TURN PLANNING FOR A DIALOGUE-BASED INTELLIGENT TUTORING SYSTEM

BY

FENG-JEN YANG

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 2001

ACKNOWLEDGMENTS

I would like to thank my mentor, Professor Martha W. Evens, so much for her patience and guidance. I will never forget that she is always there whenever I need help. I also thank all of the CIRCSIM-Tutor project members and our domain experts at Rush Medical College. They have been so helpful with research and so friendly in life. I really appreciate all of the wonderful time to work with them. Besides, I express my deepest love to my parents, sisters and brothers. They have been doing whatever they could to support and encourage me so that I could keep going for my objective. With all of the help, I have done this research smoothly.

The CIRCSIM-Tutor Project has been 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.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

CIRCSIM-Tutor is an intelligent tutoring system using a natural language interface to tutor medical students on problem-solving in the domain of reflex control of blood pressure. This negative feedback system is a difficult topic for most first-year medical students. With this concern in mind, our domain experts at Rush Medical College argued for the necessity of developing an intelligent tutoring system as an assistant outside the classroom capable of using language to help students understand this topic.

The baroreceptor reflex is the mechanism in charge of regulating blood pressure in the human body so that it will not go beyond the tolerable range. If something happens to change the blood pressure, such as a transfusion, hemorrhage or pacemaker malfunction, the baroreceptor reflex will attempt to regulate the blood pressure in a negative feedback manner so the blood pressure will go back to a stable state again.

While using this system the student is presented with a predefined procedure and then is asked to predict the qualitative changes in seven core variables at three different chronological stages of the reflex cycle. These predictions are then used as the basis of a tutoring dialogue to remediate any misconception that the student has revealed.

1.1 The Evolution of CIRCSIM-Tutor

CIRCSIM-Tutor is closely related to the other Computer Aided Instruction systems developed at the Department of Physiology at Rush Medical College. Work began in 1983 when Rovick and Brenner developed the HEARTSIM system on PLATO to help students understand the regulation of blood pressure [Rovick and Brenner 1983]. As an extension of

HEARTSIM, in 1986, Joel Michael and Allen Rovick designed CIRCSIM on DOS to teach the physiological scenario of blood pressure regulation in more detail [Rovick and Michael 1986]. In order to overcome the rigid tutoring plans hardwired into CIRCSIM, Michael and Rovick at Rush Medical and Martha Evens at the Illinois Institute of Technology proposed a joint project to develop an adaptive tutoring system called CIRCSIM-Tutor. A prototype version of CIRCSIM-Tutor was implemented in Prolog by Kim [1989]. Version 2 was implemented in LISP by [Lee 1990, Shim 1991, Woo 1991, Zhang 1991, Seu 1992, Evens et al. 2001]. The implementation of Version 3 is still in progress. My contribution is to provide some additional modules to improve the discourse and the surface text generation.

1.2 Modeling the Baroreceptor Reflex

The behavior of the baroreceptor reflex can be described by a qualitative model using seven core variables over three stages. The seven core variables as they appear in the prediction table are Central Venous Pressure (CVP), Inotropic State (IS), Stroke Volume (SV), Heart Rate (HR), Cardiac Output (CO), Total Peripheral Resistance (TPR) and Mean Arterial Pressure (MAP). The three stages in the order of occurrence are the Direct Response (DR) Stage, which is the time immediately after the perturbation and before the reflex is activated, the Reflex Response (RR) Stage, when the changes caused by the baroreceptor reflex begin to take effect, and the Steady State (SS) Stage, the time after restabilization. The causal relationships between these variables can be modeled by a concept map as shown in Figure 1. Besides showing the qualitative influence among variables, Figure 1 also takes the baroreceptor and the nervous system into consideration so that this concept map is applicable to all of the three stages.

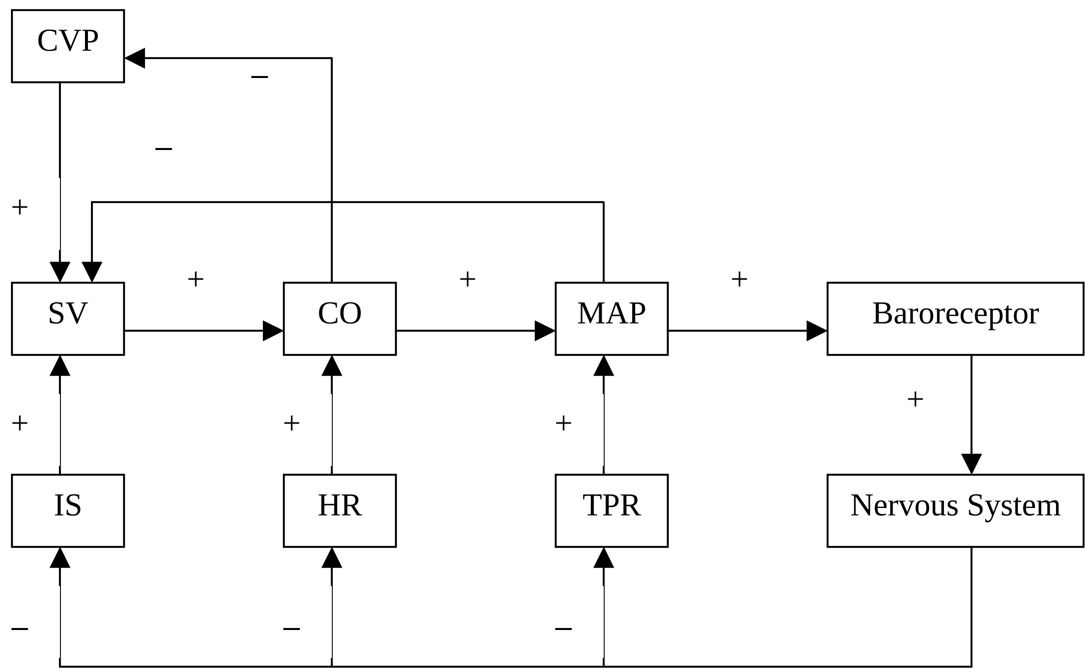

Figure 1. The Concept Map

The concept map represents the underlying knowledge that our colleagues at Rush Medical College want their students to internalize. Within the map each parameter is represented as a text box. The direction of arrows represents the causal relationships between parameters from the cause side to the effect side. Furthermore, each arrow is accompanied with a plus sign or minus sign representing a direct relationship or an inverse relationship respectively. So an arrow with a plus sign from parameter one to parameter two means that increasing parameter one results in increasing parameter two, while decreasing parameter one results in decreasing parameter two. Similarly, an arrow with a minus sign from parameter one to parameter two means that increasing parameter one results in decreasing parameter two, while decreasing parameter one results in increasing parameter two. For example, increasing the Cardiac Output (CO) results in decreasing the

Central Venous Pressure (CVP), but increasing the Cardiac Output (CO) results in increasing the Mean Arterial Pressure (MAP).

It is possible for a parameter to have two determinants. In such cases, we need to think about which determinant is stronger, since we are thinking about qualitative changes. The change in the stronger determinant will dominate the total qualitative change, even though the other determinant may change in the opposite direction. For example, the Stroke Volume (SV) has two determinants, the Central Venous Pressure (CVP) and the Inotropic State (IS), but the Inotropic State (IS) is stronger than the Central Venous Pressure (CVP). So if we have the Central Venous Pressure (CVP) decreased but the Inotropic State (IS) increased, the increase in the Inotopic State (IS) is stronger than the decrease in the Central Venous Pressure (CVP) and the Stroke Volume (SV) will still increase.

1.3 Using CIRCSIM-Tutor

After a user logs in to CIRCSIM-Tutor, the system shows the tutoring screen, which consists of the procedure description window, the dialog window, and the prediction table. The procedure description window displays a predefined perturbation to the student. The student is then asked to predict the qualitative changes of the seven core variables in the first stage (DR). The prediction table is used to store the predictions made by the student. Based on the student’s predictions, the system will conduct a natural language dialogue to remediate the student’s misconceptions. This interactive conversation between the student and the machine tutor is carried out in the dialog box. This interface with a sample prediction and dialogue is shown in Figure 2. Once this dialogue is over, the system asks for predictions for the RR Stage and then again begins a remedial dialogue. This cycle is repeated a third time for the SS Stage.

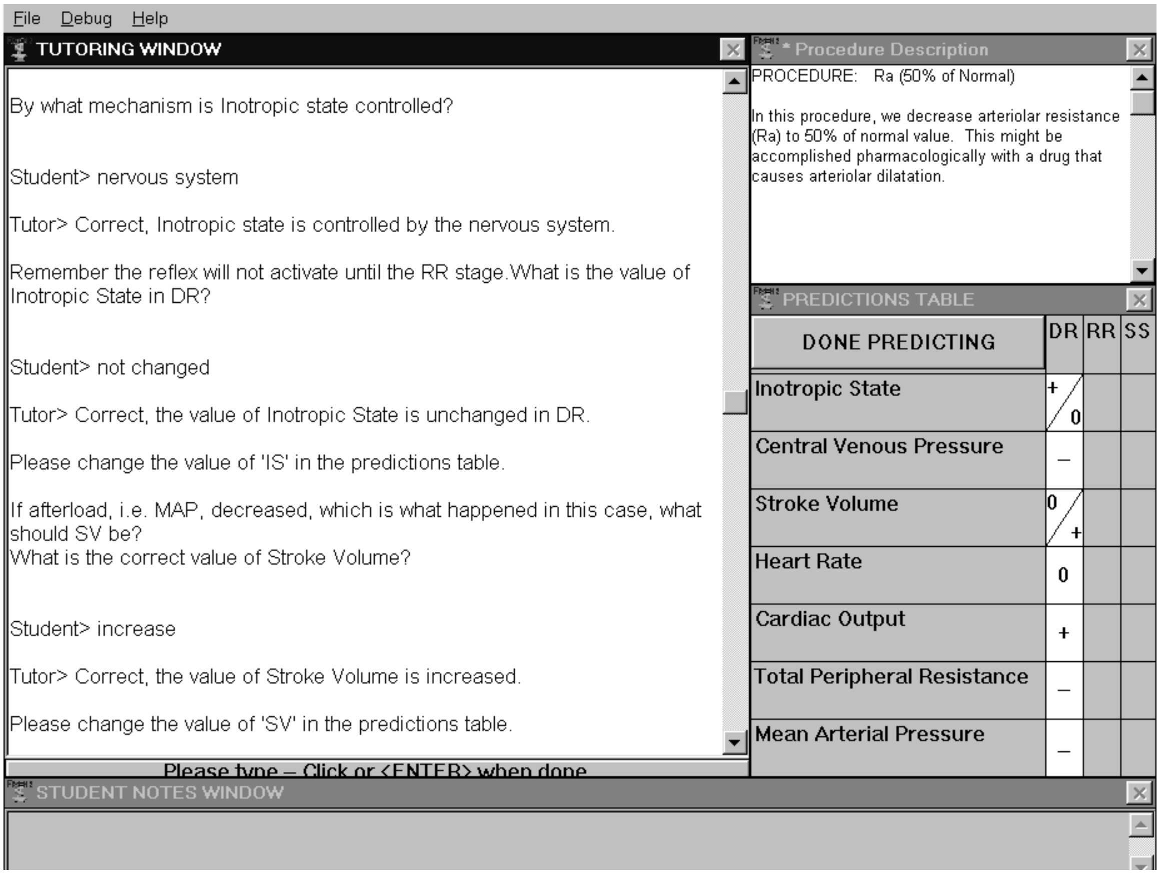

Figure 2. The User Interface

Our tutorial dialogues are based on the keyboard-to-keyboard sessions carried out by our domain experts and their first-year medical students. These dialogues largely consist of segments devoted to single variables and the relationships with other variables. The tutor introduces a variable for discussion, and the tutor and student discuss it until the student produces a correct prediction. The purpose of this research is to discover a method of planning the tutorial turns of CIRCSIM-Tutor, so that we can generate a turn-taking dialogue that is similar to a human tutoring session.

1.4 Why is Turn Planning Necessary?

If the student is to benefit from a natural language interface, the tutoring system must be provided with the properties that make human natural language so effective [Moser and Moore 1995]. To this end, CIRCSIM-Tutor tries to imitate the human tutor's language as much as possible.

Like most natural language generation systems, the current version of CIRCSIM-Tutor has a discourse planner to produce a discourse plan that specifies both the content and overall structure of a dialogue session. In terms of determining the deep structure, knowing the content and structure of a dialogue is enough and the discourse planner has been doing a good job. However, to make a dialogue fluent and coherent, knowing only the deep structure is far from enough. There is still a considerable range of refinements to be made, before feeding a discourse plan to a surface text generator. The discourse planner leaves open a certain number of decisions about the surface form of the text to be generated.

By showing examples of incoherent turns, Freedman [1996a] introduced the necessity of turn planning to the CIRCSIM-Tutor project. The current version of CIRCSIM-Tutor does not have this level of planning. After receiving a discourse plan from the discourse planner, it plans the tutor's utterances as individual sentences. Each sentence realizes some tutorial or dialogue goal, such as providing an acknowledgment of the student's answer, giving a hint, or asking the next question.

We have consulted transcripts of human tutors for guidance in the planning process. The human tutoring transcripts are the source of the rules that determine the machine's decisions on issues, such as what to teach next and how to teach it, how to adjust the tutoring to various student responses, what topics should be elicited from the student and what topics should be ignored, and the language of individual sentences. However, the dialogue is generated a sentence at a time. Without considering issues beyond the sentence level, it still generates comprehensible dialogue but it often sounds unnatural.

After looking into this issue, we decided to add a turn planner to CIRCSIM-Tutor to improve the rhetorical structure of our machine dialogue, so that it will sound more fluent and more coherent with the evolving discourse focus.

1.5 The Incoherence of the Current Version

An evaluation of CIRCSIM-Tutor by 50 students at Rush Medical College demonstrated that in the absence of a turn planner its dialogue is already comprehensible, but it is stiff and repetitive. The students may become bored and lose their interest in learning the subject. The problem is that the machine dialogue does not have the fluency and coherence provided by human tutors.

Figure 3 is an example dialogue taken from the current version of CIRCSIM-Tutor, which reveals the lack of fluency and coherence in our machine dialogues. For example the two utterances in the turn T3 are generated by two discourse plans. The first utterance is based on the evaluation of the student's input in the turn S2. In the second utterance of T3 the variable Inotropic State should be pronominalized, but the antecedent is generated by another plan.

T1:	By what mechanism is Inotropic State controlled?
S2:	nervous system
T3:	Correct, Inotropic State is controlled by the nervous system. What is the value of Inotropic State in DR?
S4:	decreased
T5:	Nope, the value of Inotropic State is not decreased in DR. Remember. Inotropic state is neurally controlled. What is the value of Inotropic State in DR?
S6:	not changed
T7:	Correct, the value of Inotropic State is unchanged in DR. Please change the value of 'IS' in the prediction table.

Figure 3. An Example Dialogue before Turn Planning

1.6 Some Discourse Features Handled by the Turn Planner

One of the important areas of research in computational discourse is finding out what information is contained in the sequence of utterances but goes beyond the meaning of individual utterances themselves [Grosz 1997]. Our goal is to make the machine dialogue fluent and coherent. We therefore have some range of options in deciding which discourse features to work on. Some features have been chosen to be handled by the turn planner to make the dialogue in Figure 3 sound more natural and more fluent.

To illustrate each feature and the corresponding improvement, the related dialogue turns will be chosen and improved step by step as we discuss each feature.

- Softeners

  The machine tutor always asks questions directly, but human tutors tend to ask questions politely instead of directly. They frequently use various politeness locutions, such as *Can you tell me* … or *Do you know* …, before the question. So, turn T1 can be improved with a softener:

  T1: *Can you tell me* by what mechanism Inotropic State is controlled?

- Abbreviated Variable Names

  The machine tutor always spells out variable names while asking questions and giving acknowledgments, but human tutors usually use abbreviated names such as IS, HR, TPR, etc. Since they sometimes do not abbreviate, and we believe there may be reasons for occasionally preferring the spelled-out form, the turn planner will have to make this decision [Yang et al. 2000b]. So, turns T3 and T5 can be improved to read:

  T3: Correct, *IS* is controlled by the nervous system.

  What is the value of *IS* in DR?

  S4: decreased

  T5: Nope, the value of *IS* is not decreased in DR.

  Remember, *IS* is neurally controlled.

  What is the value of *IS* in DR?

- Discourse Markers

  Using discourse markers can make clear the logical relationship between two sentences or the relationship of a sentence to the tutorial goal structure. In turns T3 and T5, the two consecutive utterances belong to different topics, but the machine tutor does not use discourse markers, while human tutors tend to use discourse markers like *so*, *now*, *but*, *therefore*, etc., to move from one discourse focus to another. So, turns T3 and T5 can be further improved to read:

T3: Correct, *IS* is controlled by the nervous system.

So, what is the value of *IS* in DR?

T5: Nope, the value of *IS* is not decreased in DR.

Remember, *IS* is neurally controlled.

Now, what is the value of *IS* in DR?

- Acknowledgments

In turns T3 and T5, the acknowledgments are both explicit and content-based. The sentences sound redundant. In human dialogues, acknowledgments following student answers are often reduced to a single word, appended to the next sentence, or omitted entirely [Brandle 1998, Spitkovsky and Evens 1993]. Whether an acknowledgment can be reduced and attached to the next sentence depends on the syntax of the next sentence, the relation of the next sentence to the answer being acknowledged, and whether the answer is correct or not. Usually correct answers are not repeated. Since at this point the discourse planner generates an acknowledgment separately from the succeeding utterance, it does not know whether such reduction is possible. So turns T3 and T5 can be even further improved to read:

T3: *Right*.

So, what is the value of *IS* in DR?

S4: decreased

T5: *No*, the value of *IS* is not decreased in DR.

Remember, *IS* is neurally controlled.

Now, what is the value of *IS* in DR?

A special phenomenon of acknowledging the student's answer is that human tutors tend to acknowledge the student's finally correct answer more strongly than usual, especially when the student has made some mistakes and finally got the correct answer. So, turn T7 can improved to read:

T7: *Very good*.

- Pronouns

In turn T5, the intended variable name has been mentioned in the previous turn. In this case, human tutors tend to use the pronoun *it* to refer to the variable previously mentioned. So, the turn T5 can be improved to read:

T5: *No*, *IS* is not decreased in DR.

Remember, *it* is neurally controlled.

Now, what is the value of *IS* in DR?

Generally speaking, these choices are instances of lexical selection. Since we are using schemata as planning operators, an efficient way of learning the rules for lexical selection is by searching for examples of lexical usage in transcripts marked up with tutoring schemata. We search for instances of the same schema expressed in different ways. After further in-depth analysis of these instances, we can establish rules for lexical selection.

Addressing only the five discourse features discussed above, the dialogue in Figure 3 can be transformed into Figure 4.

T1: *Can you tell me* by what mechanism Inotropic State is controlled?

S2: nervous system

T3: *Right*.

So, what is the value of *IS* in DR?

S4: decreased

T5: *No*, *IS* is not decreased in DR.

Remember, *it* is neurally controlled.

Now, what is the value of *IS* in DR?

S6: not changed

T7: *Very good*.

Figure 4. An Example Dialogue after Turn Planning

The obvious difference between Figure 3 and Figure 4 makes us believe that by adding a component that pre-processes the sentences within a single dialogue turn we can improve the rhetorical structure of our tutorial turns. This justification is also consistent with the necessity of separate paragraph planning [Hovy 1990, Mann and Moore 1981].

1.7 Research Goals

The purpose of this research is to build a turn planner as a new level of discourse planning in the current version of CIRCSIM-Tutor, so that, in the new version we can generate the structure of a tutorial turn as an integral whole, not just a sentence at a time. With the whole turn in the buffer, it is easier to consider issues beyond the sentence level. This will make the surface sentence generation easier, more natural and more fluent.

1.8 The Organization of This Thesis

Chapter 1 contains a justification for doing turn planning. Chapter 2 discusses some previous related research ranging from planning problems and discourse analyses to evaluation methods. Chapter 3 discusses the tutorial dialogue analysis in our domain and explains how the turn planner fits into the CIRCSIM-Tutor context. Chapter 4 shows the architecture of the turn planner and illustrates the turn generation process and the lexical selection process. Chapter 5 describes my method of lexical analysis and the derivation of lexical rules used by the turn planner. Chapter 6 proposes a method of evaluating the turn planner. Chapter 7 states my conclusion and describes some directions for future research.

CHAPTER II

RELATED RESEARCH

Planning discourse at the abstract level of a turn is not well-studied in natural language dialogue systems, since dialogue systems are very new. Most work on text generation has involved the planning of expository text, not dialogue. So far, we have not found any existing system that has specifically planned discourse at the turn level. Of course, there will be a trade-off between better text generation and doing an additional level of discourse planning. The justification for adding this level of planning is to have more fluent and coherent tutorial dialogue.

Since the idea is new, it is hard to find directly related previous research. Nevertheless, some methodologies from other discourse levels are applicable. Here, I have organized the related research in three parts, planning problems, approaches to discourse analysis, and evaluation methods.

2.1 Planning Problems

CIRCSIM-Tutor plans discourse to achieve a tutorial goal, and then plans additional dialogue interactively according to the student's response [Woo 1991, Freedman 1996b]. It also formulates the discourse plan in advance and then executes it incrementally. Thus, during turn planning, a pre-formulated discourse plan is often being executed partially before it completes. It can be assumed that this pre-formulated plan will take care of the scenario smoothly. The hard part is that the student's response is unpredictable; there is no guarantee that it will go along with what we have planned. To be adaptive, the turn planner has to be prepared to repair the partial plan at the same time according to the student's feedback. These two features make the turn planning job closely related to both opportunistic planning and incremental planning.

2.1.1 Opportunistic Planning. An opportunity is a possibility subject to a specific combination of circumstances [Mellish et al. 1998]. This combination of circumstances makes the next step unexpected and unpredictable. An efficient way to plan under such uncertainty is to choose simple plans and adapt them whenever unpredicted circumstances are encountered.

Our approach to turn planning assumes an environment in which the dialogue planning is opportunistic because the student's response is unexpected and there is no way that we can predict it. So the next tutorial turn has to be planned on the basis of the combination of circumstances including the student's previous progress and current response.

Two systems in this field with planning ideas applicable to the turn planner are described in the following sections.

2.1.1.1 ILEX. ILEX. (Intelligent Labeling Explorer) is an opportunistic text generation system used to generate a dialogue between a museum curator and a visitor [Mellish et al. 1998]. It searches for items in a museum gallery automatically and generates a sequence of descriptions to reflect the interests of the visitor. To be efficient and prevent redundancy, at any discourse point, the text plan consists of facts that the system knows but that have not been conveyed to the user.

In order to plan a commentary turn describing a proper sequence of facts, the domain knowledge is organized into a three-tiered structure called content potential consisting of *the entities*, *the facts* and *the relations*. The tiers are further interconnected according to thematic and rhetorical relations. When a fact has been selected, the text plan can follow the entity-based moves or the relation-based moves to select the next fact, and then generate the description of those facts as a turn.

The following two examples illustrating an entity-based move and a relation-based move respectively are taken from [Mellish et al. 1998]. Example 1 uses *Jessie M. King* as the related entity and generates a turn, while Example 2 uses the example relation between the nucleus and satellite to generate another turn. The example relation comes from Rhetorical Structure Theory (RST). It provides a good way to put related utterances into a coherent turn [Mann and Thompson 1988].

Example 1:

This jewel was designed by *Jessie M. King*.

King worked in London.

Example 2:

Arts and Craft jewelry tends to be elaborate. (*Nucleus*)

For instance, this jewel has floral motifs. (*Satellite*)

Once the new fact has been selected this fact may act as the starting point for choosing its next fact.

If we treat the facts as the primitive dialogue acts of CIRCSIM-Tutor, the similarity of ILEX to turn planning is that the choice of the next primitive is always dependent on the combination of all the previous primitives and the current primitive. The only difference is that the discourse of CIRCSIM-Tutor is organized as a set of hierarchical schemata while the discourse of ILEX is determined by entity-based or relation-based moves.

2.1.1.2 PARETO. PARETO (Planning and Acting in Realistic Environments by Thinking about Opportunities) is an opportunistic planner used to simulate the behavior of a robot delivery vehicle [Pryor and Collins 1994]. In order to adapt to the unexpected environment, when PARETO is about to carry out a new goal, it searches its library of sketchy plans and chooses one that will achieve the goal. Each sketchy plan represents a possible method of achieving a certain goal and each goal may consist of several subgoals. At this moment, the efficiency of a sketchy plan is not an issue of concern, since the environment to be encountered is still beyond prediction. After choosing a sketchy plan, PARETO adds the new task to its task agenda and executes it. If everything goes as planned, then the task will be marked as successful and removed from the agenda. Otherwise, PARETO can either discard the current task and choose another sketchy plan, or repeat some primitive actions until the task is finally successful and removed. So, when plans are executed, the task agenda may hold tasks with different levels of abstraction.

If we think of the sketchy plans as the schemata of CIRCSIM-Tutor, the similarity between PARETO and turn planning is that the discourse plan of CIRCSIM-Tutor consists of schemata at different discourse levels while the task agenda of PARETO holds tasks at different levels of abstraction. Also, the replanning or repeating strategy of PARETO is similar to the adaptive discourse strategy of CIRCSIM-Tutor.

2.1.2 Incremental Planning. As the name indicates, incremental planning uses planning rules to incrementally expand goals into subgoals. It assumes that achieving one subgoal will not destroy the effects of other subgoals. So, the original goal can be carried out incrementally.

The actions of incremental planning are executed as soon as they are planned. Once the actions are executed, there is no way to backtrack or search. Replanning happens whenever the user fails to follow the plan. Since no backtracking is allowed, the later planning decisions can not influence earlier ones. In this manner, the system is actually trying possible solutions to fulfill a given goal. So, an optimal solution cannot be guaranteed even though there may be one [Cawsey 1992]. Cawsey's system in this field has planning ideas applicable to our turn planner. It is described in the following section.

2.1.2.1 EDGE. EDGE (the Explanatory Discourse GEnerator) is an explanation generation system used to generate explanatory dialogue for electronic circuits [Cawsey 1992]. Since these explanations are interactive, assumptions about the user's background and the current focus may change during the process of the explanation. So, too much detailed planning may be unnecessary and redundant. An important planning idea of the EDGE system is not to commit to the details of the explanation before it has to.

The discourse planning in the EDGE system proceeds incrementally. The primitive action is executed as soon as it is planned. In order to avoid redundancy, the planning expands a goal into subgoals in a depth-first hierarchical manner. When the system is given a topic to explain, it places this topic on the agenda. As the planning proceeds, it selects a goal from the agenda and executes it, if it corresponds to a primitive action. Otherwise, it selects a planning rule to satisfy this goal and expands some subgoals on the agenda according to this rule. After all subgoals are satisfied in a given order, the original goal is satisfied as well.

The similarity of EDGE to turn planning is that the discourse plan is carried out in a depth-first hierarchical manner. Once the turn planner has accumulated enough dialogue primitives, it generates a tutorial turn right away. In this manner, our turn planner is fulfilling a tutorial goal by incrementally satisfying its subgoals.

2.1.3 Combining Opportunistic and Incremental Planning. Two of the most important planning methodologies applicable to the CIRCSIM-Tutor domain are opportunistic planning and incremental planning. The turn planner adopts opportunistic strategies to plan the next tutorial turn according to the student's response and uses incremental strategies both to accumulate the dialogue primitives within a tutorial turn and to carry out the pedagogical goals incrementally [Yang et al. 2000b].

2.2 Discourse Analysis

Planning discourse for expository text has one thing in common with dialogue planning: the discourse must be divided into segments. For dialogue systems a discourse segment is a chunk of utterances that the speaker uses to show some intention or to convey some information. To be considered as a discourse segment, a span of utterances must have a recognizable purpose. The meaning of a dialogue segment is an aggregation of individual utterances that may be understood according either to informational or intentional relationships among utterances.

Different researchers have identified many different factors in discourse analysis, such as attention, intention, initiative, rhetorical structure, story trees, and turn-taking behavior [Nakatani and Traum 1998]. Many methods have been proposed for analyzing the local discourse context. The most popular method is annotating a corpus of the type of discourse that you wish to generate. A set of general instructions for annotating discourse segments and identifying the purposes of discourse segments was proposed by [Nakatani et al. 1995]. By investigating the relationship between reference and segmentation, Passonneau [1994] designed a protocol for coding discourse referential noun phrases and their antecedents. Other researchers such as Allen and Core [1997], Nakatani and Traum [1998] and Brennan and Clark [1996] have also suggested methods for exploring lexical issues.

Two discourse segmentation ideas in different domains are described in the following sections.

2.2.1 SHERLOCK. The SHERLOCK system is a computer aided fault diagnosis system to determine the location of a power distribution fault in a ring network by analyzing some specified symptoms of faults [Wong et al. 1988].

In the SHERLOCK domain, the explanation consists of a student-tutor turn-taking dialogue. The transcripts of tutorial explanations are annotated with the following discourse elements [Moore et al. 1996] based on the Rhetorical Structure Theory of Mann and Thompson [1988]:

- Segment

  Typically each tutor's explanation is a segment, which answers the student's immediately previous question.

- Core

  Almost always, a segment will have at least one constituent that most directly expresses the purpose of the segment. This constituent is called the core of the segment.

- Contributor

  In addition to the core, a segment may also have some contributors, which help to achieve the purpose expressed by the core of the segment.

- Intentional relation

  An intentional relation between core and contributor describes what the speaker is trying to accomplish by including the contributor in addition to the core.

- Informational relation

  An informational relation describes how the content of the core and its contributors are related in the domain.

- Minimal unit

  Segments that have no contributors are minimal units, because they do not have further intentional structure.

2.2.2 TRAINS. The TRAINS system was built to discuss the efficient routes for trains in the Northeastern United States [Allen et al. 1995]. In the TRAINS domain, the discourse structure is annotated with the following discourse elements [Nakatani and Traum 1998]:

- Token

  The dialogues are split into utterance-tokens based on prosody and grammar. Intuitively, a token corresponds to a single intonational phrase or a single grammatical clause.

- Common ground unit (CGU)

A CGU clusters distinct tokens to achieve a mutual understanding. It concentrates on establishing what is being said at the level of information exchange.

- I-unit (IU)

  Here the "I" stands for either *informational* or *intentional*. The relationships among CGUs, can be used to group CGUs into a hierarchical topic-structure or planning-based structure called IU trees.

The marking of CGUs is a good way of getting the level of commonality between participants in dialogue. Also the marking of IUs provides a good way of identifying the hierarchical purpose/topic structure. Overall, this annotation is good at marking the mixed-initiative interaction between the user and the dialogue system. This is especially worthy of attention when the task is planning discourse for dialogue systems allowing user initiatives.

2.3 Evaluation

The behavior of a natural language dialogue system is a series of complex interactions. This makes it hard to evaluate its performance. Even the comparison of alternative systems in similar domains is virtually impossible [Fraser 1997]. Nonetheless, the evaluation of natural language systems still plays a critical role in guiding and focusing research in computational linguistics. It challenges researchers in both building advanced systems and solving hard problems.

In the past decade, some conferences and workshops, such as Message Understanding Conferences (MUCs), Spoken Language Technology Workshops, and Machine Translation Workshops, have been focused on the evaluation of natural language systems. In these workshops, three typical types of evaluation are broadly used to evaluate systems for three different purposes [Hirschman and Thompson 1997]. They are:

- Adequacy Evaluation

  The fitness of a system for a special purpose is one of the critical factors in bringing natural language systems to market. For potential users, they have to know if the products on offer in a given application domain are suitable for their particular tasks or not. If so, they have to consider further tradeoffs between fitness and cost and then choose the most suitable one.

- Diagnostic Evaluation

  For systems where the coverage is important, the developers or end-users usually construct a large test suite to cover all of the elementary linguistic phenomena and their important combinations in the input domain. By testing systems with a large test suite, they can generate diagnostic profiles. The typical systems using this evaluation are machine translation and natural language understanding systems.

- Performance Evaluation

  Most of the ideas about quantitative performance evaluations are imported from information retrieval. There are three aspects to performance evaluation. The first is *Criterion*, which addresses what to evaluate such as *precision*, *speed* and *error rate*. The second is *Measure*, which specifies the property to report in order to get the chosen criterion such as *ratio of hits to hits plus misses*, *seconds to process*, and *incorrect percentage*. The third is *Method*, which is used to determine the appropriate value for a given measure such as the *analysis of system behavior over benchmark tasks*. In natural language systems, the approaches provide a useful way for system developers to compare different implementations of a technology or different versions of the same implementation.

So far, there is no established standard for evaluating the performance of natural language dialogue systems. All of the workshops and conferences have reiterated the importance of evaluation, but failed to reach any agreement on how to do it. Although some evaluation methods have been proposed, most of them are quite domain dependent and also inconclusive.

Some examples of evaluations of natural language dialogue systems are described in the following sections.

2.3.1 JUPITER. The JUPITER system is a telephone-based conversational system used to provide world-wide weather information over the telephone [Zhu et al. 1997]. In the JUPITER domain, two types of evaluation have been adopted for the spoken language system. The research group proposed a suite of metrics to evaluate their system as follows [Polifroni et al. 1998]:

- Word/sentence accuracy

  This metric is used in evaluating the Speech Recognizer.

- Parse coverage

  This metric is used in evaluating the Parser.

- Phrase comparisons

  This metric is used in the evaluation of Content Understanding and Generation.

- Understanding score

This metric is used in the evaluation of the Recognizer, Parser and Discourse Planning.

- Static database assessment

  This metric is used in the evaluation of Understanding, Discourse planing, Dialogue, Database Access and Generation.

- Logfile Evaluation

  This metric is used in the evaluation of Recognition, Understanding, Discourse Planning, Dialogue, Database Access and Generation.

On one hand, this suite of metrics provides a good assessment of the system behavior by examining each query/response pair. On the other hand it also examines the behavior of each part of the system and shows how well each performs separately.

2.3.2 EAGLE. EAGLE (Expert Advisory Group on Linguistic Computing) is a newly launched project trying to coordinate the European efforts of both academic and industrial participants toward the creation of de facto standards for corpora, lexicons, speech data, evaluations, and formalisms. As part of the work of the EAGLE project, the research group proposes a simple and practical reporting framework for spoken dialogue systems. This approach defines three sets of parameters and specifies the range of their possible values [Fraser 1997].

The first set belongs to system metrics that are used to characterize the basic features of the spoken dialogue system to be evaluated, such as:

- Input type

This parameter characterizes the way user's dialogue is input to the system. The possible values are *Speech*, *Text*, *Pulse* and *Other*.

- Input vocabulary

  The system's overall vocabulary size should be indicated.

- Input perplexity

  The perplexity is a doubt while recognizing the input. This parameter lists the average perplexity of the recognition vocabulary.

- Output type

  This parameter characterizes the system's output to the user. The possible values are *Speech*, *Text* and *Other*.

- Dialogue type

  This parameter indicates the level of dialogue complexity supported by the system. The possible values are *Menu*, *System-Led* and *Mixed-Initiative*.

The second set belongs to test conditions that are used to characterize the basic features of the evaluation exercise, such as:

- Type of users

  This parameter characterizes the kind of users. The possible values are *Project*, those who involved in designing or building the system, *Expert*, those who are familiar with the domain and *Naïve*, those who are totally unfamiliar with the domain.

- Number of users

  In general the significance of the results increases with sample size, but counting only the number of dialogues is not an adequate sampling technique. It is important to understand whether the corpus is provided by many people or by a small number of people. This parameter indicates the number of users.

- Number of dialogues

  This parameter records the number of dialogues in the tested corpus. A dialogue is defined as a continuous session of interaction with the system.

- Number of tasks

  This parameter records the number of tasks in the evaluation exercise.

The third set belongs to test results that are used to characterize the basic features of the system's performance collected during the evaluation exercise, such as:

- Average turns per dialogue

  This parameter records the total number of system and user turns in the tested corpus divided by the number of dialogues in the corpus.

- Average dialogue duration

  This parameter is used to describe the average dialogue duration, starting from the beginning of the first utterance to the end of the last utterance.

- Average turn delay

  This parameter is used to describe the average time taken by the system to respond to a user input.

- Dialogue success rate

  This parameter is used to describe the percentage of all dialogues in the corpus where the system either succeeds in correctly satisfying all the user's requests or it correctly identifies the fact that the requested tasks cannot be performed.

- Task success rate

This parameter is used to describe the percentage of all tasks in the corpus where the system either succeeds in correctly satisfying the user's tasks or it correctly identifies the fact that the tasks cannot be satisfied.

- Crash rate

  This parameter records the percentage of all dialogues in the corpus where the system fails to complete a dialogue in a coherent manner.

An especially important feature of EAGLE's evaluation worthy of notice here is that it takes the user's views and needs into account. This kind of attention has seldom been paid by other systems to the user's satisfaction.

2.3.3 EBMT. Examples have been used by intelligent tutoring systems in a variety of domains from mathematics to programming languages. The usefulness of using examples in complex subjects is dependent on how well we understand the working of the examples in an integrated description of text and examples. As part of the research on the EBMT (Generalized Example Based Machine Translation) project, the language technology research group at Carnegie Mellon University looked into the issues of presenting examples in a useful and effective form using integrated descriptions of text and examples. They identified several critical heuristics in terms of understanding descriptions containing examples. They are *descriptions with and without examples, positioning the example, presentation of different example types*, *complexity and number of examples*, and *presentation orders of examples*. A further verification was shown in an empirical evaluation to see how each heuristic can help in gaining a better understanding of tutorial text [Mittal 1999].

The experiment was conducted by presenting different tutorial descriptions to different groups of participants. Each description takes a heuristic into account, while other descriptions disregard that heuristic on purpose. After reading these descriptions in a limited time, the participants were asked to answer a set of questions designed to measure how much a heuristic can help improve the understanding of that tutorial description.

The evaluation showed the following results:

- Descriptions with and without Examples

  The usefulness of examples in tutoring context is almost indubitable. The group given a description without examples made between four and eleven mistakes out of twenty one questions with an average of six mistakes. The other group, the group given descriptions containing examples, made between zero and four mistakes out of twenty questions with an average of two mistakes. The result shows that the inclusion of examples does help in understanding a concept.

- Positioning the Example

  It is important for examples to be placed in appropriate places whether *before* the text, *within* the text or *after* the text. In the group given *interleaved* examples, only one person made a mistake out of ten questions. In the group with examples *after* the description five participants made an average of three mistakes. In the group with the examples *before* the description, six participants made an average of three mistakes. The participants showed that the best placement for examples is immediately following the point they are supposed to illustrate.

- Presentation of different example types

The research group categorized the variation of examples in three dimensions, their *polarity* with respect to the definition they accompany, the *text type* for which they are generated, and the *knowledge type* of which they happen to be instances. The polarity of an example can be *positive*, *negative* or *anomalous*. Anomalous examples are defined as including instances that are examples not covered by the definition. In this experiment, they only consider the difference of presenting anomalous examples together with and apart from the normal examples. In the group given a description with unmarked anomalous examples, all participants got all questions wrong. In the group given a description with marked anomalous examples, only two out the six people got questions wrong. Therefore, it is important to separate anomalous examples from others and present them explicitly.

- Complexity and number of examples

  The complexity and number are two factors working together to help understand a concept. Two descriptions with the same number but different complexity of examples or the same complexity but different number of examples may lead to different extents of understanding. To see the difference, two experiments were conducted. The first experiment tested both the complexity and number of examples. Its results show that in the group given a description with three simple examples, all participants got all ten questions right. In the group given a description with three complex examples, the participants made an average of two mistakes out of ten questions. In the group given a description with only the last example, the participants made an average of 3.25 mistakes out of ten questions. The second experiment was designed to measure the number of examples required.

The results showed that giving the participants more than enough examples did not raise the success rate significantly.

- Presentation orders of examples

  It is important that related examples appear in an appropriate sequence. The generation of examples has to take into account associated information such as prompts, background information, and contrasting information. A different sequence of examples will result in a different sequence of associated information. The results show that the group given a description with ordered examples made an average of two mistakes out of ten questions. The group given descriptions with unordered examples made an average of six mistakes out of ten questions. This shows that the ordering of examples is an important factor ensuring the coherence and usefulness of the overall description.

This evaluation leads to a very good reflection of how closely machine-generated descriptions can be matched to texts made by humans. Especially, the idea of testing the efficiency of each heuristic in increasing the user's comprehension of a concept is applicable to the evaluation of the turn planner, since the turn planner considers several issues in improving the fluency and coherence of our machine dialogue. It would be useful to conduct an evaluation to see how much the machine dialogue has improved with turn planning.

2.3.4 TRAINS-96. The TRAINS-96 system was constructed from the TRAINS-95 system by adding distances and times to the train route, allowing users to modify routes, adding robust rules in the parser to prevent incorrect understanding, and adding a template-based post-parser module to handle more domain-specific and less well-defined examples [Stent and Allen 1997].

During the formal evaluation of the TRAINS-95 system, two parameters, *time to task completion* and *quality of the solution,* were used to evaluate the general criteria from the task-based perspective. The quality of the solution was measured by whether the stated goals for a task were met, and if so, how much time was taken to complete.

The evaluation of the TRAINS-96 system involved sixteen subjects in a one-hour session with the TRAINS system. Of the sixteen subjects, three were recent college graduates, two were high school students and seven were undergraduates. None of them had experience with the TRAINS systems before. The evaluation used the same five tasks used in evaluating the TRAINS-95 system plus a sixth task for data collection. Each of the first five tasks comes with its own restrictions to simulate different scenarios. In the sixth task the user was given seven trains at different cities and asked to move as many trains as possible to a same destination. After each task the subject was asked to complete a questionnaire and see if the subject had difficulty in completing the task. If so, what caused the difficulty. After completing the final task, the subject completed a more general questionnaire allowing the subject to comment on the system in general [Stent and Allen 1997].

The results of task performance are:

- Tasks with robustness

  The time to completion in four out of five tasks is lower and the length of the route is longer in four out of five tasks.

- Tasks with speech feedback

The time to completion in four out of five tasks is lower and the length of route is less in three out of five tasks.

- Tasks with combinations

  In two of the tasks the time to completion is lowest with robustness but not speech feedback. In another two the time to completion is lowest with robustness and speech feedback. Overall the best time to completion is obtained when both robustness and speech feedback are used.

The subject questionnaire responses show:

- With robustness

  Subjects are less likely to blame the route planner for difficulties.

- With speech feedback

  Subjects are more likely to blame the natural language parts of the system for difficulties.

- Overall

  Subjects are less likely to blame the route planner then to blame the language understanding parts of the system.

This evaluation showed some preliminary results indicating performance differences with and without the robust parsing rules and speech feedback. The results did match their hypotheses but the small sample size also caused a large variance. An experiment like this should be performed with more subjects.

2.3.5 Pretest and Post-test. Conducting a pretest and a post-test before and after the use of a tutoring system is the most popular way of evaluating how much a system can help users to learn a specific topic. This evaluation is easy to apply and good for evaluating the efficiency of a system as a whole without looking into the behavior of each component. Human factors can influence the result to a certain extent. CIRCSIM-Tutor has been evaluated in this way in some of our experiments with students at Rush Medical College.

CHAPTER III

ANALYSIS OF TUTORIAL DIALOGUE

One of the most important research resources in the CIRCSIM-Tutor project is a set of tutoring transcripts numbered from K1 to K76. These sessions were carried out in a keyboard-to-keyboard manner between our domain experts and their students. Most of the previous research in this project is based on the study and analysis of these transcripts.

Our dialogue analysis is derived from a fundamental assumption made by McKeown saying that a hierarchical organization of discourse around fixed schemata can guarantee good coherence and proper content selection [McKeown 1985]. Based on this assumption, Freedman [1996b] analyzed these transcripts and came up with a structured description of the different pedagogical goals that provides a basis for tutorial and discourse planning in the computer tutor. As an extension of Freedman's theory, Kim [1998] annotated the portion of the transcripts involving the DR stage and produced a set of hierarchical tutoring schemata that are being used as plan operators in the new version of CIRCSIM-Tutor.

3.1 Transcript Annotation

We have been using the Standard Generalized Markup Language (SGML) to annotate the hierarchical structure of our tutoring sessions. The main reason to use SGML is that it has been standardized based on an international agreement. So there is no problem with porting the annotation between different parsers. Also the users have all of the necessary flexibility to define their own annotation tags for either content analysis or format arrangement. This flexibility provides us with a good way of marking the structural and content information at the same time, which is especially useful in terms of extracting discourse information and analyzing surface linguistic phenomena.

The initial work of transcript annotation was performed by Kim while analyzing the DR stage in the first fifty one transcripts [Kim 1998, Kim et al. 1998a]. Continuing work on annotating the RR and SS stages is in progress by our project members. An example of transcript annotation is shown in Figure 5 where the hierarchical structure is represented by indenting annotation tags at different levels.

In terms of dialogue analysis, this annotation not only provides us with a deep insight into our tutoring sessions, but also leads to a set of tutorial schemata reflecting the hierarchical tutorial goals described in next section.

3.2 Tutoring Schemata

In CIRCSIM-Tutor the discourse planning and text generation are based on a set of hierarchical tutoring schemata as shown in Figure 6 [Kim 1998]. The abstract high level schema named T-tutors-procedure is used for teaching each predefined perturbation. Each perturbation is then divided into three stages; the schemata for these stages are designated T-tutors-stage. In each stage, the tutor carries on a remedial dialogue about any wrong prediction the student has made. The schemata at this level are named T-corrects-variable. The tutoring of each variable can be further expanded to several methods, which are named T-does-neural-DLR, T-tutors-via-determinants, etc. Again, each method consists of one or more topics, which are designated T-tutors-mechanism,

```
<T-tutors-procedure  proc=proc-pacemaker>
    <T-tutors-stage  stage=DR>
        <T-introduces-stage>
            <T-informs>
                T: Let's take a look at some of your predictions.
            </T-informs>
        </T-introduces-stage>
        <T-corrects-variable  var=TPR>
            <T-introduces-variable>
                <T-informs>
                    T: Take the last one first.
                </T-informs>
            </T-introduces-variable>
            <T-Tutors-variable>
                <T-does-neural-DLR>
                    <T-tutors-mechanism>
                        <T-elicits  softener="can you tell me">
                            T: Can you tell me how TPR is controlled?
                            <S-ans catg=correct>
                                S: Autonomic nervous system.
                            </S-ans>
                            <T-ack  type=positive>
                                T: Yes.
                            </T-ack>
                        </T-elicits>
                    </T-tutors-mechanism>
                    <T-tutors-DR-info>
                        <T-informs  DM="and" type=explain-DR>
                            T: And the predictions that you are making are for the period
                                before any neural changes take place.
                        </T-informs>
                    </T-tutors-DR-info>
                    <T-tutors-value>
                        <T-elicits  DM="so">
                            T: So what about TPR?...
                            <S-ans  catg=correct >
                                S: I would like to change my response re TPR to zero change.
                            </S-ans>
                            <T-ack  type=positive>
                                T: Good.
                            </T-ack>
                        </T-elicits>
                    </T-tutors-value>
                </T-does-neural-DLR>
            </T-tutors-variable>
        </T-corrects-variable>
    </T-tutors-stage>
</T-tutors-procedure>
```

Figure 5. An Example of Transcript Annotation

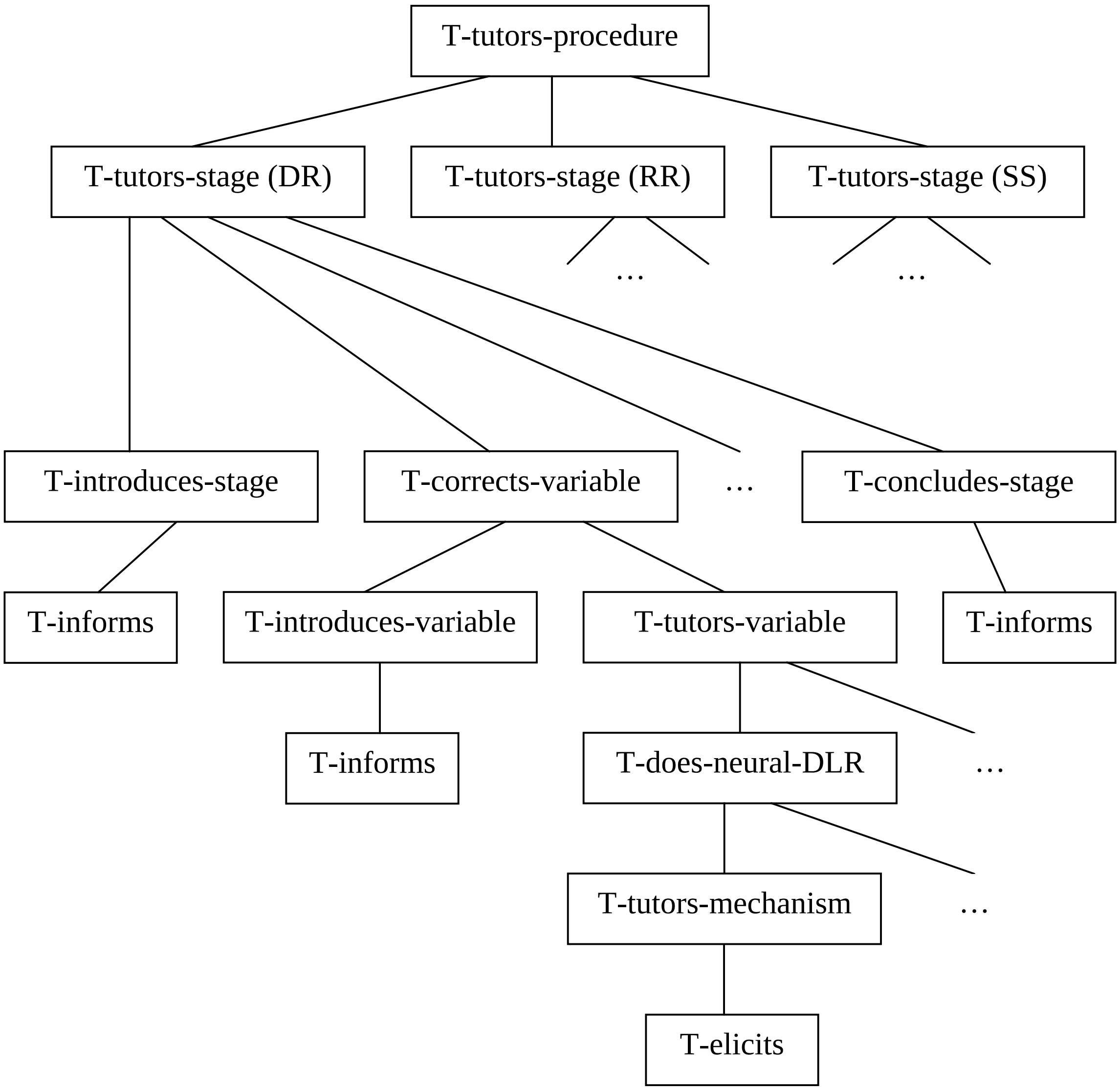

Figure 6. The Tutoring Schemata of CIRCSIM-Tutor [Adapted from Kim 1998]

T-tutors-DR-info, etc., and each topic consists of primitive dialogue acts, which are designated as T-elicits or T-informs.

If we think of a schema as a tree, then the internal nodes can be used to specify the structural information of our tutoring sessions and the leaf nodes can be used to represent the content of surface text generation. To this end, the internal nodes from the procedure level down to the topic level can be used as operators for planning curriculum and discourse. On the other hand, the leaf nodes, dialogue primitives, can be used for guiding the generation of the surface utterances [Kim 2000].

3.3 Turn Planner in the CIRCSM-Tutor Context

With the tutoring schemata described in the previous section, we can go on to plan the discourse at different pedagogical levels, to handle the issues of *what to say*, *when to say it*, and *how to say it*. The current version of CIRCSIM-Tutor has only two levels of planning: lesson planning and discourse planning [Woo 1991]. When this version was developed, we had only four procedures with the same difficulty level.

As the system evolves, we have increased the number of procedures to 83. These procedures are further categorized into five content categories, five procedure difficulty levels and four procedure description levels [Cho et al. 1999]. This has required the development of a complex curriculum plan. To make the new version more modular and efficient, we have developed three separate sets of planning rules one for each of the three different levels of planning. The pipelined structure of these planning operators is shown in Figure 7.

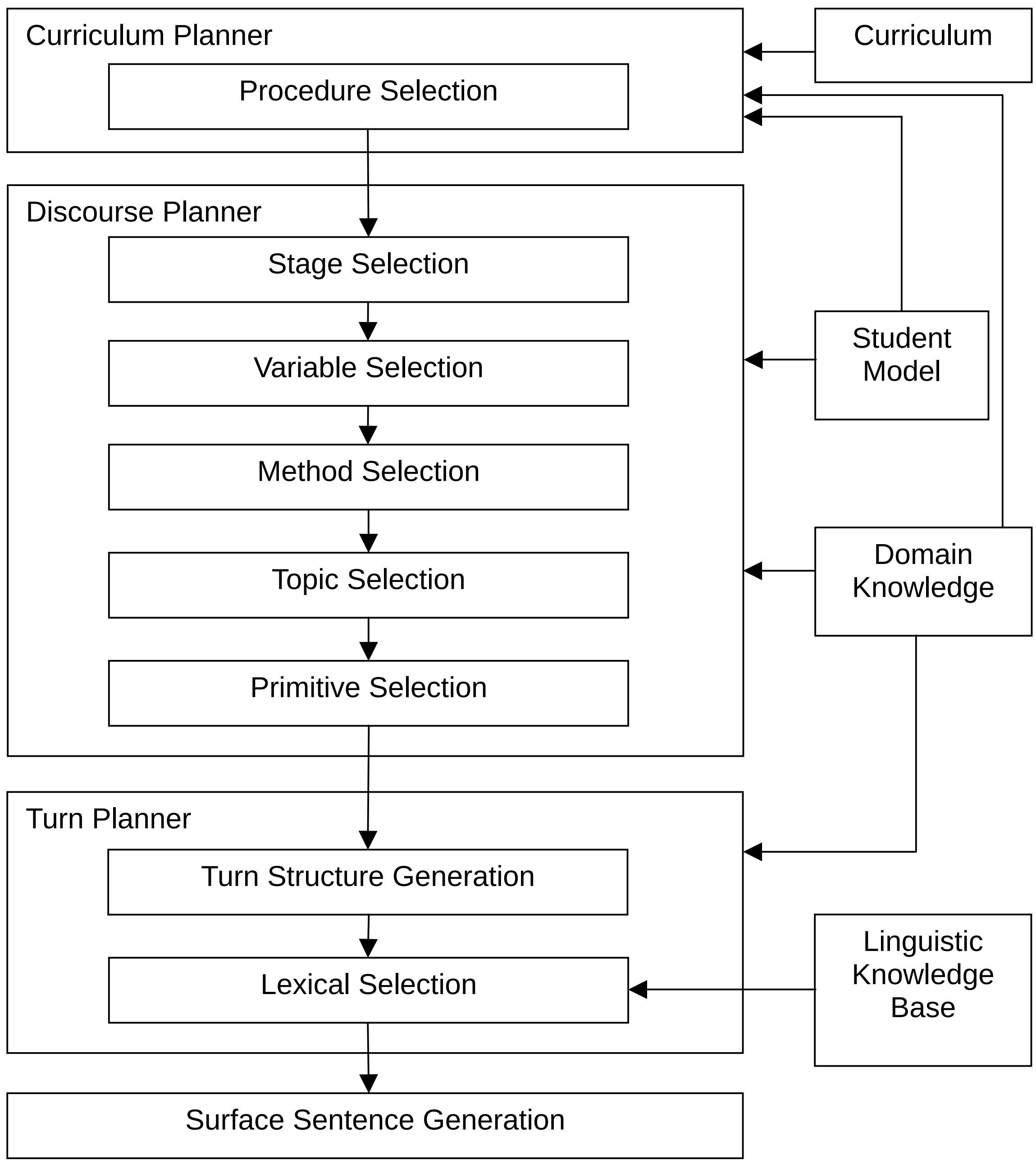

Figure 7. The Three Planners of CIRCSIM-Tutor

In this context, the curriculum planner is responsible for the arrangement of procedures according to the progress of the student [Cho et al. 1999]. The discourse planner is then responsible for the arrangement of tutoring stages, variables, methods, topics, and primitives, according to the tutoring protocols [Mills 2001]. The curriculum planner and discourse planner are closely related to the interactive response of the student. So, it is essential to consult the student model [Zhou and Evens 1999] and input processing module [Glass 2001]. If the student does not respond as expected or can not follow the curriculum, the planners have to replan the discourse or curriculum. Finally, the turn planner is responsible for generating the structure of each tutorial turn. It also selects the semantic forms and lexical items according to the current intra-turn focus [Yang et al. 2000a]. With these features, the surface text generator can generate the sentences that make up a coherent turn. This process will be further explained in the following chapter.

3.4 Atlas Planning Engine

The Atlas Planning Engine (APE) is a reactive and adaptive planner designed to build dialogue-based systems [Freedman 2000a, b, Freedman et al. 2001]. It allows the system to have a dialogue plan in advance, but also be prepared to change the dialogue plan when the user has a problem with following this plan. It is an incremental planner that avoids planning too much detail ahead, since we cannot predict the user's input.

3.4.1 Planning Example. The new version of CIRCSIM-Tutor uses APE as the planning agent for our *curriculum*, *discourse,* and *dialogue turns*. It uses a stack as agenda to store its tutorial goals. To initiate a planning session, the system pushes the initial goal onto the stack and then searches for plan operators that can be applied to fulfill this goal. It is possible that more than one plan operator can be applied and the system will adopt the last one found. After deciding which plan operator to apply, the original goal in the stack is replaced with the recipe for this plan operator. At this point the system forms another version of the agenda. The whole session evolves in this recursive manner until the bottom level within the hierarchical discourse structure is reaching. As soon as a goal is fulfilled it is popped off the stack. Finally the initial goal is fulfilled and popped out of the stack. At this point, the system has completed a session and it is ready to conduct another session.

In the new version of CIRCSIM-Tutor, we use three sets of APE plan operators to plan contents at the *curriculum level*, *discourse level*, and *dialogue turn level* respectively. These three sets of operators form the operations of our curriculum planner, discourse planner, and turn planner. For example, to conduct a tutoring session, the system pushes the initial goal, *did-session*, onto the stack. To fulfill this goal the system decides to apply the plan operator, *do-session*. It expands this goal into three subgoals, *did-beginning-of-session*, *did-one-student*, *did-end-of-session*, as follows:

```
(def-operator do-session
        :goal       (did-session)
        :filter     ()
        :precond    ()
        :recipe     ((goal (did-beginning-of-session))
                    (goal (did-one-student))
                    (goal (did-end-of-session)))
        :temp       ())
```

After this operator is executed, the initial goal is popped off and the stack contains these three goals, *did-beginning-of-session*, *did-one-student*, *did-end-of-session*, from top to bottom. So the system keeps going to fulfill the top goal, *did-beginning-of-session*, by applying the plan operator, *do-beginning-of-session*, which displays a welcome message on the screen, as follows:

```
(def-operator do-beginning-of-session
        :goal          (did-beginning-of-session)
        :filter        ()
        :precond       ()
        :recipe        ((goal (did-utter "Welcome to CST")))
        :temp          ())
```

After showing the welcome message, the top goal is popped out of the stack. So the stack has two goals, *did-one-student*, *did-end-of-session*, from top to bottom. To fulfill the top goal, the system applies the plan operator, *do-one-student*, which asks for the student's name and starts the tutoring session with a student as follows:

```
(def-operator do-one-student
        :goal (did-one-student)
        :filter  ()
        :precond ()
        :recipe  ((goal (did-obtain student-name))
                (fact (w-student-name-is ?student))
                (goal (w-student-name-is ?student))
                (goal (did-student ?student)))
        :temp    ())
```

The system keeps expanding and fulfilling each goal in the same manner. After the goal *did-one-student* is satisfied, it is popped off the stack. So the stack has one goal, *did-end-of-session*, left. To fulfill this goal, the system applies the plan operator, *do-end-of-session*, which displays a good bye message as follows:

```
(def-operator do-end-of-session
    :goal       (did-end-of-session)
    :filter     ()
    :precond    ((f-close-log))
    :recipe     ((goal (did-utter ("Goodbye"))))
    :temp       ())
```

After the last goal on the stack is fulfilled, the stack is empty and system is ready for another tutoring session.

CHAPTER IV

THE TURN PLANNER

If we model the behavior of the turn planner in terms of a discourse tree, it deals with integrating the leaf nodes into a coherent turn-taking dialogue. This integration is related both to discourse planning and to surface sentence generation. So a central problem with the development of the turn planner is how to make a smooth connection among the semantic representation, the pragmatic information, and the surface linguistic phenomena. In other words, the turn planner has to consider the alternatives in terms of representing the content of the participants' utterances, performing the dialogue acts, and generating the surface language. These alternatives not only provide a certain level of implementation flexibility, but also introduce the possibility of optimization at some level. Since we are using schemata as planning operators, having a coherent movement of discourse focus is no longer a problem. My goal is to produce a coherent dialogue.

4.1 Textual Overlapping

Like most Socratic dialogue patterns, the typical structure and content of our tutorial turns include both an acknowledgment of the student's previous statement and an introduction of some new material, i.e., the end of the current topic and the beginning of the next topic. In a sense, there is some overlap between the span of dialogue turns and the coverage of tutorial goals. If we try to map the span of tutorial turns onto the coverage of a discourse plan, we will find an inconsistency between the boundaries of dialogue turns and the boundaries of tutorial goals. It is possible that a tutorial goal needs more than one dialogue turn for its realization or a dialogue turn can realize more than one tutorial goal.

This inconsistency brings up a potential challenge to the development of the turn planner in terms of fitting the turn planner into the current system. The turn planner has to overcome this inconsistency and organize the utterances into a coherent dialogue turn, whether or not they belong to the same discourse plan.

Following the tutoring schemata introduced in the previous chapter, the plans made by the curriculum and the discourse planner are neatly nested in a hierarchical fashion. However, since many tutorial turns contain the end of one tutorial schema and the beginning of the next, we need to make coherent turns out of material from different plans. Here coherence is a main issue of concern.

Several declarative markup languages have been developed to represent the overlap of textual features electronically. These languages are either based on the Standard Generalized Markup Language (SGML) [Barnard et al. 1988, Barnard et al. 1995, Sperberg-McQueen and Burnard 1994, Sperberg-McQueen and Burnard 1995] or on its subset the Extensible Markup Language (XML) [Sperberg-McQueen and Huitfeldt 1999]. These languages provide a good way of identifying textual overlaps. But, in terms of planning the content of overlapped text, they do not help. We need another approach to interface the turn structure with the current discourse plan and make it plan dialogue turns in the evolving conversation.

4.2 Turn Structure Generation

In order to overcome the intrinsic inconsistency between tutorial goals and dialogue turns, we decided to design the turn planner to accumulate tutorial primitives from the discourse planner until the system is ready to ask a question. This is a good breakpoint for turns in a

Socratic dialogue pattern. Once the turn planner has enough dialogue primitives to form a turn, it goes on to select the proper lexical items according to intra-turn and other contextual considerations, such as spelling out the variables or not, using discourse markers or not, using softeners or not, and using pronouns or not, etc. These selections are then kept as a set of features and passed to the surface sentence generator. Finally the surface sentence generator generates the dialogue turn as a whole instead of generating and displaying separate sentences one at a time.

In this manner, we can really have a good pipeline between the discourse planner and surface sentence generator without disturbing the current structure of discourse planning. Also by taking intra-turn focus and contextual information into consideration, it is easier to guarantee coherence. This process is illustrated in Figure 8.

In this example, the turn planner accumulates the first two tutorial primitives from the discourse planner. The first primitive introduces a variable to tutor and the second primitive asks for the determinant of that variable. These two primitives thus form a typical pattern for tutorial turns in a Socratic dialogue. At this moment, the turn structure is set and the system is ready to think about choosing lexical items. It decides to use the abbreviated variable name *CVP*, use the discourse marker *first*, use the softener *can you tell me*, and use the pronoun *it*. Those decisions are reflected in the final dialogue turn being generated.

The advantage of this process is that the turn planner can keep its own turn structure in a buffer, which does not disturb the hierarchical plans generated by other planners. Also, with the whole turn in its buffer, it is easier for the turn planner to

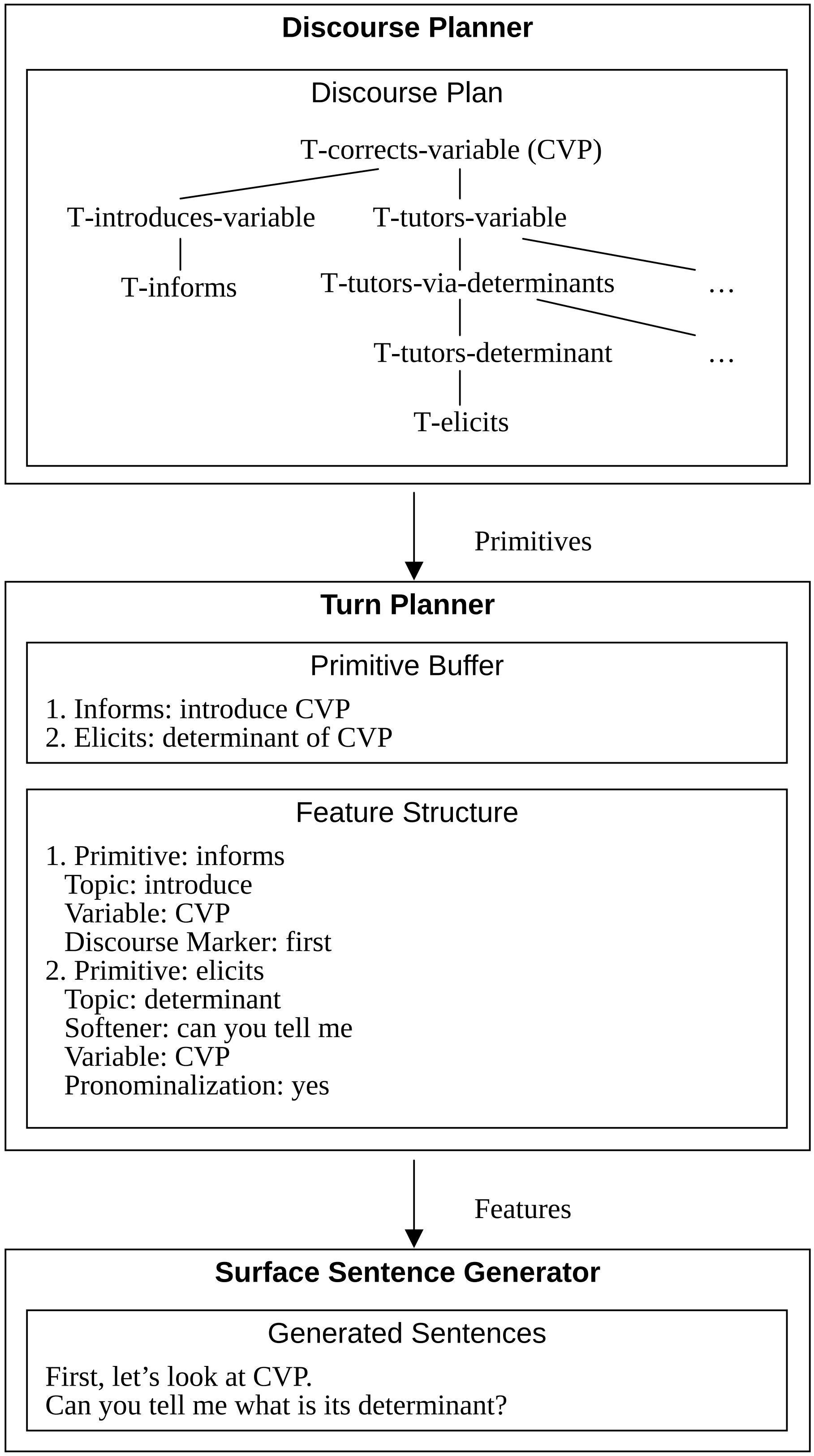

Figure 8. Generating a Tutorial Turn

maintain intra-turn coherence. We believe that without thinking of the problem of planning ahead, this is the best way of planning our tutorial turns.

Another possible method of generating turn structures is applying the strategy of systemic text generation [Patten 1988]. The gate feature is a possible mechanism for generating turn structures. In the systemic methodology, the gate feature works like the *AND* gate in the logic circuits. It is triggered by a combination of other features. We thus can design a systemic grammar with a gate to generate a turn structure whenever there are enough primitives to form a tutorial turn. However, the systemic approach is not compatible with the surface text generation approach that we are committed to work with, so, we have not applied it in the new version.

4.3 Lexical Selection

Lexical selection is another important issue in turn planning. The surface text generator combines lexical items into sentences. With better choice of lexical items, the sentences will sound more natural and more fluent.

One of the main concerns in designing CIRCSIM-Tutor is trying to imitate the dialogue behavior of human tutors. This forces us to do a careful selection among all of the lexical alternatives. Another issue involved in making the machine dialogue human-like is to find an efficient way of simulating the special phenomena of lexical usage in the tutoring schemata. This work starts with identifying these lexical phenomena. A further in-depth analysis of these phenomena can then lead to rules of lexical selection. So far, we have found that variable descriptions, discourse markers, and acknowledgment choices are closely related to the tutoring schemata and we have established some rules for these three categories. These analyses are described in the next chapter.

CHAPTER V

LEXICAL ANALYSIS

We have chosen to explore lexical issues in contextual discourse by marking up the corpus of tutoring transcripts. Our analysis is based on the assumption that a good discourse theory must be able to account for the ordering of major dicourse constituents and predict the surface linguisitc phenomena that depend on structural aspects of discourse [Reichman 1985, Chapter 5]. Another useful idea comes from Passonneau's protocol, especially for the problem of finding the inference relationships between different discourse segments [Passonneau 1994]. The draft of DAMSL [Allen and Core 1997], which uses a backward looking function to capture how the current utterance relates to its antecedent, is also a helpful reference.

The lexical analysis described in this chapter is focused on the semantic and pragmatic relationships among the tutoring schemata as well as looking for special phenomena of lexical usage in the CIRCSIM-Tutor domain.

5.1 Presentation of Lexical Usage for Visualization

Since we want to predict the surface linguistic phenomena from the structural aspects of discourse, it is more useful to have a method that shows discourse structure and lexical usage at the same time. In this way, it is easier for the analysis to take both issues into consideration. I have developed a new representation for lexical usage that allows the researcher to visualize lexical research. This method begins by representing the hierarchical tutoring schemata as tables and then maps the lexical items of interest onto those table entries according to their original positions in the schemata. In this way we can visualize both the discourse structure and lexical usage simultaneously.

Figure 9 illustrates the visualization of the variable descriptions used while tutoring the variable TPR in the transcript K12. More examples of variable descriptions are shown in Appendix A.

<table>
<tr><td colspan="4">T-corrects-variable var=TPR</td></tr>
<tr><td>T-introduces-variable</td><td colspan="3">T-tutors-variable</td></tr>
<tr><td rowspan="3">T-informs</td><td colspan="3">T-does-neural-DLR</td></tr>
<tr><td>T-tutors-mechanism</td><td>T-tutors-DR-info</td><td>T-tutors-value</td></tr>
<tr><td>T-elicits</td><td>T-informs</td><td>T-elicits</td></tr>
<tr><td>T: Now how about TPR?
S: …</td><td>T: By what mechanism will it increase?
S: …</td><td>…</td><td>T: So what do you think about TPR now?
S: …</td></tr>
</table>

Figure 9. Visualization of Variable Descriptions

In this example, we used typography to indicate the lexical features that interest us. The variable *TPR* is marked, along with the anaphoric references to it. The tutor first uses the abbreviated variable name *TPR* to bring up this variable to teach. In the immediately following topic, the tutor uses the pronoun *it* to refer to the previous mentioned *TPR*. After that the tutor goes on to convey some other related explanations and in the final topic the tutor uses the abbreviated variable name *TPR* again to bring back the discourse focus.

A discourse planned using this schema structure will always have the variable introduced in the first topic. So, in the second topic it will always be safe to use a pronoun to refer to the same variable and maintain the same discourse focus. Also, in the sense of making a conclusion, it is appropriate to use abbreviation for the parameter name to bring back focus.

Figure 10 is designed to help us visualize the usage of discourse markers while tutoring the variable TPR in the transcript K10. More examples of the use of discourse markers in human tutoring sessions are described in Appendix B [Kim et al. 2000].

<table>
<tr><td colspan="4">T-corrects-variable var=TPR</td></tr>
<tr><td>T-introduces-variable</td><td colspan="3">T-tutors-variable</td></tr>
<tr><td rowspan="3">T-informs</td><td colspan="3">T-does-neural-DLR</td></tr>
<tr><td>T-tutors-mechanism</td><td>T-tutors-DR-info</td><td>T-tutors-value</td></tr>
<tr><td>T-elicits</td><td>T-informs</td><td>T-elicits</td></tr>
<tr><td>T: Take the last one first.</td><td>T: Can you tell me how TPR is controlled?
S: …</td><td>T: And the predictions that you are making are for the period before any neural changes take place.</td><td>T: So what do you think about TPR now?
S: …</td></tr>
</table>

Figure 10. Visualization of Discourse Marker Usage

In this example, the tutor uses the discourse marker *And* to move from one topic to a semantically continuous topic and uses the discourse marker *So* to mark the final topic as an appropriate conclusion [Yang et al. 2000a].

A discourse planned according to the schema T-does-neural-DLR will always have the first two topics semantically continuous. So, it will be appropriate to use the discourse marker *And* to connect these two topics. Also in the last topic the tutor has to make a conclusion and the discourse marker *So* is a good way to make this conclusion.

Figure 11 is an illustration of the way acknowledgments are used while tutoring TPR in the transcript K48. More examples of the use of acknowledgments are displayed in Appendix C.

<table>
<tr><td colspan="4">T-corrects-variable var=TPR</td></tr>
<tr><td>T-introduces-variable</td><td colspan="3">T-tutors-variable</td></tr>
<tr><td rowspan="3">T-informs</td><td colspan="3">T-does-neural-DLR</td></tr>
<tr><td>T-tutors-mechanism</td><td>T-tutors-DR-info</td><td>T-tutors-value</td></tr>
<tr><td>T-elicits</td><td>T-informs</td><td>T-elicits</td></tr>
<tr><td>T: You predicted that TPR would increase.</td><td>T: What mechanism does this?
S: Autonomic nervous system.
T: Right.</td><td>T: And during DR what changes in ANS activity occur?
S: none.
T: Right</td><td>T: So do you want to change your prediction:
S: Yes.
TPR has no change.
T: Great!</td></tr>
</table>

Figure 11. Visualization of the Choice of Acknowledgments

In this example, the tutor uses the explicit positive acknowledgments *Right* and *Great* to accept the student's answers. For the first two questions the tutor gives a hint by asking some background knowledge and moving toward the final question. Fortunately, the student answers these two hints right. So the tutor uses the explicit word *Right* to accept these answers. Finally the student figures out the correct answer and the tutor acknowledges it in a stronger manner than usual and says *Great*.

5.2 Result of Lexical Analysis

The purpose of visualization is to gather together all the instances of these phenomena showing the contexts in which they occur. We look at two types of context: the surrounding text and the position within the tutorial dialogue schema. Ultimately we can find rules governing lexical usage in the CIRCSIM-Tutor domain.

In Figure 9, we can see the usage of pronominal forms. The tutor uses *it* to refer to *TPR*, which was previously mentioned. By showing this, we can further make rules for describing variables with pronominal forms.

Similarly, we can find lexical rules for other categories. In Figure 10, we see the usage of discourse markers. The tutor uses *and* to initiate a semantically continuous topic and then uses *so* to conclude the tutoring of TPR. Also, in Figure 11, we see the usage of acknowledgments. A common phenomenon is that the tutor tends to use stronger acknowledgments such as *good*, *great*, or *absolutely* when the student finally gets the right answer after making some mistakes.

5.3 Lexical Rules and Discussions

Using the method explained in Section 5.1, we developed lexical rules for *variable references*, *discourse markers*, and *acknowledgment choices*. They are further described in the following sections.

5.3.1 Lexical Rules for Variable Descriptions. The following rules are derived for variable descriptions and references.

Rule 1: Use abbreviated variable names

Case 1: Within the topic *T-introduces-variable*, the tutor uses the *abbreviated name* to introduce a new variable.

Example:

<T-introduces-variable>

K11-tu-41-1: You only have **TPR** left.

</T-introduces-variable>

Case 2: Within the topic immediately following *T-introduces-variable*, the tutor keeps using the *abbreviated name* of the variable to maintain the same discourse focus.

Example:

<T-introduces-variable>

K11-tu-41-1: You only have **TPR** left.

</T-introduces-variable>

…

<T-tutors-variable>

<T-does-neural-DLR>

<T-tutors-mechanism>

K11-tu-49-3: How is **TPR** controlled?

…

</T-tutors-mechanism>

</T-does-neural-DLR>

</T-tutors-variable>

Case 3: Within the last topic of *T-tutors-variable*, the tutor uses the *abbreviated name* of the variable to end digressions and bring back the discourse focus.

Example:

<T-tutors-variable>

…

<T-does-neural-DLR>

K10-tu-29-4: Can you tell me how TPR is controlled?

…

K10-tu-31-2: And the predictions that you are making are for the period before any neural changes take place.

<T-tutors-value>

K10-tu-31-3: So what about **TPR**?

…

</T-tutors-value>

</T-does-neural-DLR>

</T-tutors-variable>

Rule 2: Use pronominal descriptions

Case 1: Within the topic immediately following *T-introduces-variable*, the tutor uses *it* to refer to the variable and maintain the same discourse focus.

Example:

<T-introduces-variable>

K12-tu-31-1: Now how about **TPR**?

</T-introduces-variable>

…

<T-tutors-variable>

<T-does-neural-DLR>

<T-tutors-mechanism>

K12-tu-33-1: By what mechanism will **it** increase?

…

</T-tutors-mechanism>

</T-does-neural-DLR>

</T-tutors-variable>

Case 2: Within the topic immediately following *T-introduces-variable*, the tutor uses *this* to refer to a proposition and maintain the same discourse focus.

Example:

<T-tutors-variable>

…

<T-explores-anomaly>

<T-presents-anomaly>

K26-tu-76-2: So, co decreases even though sv increases.

</T-presents-anomaly>

<T-tutors-anomaly>

K26-tu-76-3: How can you explain **this**?

</T-tutors-anomaly>

</T-explores-anomaly>

</T-tutors-variable>

Rule 1 and Rule 2 are similar to the general results found by Walters [1992]. By marking up the anaphor and the item being referenced, she made a general conclusion that pronouns tend to be used to refer to items that appeared in the same tutoring schema while definite descriptions referring noun phrases tend to be used to refer to items in new plans.

Rule 3: Use definite descriptions

Case 1: Within the topic of *T-introduces-variable*, the tutor uses *the last one* or *this issue* to introduce the variable.

Example:

K10-tu-29-2: Let's take a look at some of your predictions.

<T-introduces-variable>

K10-tu-29-3: Take **the last one** first.

</T-introduces-variable>

Case 2: Within the topic immediately following *T-introduces-variable*, the tutor uses *that prediction* to refer to both the variable and its change and maintain the same discourse focus.

Example:

<T-introduces-variable>

K48-tu-44-3: you predicted that **TPR would increase**.

</T-introduces-variable>

…

<T-tutors-variable>

<T-does-neural-DLR>

<T-tutors-mechanism>

K48-tu-44-4: Can you explain how you arrived at **that prediction**?

…

</T-tutors-mechanism>

</T-does-neural-DLR>

</T-tutors-variable>

Case 3: Within the last topic of *T-tutors-variable*, the tutor uses *your prediction* to end digressions and bring back the discourse focus.

Example:

<T-tutors-variable>

<T-does-neural-DLR>

K48-tu-44-4: Can you explain how you arrived at that prediction?

…

K48-tu-48-2: and during DR what changes in ANS activity occur?

…

<T-tutors-value>

K48-tu-50-1: So do you want to change **your prediction**?

</T-tutors-value>

</T-does-neural-DLR>

</T-tutors-variable>

5.3.2 Lexical Rules for Discourse Markers. The following rules are derived for applying discourse markers.

Rule 1: Use *so* and *now*

Case 1: *so* and *now* are used in *T-introduces-variable* to initiate a discourse focus. This is similar to behavior observed by Schiffrin [1987, p261].

Example:

<T-introduces-variable>

K11-tu-53-2: **So** let me ask you, are there any other of these variables that are primarily under neural control?

</T-introduces-variable>

Case 2: *so* and *now* are used to conclude *T-tutors-variable*. This is similar to the idea of marking results discussed by Schiffrin [1987, Ch 7].

Example:

<T-tutors-variable>

<T-does-neural-DLR>

…

<T-tutors-value>

K10-tu-31-3: **So** what about TPR?

…

</T-tutors-value>

</T-does-neural-DLR>

</T-tutors-variable>

Rule 2: Use *first* in *T-introduces-variable* to introduce the first topic of the first variable being tutored.

Example:

<T-introduces-variable>

K13-tu-37-3: **First**, what parameter determines the value of rap?

</T-introduces-variable>

Rule 3: Use *but* in *T-presents-contradiction* to contrast two ideas.

Example:

<T-shows-contradiction>

<T-presents-contradiction>

K10-tu-41-2: You predicted that it would go up.

…

K10-tu-43-1: **But** remember that we’re dealing with the period before there can be any neural changes.

</T-presents-contradiction>

</T-shows-contradiction>

Rule 4: Use *and* to initiate a semantically continuous topic.

Example:

<T-does-neural-DLR>

<T-tutors-mechanism>

K10-tu-29-4: Can you tell me how TPR is controlled?

…

</T-tutors-mechanism>

<T-tutors-DR-info>

K10-tu-31-2: **And** the predictions that you are making are for the period before any neural changes take place.

</T-tutors-DR-info>

…

<T-does-neural-DLR>

Rule 5: Use *therefore* to summarize *T-tutors-via-deeper-concepts.*

Example:

<T-tutors-via-deeper-concepts>

<T-tutors-determinant>

K27-tu-52-1: If I have a single blood vessel, what parameter most strongly determines its resistance to flow?

…

<T-moves-to-previous-concepts>

<T-tutors-determinant>

K27-tu-54-1: And physiologically, what determines the diameter of the blood vessels?

</T-tutors-determinant>

</T-moves-to-previous-concepts>

</T-tutors-determinant>

<T-tutors-determinant>

K27-tu-56-2: **Therefore**, what determines TPR?

</T-tutors-determinant>

</T-tutors-via-deeper-concepts>

5.3.3 Lexical Rules for Acknowledgments. The following rules are derived for giving acknowledgments.

Rule 1: Use a negative acknowledgment such as *no* or *not quite* to reject the student’s first wrong answer.

Example:

K12-tu-31-1: Now how about TPR?

<T-elicits>

K12-tu-33-1: By what mechanism will it increase?

<S-ans catg=incorrect>

K12-st-34-1: If you increase pressure will you momentarily increase resistance

</S-ans>

<T-ack type=negative>

K12-tu-35-1: **No**.

</T-ack>

</T-elicits>

Rule 2: Use a partial acknowledgment, such as *partly correct*, to partially accept the student's answer.

Example:

<T-elicits>

K47-tu-56-5: Can you tell me what you think that IS means?

<S-ans catg=near-miss>

K47-st-57-1: the contractility of the heart caused by preload and sympathetic stimulation

</S-ans >

<T-ack type= partially-correct >

K47-tu-58-1: **Partly correct**.

</T-ack >

</T-elicits>

Rule 3: Use of positive acknowledgments

Case 1: Use *yes* or *right* to accept the student's first correct answer.

Example:

K10-tu-29-2: Let's take a look at some of your predictions.

K10-tu-29-3: Take the last one first.

<T-elicits>

K10-tu-29-4: Can you tell me how TPR is controlled?

<S-ans catg=correct>

K10-st-30-1: Autonomic nervous system

</S-ans>

<T-ack type=positive>

K10-tu-31-1: **Yes**.

</T-ack>

</T-elicits>

Case 2: Use a strong positive acknowledgment, such as *good*, *very good*, *absolutely*, *exactly*, or *great* to accept the student's final correct answer, especially when the student had some difficulty in reaching this goal.

Example:

<T-elicits>

K27-tu-72-2: How is this possible?

<S-ans catg=correct>

K27-st-73-1: Hr is down more than sv is up

</S-ans>

<T-ack type=positive>

K27-tu-74-1: **Very good**.

</T-ack>

</T-elicits>

Rule 4: Acknowledgment is omitted in some special situations, such as when the tutor is identifying the student's problem, or the student has a near miss answer.

Case 1: the tutor tries to identify the student’s problem without giving any acknowledgment.

Example:

<T-diagnoses-errors>

<T-identifies-problem>

<T-elicits>

K27-tu-50-2: Why do you think that TPR will decrease?

<S-ans catg=incorrect>

K27-st-51-1: Since HR decreased, CO will decrease and the direct response would be decreased TPR.

</S-ans>

</T-elicits>

</T-identifies-problem>

</T-diagnoses-errors>

K27-tu-52-1: If I have a single blood vessel, what parameter most strongly determines its resistance to flow?

(Acknowledgment omitted)

Case 2: The tutor doesn’t give any acknowledgment when the student gives a near-miss answer, but tries other methods to guide the student toward the correct answer.

Example:

<T-tutors-via-determinants>

<T-tutors-determinant>

<T-elicits>

K25-tu-48-3: What parameter determines rap?

<S-ans catg=near-miss>

K25-st-49-1: Central venous pressure.

</S-ans>

</T-elicits>

<T-moves-toward-PT method-type=inner>

<T-tutors-determinant>

<T-elicits>

K25-tu-50-1: And what determines cvp?

(Acknowledgment omitted)

<S-ans catg=correct>

K25-st-51-1: Blood volume and "compliance" of the venous side of the circ.

</S-ans>

<T-ack type=positive>

K25-tu-52-1: Right.

</T-ack>

</T-elicits>

<T-moves-toward-PT>

</T-tutors-determinant>

</T- tutors-via-determinants>

5.4 Other Lexical Problems

This method of analysis works very well for selecting lexical categories that are closely related to the tutoring schema, such as variable descriptions, discourse markers and acknowledgments. However, we still have other categories that are less related to the tutoring schema, such as the lexical choice of verbs for qualitative change and the decision about when the tutor repeats what the student has said. As part of this research, we need to explore more issues and discover more rules to make this work more complete.

5.4.1 Verb Phrases for Qualitative Change. One of the special lexical phenomena in our tutoring sessions is that the human tutor tends to use some verb phrases as a pair while describing the qualitative changes in variables [Zhang 1991]. The tutor tends to use *go up* and *go down*, *rise* and *fall*, + and -, or *increase* and *decrease* in pairs to describe these qualitative relations. These pairs are important in making cohesive lexical selections.

Example:

K26-tu-76-2: So, CO **decreases** even though SV **increases**.

5.4.2 User Driven Choice. Another special lexical phenomenon in our domain is that the human tutor tends to adopt the student's words or phrases in the immediately following dialogue. Ramachandran [1994] was the first to notice this phenomenon. He named it user driven lexical choice. In terms of imitating the human tutor's language this is an important consideration and we need to apply it to our lexical selections.

Example:

K20-tu-36-8: So what do you think happens to SV, given this info?

K20-st-37-1: Since CC is not changing then I would think no change in SV.

K20-st-37-2: But since **CO D** then **RAP I**. I think **SV I**

K20-tu-38-1: Correct.

K20-tu-38-2: When **CO D**, **RAP** (which determines EDV) **I** and **SV I**.

5.5 Other Coherence Issues

Shifts in discourse focus are an important consideration in terms of generating coherent text. Maintaining the coherence of a conversation means ensuring that the discourse focus evolves in a reasonable way [Reichman 1985]. In CIRCSIM-Tutor, a tutorial turn may be able to realize several tutorial goals or a tutorial goal may need several turns to realize. These problems have raised a number of issues involving intra-turn and inter-turn coherence.

- Discourse focus movement

  Schema-based planning is a good way of modeling discourse behaviors [McKeown 1985]. An alternate idea is using plan operators to model discourse behaviors [Moore and Paris 1993]. Our schema-based plan operators provide us with a smooth approach to moving the discourse focus. As long as we can choose a suitable schema to continue the tutoring sessions and generate dialogue accordingly, we can generate coherent discourse.

- Global discourse structure

CIRCSIM-Tutor uses a global history list to enhance coherence. Based on the history list, we can take care of a lot of lexical issues such as anaphoric reference, ellipsis, verb phrase anaphora, etc.

CHAPTER VI

EVALUATION

The justification for adding a turn planner as a new module to the current architecture is to improve the rhetorical structure and have a more natural and more fluent machine dialogue. In terms of evaluation, fluency and coherence are subjective ideas related to the user acceptance of this system. So, contrary to most of the published literature on evaluation, the evaluation of the turn planner should be mostly focused on user acceptability.

6.1 Evaluation Method

Concerning the issue of user acceptance, I recommend a statistical evaluation, representing the quality of our machine dialogue by mean and variance. This approach is similar to the idea of evaluating the validity of each improvement applied in the EBMT system and the TRAINS-96 system. For each of the improvements made by the turn planner, we will generate two dialogues from different contexts. One of the dialogues takes an improvement into account while the other disregards that improvement on purpose. After constructing these two dialogues, we can show them to our potential users, the first-year medical students, and have them grade the dialogue quality on a scale from one to five. Then, the grading difference can be used as an indication of the validity of applying each improvement [Yang et al. 2001].

6.2 Evaluating the Validity of Applying Discourse Markers

To evaluate the validity of applying discourse markers, we will have the medical students grade the following two dialogues. One takes the discourse markers into account and the other disregards discourse markers. By analyzing the difference of their grading, we can have an indication of how much the discourse markers improve the tutorial dialogue.

The following example takes discourse markers into account:

T1: Can you tell me by what mechanism Inotropic State is controlled?

S2: nervous system

T3: Right.

So, what is the value of IS in DR?

S4: decreased

T5: No, it is not decreased in DR.

Remember, it is neurally controlled.

Now, what is the value of IS in DR?

S6: not changed

T7: Very good.

The following example disregards discourse markers:

T1: Do you know what are the determinants of Cardiac Output?

S2: hr sv

T3: Which determinant is dominant in this case?

S4: hr

T5: No, the important determinant of CO in this case is SV.

HR didn’t change yet.

Is the relationship from SV to CO direct or is it inverse?

S6: direct

T7: Good.

What is the correct value of CO?

S8: down

T9: Very Good.

6.3 Evaluating the Validity of Improving Variable Descriptions

To evaluate the validity of improving variable descriptions, we will have the medical students grade the following two dialogues. One is generated with improved variable descriptions while the other is generated without this consideration. The grading difference can then be an indication of the validity of improving variable descriptions.

The following example is generated with improvements in variable descriptions:

T: Do you know by what mechanism Total Peripheral Resistance is controlled?

S: neural

T: Right, it is controlled by the nervous system.

Remember, the reflex will not activate until the RR stage.

So, what is the value of TPR in DR?

S: unchanged

T: Great.

The following example is generated without any improvement in variable descriptions:

T: Can you tell me what are the determinants of Cardiac Output?

S: sv and hr

T: Right, Cardiac Output is determined by Stroke Volume and Heart Rate.

Now, which determinant is dominant in this case?

S: sv

T: Right, the important determinant of Cardiac Output in this case is Stroke Volume.

And is the relationship from Stroke Volume to Cardiac Output direct or inverse?

S: direct

T: Good.

So, what is the correct value of Cardiac Output?

S: decreased

T: Absolutely.

6.4 Evaluating the Validity of Improving Acknowledgments

To evaluate the validity of improving acknowledgments, we will have the medical students grade the following two dialogues. Again, the difference can be an indication of the validity of improving acknowledgments.

The following example is generated with improvements in acknowledgments:

T: Can you tell me what are the determinants of Mean Arterial Pressure?

S: co and tpr

T: Right.

And which determinant is dominant in this case?

S: tpr

T: No, it is not.

The important determinant of MAP in this case is CO.

TPR didn't change yet.

Do you know is the relationship from CO to MAP direct or inverse?

S: direct

T: Good.

So, what is the correct value of MAP?

S: decreased

T: Absolutely.

The following example is generated without improvements in acknowledgments:

T: Can you tell me what is the determinant of Central Venous Pressure?

S: co

T: Correct, CVP is determined by CO.

And is the relationship from CO to CVP direct or is it inverse?

S: direct

T: Sorry, you got the wrong answer.

A change in CO has an inverse effect on CVP.

So, what is the correct value of CVP?

S: decreased

T: Correct, the value of CVP is decreased.

6.5 Other Concerns

The sample size is an important concern in determining the precision of statistical evaluations. For this kind of evaluation, the sample size should be large enough, so that any special result will not lead to a big variance. Also, to get an intuitive response concerning the user acceptance, the time of reading and grading sample dialogues should be limited to allow the subjects to work at a regular pace.

CHAPTER VII

CONCLUSIONS

Computers help people process massive information efficiently on one hand, but tend to respond to people with more information than necessary on the other hand. The reason is that the exchange of information between human and computer is very different from the exchange of information between human and human. While most information systems respond to users with preprogrammed information, people tend to exchange information incrementally. After giving the listener a certain amount of information, the speaker tends to stop and wait for some response from the listener. So that the speaker knows if the listener is really following the intended communication goal or not.

If a user asks an information system the same question twice, it is very likely that the system will respond with the same answer twice, but people will hardly have this kind of dialogue. Instead of repeating the same answer, people tend to adapt answers according to the listener's understanding and try to solve the unspoken problems in the way of reaching the dialogue goal [McRoy et al. 1999]. The invention of dialogue systems is motivated by the desire to reform the interaction between human and computer to change the typical dysfunctional exchange into an incremental information exchange. To this end, implementing a turn taking dialogue between human and computer is the best way to fulfill the goal of an incremental information exchange.

7.1 Summary

The planning of turn taking dialogue is not well-studied in natural language dialogue systems. Part of the reason is that conducting turn-taking dialogues is a real-time, responsive, and adaptive process. A dialogue progresses interactively planned by both participants. Thus too much prediction and planning ahead may be counter productive.

In this research, I focus on the task of refining the discourse plans and producing a more detailed dialogue specification that can be passed to a surface sentence generator that produces the actual words. To make the dialogue more fluent and more natural, a turn buffer is used to accumulate the dialogue primitives from the discourse planner until there are enough to form a turn. Once the turn planner has enough dialogue primitives to form a turn, it goes on to select the proper lexical items according to intra-turn and other contextual considerations. This makes it possible to continue the pipeline between the discourse planner and the surface sentence generator in the CIRCSIM-Tutor domain without disturbing the current structure of discourse planning and curriculum planning. Also by taking intra-turn focus and contextual information into consideration, it is easier to guarantee coherence.

Producing well chosen lexical items is an important factor in generating coherent dialogue. In the turn planner, the lexical items are selected on the basis of the structural aspects of the discourse. I developed a method that shows discourse structure and lexical usage at the same time, which makes it easier to take both issues into consideration while analyzing lexical phenomena. This method of analysis works very well for making choices from lexical categories that are closely related to the tutoring schema, such as variable descriptions, discourse markers, and acknowledgments. Based this analysis, I derived three sets of rules for lexical choice.

7.2 Significance

As we have seen turn planning can make dialogue seem more fluent and more nature. This is a new problem and we have taken the first step toward it. In many systems as in the old version of CIRCSIM-Tutor the result of discourse planning is a tree in which a leaf node is a simplified structure including a predicate and some arguments. The predicate is usually named by the verb and the arguments are often some symbols representing the intended dialogue primitive. This structure is enough for generating dialogue at the sentence level, but, in terms of generating a coherent dialogue, some other issues beyond the sentence level should also be taken into consideration. My new turn planner takes some of these intra-turn and inter-turn issues into consideration. The result of turn planning is still not a text, but by providing the surface sentence generator with better dialogue specifications we can improve the rhetorical structure to a reasonable level.

7.3 Future Study

Many natural language research groups have found that a certain number of natural language generation issues are beyond the consideration of discourse planning and surface generation, but they are nonetheless important in building high-quality text generation systems. My research on turn planning has touched only some of the issues. Other further in-depth research can make a significant contribution to dialogue generation.

In future study, I hope to learn more about how human dialogue works, especially tutorial dialogue. I want to find some more general contingencies that form the specification of dialogue turns and more clues for selecting lexical items according to contextual information.

The evaluation of natural language dialogue systems is still evolving toward a standard. Some systems have been evaluated on the basis of application specific metrics, while others have been evaluated from a technology perspective. For the evaluation of the turn planner, I proposed a method to evaluate the improvement of dialogue from the perspective of user acceptance. In my proposal the user is asked to compare the difference between two dialogues: one takes an improvement into account while the other disregards that improvement on purpose. The result can be used as an indication of the validity of applying each improvement.

Another concern for my future study is searching for a more general method that can be applied to the evaluation of most natural language systems. I want to be involved in actual evaluations of CIRCSIM-Tutor.

Gregory Sanders [1995] identified multiturn discourse patterns, which he called “directed lines of reasoning.” More work is needed on planning turns in directed lines of reasoning.

APPENDIX A

VARIABLE DESCRIPTIONS AND SCHEMATA

VARIABLE DESCRIPTIONS AND SCHEMATA

The following sections show the phenomena of variable descriptions in our tutoring schemata during the DR Stage.

A.1 Variable Descriptions while Tutoring TPR

	T-corrects-variable var=TPR			
	T-introduces-variable	T-tutors-variable		
		T-does-neural-DLR		
		T-tutors-mechanism	T-tutors-DR-info	T-tutors-value
K10	**the last one**	TPR		TPR
K11	TPR	TPR		TPR
K48	**TPR**	**that prediction**		**your prediction**

	T-corrects-variable var=TPR					
	T-introduces-variable	T-tutors-variable				
		T-does-neural-DLR				
		T-tutors-mechanism			T-tutors-DR-info	T-tutors-value
		Attempt 1	Attempt 2	T-moves-toward-PT		
				T-tutors-mechanism		
K12	**TPR**	**it**				**TPR**

	T-corrects-variable var=TPR				
	T-tutors-variable				
	T-diagnoses-errors	T-tutors-via-deeper-concepts		T-does-neural-DLR	
	T-identifies-problem	T-tutors-determinant	T-tutors-determinant	T-tutors-DR-info	T-tutors-value
		T-moves-to-previous-concepts			
K27	**TPR**		**TPR**		**TPR**

A.2 Variable Descriptions while Tutoring IS

	T-corrects-variable var=CC		
	T-introduces-variable	T-tutors-variable	
		T-shows-contradiction	
		T-presents-contradiction	T-tutors-contradiction
K10	**CC (S-ans)**	**it**	**CC**

	T-corrects-variable var=CC					
	T-introduces-variable	T-tutors-variable				
		T-shows-contradiction		T-does-neural-DLR		
		T-presents-contradiction	T-tutors-contradiction	T-tutors-determinant	T-tutors-DR-info	T-tutors-value
K11	**CC (S-ans)**	**CC**	**it**	**CC**		**CC**

	T-corrects-variable var=CC		
	T-tutors-variable		
	T-does-neural-DLR		
	T-tutors-mechanism	T-tutors-DR-info	T-tutors-value
K16	contractility		CC

	T-corrects-variable var=IS				
	T-introduces-variable	T-tutors-variable			
		T-does-neural-DLR			
		T-tutors-definition	T-tutors-mechanism	T-tutors-DR-info	T-tutors-value
K47	**IS**	IS	IS		IS

A.3 Variable Descriptions while Tutoring SV

	T-corrects-variable var=SV
	T-tutors-variable
	T-moves-forward
	T-tutors-value
K12	**SV**

	T-corrects-variable var=SV		
	T-tutors-variable		
	T-via-determinant		
	T-tutors-determinant	T-tutors-value	
		Attempt 1	Attempt 2
K14	**SV**	**SV**	**SV**
K20	**SV**	**SV**	

	T-corrects-variable var=SV		
	T-tutors-variable		
	T-moves-forward	T-explores-anomaly	
		T-presents-anomaly	T-tutors-anomaly
K25	**SV**	**SV**	**SV**
K26	**SV**	**SV**	**this**

	T-corrects-variable var=SV						
	T-introduces-variable	T-tutors-variable					
		T-tutors-via-determinant		T-shows-contradiction		T-explores-anomaly	
		T-tutors-determinant	T-moves-toward-PT	T-presents-contradiction	T-tutors-contradiction	T-presents-anomaly	T-tutors-anomaly
			T-tutors-PT-entry				
K27	**SV**	**SV**		**SV**	**SV**	**SV**	**this**

A.4 Variable Descriptions while Tutoring CVP

	T-corrects-variable var=RAP					
	T-introduces-variable	T-tutors-variable				
		T-tutors-via-determinant		T-tutors-via-deeper-concepts		
		T-tutors-determinant	T-tutors-relationship	T-tutors-relationship	T-tutors-relationship	T-tutors-value
K11	**RAP**		**RAP**			**RAP**

	T-corrects-variable var=RAP		
	T-tutors-variable		
	T-tutors-via-determinant		
	T-tutors-determinant	T-tutors-relationship	T-tutors-value
K13	**RAP**	**RAP**	RAP

<table>
<tr><td rowspan="6"></td><td colspan="8">T-corrects-variable var=RAP</td></tr>
<tr><td>T-introduces-variable</td><td colspan="7">T-tutors-variable</td></tr>
<tr><td></td><td colspan="7">T-tutors-via-determinant</td></tr>
<tr><td></td><td colspan="2">T-tutors-determinant</td><td colspan="3">T-tutors- determinant</td><td>T-tutors-relationship</td><td>T-tutors-value</td></tr>
<tr><td></td><td>Attempt 1</td><td>Attempt 2</td><td colspan="2">T-tutors-via-deeper-concepts</td><td>T-moves-towards-PT</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>T-tutors-compliance-info</td><td>T-tutors-determinant</td><td>T-tutors-PT-entry</td><td></td><td></td></tr>
<tr><td>K14</td><td>this issue</td><td>RAP</td><td>RAP</td><td></td><td></td><td></td><td>RAP</td><td>RAP</td></tr>
</table>

<table>
<tr><td rowspan="6"></td><td colspan="5">T-corrects-variable var=RAP</td></tr>
<tr><td>T-introduces-variable</td><td colspan="4">T-tutors-variable</td></tr>
<tr><td></td><td colspan="4">T-tutors-via-determinant</td></tr>
<tr><td></td><td colspan="3">T-tutors-determinant</td><td>T-tutors-consequence-value</td></tr>
<tr><td></td><td></td><td>T-moves-towards-PT</td><td>T-moves-towards-PT</td><td></td></tr>
<tr><td></td><td></td><td>T-tutors-determinant</td><td>T-tutors-determinant</td><td></td></tr>
<tr><td>K25</td><td>RAP</td><td>RAP</td><td></td><td></td><td>RAP</td></tr>
</table>

<table>
<tr><td rowspan="6"></td><td colspan="5">T-corrects-variable var=RAP</td></tr>
<tr><td colspan="5">T-tutors-variable</td></tr>
<tr><td colspan="5">T-tutors-via-determinant</td></tr>
<tr><td colspan="3">T-tutors-determinant</td><td>T-tutors-relationship</td><td>T-tutors-value</td></tr>
<tr><td></td><td colspan="2">T-moves-towards-PT</td><td></td><td></td></tr>
<tr><td></td><td>T-tutors-determinant</td><td>T-tutors-determinant</td><td></td><td></td></tr>
<tr><td>K26</td><td>RAP</td><td></td><td></td><td>RAP</td><td>RAP</td></tr>
</table>

APPENDIX B

DISCOURSE MARKERS AND SCHEMATA

DISCOURSE MARKERS AND SCHEMATA

The following sections show the phenomena of using discourse markers in our tutoring schemata during the DR Stage.

B.1 Discourse Marker Usage while Tutoring TPR

	T-corrects-variable var=TPR			
	T-introduces-variable	T-tutors-variable		
		T-does-neural-DLR		
		T-tutors-mechanism	T-tutors-DR-info	T-tutors-value
K10			**and**	so
K11				now
K48			**and**	so

	T-corrects-variable var=TPR					
	T-introduces-variable	T-tutors-variable				
		T-does-neural-DLR				
		T-tutors-mechanism			T-tutors-DR-info	T-tutors-value
		Attempt 1	Attempt 2	T-moves-toward-PT		
				T-tutors-mechanism		
K12				**and**	**and**	**so**

	T-corrects-variable var=TPR				
	T-tutors-variable				
	T-diagnoses-errors	T-tutors-via-deeper-concepts		T-does-neural-DLR	
	T-identifies-problem	T-tutors-determinant	T-tutors-determinant	T-tutors-DR-info	T-tutors-value
		T-moves-to-previous-concepts			
K27		**and**	**therefore**		**so**

B.2 Discourse Marker Usage while Tutoring IS

<table>
<tr><td rowspan="4"></td><td colspan="3">T-corrects-variable var=CC</td></tr>
<tr><td>T-introduces-variable</td><td colspan="2">T-tutors-variable</td></tr>
<tr><td></td><td colspan="2">T-shows-contradiction</td></tr>
<tr><td></td><td>T-presents-contradiction</td><td>T-tutors-contradiction</td></tr>
<tr><td>K10</td><td></td><td>but</td><td>so</td></tr>
</table>

<table>
<tr><td rowspan="4"></td><td colspan="6">T-corrects-variable var=CC</td></tr>
<tr><td>T-introduces-variable</td><td colspan="5">T-tutors-variable</td></tr>
<tr><td></td><td colspan="2">T-shows-contradiction</td><td colspan="3">T-does-neural-DLR</td></tr>
<tr><td></td><td>T-presents-contradiction</td><td>T-tutors-contradiction</td><td>T-tutors-determinant</td><td>T-tutors-DR-info</td><td>T-tutors-value</td></tr>
<tr><td>K11</td><td>so</td><td>but</td><td></td><td></td><td></td><td>so</td></tr>
</table>

<table>
<tr><td rowspan="4"></td><td colspan="3">T-corrects-variable var=CC</td></tr>
<tr><td colspan="3">T-tutors-variable</td></tr>
<tr><td colspan="3">T-does-neural-DLR</td></tr>
<tr><td>T-tutors-mechanism</td><td>T-tutors-DR-info</td><td>T-tutors-value</td></tr>
<tr><td>K16</td><td></td><td></td><td>so</td></tr>
</table>

<table>
<tr><td rowspan="4"></td><td colspan="5">T-corrects-variable var=IS</td></tr>
<tr><td>T-introduces-variable</td><td colspan="4">T-tutors-variable</td></tr>
<tr><td></td><td colspan="4">T-does-neural-DLR</td></tr>
<tr><td></td><td>T-tutors-definition</td><td>T-tutors-mechanism</td><td>T-tutors-DR-info</td><td>T-tutors-value</td></tr>
<tr><td>K47</td><td>however</td><td></td><td></td><td></td><td></td></tr>
</table>

B.3 Discourse Marker Usage while Tutoring SV

<table>
<tr><td rowspan="4"></td><td>T-corrects-variable var=SV</td></tr>
<tr><td>T-tutors-variable</td></tr>
<tr><td>T-moves-forward</td></tr>
<tr><td>T-tutors-value</td></tr>
<tr><td>K12</td><td>and</td></tr>
</table>

	T-corrects-variable var=SV			
	T-tutors-variable			
	T-via-determinant			
	T-tutors-determinant		T-tutors-value	
	Attempt 1	Attempt 1	Attempt 1	Attempt 2
K14	**now**		**Well (2)**	**so**
K20			**so**	

	T-corrects-variable var=SV		
	T-tutors-variable		
	T-moves-forward	T-explores-anomaly	
		T-presents-anomaly	T-tutors-anomaly
K25	**so**	**so**	**now**
K26	**and**	**so**	

	T-corrects-variable var=SV						
	T-introduces-variable	T-tutors-variable					
		T-tutors-via-determinant		T-shows-contradiction		T-explores-anomaly	
		T-tutors-determinant	T-moves-toward-PT	T-presents-contradiction	T-tutors-contradiction	T-presents-anomaly	T-tutors-anomaly
			T-tutors-PT-entry				
K27				**so**	**so**	**so**	

B.4 Discourse Marker Usage while Tutoring CVP

	T-corrects-variable var=RAP					
	T-introduces-variable	T-tutors-variable				
		T-tutors-via-determinant		T-tutors-via-deeper-concepts		
		T-tutors-determinant	T-tutors-relationship	T-tutors-relationship	T-tutors-relationship	T-tutors-value
K11						**so**

	T-corrects-variable var=RAP		
	T-tutors-variable		
	T-tutors-via-determinant		
	T-tutors-determinant	T-tutors-relationship	T-tutors-value
K13	**first**		**so**

<table>
<tr><td rowspan="6"></td><td colspan="8">T-corrects-variable var=RAP</td></tr>
<tr><td>T-introduces-variable</td><td colspan="7">T-tutors-variable</td></tr>
<tr><td></td><td colspan="7">T-tutors-via-determinant</td></tr>
<tr><td></td><td colspan="2">T-tutors-determinant</td><td colspan="3">T-tutors- determinant</td><td>T-tutors-relationship</td><td>T-tutors-value</td></tr>
<tr><td></td><td>Attempt 1</td><td>Attempt 2</td><td colspan="2">T-tutors-via-deeper-concepts</td><td>T-moves-towards-PT</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>T-tutors-compliance-info</td><td>T-tutors-determinant</td><td>T-tutors-PT-entry</td><td></td><td></td></tr>
<tr><td>K14</td><td></td><td></td><td></td><td></td><td></td><td>well</td><td></td><td>since</td></tr>
</table>

<table>
<tr><td rowspan="6"></td><td colspan="5">T-corrects-variable var=RAP</td></tr>
<tr><td>T-introduces-variable</td><td colspan="4">T-tutors-variable</td></tr>
<tr><td></td><td colspan="4">T-tutors-via-determinant</td></tr>
<tr><td></td><td colspan="3">T-tutors-determinant</td><td>T-tutors-consequence-value</td></tr>
<tr><td></td><td></td><td>T-moves-towards-PT</td><td>T-moves-towards-PT</td><td></td></tr>
<tr><td></td><td></td><td>T-tutors-determinant</td><td>T-tutors-determinant</td><td></td></tr>
<tr><td>K25</td><td>now</td><td></td><td>and</td><td>but</td><td></td></tr>
</table>

<table>
<tr><td rowspan="6"></td><td colspan="5">T-corrects-variable var=RAP</td></tr>
<tr><td colspan="5">T-tutors-variable</td></tr>
<tr><td colspan="5">T-tutors-via-determinant</td></tr>
<tr><td colspan="3">T-tutors-determinant</td><td>T-tutors-relationship</td><td>T-tutors-value</td></tr>
<tr><td></td><td colspan="2">T-moves-towards-PT</td><td></td><td></td></tr>
<tr><td></td><td>T-tutors-determinant</td><td>T-tutors-determinant</td><td></td><td></td></tr>
<tr><td>K26</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

APPENDIX C

ACKNOWLEDGMENTS AND SCHEMATA

ACKNOWLEDGMENTS AND SCHEMATA

The following sections show the phenomena of using acknowledgments in our tutoring schemata during the DR Stage.

C.1 Acknowledgments while Tutoring TPR

	T-corrects-variable var=TPR			
	T-introduces-variable	T-tutors-variable		
		T-does-neural-DLR		
		T-tutors-mechanism	T-tutors-DR-info	T-tutors-value
K10		**Yes**		**Good**
K11		**Right**		**Right**
K48		**Right**	**Right**	**Great**

	T-corrects-variable var=TPR					
	T-introduces-variable	T-tutors-variable				
		T-does-neural-DLR				
		T-tutors-mechanism			T-tutors-DR-info	T-tutors-value
		Attempt 1	Attempt 2	T-moves-toward-PT		
				T-tutors-mechanism		
K12		**No**	**Yes**	**Yes**		**Correct**

	T-corrects-variable var=TPR				
	T-tutors-variable				
	T-diagnoses-errors	T-tutors-via-deeper-concepts		T-does-neural-DLR	
	T-identifies-problem	T-tutors-determinant	T-tutors-determinant	T-tutors-DR-info	T-tutors-value
		T-moves-to-previous-concepts			
K27	**Omitted negative ack**	**Right**	**Omitted positive ack**		**Right**

C.2 Acknowledgments while Tutoring IS

<table>
<tr><td rowspan="4"></td><td colspan="3">T-corrects-variable var=CC</td></tr>
<tr><td>T-introduces-variable</td><td colspan="2">T-tutors-variable</td></tr>
<tr><td></td><td colspan="2">T-shows-contradiction</td></tr>
<tr><td></td><td>T-presents-contradiction</td><td>T-tutors-contradiction</td></tr>
<tr><td>K10</td><td>Yes</td><td></td><td>Omitted negative ack, Good</td></tr>
</table>

<table>
<tr><td rowspan="4"></td><td colspan="6">T-corrects-variable var=CC</td></tr>
<tr><td>T-introduces-variable</td><td colspan="5">T-tutors-variable</td></tr>
<tr><td></td><td colspan="2">T-shows-contradiction</td><td colspan="3">T-does-neural-DLR</td></tr>
<tr><td></td><td>T-presents-contradiction</td><td>T-tutors-contradiction</td><td>T-tutors-determinant</td><td>T-tutors-DR-info</td><td>T-tutors-value</td></tr>
<tr><td>K11</td><td>Right again</td><td></td><td>You can’t have it both way</td><td></td><td></td><td>Right again</td></tr>
</table>

<table>
<tr><td rowspan="4"></td><td colspan="3">T-corrects-variable var=CC</td></tr>
<tr><td colspan="3">T-tutors-variable</td></tr>
<tr><td colspan="3">T-does-neural-DLR</td></tr>
<tr><td>T-tutors-mechanism</td><td>T-tutors-DR-info</td><td>T-tutors-value</td></tr>
<tr><td>K16</td><td>Right</td><td></td><td>Omitted positive ack</td></tr>
</table>

<table>
<tr><td rowspan="4"></td><td colspan="5">T-corrects-variable var=IS</td></tr>
<tr><td>T-introduces-variable</td><td colspan="4">T-tutors-variable</td></tr>
<tr><td></td><td colspan="4">T-does-neural-DLR</td></tr>
<tr><td></td><td>T-tutors-definition</td><td>T-tutors-mechanism</td><td>T-tutors-DR-info</td><td>T-tutors-value</td></tr>
<tr><td>K47</td><td></td><td>Partly correct</td><td></td><td></td><td>Correct</td></tr>
</table>

C.3 Acknowledgments while Tutoring SV

<table>
<tr><td rowspan="4"></td><td>T-corrects-variable var=SV</td></tr>
<tr><td>T-tutors-variable</td></tr>
<tr><td>T-moves-forward</td></tr>
<tr><td>T-tutors-value</td></tr>
<tr><td>K12</td><td>Absolutely</td></tr>
</table>

	T-corrects-variable var=SV			
	T-tutors-variable			
	T-via-determinant			
	T-tutors-determinant		T-tutors-value	
	Attempt 1	Attempt 1	Attempt 1	Attempt 2
K14	**No**	**Good**	**Omitted negative ack**	**Right**
K20	**Well that's partly correct**		**Correct**	

	T-corrects-variable var=SV		
	T-tutors-variable		
	T-moves-forward	T-explores-anomaly	
		T-presents-anomaly	T-tutors-anomaly
K25	**Omitted positive ack**		**Exactly**
K26	**Right!**		**Right**

	T-corrects-variable var=SV						
	T-introduces-variable	T-tutors-variable					
		T-tutors-via-determinant		T-shows-contradiction		T-explores-anomaly	
		T-tutors-determinant	T-moves-toward-PT	T-presents-contradiction	T-tutors-contradiction	T-presents-anomaly	T-tutors-anomaly
			T-tutors-PT-entry				
K27		**Right**	**Right**		**Not quite**		**Very good**

C.4 Acknowledgments while Tutoring CVP

	T-corrects-variable var=RAP					
	T-introduces-variable	T-tutors-variable				
		T-tutors-via-determinant		T-tutors-via-deeper-concepts		
		T-tutors-determinant	T-tutors-relationship	T-tutors-relationship	T-tutors-relationship	T-tutors-value
K11			**Omitted negative ack**		**Yes**	**Great**

	T-corrects-variable var=RAP		
	T-tutors-variable		
	T-tutors-via-determinant		
	T-tutors-determinant	T-tutors-relationship	T-tutors-value
K13	**Not in the way you seem to think**	**Omitted positive ack**	**Omitted positive ack**

<table>
<tr><td rowspan="6"></td><td colspan="8">T-corrects-variable var=RAP</td></tr>
<tr><td>T-introduces-variable</td><td colspan="7">T-tutors-variable</td></tr>
<tr><td></td><td colspan="7">T-tutors-via-determinant</td></tr>
<tr><td></td><td colspan="2">T-tutors-determinant</td><td colspan="3">T-tutors- determinant</td><td>T-tutors-relationship</td><td>T-tutors-value</td></tr>
<tr><td></td><td>Attempt 1</td><td>Attempt 2</td><td colspan="2">T-tutors-via-deeper-concepts</td><td>T-moves-towards-PT</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>T-tutors-compliance-info</td><td>T-tutors-determinant</td><td>T-tutors-PT-entry</td><td></td><td></td></tr>
<tr><td>K14</td><td></td><td>Omitted negative ack</td><td>Omitted negative ack</td><td></td><td>Omitted negative ack</td><td>Omitted partial correct ack</td><td>No</td><td>Right</td></tr>
</table>

<table>
<tr><td rowspan="6"></td><td colspan="5">T-corrects-variable var=RAP</td></tr>
<tr><td>T-introduces-variable</td><td colspan="4">T-tutors-variable</td></tr>
<tr><td></td><td colspan="4">T-tutors-via-determinant</td></tr>
<tr><td></td><td colspan="3">T-tutors-determinant</td><td>T-tutors-consequence-value</td></tr>
<tr><td></td><td></td><td>T-moves-towards-PT</td><td>T-moves-towards-PT</td><td></td></tr>
<tr><td></td><td></td><td>T-tutors-determinant</td><td>T-tutors-determinant</td><td></td></tr>
<tr><td>K25</td><td></td><td>Omitted near miss ack</td><td>Right</td><td>Omitted positive ack</td><td>No</td></tr>
</table>

<table>
<tr><td rowspan="6"></td><td colspan="5">T-corrects-variable var=RAP</td></tr>
<tr><td colspan="5">T-tutors-variable</td></tr>
<tr><td colspan="5">T-tutors-via-determinant</td></tr>
<tr><td colspan="3">T-tutors-determinant</td><td>T-tutors-relationship</td><td>T-tutors-value</td></tr>
<tr><td></td><td colspan="2">T-moves-towards-PT</td><td></td><td></td></tr>
<tr><td></td><td>T-tutors-determinant</td><td>T-tutors-determinant</td><td></td><td></td></tr>
<tr><td>K26</td><td>Omitted don’t know ack</td><td></td><td>No</td><td></td><td>Omitted positive ack</td></tr>
</table>

BIBLIOGRAPHY

Allen, J., and Core, M. 1997. Draft of DAMSL: Dialog Act Markup in Several Layers, *The Multiparty Discourse Group at the Discourse Research Initiative (DRI) Meeting*, Schloss Dagstuhl, np.

Allen, J., Schubert, L., Ferguson, G., Heeman, P., Hwang, C., Kato, T., Light, M., Martin, N., Miller, B., Poesio, M., and Traum, D. 1995. The TRAINS Project: A Case Study in Defining a Conversational Planning Agent, *Journal of Experimental and Technical Artificial Intelligence*, Vol. 7, No. 1, pp. 7 - 48.

Barnard, D., Hayter, R., Karababa, M., Logan, G., and McFadden, J. 1988. SGML-Based Markup for Literary Texts: Two Problems and Some Solutions, *Computers and the Humanities*, Vol. 22, No. 4, pp. 265 - 276.

Barnard, D., Burnard, L., Gaspart, J., Price, L., and Sperberg-McQueen, C.M. 1995. Hierarchical Encoding of Text: Technical Problems and SGML Solutions, *Computers and the Humanities*, Vol. 29, No. 3, pp. 211 - 231.

Brandle, S. 1998. *Using Joint Actions to Explain Acknowledgments in Tutorial Discourse: Application to Intelligent Tutoring Systems*, Ph.D. Dissertation, Computer Science Department, Illinois Institute of Technology, Chicago, IL.

Brennan, S.E., and Clark, H.H. 1996. Conceptual Pacts and Lexical Choice in Conversation, *Journal of Experimental Psychology: Learning, Memory, and Cognition*, Vol. 22, pp. 1482 - 1493.

Cawsey, A., 1992. *Explanation and Interaction*, Cambridge, MA: MIT Press.

Cho, B., Michael, J., Rovick, A., and Evens, M. 1999. A Curriculum Planner Model for an Intelligent Tutoring System, *Proceedings of the 12th International Florida Artificial Intelligence Research Society Conference (FLAIRS 1999)*, Orlando, FL, pp. 197 - 201.

Evens, M., Brandle, S., Chang, R., Freedman, R., Glass, M., Lee, Y., Shim, L., Woo, C., Zhang, Y., Zhou, Y., Michael, J., and Rovick, A. 2001. CIRCSIM-Tutor: An Intelligent Tutoring System Using Natural Language Dialogue, *Proceedings of the 12th Midwest AI and Cognitive Science Conference (MAICS 2001)*, Oxford, OH, pp. 16-23.

Fraser, N.M. 1997. Spoken Dialogue System Evaluation: A First Framework for Reporting Results, *Proceedings of the 5th European Conference on Speech Communication and Technology*, Rhodes, Greece, pp. 1907 - 1910.

Freedman, R. 1996a. Using Tutoring Patterns to Generate More Cohesive Text, *Proceedings of the 2nd International Conference on the Learning Sciences*, Evanston, IL, pp. 75 - 82.

Freedman, R. 1996b. *Interaction of Discourse Planning, Instructional Planning and Dialogue Management in an Interactive Tutoring System*, Ph.D. Dissertation, Northwestern University, Evanston, IL.

Freedman, R. 2000a. Using a Reactive Planner as the Basis for a Dialogue Agent, *Proceedings of the 13th International Florida Artificial Intelligence Research Society Conference (FLAIRS 1999)*, Orlando, FL, pp. 203 - 208.

Freedman, R. 2000b. Plan-Based Dialogue Management in a Physics Tutor, *Proceedings of the 6th Applied Natural Language Processing Conference*, Seattle, WA, pp. 52 - 59.

Freedman, R., Cho, B., Glass, M., Zhou, Y., Kim, J., Mills, B., Yang, F., and Evens, M., 2001. Adaptive Processing in a Medical Intelligent Tutoring System, *Proceedings of the NAACL Workshop on Adaptation in Dialogue Systems*, Pittsburgh, PA, pp. 17 - 24.

Glass, M. 2001. Processing Language Input in the CIRCSIM-Tutor Intelligent Tutoring System, Moore, J., Luckhardt Redfield, C., and Johnson, W.L. (editors), *Artificial Intelligence in Education*, pp. 210-221.

Grosz, B. 1997. Discourse and Dialogue: Overview, In Varile, G., Zampolli, A., Cole, R., Mariani, J., Uszkoreit, H., Zaenen, A., and Zue, V. (editors), *Survey of the State of the Art in Human Language Technology*, Chapter 6.1, Cambridge, UK: Cambridge University Press.

Hirschman, L., and Thompson, H. 1997. Overview of Evaluation in Speech and Natural Language Processing, In Varile, G., Zampolli, A., Cole, R., Mariani, J., Uszkoreit, H., Zaenen, A., and Zue, V. (editors), *Survey of the State of the Art in Human Language Technology*, Chapter 13.1, Cambridge, UK: Cambridge University Press.

Hovy, E. 1990. Unresolved Issues in Paragraph Planning. In Dale, R., Mellish, C., and Zock, M., (editors.), *Current Research in Natural Language Generation*, San Diego, CA: Academic Press, pp. 17 - 45.

Kim, J., 1998. *The SGML Markup Manual for CIRCSIM-Tutor*. Technical Report, Computer Science Department, Illinois Institute of Technology, Chicago, IL.

Kim, J, Freedman, R., and Evens, M. 1998a. Relationship Between Tutorial Goals and Sentence Structure in a Corpus of Tutoring Transcripts, *Proceedings of the 9th Midwest AI and Cognitive Science Conference*, Dayton OH, pp. 124-131.

Kim, J., Freedman, R., and Evens, M. 1998b. Responding to Unexpected Student Utterances in CIRCSIM-Tutor v.3: Analysis of Transcripts, *Proceedings of the 11th International Florida Artificial Intelligence Research Society Conference (FLAIRS 1998)*, Sanibel Island, FL, pp. 153 - 157.

Kim, J., Glass, M., Freedman, R., and Evens, M. 2000. Learning the Use of Discourse Markers in Tutorial Dialogue for an Intelligent Tutoring System, *Proceedings of the 22nd Annual Conference of the Cognitive Science Society (MAICS 2000)*, Philadelphia, PA, pp. 262 - 267.

Kim, J. 2000. *Natural Language Analysis and Generation for Tutorial Dialogue,* Ph.D. Dissertation, Illinois Institute of Technology, Chicago, IL.

Kim, N. 1989. *CIRCSIM-Tutor: An Intelligent Tutoring System for Circulatory Physiology*, Ph.D. Dissertation, Computer Science Department, Illinois Institute of Technology, Chicago, IL.

Lee, Y. 1990. *Handling Ill-Formed Natural Language Input for a Intelligent Tutoring System*, Ph.D. Dissertation, Illinois Institute of Technology, Chicago, IL.

Mann, W., and Moore, J. 1981. Computer Generation of Multiparagraph English Text, *The American Journal of Computational Linguistics*, Vol. 7, No. 1, pp. 17 - 29.

Mann, W., and Thompson, S. 1988. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization, *Text*, Vol. 8, No. 3, pp. 243 - 281.

McKeown, K. 1985. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*, Cambridge, UK: Cambridge University Press.

McRoy, S, Ali, S., Restificar, A., and Channarukul, S. 1999. Building Intelligent Dialogue Systems, *Intelligence*, Vol. 10, No. 1, pp. 14 – 23.

Mellish, C., O'Donnell, M., Oberlander, J., and Knott, A., 1998. An Architecture for Opportunistic Text Generation, *Proceedings of the 9th International Workshop on Natural Language Generation*, Niagara-on-the-Lake, Ontario, Canada, pp. 28 - 37.

Mills, B. 2001. Using the Atlas Planning Engine to Drive an Intelligent Tutoring System: CIRCSIM-Tutor Version 3, *Proceedings of the 14th International Florida Artificial Intelligence Research Society Conference (FLAIRS 2001)*, Key West, FL, 211 - 215.

Mittal, V.O. 1999. *Generating Natural Language Descriptions with Integrated Text and Examples*, Mahwah, NJ: Lawrence Erlbaum Associates.

Moore, J., and Paris, C. 1993. Planning Text for Advisory Dialogues: Capturing Intentional and Rhetorical Information, *Computational Linguistics*, Vol. 19, No. 4, pp. 651 - 694.

Moore, J., Lemaire, B., and Rosenblum, J. 1996. Discourse Generation for Instructional Applications: Identifying and Exploiting Relevant Prior Explanations, *The Journal of the Learning Sciences*, Vol. 5, No. 1, pp. 49 - 94.

Moser, M., and Moore, J.D. 1995. Investigating Cue Selection and Placement in Tutorial Discourse, *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, Cambridge, MA, pp. 130 - 135.

Nakatani, C., Grosz, B., Ahn, D., and Hirschberg, J. 1995. Instructions for Annotating Discourses, Technical Report Number TR-21-95, Center for Research in Computing Technology, Harvard University, Cambridge, MA.

Nakatani, C., and Traum, D. 1998. *Draft: Discourse Structure Coding Manual*, *The 3rd Discourse Resource Initiative (DRI) Meeting*, Chiba, Japan, np.

Passonneau, R. 1994. *Protocol for Coding Discourse Referential Noun Phrases and Their Antecedents*, Revised November 1994, Technical Report, CARD Project, Department of Computer Science, Columbia University, New York City, NY.

Patten, T. 1988. *Systemic Text Generation as Problem Solving*, Cambridge, UK: Cambridge University Press.

Polifroni, J., Seneff, S., Glass, J., and Hazen, T.J. 1998. Evaluation Methodology for a Telephone-Based Conversation System, *Proceedings of the 1st International Conference on Language Resources and Evaluation*, Paris, France: European Language Resources Association, pp. 43 - 49.

Pryor, L., and Collins, G. 1994. Opportunities: A Unifying Framework for Planning and Execution, *Proceedings of the 2nd International Conference on Artificial Intelligence Planning Systems*, Chicago, IL, pp. 329 - 334.

Ramachandran, K. 1994. *Lexical Choice in Natural Language Text Generation for an Intelligent Medical Tutoring System*, M.S. thesis, Illinois Institute of Technology, Chicago, IL.

Reichman R. 1985. *Getting Computers to Talk like You and Me*, Cambridge, MA: The MIT Press.

Rovick, A., and Brenner, L. 1983. HEARTSIM: A Cardiovascular Simulation with Didactic Feedback, *The Physiologist*, Vol. 26 No. 4, pp. 236-239.

Rovick, A., and J. Michael. 1986. CIRCSIM: IBM-PC Computer Teaching Exercise on Blood Pressure Regulation, *Proceedings of the XXX Congress of International Union of Physiological Sciences*, Vancouver, p. 138.

Sanders, G. 1995. *Generation of Explanations and Multi-Turn Discourse Structures in Tutorial Dialog, Based on Transcript Analysis*, Ph.D. Dissertation, Illinois Institute of Technology, Chicago, IL.

Schiffrin, D. 1987. *Discourse Markers*, Cambridge, UK: Cambridge University Press.

Seu, J. 1992. *The Development of an Input Understander for an Intelligent Tutoring System Based on a Sublanguage Study*, Ph.D. Dissertation, Illinois Institute of Technology, Chicago, IL.

Shim, L. 1991. *Student Modeling for an Intelligent Tutoring System: Based on the Analysis of Human Tutoring Sessions*, Ph.D. Dissertation, Illinois Institute of Technology, Chicago, IL.

Sperberg-McQueen, C.M., and Burnard, L. 1994. *Guidelines for Elecronic Text Encoding and Interchange*, Chicago and Oxford: Text Encoding Initiative.

Sperberg-McQueen, C.M., and Burnard, L. 1995. The Design of the TEI Encoding Scheme, *Computers and the Humanities*, Vol. 29, No. 1, pp. 17 - 39.

Sperberg-McQueen, C.M., and Huitfeldt, C. 1999. Concurrent Document Hierarchies in MECS and SGML, *Literary and Linguistic Computing*, Vol. 14, No. 1, pp. 29 - 42.

Spitkovsky, J.A., and Evens, M. 1993. Negative Acknowledgments in Natural Language Tutoring Systems, *Proceedings of the 4th Midwest AI and Cognitive Science Conference*, pp. 41 - 45.

Stent, A., and Allen, J. 1997, *TRAINS-96 System Evaluation*, TRAINS Technical Note 97-1, Department of Computer Science, University of Rochester, Rochester, NY.

Walters, J. 1992. *The Markup of Anaphora and Referenced Items for Tutorial Transcripts*, Unpublished Paper, Computer Science Department, Illinois Institute of Technology, Chicago, IL.

Wong, K., Tsang, C., and Chan, W. 1988. SHERLOCK – A System for Diagnosing Power Distribution Ring Network Faults, *Proceedings of the 1st International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Tullahoma, TN, Vol. 1, pp. 109 - 113.

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

Yang, F., Kim, J., Glass, M., and Evens, M. 2000a. Lexical Usage in the Tutoring Schema of CIRCSIM-Tutor: Analysis of Variable References and Discourse Markers, *Proceedings of the 5th International Conference on Human Interaction with Complex Systems (HICS 2000)*, Urbana, IL, pp. 27 - 31.

Yang, F., Kim, J., Glass, M., and Evens, M. 2000b. Turn Planning in CIRCSIM-Tutor, *Proceedings of the 13th International Florida Artificial Intelligence Research Society Conference (FLAIRS 2000)*, Orlando, FL, pp. 60 - 64.

Yang, F., Glass, M., and Evens, M. 2001. Evaluation of the Turn Planner in CIRCSIM-Tutor, *Proceedings of the 17th International Conference on Advanced Science and Technology (ICAST 2001)*, Chicago, IL, to appear.

Zhang, Y. 1991. *Knowledge-Based Generation for an Intelligent Tutoring System*, Ph.D. Dissertation, Illinois Institute of Technology, Chicago, IL.

Zhou, Y., and Evens, M. 1999. A Practical Student Model in an Intelligent Tutoring System, *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 1999)*, Chicago, IL, pp. 13 - 18.

Zhu, V., Seneff, S., Glass, J., Hetherington, L., Hurley, E., Meng, H., Pao, C., Polifroni, J, Scholming, R., and Schmid, P. 1997. From Interface to Content: Translingual Access and Delivery of On-line Information, *Proceedings of the 5th European Conference on Speech Communication and Technology*, Rhodes, Greece, pp. 2227 - 2230.