Understanding Discourse Conventions in Tutoring

Beverly P. Woolf
David D. McDonald

COINS Technical Report #85-22
30 August 1985

Department of Computer and Information Science
University of Massachusetts
Amherst, Massachusetts 01003 [1]

Published in the *Proceedings of the Expert Systems in Government Symposium*, sponsored by IEEE & Mitre Corp, Oct 23-25, 1985, McLean, Virginia.

[1] This work was supported in part by Rome Air Development Center (RADC) Grant Number SU353-9023-3

1. Abstract

Speakers have expectations about listeners that enable them to produce coherent discourse. These expectatations should be incorporated into machine tutors so that they too can generate expectations about their student users. We intend to show how expectations can be used to anticipate a user's *choice* of responses based upon the dynamics of the speaker/listener interaction. The paper describes a way to formalize the constraints and operations in discourse and how to use these constraints to transform interpretation and speech act knowledge into computational elements, such as plans and rules.

2. Discourse conventions

One of the largest theoretical stumbling blocks in the design of effective machine discourse systems is the lack of an adequate representation or understanding of discourse conventions. *Human* speakers employ subtle linguistic cues to shift topics or provide supplementary knowledge. Listeners use these cues to set up expectations about the underlying structure of the discourse and to relate current utterances to preceding ones. The listener's expectations are what the speaker tries to anticipate and to deliberately control.

The aim is to build a machine speaker that represents these conventions and responds to its user based on inferences about a model of the user or the discourse history. Early computer discourse systems controlled the flow of discourse producing canned texts that were typically the same regardless of the user's knowledge or the discourse history.[1,2] More recent interface systems have begun to tailor their responses to the user and discourse context.[3,4] The basic problem in designing machine discourse is how to make inferences about the user and how to have these inferences govern the form of the text produced*.[2]

For instance, the adjustments that a *computer tutor* would make are dependent upon its specific experience with a student and a variety of experiences would lead to a variety of responses. Thus we would want a computer tutor to interact with a knowledgeable student in a way that is fundamentally different, both in style and content, from the way it would engage a confused one. It is not intended that the computer simply produce correct answers in response to a student's wrong answers; rather before responding to a wrong answer, the machine should resolve issues such as:

---

[2] *We recognize that a machine cannot know with certainty what a listener knows, neither can a listener know what a machine knows; a machine cannot be omnicient or clairvoyant. However, the machine can deduce, on the basis of evidence, something about what the listener assumes. These assumptions can be used to govern the form of the text generated.

>> when and how to stop to explain the wrong answer;

>> whether it is preferable to explain the error or to start a lengthy exploration of the student's knowledge;

>> whether to allow uncertainty about the student's knowledge to persist temporarily while it explores a potential misconception; and

>> how hard it should work to understand why a student answered a question incorrectly or how much effort should be exerted to resolve questions about the student's presumed knowledge or misconceptions.

Though many areas of research on understanding discourse conventions are interesting and several problems are ripe for a solution, we have focused on the role of the speaker because we want to study discourse in the context of tutoring. In tutoring, perhaps more than in other types of discourse, the speaker (the tutor) chooses conversational moves based on the responses of the student. There are many occasions in which a tutor will interpret what a student says and "read into" the answer additional material to update his current model of the student's knowledge. Based on these considerations, a <u>human</u> tutor would adjust the discourse; a machine tutor should do the same. Tutoring provides us with a rich, well-contained field in which to study discourse conventions from a speaker's point of view.

A second reason to work in tutoring is because of the wealth of research on language and tutoring in our environment at UMass. One research effort has focused on natural language comprehension,[5,6] generation,[7] discourse control,[8] and legal reasoning.[9] Another effort has focused on tutoring discourse[10] and a related effort at Yale has focused on student errors[11] and the learning and teaching of Pascal looping constructs.[12,13] As a result of this extended research environment we have been able to formalize knowledge about human tutoring protocols, understand the epistemology of Pascal looping constructs and have a realistic way to accumulate a rich model of the user. Therefore, we have a domain where the system can select the appropriate content to discuss with the student based on an understanding of its audience, in addition to in-depth knowledge of language and tutoring.

This paper discusses several research areas being pursued, including problems to be solved, recent research in the area, and conclusions that might be drawn about discourse conventions as a result of our studies. It also presents an example of how this computational model is being used to build a robust tutor for Pascal programming. Some of the research areas to be discussed are

o <u>discourse control</u> - how to focus on appropriate topics, errors or examples;

o <u>knowledge representation</u> - how to create a data structure for the codification and interpretation of utterances; and

o <u>natural language generation</u> - how to produce appropriate text for the situation.

3. Qualitative reasoning about discourse

Discourse is often described in qualitative terms:

"the speaker was [helpful] [abrupt] [angry]"
"the student was [confused] [unprepared] [sharp]"
"the topic was [important] [understood] [trivial]"

Awareness of the *effect* of an utterance on the overall interpretation of the discourse is also described in qualitative terms such as:

"the example was useful"
"the argument was weak"

Figure 1 contains other analyses of discourses from a psychologist, computational linguist and psychiatrist. In each case, the researchers have teased out implicit rules of discourse based on how a speaker should interpret his listener's level of knowledge or understanding. The impact of these rules suggests that people would be better speakers or tutors if they followed implicit rules. To represent these rules in the machine tutor and to enable it to demonstrate the same aspects of good discourse conventions alluded to in the analyses, is the present goal of our research.

Toward this end, we have begun to capture several of the features we recognized in the analyses of Figure 1. For example, the analyses refer to inferences (in italics) about the student's prior knowledge, or the "mutual" knowledge of the two conversants. Our system will recognize qualitative inferences such as when a ***topic is generally known**, ***student has background information**, or ***student is confused**.

Also represented in the analyses are qualitative inferences about knowledge, particularly mutual knowledge:

"what the student already knows,"
"deeper level of analysis,"
"shared focus of attention."

These inferences are not defined or explained in the analyses and their casual use suggests a degree of subjectivity about quantities such as "knowledge", "confusion", or "attention". In addition, to understand these metrics the reader is expected to understand processes such as "*building* on what a student knows", "*raising* issues", and "*establishing* a shared focus of attention."

Representing these complex discourse conventions and metrics requires using qualitative expressions of knowledge. There is evidence from other fields that qualitative reasoning and representations are useful: e.g., teaching,[17] Artificial Intelligence (AI),[18] and the domain of physics.[19] Tracing qualitative inferences in a discourse model will be relatively intractable, compaired with, for example, tracing speech acts. Qualitative inferences will be multiplexed between and within other streams of inferences, some being initiated or continued while others are simultaneously being started. The result is that the intent of a particular stream of inferences can become confounded. Yet, we suggest that it is worth the

From Analysis and Synthesis of Tutoring Discourse:[14]

> [A tutor] builds on what *the student already knows* [and] can question him about *his previous knowledge*. Then he can teach new material by relating it to that *previous knowledge* [pg 50].
>
> [A tutor] can respond directly to student errors, . . . question him to *diagnose the confusion* and can provide *relevant information* to *straighten him out*. [pg 50].
>
> The question *raised the issue* of . . . moving [the discourse] to a *deeper level of analysis* than made so far [pg 67].

From Plain Speaking: A Theory and Grammar of Spontaneous Discourse:[15]

> Much of the implicit *knowledge speakers and listener s share* is knowledge of the particular components of various conversational moves -- what kinds of utterances must be made in order to fulfill various discourse functions [chapter 3, pg 1].

From Parental Communication Deviance and Schizophrenia: A Cognitive-Developmental Analysis:[16]

> A failure on the part of the speaker to *establish and maintain a shared focus of attention* with one's listener [pg 68].
>
> A tendency to *equivocate concerning one commitment* to one's statements and a tendency to *vacillate concerning the content* of one's statement [pg 68].
>
> A *lack of specificity* with regard to the referent, *unexplained contradictions* . . . inappropriate responses suggestive of a failure to grasp the *intent of a question by the interlocutor* [pg 62].
>
> [A failure] to take into account the *cognitive needs* of the listener [pg 62].

Figure 1: Analyses of discourse conventions from the literature.

---

effort to try to make qualitative inferences because they provide a more powerful representation of the intention of the speaker than do speech acts. In particular, they are more predictive of subsequent utterances and can be used to propose and elucidate a speaker's intent or the direction of the discourse. We suggest that tracing implications to evaluate the effect or goal of a discoure provides a sound framework for understanding discourse.

4. Maxims of tutoring

We suggest that tutoring consists of following certain maxims of discourse conventions (in the same sense used by Grice[20]) and we analyze research such as that in Figure 1 to identify these maxims. We expect to be able to evaluate the reasonableness the tutoring discourse we produce by recognizing whether the maxims are satisfied. In this section we define some tutoring maxims and outline how we intend to monitor discourse based on a notion of maxim satisfaction.

In order to model the qualitative effect of utterances we first define conversational move-classes as groups of utterances that have the same rhetorical effect, such as **question topic, summarize topic, acknowledge correct answer** and **provide example**. We suggest that a tutor's choice of conversational move indicates his (its) "intention" in the sense that a move sets up expectations in a listener. For instance, a conversational move such as **make accusation** typically would elicit negative responses from a listener. For instance, consider some queries a tutor might pose to a student about loop execution in a Pascal program, as suggested in Figure 2.

(*provide-example*)
(*question-hypo*)
If the input is 10, how many times would your loop execute?

(*question-topic-value*)
Do you know how many times your loop would execute?

(*make-a-claim*)
I bet you don't know how many times your loop will iterate.

(*make-an-accusation*)
You couldn't possibly understand loop execution.

Figure 2: Reading implications from utterances.

Each sentence has a similar locutionary force, yet each conveys a different intention on the part of the speaker. Further, there is a continuum such that a tutor may couch his statements at any place along the higher end. The implication drawn would be of close attention, even commitment, to the student. On the other hand, a statement at the lower end would imply non-commitment, non-involvement and possibly antagonism. Relative to the four utterances above, we say that use of a phrase representing a certain point on the scale implies that the tutor chose not to phrase the utterance by another expression lower on the list. This reasoning on the part of the listener is licensed by the Gricean[20] maxim of manner. Grice has defined very general maxims for discourse, that are evocative, yet not detailed enough to provide a basis for a computational theory of discourse by themselves.

Our goal is to propose a computational model of tutoring discourse that elucidates and refines these maxims and links them with specific conversation moves. Ultimately inferences about conversational moves will be used to guide the system's choice of utterances. The tutoring maxims that we propose are derived from Gricean maxims for discourse and are tailored for tutoring. They include:

Quality: be committed and interested in the student's knowledge;
be supportive and co-operative;
do not take the role of "antagonist"

Quantity: be specific and perspicious;
use a minimum of attributes to describe a known concept;

Relation: be relevant;
find a student's threshold of knowledge;
bring up new topics and viewpoints as appropriate

Manner: be in control;
allow a student to determine a new topic;
allow context to determine a new topic.

Figure 3 further discriminates these maxims in terms of move-classes that support each one. Maxims are listed on the left and the sequence of move-classes that supports them on the right. By being attentive to moves during discourse, a system can monitor its own behavior and guide subsequent moves so as to be consistent with the maxims of good tutoring. The system can identify maxims on the left, and invoke the move-class on the right that are associated with them. For instance, if the system plans to be more organized, it can **outline topics, introduce topics, terminate topics**, and then **review topics**. Alternatively, if the system needs to record the "effect" of its actions on the listener, it can list the actions taken by the tutor and determine if its own actions are consistent with certain maxims. For instance, if the interaction with the student could be described as an ordered set of utterances, such as **question student, acknowledge answer, propose misconception**, and **provide example**, the overall effect of the actions could be to determine the student's threshold of knowledge. Whether or not that threshold was determined is a non trivial, and as yet unanswerable, question.

The table in Figure 3 can be read in two directions: from left to right it allows the system to select a maxim and plan subsequent tutoring discourse by invoking the associated sequence of move-class; from right to left it provides an abstraction of the system's activities so that the effect of the system, interms of the expectation of the listener and the maxims of good tutoring, can be expressed.

Maxims	Conversational move-classes
Be co-operative: –work with student	explain topics summarize topics clearly terminate topics review or repeat topics release control of dialogue
Be committed: –show interest	acknowledge answer explain topics
–support student	outline topics introduce topics
Be relevant: –find student's threshold	question student evaluate student hypotheses propose and verifymisconceptions
–teach at threshold	provide analogy example summarize topic
Be organized: –structure domain	outline topics introduce topics terminate topics review topics
–complete information	clearly terminate topics teach subtopics after topic teach attributes after topic teach subgoals after goal
Be in control: –strictly guide discourse	introduce topic describe topic question student

Figure 3: Tutoring Maxims supported by move-classes.

5. Maxims and move-classes

In order to computationally associate maxims with sequences of move-class, we need to make inferences about the qualitative effect of each move-class on the discourse. To do this, we suggest the effect that each move-class has on discourse entities, such as topics or a student's knowledge. Each conversational move is defined as a data structure and two inferences are made from it. The first inference or *implication*[*20h] is linked *directly* to a move-class. It represents an assesment made about the move-class itself and is fixed and non-negotiable. The second kind of inference or *global implication* is linked to indirectly on sequences of move-classes. It represents an inference made about the effect of several move-classes and is volatile over the life of the dialogue. Global inferences are dynamically modified by the sequence of move-classes. Each inference type is discussed below.

5.1 Implications

Implications are bound to the move-class itself. They exist independent of the "truth" or "meaning" of the utterance and define what the listener receives in addition to the spoken words. In our model, a qualitative implication bound to the move-class is placed on a stack whenever its move-class is invoked.

Typical objects in our ontology

(define-move-class QUESTION-TOPIC
Evidence:
Q+ *topic is important*
Q+ *topic is within threshold of knowledge*
Q+ *topic is learnable through discourse*)

(define-move-class PRESENT-TOPIC
Evidence:
Q+ *topic is generally known*
Q+ *topic is background information*
Q++ *topic is less important*
Q++ *topic is impact material*)

Figure 4: Implications bound to move-classes.

Figure 4 lists the implications bound to two move-classes, **question topic** and **present topic**. For instance, if a tutor questions a student about a topic, the implications of this

[1] *An implication was originally called an implicature by Grice and was attached to specific words, not to groups of words.

are that the tutor 1) knows (or is trying to learn) the student's threshold of knowledge, 2) assumes the student can answer the question, 3) thinks the topic is important or is learnable through the discourse. These implications can be assumed by a listener independent of the content of the query.

We speak of implications in the same sense as Grice's implications, but in reference to sequences of words perceived as a single conversational move. Grice's implications originally referred to inferences made over single words. For example, the italicized words in Figure 5 have explicit implicatures. The word *and* in the first sentence carries an implication that the activity of going to jail preceeded, and possibly caused the second activity, that George became a criminal. The use of the word *tried* in the second sentence carries with it an entailment that Millie failed to swim the English channel, and the use of the phrase *one leg* in the third sentence, implies that the speaker *does not* in fact have two legs.

---

1) George went to jail *and* became a criminal.
2) Millie *tried* to swim the English channel.
3) I have *one* leg.

Figure 5: Implicatures in text.

Implications, as we use them, define each participant's common-sense reasoning about a conversational move. They include the desiderata normally accepted by a rational discourser. We would like to think that implications embody a speaker's motivation, intention, and involvement in the discourse.

5.2 Global implications

Global implications are based on extended reasoning about sequences of move-classes. They include assesments such as ***student is confused, *topic is known** or ***misconception is resolved** and are modified with each new tutor/student interactions. Global implications are uncertain and represent the system's best estimate about the state of affairs of knowledge of the student or topic at the current time. Whereas, implications were known with certainty at the time a move-class was invoked, global implications require reasoning under uncertainty to deduce which one a number of competing global implications might take effect. Reasoning with uncertainty must allow for the accumulation of support for or against a number of global implications.

Figure 6 presents an example of how global implications can be inferred over the course of a tutoring dialogue. In the example, the tutor's goal is to determine the breadth of the student's understanding of primitive topics about Pascal loops. Three questions are presented that might be asked of a student who had submitted an incorrect Pascal program. After the first correct response certain immediate inferences can be made; the student has definitional knowledge of the topic, the topic is generally learnable through other efforts (i.e., textbooks or lectures), and the topic was studied as

Tutor: Do you know that *GRADE* in line 8 is a control variable?
Student: Yes

; *IMPLICATIONS*
; *student_has_definitional_knowledge
; *topic_is_generally_known
; *topic_is_learnable_elsewhere
; *topic_is_background_material

Tutor: Good. What is the value of grade before leaving the loop in line 13?
Student: 9999

; *IMPLICATIONS*
; *topic_is_generally_known
; *topic_is_learnable_elsewhere
; *topic_is_background_material

; *GLOBAL IMPLICATIONS*
; *student/domain_agreement
; *student_knows_the_topic

Tutor: That's right. What is the value of grade after leaving the WHILE loop, in line 13?
Student: I don't know.

; *IMPLICATIONS*
; *tell_tale_signs_lack_of_knowledge
; *student_does_not_know_the_topic

; *GLOBAL IMPLICATIONS*
; *topic_is_on_student's_threshold
; *student/domain_disagreement
; *student_is_confused

Figure 6: Analysis in a tutoring interaction.

---

background material. After two correct answers, the tutor has reinforced its initial evaluation of the student's knowledge but now is licenced to make more extensive inferences about the student or the topic. In this case, the global implication might be that there is some agreement between the student's information and the domain knowledge base. This inference is possible because evidence from the additional correct answer provides support for the global implication.

The student's third response is wrong and the tutor now is forced to reverse its current evaluation. After a single wrong answer, several immediate implications are available since they are bound to the conversational move: either the student does not know the material in question or he made a careless error. If we assume the former and recognize that the wrong answer came on the heels of two correct answers, we have a more complex implication: now it is possible to say that the topic might lie on the student's threshold of knowledge. This is because the student knows some attributes

***Student-has-tell-tale-signs-of-knowledge** - assume student has indirectly used the topic.
***Student-has-definitional-knowledge** - assume student knows the definition of the topic.
***Student-has-background-information** - assume student knows topic through prior experience.
***Student-is-actively-forming-knowledge** - assume student is forming a model of the information.
***Student-is-confused** - assume student is confused.
***Student-knows-the-topic** - assume student has used topic correctly.
***Student-understands** - assume student understands the topic.
***Student/domain-agreement** - assume agreement between student's knowledge and domain knowledge.
***Student s-knowledge-threshold-known** - assume student's threshold of knowledge is known.

Figure 7: Global implications about a student.

***Topic-is-important** -- assume topic is important.
***Topic-is-generally-known** -- assume topic is generally known.
***Topic-is-learnable-elsewhere** -- assume topic hassume been learned at another time.
***Topic-is-learnable-through-dialogue** -- assume topic can be learned during the dialogue.
***Topic-is-on-student-threshold** -- assume topic lies on the student threshold of knowledge.
***Topic-is-background-material** -- assume topic wassume learned before the discourse.
***Topic-is-less-important** -- assume topic is less important.
***Topic-is-irrelevant** -- assume topic is irrelevant.
***Topic-was-complete** -- assume topic was fully developed.
***Topic-has-been-popped** -- assume topic lies at a higher level in knowledge base.

Figure 8: Global implications about topic.

about the topic, control variables, e.g., its definition and value before loop exit, yet he does not know at least one attribute e.g., its value after loop exit.

The example shows how the system can infer more general knowledge about the student and the topic by using global implications. Since global interpretations are made over several interactions, additional evidence brought from earlier responses, can be weighed along with current implications to generate a more global view. In this way the system can achieve a broader view of student knowledge and topic complexity.

Additional global inferences that we expect the machine to make are presented in Figures 7 and 8. In each figure, global implications are listed on the left and the assessments of which they are a "gloss" are on the right.

5.3 Managing discourse using global implications

One of the primary objectives of this model is to use support for global implications to influence discourse behavior. Discourse management is handled by an ATN-like mechanism that allows both default and exceptional behavior.[10] Default behavior is based on traversal of the arcs of the ATN; exceptional behavior is achieved by activated meta-rules that move the system from one set of discourse states to a new set of states. Transitions within discourse states define the system's default behavior. For instance, the default response to a wrong answer might be a two state sequence: **explicitly acknowledge incorrect answer**, followed by **teach topic attribute**. This sequence can be abandoned if a meta-rule fires and replaces it with a sequence such as **provide example** and **question topic**. A meta-rule is a structure defined by preconditions, prior states, actions, and post-processing actions (see Appendix 1). Preconditions are largely built from global implications. Once a global implication passes threshold and triggers a meta-rule, the discourse manager will move to a new state sequence by a method described in detail by Woolf and McDonald[10].

Discourse behavior is determined by meta-rules, which in turn are enabled by global implications reaching threshold. Global implications will reach threshold as a result of support from the on-going discourse. The system supports global implications in a process that is analogous to the read-eval-print cycle of LISP. The top-level "thinking" of the machine is suggested in Figure 9.

STEP1: Tutor behaves according to default state sequence (consistent with current implications).
A) generate text
B) parse student's response

STEP2: Tutor identifies implications of student's utterance and endorses evidence for global implications

STEP3: Some endorsed global implications may reach threshold.

STEP4: Global implications at threshold may trigger meta-rules taking the system to a new state sequence.

STEP5: Go to 1.

Figure 9: Steps to manage discourse.

After a student responds, implications from his response will be placed on a stack and certain global implications will be activiated. Global implications that gain support from the newly activated implications are endorsed, i.e., given reasons to be believed or disbelieved.[21] The endorsements are associated with an applicability condition: "correct answer indicates correct information", is always possible when the response is correct; "correct answer indicates a quess" is applicable when the response is correct but earlier responses were wrong; and "could be a mistake" is applicable for any response. Global implications that pass beyond a threhold level are viable assesments of the topic or student; they can be used to activate changes in the system's discourse behavior. Some endorsements are *positive*, meaning they support the interpretation with which they are associated. Others are *negative*, meaning they provide reasons to disbelieve their associated interpretations.

In sum, the state of affairs of the discourse is given by support for or against global implications. When evidence for a change in interpretation exceeds threshold and the system has reason to endorse a new interpretation, it takes action and changes its teaching strategy. Customized tutoring behavior is achieved through recognition of the effects of these global implications.

6. Proposed Tutoring Discourse

As an example of the kind of high-performance tutoring system we intend to build using interpretation knowledge, we present a scenario of how our program would tutor a student in Pascal. Figure 10 shows the kind of problem students receive in our department's introductory Pascal course. Below the problem statement is a program actually written by a student.

PROBLEM: Write a program that finds the average grade for a student who types his grades in at the keyboard. After the last grade is typed in the student will type 9999. Please print out the average grade.

```
Program Student29 (input, output);
Var
    sum, num, grade, ave : integer;
Begin
    sum := 0;
    num := 1;
    read (grade);
    while grade <> 9999 do
      begin
       read (grade);
       sum := sum + grade;
       num := num + 1
      end;
     while grade = 9999 do
        begin
        ave := sum/num;
         writeln (ave)
        end;
  end.
```

Figure 10: A Student Program.

The program is syntactically correct but does not produce the desired result. [*,1] It reveals at least four underlying misconceptions about control variables, looping constructs, and flow of control. Cognitive studies of programming[13,22] suggest for instance, that the student may believe that:

>> the order of operations inside the loop should be READ/PROCESS (a Pascal WHILE loop is designed to be PROCESS/READ);

>> the value of a variable inside the loop is distinct from its value outside the loop;

>> the loop construct does not test the last value of the control variable; and

>> a single iteration of a sequence of commands requires a looping construct.

There is evidence for each misconception.[**,2] The task of the discourse that the tutor engages in is to resolve the ambiguity among the possible misconceptions and to focus the discourse on the repair of a single misconception. The discourse in Figure 11 is an example of what we expect our tutor to do and how it might unravel the student's misunderstandings and make him "verbalize" his malformed ideas. In order to generate that discourse, the tutor:

focused on several topics in succession (e.g., control variables and while loops);

handled several errors (e.g., value the control variable and the number of loop iterations); and

customized new examples to present to the student (e.g., hypothetical program and a new problem).

---

[1] The program is flawed in several ways. The value of the "read" command in line 7 will not be processed, because the "read" command in line 10 immediately replaces it with a new value. The second loop will never terminate because the control variable "grade" is never modified inside the loop body. We also suggest, that the student assumes that the second loop will execute only once since its only action is to print out the average. We call such a loop an "if-loop" and suggest that the student has confused the "if" conditional construct with the family of looping constructs.

[2] For instance, the fact that "read (grade)" in line 10 preceeds the processing of grade in line 11 is evidence for the first misconception. In addition, the use of two loops, both controlled by the same variable, "grade", is evidence for the second and third misconceptions since they suggest that the student does not realize that the value of "grade" will be available at the end of the first loop and therefore does not have to be retested.

1. *Tutor*: Do you know that *GRADE* in line 8 is a control variable?
   Student: Yes

2. *Tutor*: O.K. What is the value of *GRADE* immediately before leaving the loop in line 13?
   Student: 9999

3. *Tutor*: Right, and what is its value immediately after leaving the WHILE loop, in line 13?
   Student: I don't know.

4. *Tutor*: O.K, suppose the values read in were 1 5 9 10 9999.
   How many times would your WHILE loop execute.
   Student: I don't know. Maybe 5 times.

5. *Tutor*: Well, what would be the value of *GRADE* immediately after leaving the loop in line 13?
   Student: I don't know.

6. *Tutor*: You seem to have a misconception. The loop would execute 4 times. Although 9999 would be read inside the loop during the fourth iteration and tested at the top of the fifth iteration, the entrance test for that iteration would fail since *GRADE* is now unequal to 9999. The fifth loop execution would never occur and control would pass to line 14 immediately after the loop.

   You probably assumed that the value of the variable inside the loop was different from its value outside the loop. In fact, whether inside or outside of the loop, variables in a WHILE loop have the same value. If *GRADE* equals 9999 before the loop terminates, it will still equal 9999 after the loop terminates. Testing the value of *GRADE* immediately after the WHILE loop is superfluous.

   Now, let me give you a new problem: Compute and print the average number of hours worked each day by a student employed during a month if hours per day is typed in and averaged by the program. Assume that hours per day are typed on a single line and followed followed by -1.

Figure 11: Proposed tutoring discourse for the Program in Figure 9.

Note that in Figure 11 the tutor asks one question (line 1) to establish that both it and the student share a common vocabulary about control variables. In the next two questions (lines 2-3) the tutor asks enough questions concerning misconceptions about variable values and control flow to establish that the student does, in fact believe that the value of *GRADES* is not available after the first loop terminates. In line 4 the tutor presents some example input custom-tailored to the problem and the student's history in order to verify its hypothesis that the student did not realize that the value of *GRADES* was available after the loop exited. Based on the student's response thus far the tutor (line 6) explains its diagnosis of the misconception in terms of characteristics of the presenting program: *GRADES* had a value of 9999 when the first loop terminated and after it terminated *GRADES* will retain that value. Thus, *GRADES* is available between the first and second loops.

7. Example Generation

Generating examples is a key feature of the proposed system and we will be working closely with Rissland[23] to enrich explanations with examples. Generating illuminating examples tailored to a student's level of knowledge, requires knowing the student's activities, background and particularly his history of errors. Generation and modification of examples is a powerful technique both to refine the model of the student and as a tool for defining the student's level of understanding of the domain.[24] We propose to extend and apply previous work on *constrained example generation* in which new examples are generated from old primarily by domain-specific modifications of existing examples. Some of these constraints are generated by general principles such as "Look at extreme cases," "Look at a simpler case." Other constraints will come from specific knowledge of an individual student, his context, past history, cognitive style, etc.

8. Summary

We have suggested a way to represent the implications and intentions of a speaker as distinct from representing actual utterances. In our model, a computer tutor makes inferences about a student's knowledge or domain topics based on constraints about the type of utterances spoken. Support for or against each implication is given by the type of conversational move. The tutor's control structure allows the systems to review or redirect the discourse based on the system's evalaution of implications it can make about the student's knowledge or the topic. We expect that evaluating implications will allow us to make predictions about managing subsequent discourses and judgements about the quality of the current discourse.

9. References

1. W. Mann, J. Moore, and J. Levin, "A comprehension model for human dialogue," *International Joint Conference on Artifical Intelligence*, 1977.

2. K. McKeown, "Generating relevant explanations: Natural language responses to questions about data base structure," *National Proceedings of the Association of Artificial Intelligence*, 1980.

3. T. W. Finin, "Providing help and advice in task oriented systems," *Proceedings*

IJCAI-83, Karlsruhe, W. Germany, 1983.

4. R. Wilensky, "Talking to UNIX in english: An overview of UC," *Proceedings AAAI-82*, Pittsburgh, PA, August 1982.

5. W. Lehnert, M. Dyer, P. Johnson, C. Yang, and S. Harley, "Boris--An experiment in in-depth understanding of narratives," *Artificial Intelligence*, 20, pp. 15-62, 1983.

6. W. Lehnert, and C. Loiselle, "An introduction to plot units," COINS Technical Report 84-13, Computer and Information Sciences, University of Massachusetts, Amherst, 1984.

7. D. McDonald, and J. Pustejovsky, "TAGs as a grammatical formalism for generation," *Proceedings of the Association of Computational Linquistics*, 1985.

8. B. Woolf, *Context-dependent planning in a machine tutor*, Ph.D. Dissertation, Computer and Information Sciences, University of Massachusetts, Amherst, MA, 1984.

9. K. Ashley, and E. Rissland, "Toward modelling legal arguments," *Proceedings of the Second International Congress, Logica, Informatica, Diritto, Automated Analysis of Legal Texts*, Florence, Italy, 1985.

10. B. Woolf, and D. McDonald, "Design issues in building a computer tutor," in *IEEE Computer*, special issue on "Artificial Intelligence for Human-Machine Interaction" Sept 1984.

11. L. Johnson, and E. M. Soloway, "PROUST: Knowledge-based program debugging," *Proceedings Eighth International Software Engineering Conference*, Orlando, FL, March 1984.

12. J. Bonar, "Collecting and analyzing on-line protocols from novice programmers," *Behavioral Research Methods and Instrumentation*, 1982a.

13. J. Bonar, "Natural problem solving strategies and programming language constructs," *Proceedings of the Fourth Annual Conference of the Cognitive Science Society*, 1982b.

14. A. Collins, E. Warnock, and J. Passafiume, "Analysis and synthesis of tutorial dialogues," *Psychology of Learning and Motivation*, vol. 9, Academic Press, Inc., 1975.

15. R. Reichman, "Plain speaking: A theory and grammar of spontaneous discourse," Ph.D. thesis, Harvard University, Department of Mathematics, also Bolt, Beranek and Newman, Technical Report #4681, 1981.

16. L. Sass, "Parental communication deviance and schizophrenia: A cognitive-developmental analysis" in L. Vaina and H. Hintikka (eds.), *Cognitive Constraints on Communication*, 1984.

17. J. Larkin. J. McDermot, D. P. Simon, and H. Simon, "Expert and novice performance in solving physics problems," *Science*, Vol. 208, 20, 1980.

18. K. Forbus, and A. Stevens, "Using qualitative simulation to generate explanations," Report #4480, Bolt, Beranek and Newman, Inc., 1981.

19. J. deKleer, and J. S. Brown, "Foundations of envisioning," *Proceedings of the National Association of Artificial Intelligence*, 1982.

20. H. Grice, "Logic and conversation," in P. Cole and J. Morgan (eds.) *Syntax and Semantics*, Academic Press, New York, pp. 41-58, 1975.

21. M. Sullivan, and P. Cohen, "An endorsement-based plan recognition program" *International Joint Conference on Artifiical Intelligence*, Los Angeles, 1985.

22. E. Soloway, J. Bonar, B. Woolf, P. Barth, E. Rubin, and K. Erlich, "Cognition and programming: Why your students write those crazy programs," *Proceedings of the National Educational Computing Conference*, NECC, No. Denton, TX, 1981.

23. E. Rissland, and E. Soloway, "Constrained example generation: A testbed for studying learning," *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, 1981.

24. E. Rissland, "Understanding Mathematics," *Cognitive Science*, Vol. 2, No. 4, 1978.

10. Appendix

```
;*****************************************************************************
;
;
;  Global implications are written in bold font and prefaced with an asterisk (*)
;  Meta-rules are written in bold font
;  Discourse states are prefaced with a dollar sign ($)
;
;
;*****************************************************************************
make_$rule_struct 'PRESENT-EXAMPLE
      preconditions
                  (FC-and *found-threshold-of-knowledge
                  *teach-at-threshold-of-knowledge
                  *confused-student
                  *tutor-is-co-operative
                  *topic-is-important)
      action '(setg next_state '($present-example))
      post_processing '()
      prior_state '($question-model
                  $question-topic
                  $question-role-value)


make_$rule_struct 'PROBE-MISCONCEPTION
      preconditions
                  (FC-AND *evidence-of-misconception
                  *confused-student
                  *known-student-threshold)
      action '(setg next_state '$probe-misconception)
      post_processing '()
      prior_state '($teach)


make_$rule_struct  'JETTISON
      precondition *dialogue-is-ineffective
      action '(setg next_state (find-parent state))
      post-process '()
      prior_state '($ all states)


make_$rule_struct  'TEACH-TOPIC
      preconditions '(*topic-is-learnable-elsewhere)
                  not *known-student-threshold
                  *direct-sign-of-student-knowledge)
      action '(setg next_state '($teach-topic))
      post_processing '()
      prior_state '($explore-knowledge)
```

Figure 12: Meta-rules to generate exceptional discourse behavior.