives and Tutor Responses in a Medical Tutoring System. In Haller, S. and McRoy, S. (eds.), Working Notes of AAAI97 Spring Symposium on Mixed-Initiative Interaction, Stanford, CA. pp. 138-144. http://www.csam.iit.edu/~circsim/

Shah, Farhana, Michael, Joel, Rovick, Allen, and Evens, Martha. 2000. Classifying Student Initiatives and Tutor Responses in Human Keyboard-to-Keyboard Tutoring Sessions. Submitted to Discourse Processes.

Shim, LeemSeop. 1991. Student Modelling for an Intelligent Tutoring System: Based on the Analysis of Human Tutoring Sessions, Ph. D. Dissertation, Illinois Institute of Technology.

Smith, Ronnie. 1997a. Practical Issues in Mixed-Initiative Natural Language Dialog: An Experimental Perspective. Proceedings of the 1997 AAAI Spring Symposium on Computational Models for Mixed Initiative Interaction, pp. 158-162

Smith, Ronnie. 1997b. Performance Measures for the Next Generation of Spoken Natural Language Dialog Systems. Proceedings of the ACL'97 Workshop on Interactive Spoken Dialog Systems, pp. 37-40

Sycara, Katia. 1988. Using Case-Based Reasoning for Plan Adaptation and Repair, (download:1999.Feb.3.) http://online.loyno.edu/cisa494/papers/Sycara.html

Tabak, Iris and Reiser, Brian. 1999. Steering the Course of Dialogue in Inquiry-based Science Classrooms. Annual Meeting of the American Educational Research Association (AERA 1999), April 1999. Montreal Canada. (download:2000.Apr.5.) http://www.ls.sesp.nwu.edu/bguile/papers.html

Travers, A. and Decker, E. 1999. New Technology and Critical Pedagogy. Radical Pedagogy, Volume 1: Issue 2, Summer 1999.

VanLehn, Kurt. 1988. Student Modeling, In M. Polson, (ed.), Foundations of Intelligent Tutoring Systems. Hillsdale, NJ: Lawrence Erlbaum Associates. pp 55-78.

Veloso, Manuela, and Blythe, Jim. 1994. Linkability: Examining Causal Link Commitments in Partial-Order Planning. Proceedings of the Second International Conference on AI Planning Systems. pp. 170-175.

Weber, Gerhard and Specht, Marcus. 1997. User Modeling and Adaptive Navigation Support in WWW-Based Tutoring Systems, Proceedings of the 6th International Conference on User Modeling. Sardinia, Italy. pp. 289-300.

Wilensky, Robert. 1983. Planning and Understanding, Reading MA: Addison-Wesley. pp. 1-17.

Wilensky, Robert, Chin, David, Luria, Marc, Martin, James, Mayfield, James and Wu, Dekai. 1989. The Berkely UNIX Consultant Project (CSD-89-520). (download:1998.Jun.4.) http://sunsite.berkeley.edu:80/Dienst/UI/2.0/Describe/ncstrl.ucb%2fCSD-89-520?abstract.

Woo, Chong. 1991. Instructional Planning in an Intelligent Tutoring System: Combining Global Lesson Plans with Local Discourse Control, Ph. D. Dissertation, Illinois Institute of Technology.

Woolf, Beverly. 1984. Context-Dependent Planning in a Machine Tutor. Ph.D. Thesis. University of Massachusetts at Amherst, Amherst, MA. COINS Technical Report 84-21.

Yang, Feng-Jen, Kim, JungHee, Glass, Michael, and Evens, Martha. 2000a, Lexical Usage in the Tutoring Schema of CIRCSIM-Tutor: Analysis of Variable References and Discourse Markers, Proceedings of Human Interaction with Complex Systems (HICS-2000), Urbana, IL, pp. 27-31.

Yang, Feng-Jen, Kim, JungHee, Glass, Michael, and Evens, Martha. 2000b, Turn Planning in CIRCSIM-Tutor, Proceedings of Florida Artificial Intelligence Research Symposium, Orlando, FL, pp. 60-64.

Yang, Qiang. 1997. Intelligent Planning: A Decomposition and Abstraction Based Approach (Artificial Intelligent). Springer-Verlag Berlin Heidelberg. pp. 163-188.

Zhou, Yujian, Freedman, Reva, Glass, Michael, Michael, Joel, Rovick, Allen, and Evens, Martha. 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, pp. 187-191.

Zhou, Yujian, Freedman, Reva, Glass, Michael, Michael, Joel, Rovick, Allen, and Evens, Martha. 1999b. Delivering Hints in a Dialogue-Based Intelligent Tutoring System. Proceedings of the 16th National Conference on Artificial Intelligence (AAAI–99), Orlando, FL, pp. 128-134.

Zhou, Yujian. and Evens, Martha. 1999c. A Practical Student Model in an Intelligent Tutoring System. Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence. Chicago, IL, pp. 13-18.