Intelligent Tutoring Systems have Forgotten the Tutor: Adding a Cognitive Model of Human Tutors

Neil T. Heffernan III

March 2001
CMU-CS-01-127

School Of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy*

Thesis Committee:
Kenneth, R. Koedinger, Co-Chair
John R. Anderson, Co-Chair
Jaime G. Carbonell
Herbert A. Simon (deceased)
Kurt VanLehn, University of Pittsburgh

Copyright © 2001 Neil T. Heffernan III

This research is sponsored in part by the Spencer Foundation, and the National Science Foundation under grant number 9720359.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies or endorsement, either expressed or implied, of NSF or the Spencer Foundation.

Keywords: Model-tracing, cognitive model, intelligent tutoring system, artificial intelligence, tutoring, mathematical education.

ABSTRACT

This dissertation makes a contribution to 1) the cognitive science of algebra learning as well as to 2) intelligent tutoring systems architectures. I present a new intelligent tutoring system for the domain of writing expressions for algebra "story" problems. This system is novel, because it is the first intelligent algebra tutor that combines a cognitive model of the domain with a rich pedagogical model of dialog-based tutoring. The algebra model is novel because contrary to prior work that has emphasized the difficulties of using **variables**, as well as the difficulties of **comprehending** the text of a word problem, I establish the empirical result that ***articulating*** a complete expression (e.g., 800-40*3) is a major determinant of problem difficulty. The tutorial model is also novel because it is based on the observation of an experienced human tutor and captures the rich tutorial strategies specific to the domain of symbolization. The resulting system, called Ms. Lindquist, has been demonstrated to improve student learning. Over 350 students have used the system available at www.AlgebraTutor.org.

ACKNOWLEDGEMENTS

I would like to thank my wife. Her patience with me while in graduate school is much appreciated. Now that we are having a child, its time for me to stop being a graduate student, but before I do I would like to thank some folks that have made the experience so enjoyable.

First, is my advisor Ken. Though it is customary to use the first person singular pronoun in dissertation (e.g., "I argue …", "I did such and so"), there is no true way to separate out the contributions that a good advisor makes to a thesis.

I am an awful editor, but luckily, I have patient friends and family that have helped me proofread aspects of this dissertation. I would like to thank, Doug B, Doug R, Carl, Adam, John P, Ken, Cris, Neilie, Eileen, Lynsey, and Lisa H.

Thanks to Carl, the Dougs, Santy, Philip, Adam and Stan for making graduate school so much fun.

My office mates (including Bumba, Dafna, Srini, Doug, Nick, Oleg, Chris, Chuck, and John) over the ~6 years of graduate school have been excellent companions as well as sources of technical advice.

My mom might have wondered when I was going to enter the real world, but at present, I show no signs of doing so anytime soon. Thanks Mom.

- Neil (Feb, 2001)

TABLE OF CONTENTS

Introduction and Contributions of this Dissertation 1
Chapter 1: An Analysis of Student Behavior 3
Chapter 2: An Experienced Human Tutor that Motivated the Tutorial Model 35
Chapter 3: The Architecture of Ms. Lindquist 53
Chapter 4: Ms. Lindquist’s Tutorial Strategies and Comparisons 75
Chapter 5: A Formative Analysis Comparing Ms. Lindquist with a Computer Aided Instruction Control 107
Chapter 6: Conclusions and Future Work 119
References 131
Appendices 139

Introduction and Contributions of this Dissertation

This dissertation makes a contribution to the fields of 1) algebra learning, as well as to 2) intelligent tutoring systems (ITS) architectures. For the last 6 years, I have been focused on building a better algebra tutor. I choose to focus on the hardest and most important topic; a task I call *symbolization*. The symbolization task is when a student is asked to translate an algebra word problem statement into an algebra expression. I have made an important contribution about what makes symbolization so hard for students (Contribution #1 below). I also made a contribution (Contribution #2) by characterizing the tutorial strategies used by an experienced human tutor and have used these results, along with my results on student thinking and development, to create a detailed cognitive model of tutoring symbolization.

In addition, this dissertation makes a contribution to ITS architectures (Contributions #3 and #4). Model-tracing intelligent tutoring systems, which have been shown to be effective (e.g., Anderson et al., 1995), owe their success to doing a good job of modeling student thinking. However, they have largely ignored modeling tutoring itself. Why? The first reason is that there is a technical challenge to surmount that involves building systems that understand aspects of dialog planning and management. But a second, and probably more important reason, is that not much is known about what makes for effective tutoring, in general, or for the specific domain that is the subject of this dissertation. In developing a detailed model of effective tutoring of symbolization, I have employed two key strategies. First, I have performed experimental studies to develop a rich understanding of student learning difficulties that serves as a basis for designing specific tutorial strategies that assist students in constructing knowledge to overcome these difficulties. Second, I have performed an empirical study of an experienced human tutor and built a model that captures key aspects of her behavior, including specific *tutorial strategies* she used for this domain.

Contributions

1. Contrary to prior work that has emphasized the difficulties of using **variables**, as well as the difficulties of **comprehending** the text of a word problem, I establish the empirical result that ***articulating*** (i.e., the writing out a complete expression such as 800-40*3) is a major determinant of problem difficulty. I establish that many students who can comprehend problems well enough to solve them, can, nevertheless, fail to be able to symbolize them. I also show that, the presence of a variable is not a significant source of problem difficulty. To support my hypothesis that students need to learn how to produce symbols, I showed that if students were given practice using the grammar of algebra expressions, they were able to transfer this knowledge to improve their symbolization skills. I present a cognitive model of symbolization learning that captures and explains these distinctions as well as being consistent with the errors that students made.
2. I present a cognitive model of tutoring based on the analysis of an experienced human tutor and inspired by my model of student thinking. This model of tutoring is able to replicate aspects of the human tutor's behavior, including the multi-turn tutorial strategies she used. This model also replicates some of the domain independent aspects of dialog planning and scaffolding. However, unlike other tutoring models that have focused on general pedagogical strategies, my model also incorporates domain-specific pedagogical

strategies that are characteristic of expert human tutors and arguably critical to optimizing student learning.

3. I present the first Andersonian model-tracing tutor that has both a student model and a tightly integrated tutorial model. This general architecture (and implementation) advances the model-tracing framework. My system incorporates my tutorial model with my model of student thinking. The student model is used to diagnose student responses. The tutorial model uses the results of this diagnosis to plan a multi-turn tutorial dialog to deal with the particular errors the student made. The dialogs created include embedded feedback (both positive & negative), sub-dialogs, hints and spliced in conversational cues to ensure coherence.
4. The complete working tutor, called Ms. Lindquist, is available on the world wide web (www.AlgebraTutor.org), has been used by over 350 students, and the web-site has been designated “best of the web” by at least five educational sites. A formative evaluation of Ms. Lindquist demonstrates that students can learn effectively with Ms. Lindquist.

An Outline of this Dissertation

In Chapter 1, I develop a theory about what makes symbolization such a difficult task. I will present empirical results of students’ behavior on this task. I will argue that contrary to prior research in this area, which has emphasized compression difficulties, students' symbolization difficulties have more to do with learning the grammar of the "foreign language" of algebra expressions. I will also present an empirical result that shows that practice at using the grammar of algebra expressions improves students' symbolizing skills. Then I present an analysis of the errors students made. Finally, I conclude by making two instructional suggestions.

In Chapter 2, to build a more human-like intelligent tutoring system for this domain, I report on observations of an experienced human tutor. I argue that the tutor uses what I call *dynamic scaffolding* that involves 1) diagnosing the student's answer, 2) giving positive feedback on portions of that answer that are correct, and 3) focusing the dialog on portions of the problem that the student got wrong. I model this third step with *tutorial strategies* that specify a plan of questions to ask the student.

In Chapter 3, I present a general architecture that expands the existing *model-tracing* framework to allow for a more dialog-based system. This new architecture combines the student model from Chapter 1 with the tutorial model from Chapter 2. I describe this architecture with reference to a new tutoring system, named *Ms. Lindquist*, which is built using this architecture. I argue that the architecture I have created is general and the implementation easily extendable.

In Chapter 4, I present the tutorial strategies that are incorporated into Ms. Lindquist. I also compare Ms. Lindquist to the human tutor from Chapter 2, as well as comparing Ms. Lindquist to an existing model-tracing tutor. Finally, I conclude by assessing how well this project met its design goals.

In Chapter 5, I present a formative analysis of Ms. Lindquist that shows that students can learn effectively with the system.

Chapter 1: An Analysis of Student Behavior

Chapter Overview:
After reviewing prior work, this chapter will present the design and results of two difficulty factor assessments (Heffernan & Koedinger, 1997, Heffernan & Koedinger, 1998). These two studies will help to answer the question of what makes the task of symbolization so hard. This chapter will argue that contrary to the prior research that has emphasized both comprehension and "understanding of variables", much of the difficulty students experience is due to their problems articulating algebra expressions. This result will help to explain another empirical result we establish: a two-operator problem is harder than the sum of its parts. This chapter will then present an analysis of the errors made on the second difficulty factor assessment; I find that many of the errors student made were consistent with the fact that their difficulties have more to do with production difficulties than comprehension difficulties. As causal evidence for the claim that symbol production is a major difficulty factor in symbolization, I present a transfer study that shows that training on the grammar of algebra expressions can transfer to symbolization. Finally, this chapter will conclude with two instructional suggestions.

The Task Domain: Symbolization

Symbolization is a difficult task for students. The Third International Study of Math and Science (Beaton, et al., 1996) presented students with the following symbolization problem that only 49% of U.S. eighth graders we able to correctly answer.

> Juan has 5 fewer hats than Maria, and Clarissa has 3 times as many hats as Juan. If Maria has *n* hats, which of these represents the number of hats that Clarissa has?
>
> A: 5-3n B: 3n C: n-6 D: 3n-5 E: 3(n-5)

For another example, the National Assessment of Educational Progress (Carpenter et al., 1981) asked 17-year-olds the following question:

> Carol earned *D* dollars during the week. She spent *C* dollars for clothes and *F* dollars for food. Write an expression using *D*, *C* and *F* that shows the number of dollars she has left.

A third of the students could not correctly answer "*D-C-F*". Moreover, a quarter of students that had two years worth of algebra could not answer it correctly.

Arguably, the single most important skill students learn in an algebra class is the ability to translate a problem situation into an algebraic expression. We call this skill *symbolization*. If a student cannot translate a real-world problem situation into the language of algebra, then all the time spent practicing symbol manipulation skills is a waste of time (and that practice comprises the bulk of most algebra classes[1]). Even if a student has access to a spreadsheet or graphing calculator, the student will still need to be able to translate a problem into the language of mathematics (i.e., an algebraic expression). This thesis is not concerned with symbolic manipulation, but with the translation of verbal problem statements into algebraic expressions. Such algebraic expressions can be combined to form equations that can then be solved (or used

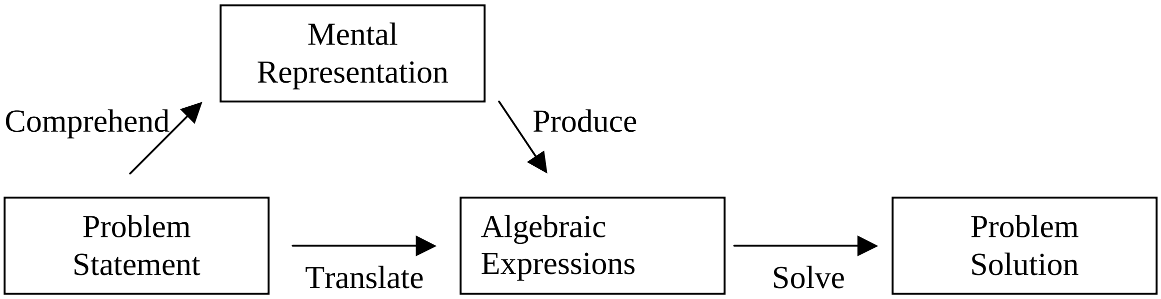
---

[1] The National Council of the Teachers of Mathematics (2000) has not been specific on how much of an algebra class should be focused on symbolic manipulation. However, a commonly used textbook (I taught from it.) shows the emphasis it gets in practice; Brown, Dolciani et al.'s (1990) textbook has each chapter arranged around a different symbolic manipulation skill. Each chapter has on average of about 9 sections, with usually 2-3 sections that show how to use that skill in the context of an algebra word problem.

to program a spreadsheet or graphing calculator), but in this study, we focus exclusively on the step of generating the expressions themselves.

Prior Work on Symbolization

In the research reviewed below (e.g., Paige & Simon,1966), symbolization is often viewed as a translation process that has a **comprehension** component as well as a **production** component (i.e., producing symbols in the target language of algebra) as shown in this schematic:



Most past research has focused on the comprehending and ignored the producing. Sometimes "translate" and "comprehend" are used synonymously, thus diminishing the role of production.

In this review, I will develop three hypotheses about what makes symbolization so difficult. They are:

1. Comprehension
2. Variables (their understanding and use, there of)
3. Production

Research on comprehension difficulties has a long history, and we will start with work on **arithmetic** story problems that then lead into work on comprehending **algebra** problems. The second hypothesis is that students lack an understanding of variables. The third hypothesis is that the difficulties lie mainly on the production side.

Prior Work on Comprehending Arithmetic Word Problems

There has been a great deal of work on arithmetic word problems (Carpenter & Moser, 1982; Carpenter, Moser, et al., 1982; Kintsch & Greeno, 1985; Lewis & Mayer, 1987; Riley & Greeno, 1988; Cummins Dellarosa, 1991; Stern, 1993; Hegarty, Mayer, et al., 1995; LeBlanc & Weber-Russell, 1996; Roth, 1996; Sakamoto, 1999; Schwarz, Nathan et al., 1996; Mwangi & Sweller, 1998; Passolunghi, Cornoldi et al., 1999; Sakamoto, 1999). Much of this work comes out of the text-comprehension tradition. For instance, van Dijk (van Dijk 1977) is interested the process of story understanding, which involves, not only comprehending individual sentences, but also the "macro-structure" that unites the sentences together. Van Dijk argues, "the same basic principles can be extended to other domains, such as … problem solving (p. 3)". Kintsch (Kintsch 1977) was interested in how people comprehended short stories, and made a distinction between the linguistic structure and the semantic structure. He represented the semantic structure with a list of propositions, which he called its *text base*. *Macro-operators* (e.g., that generalize propositions or construct new propositions) can then infer other propositions, that he calls *macro-propositions*. Kintsch has a "model that likens story comprehension to a process of labeling slots in a previously acquired, but flexible, schema (p. 55)".

The first models of arithmetic word problem solving by Riley, Greeno and Heller (1983) and Briars & Larkin (1984) had no explicit language-processing component. Instead, they dealt with a problem representation the researchers believed the children constructed. In particular, Briar & Larkin's model compared different problems and classified them according to the general schema of what happened to the quantities as sets (e.g., a part-whole schema or a more-than schema). Similarly, Riley, Greeno and Heller (1983) classified arithmetic word problems according the general schemas used, which ignored some linguistic differences. However, Hudson (1983) showed, differences in the difficulty of word problems depended greatly upon subtle linguistic variations. For instance, only 38% of students in his study could solve, "Joe has 8 Sticks. He has 5 more sticks than Tom has. How many sticks does Tom have?" However, over 79% of children could answer, "There were 8 birds and 5 worms. How many birds did not get a worm?"

In order to address this, Kintsch & Greeno (1985) provided a model that combined a model of problem solving (Riley & Greeno, 1983) with a model of text comprehension (van Dijk & Kintsch, 1983). The Kintsch & Greeno model was followed by computer simulations (e.g., Fletcher, 1985; Dellorosa, 1986; Cummins, Kintsch, Reusser & Weimer, 1988) that were able to account for many of the differences in difficulty between problems types. See Kintsch (1998) for a summary of his work as it relates to comprehending mathematics problems.

Cummins, Kintsch, Reusser, & Weimer (1988) suggest "that much of the difficulty children experience with word problems can be attributed to difficulty in comprehending abstract or ambiguous language." Cummins et al. (1988) compared two views about what children learn when solving these word problems. The first view Cummins et al. (1988) considered has its roots in Piaget (1970), and argues that children learn conceptual knowledge about these problems. Cummins et al. (1988) contrasted this with the view that students learn the linguistic forms associated with these types of problems. For instance, a student might understand the part-whole schema but not understand how to map "How many more Xs than Ys?" onto that schema. Cummins et al. (1988) argued that the second view does a better job of explaining student performance. Cummins et al. (1988) argued that the results of lesion studies on their model are consistent with the same types of errors that real students make. This focus on comprehension led to interventions based on the suggestion that comprehension was a major difficulty factor; Cummins (1991) "found [student performance to] be significantly improved by rewording problems to avoid ambiguous linguistic forms."

Reusser (1989) extended the Cummins et al (1988) model by building in an explicit *situational* model. This model sits between the text and mathematization of the problem. The situational model specifies actors, actions, states and events in the problem in terms of every day concepts. The situational model is used to constrain the formal problem-solving model.

The importance of linguistic knowledge was also emphasized by Cardelle-Elawar (1992) who comparing native speakers of English with students who spoke English as a second language. Cardelle-Elawar found that non-native speakers had difficulties with terms (such as "equals") and that this often prevented them from solving problems.

LeBlanc & Weber-Russell (1996) also focused on the role of comprehension. They presented

> "A computer simulation designed to capture the working memory demands required in 'bottom up' comprehension of arithmetic word problems. The simulation's sentence-level parser and text integration component reflect the importance of processing the problem from its

> original natural language presentation. … Consistent with previous results, which highlighted the significance of small changes in problem wording, the simulation offers a process-oriented perspective as to why natural language presentation constrains the comprehension of mathematical relationships."

Lewis & Mayer (1987) presented "a model of word problem comprehension processes that [used] schemata as guides to comprehension."

Judd & Bilsky (1989) argued that most models of verbal arithmetic problem solving view the process as including problem comprehension and problem solution. They went on to say that, "Available research suggests that, of the two components, comprehension is the more important source of problem difficulty." This claim might well be true for arithmetic word problems for elementary school children, but might not be true for ninth grade algebra students, as we will next explore.

Prior Work on Comprehending Algebra Word Problems

The work on arithmetic word problem comprehension lead naturally into work on similar algebraic problems (Hinsley, Hayes, & Simon, 1977; Mayer, 1982; Rapp, 1986; Reed & Ettinger, 1987; Hall, Kibler, Wenger & Truxaw, 1989; Bassok, 1990; Cooney & Swanson, 1990; Hoz & Harel, 1990; Kintsch, 1991; Nathan, Kintsch & Young, 1992; Aziz, Pain & Bna 1995). Not surprisingly, many researchers looked at the difficulties of the comprehension process. For instance, Kintsch, who was one of the early researchers that focused on comprehension of arithmetic word problems, moved into research on algebra word problems. He brought the same assumptions about what causes difficulties for students that had been shown to be a major obstacle for students working on arithmetic word problems. Kintsch (1991) asks us to "assume here that the theory developed for word arithmetic problems can be analogically extended into the algebra domain." Kintsch goes on to say that he will "explore which steps in the process of understanding a word problem are the difficult ones." Kintsch (1991) also states the "the premise of [his work] is that comprehension failures are central to the difficulty of word algebra problems." With these assumptions, it is not surprising that he focused on the comprehension of the problem text.

Herbert Simon and colleagues also performed important early work on algebra word problems solving. Paige & Simon (1966), and Hinsley, Hayes & Simon (1977) took Bobrow's (1968) computer program STUDENT as a foundation for their cognitive model. STUDENT had a process called *direct translation*. The direct translation process was one

> in which it translates successive sentences of the problem text into equations and then tries to solve the equations. The direct translation process involves no choice [of a schema] at the time of reading. … If the direct translation process fails to yield a solution, STUDENT will search its memory for relevant global knowledge. For example, if the problem contains a keyword such as 'distance' or 'miles,' STUDENT will retrieve the equation 'distance = rate * miles'

Paige & Simon (1966) compared symbolization to translation from English to French, which they said involved taking each French word, looking it up in a French to English dictionary, and writing down the answers with some possible changes to inflections, and rearrangements due to syntax rules. Hinsley et al. (1977) observed students' performance to try to determine if students used either a direct translation approach or a schema approach. They concluded that students used both. They observed the use of the direct translation approach more often on unfamiliar problems that did not fall easily into a familiar schema.

Mayer (1981) extended the study of schemata and classified a large number of story problems into 90 different schemata. Mayer argued that students first identify the general class of a problem and then bring to bear schemata that pull out of the situation some of the numbers to fill-in expected slots. He suggested teaching children to recognize schemata. Later he studied older student's comprehension of *algebra* story problems (Mayer, Lewis & Hegarty, 1992). Mayer later joined with Haggerty to study the eye movements performed as students comprehended text (Hegarty, Mayer & Monk, 1995)[2].

Another researcher who studied comprehension was Rapp (1986) who built an expert system that comprehended word problems in a textual form. Hall, Kibler, Wenger and Truzaw (1989) also focused on comprehension and found that problem solving and comprehension were complimentary activities that students moved back and forth between.

As was mentioned above in the section on arithmetic problems, Kintsch went on to extend his work to algebra problem solving by incorporating a situational model of (Reusser, 1989). This work also led to the Heron system (Reusser, 1996) that comprehended word problems.

In summary, a large amount of work has focused on problem comprehension as the major difficulty student face when symbolizing. It is fitting to end this section with a quote from Nathan, Kintsch, & Young (1992) who "claim that [the] symbolization [process] is a highly reading-oriented one in which poor comprehension and an inability to access relevant long term knowledge leads to serious errors." The general claim from much of the above literature, is that problem comprehension, including schema detection, are key knowledge components that cause difficulty for students while symbolizing. I will refer to this as the *comprehension hypothesis.*

Prior Work on the Understanding of Variables

Kuchemann (1981) worked on large-scale evaluation of student's mathematics performance (ages 13-15) in Britain, and wanted to characterize the level of difficulty of the individual test items by using two criteria:

1) Structural complexity (as defined by Collis (1975a, 1975b))
2) The level of understanding of letters.[3]

Here I will ignore the first criterion and instead focus on Kuchemann's second criterion. Kuchemann argued that there were six cognitive levels of understanding the use of letters in the context of an algebra problem and is shown in Figure 1.

---

[2] See also Gluck (1999) for a study of eye-movements and algebra tutoring.

[3] Kuchemann choose to use the term letter, reserving the term variable only for those students who understood at the deepest level.

Children’s Interpretations of Letters	
Letter Evaluated:	This category applies to response where the letter is assigned a numerical value from the outset.
Letter not used:	Here the children ignore the letter, or at best acknowledge its existence but without giving it a meaning.
Letter used as an object:	The letter is regard as shorthand for an object or as an object in its own right.
Letter used a specific unknown:	Children regard a latter as a specific unknown number, and can operate upon it directly.
Letter Used as generalized number:	The letter is seen as representing, or at least as being able to take, several values, rather then just one.
Letter used as a variable:	The letter is seen as representing a range of unspecified values, and a systematic relationship is seen to exist between two such sets of values.

Figure 1: The table from Kuchemann (1981) showing students level of understanding of variables.

The most proficient level was "Letter used as a variable". Kuchemann argued that students in any of the first three levels are lacking any "real understanding of even the beginning of algebra (p. 105)". He argued that "most" 13-15 years olds fell into these first three levels. The first level is when students simply replace the letter with a random number. Kuchemann suggests that students at the second level might replace all the variables in the problem with the same number, or students might replace a variable by the number representing its place in the alphabet (e.g., A=1, B=2, C=3, etc). Students at the third level were "reducing the letter’s meaning from something quite abstract to something far more concrete and ’real’ (p.107)". For an example, consider this problem that Kuchemann analyzed:

> Blue pencils cost 5 pence each and red pencils cost 6 pence each. I buy some blue and some red pencils and altogether it costs me 90 pence. If *b* is the number of blue pencils bought, and *r* is the number of red pencils bought, what can you write down about *b* and *r*?

Only 13% of students got it correct (i.e., b*5+r*6=90). In addition, 17% of students gave the most common error of $b+r=90$. Kuchemann argued that students making this error were at the third level because they treated the letters as objects. Kuchemann suggests, "… the extent to which the letters are meaningful to children will be of vital importance in determining item difficulty." I will call Kuchemann's explanation of the difficulty of algebra problems the "lack of understanding of variables" hypothesis (or simply the *Variables Hypothesis).* Kuchemann's work does not address the role of comprehension of word problems. Therefore, Kuchemann's work is of interests to this dissertation, not for what it establishes but for what it emphasizes as a source of difficulty for students.

Kuchemann's work influenced the National Council of Teachers of Mathematics (NCTM) in the United States. The *Principles and Standards for School Mathematics* from the NCTM (NCTM, 2000) is probably the most influential book affecting the practice of the nations teachers and teacher educators. In its algebra section for middle school students grades 6-8 (most students now take algebra in eighth grade) it argues that:

> Student's understanding of variable should be far beyond simply recognizing that letters can be used to stand for unknown numbers …

> Most student will need extensive experience in interpreting relationships among quantities in a variety of problem context before that can work meaningfully with variables and symbolic expressions. An understanding of the meaning and uses of variables develops gradually as students create and use symbolic expressions and relate them to verbal, tabular, and graphical representations. (p. 225)

The *Principles* cites four levels of understanding the use of variables that is very similar to Kuchemann's level. The *Principles* doesn't specifically specify what is the most difficult aspect of symbolization, but we can reasonable infer that what they choose to emphasis is probably what they think is hard (If the NCTM believed it was easy for student's, they probably wouldn't have to emphasis it so much.) Since the *Principles* argue that students should "know" hundreds of things, it is important to note how highly the NCTM values the "understanding of variables". In the back of the *Principles* the NCTM breaks up algebra into 4 components

1. Understand patterns, relations and functions.
2. Represent and analyze mathematical situations and structures using algebraic symbols.
3. Use mathematical models to represent and understand quantitative relationships.
4. Analyze changes in various contexts.

The 2nd item is the component that is of relevance to this dissertation and is further broken the into four "expectations", one of which is that students "develop an initial conceptual understanding of different uses of variables." I argue that this emphasis certainly suggests that student's lack of understanding of variables could explain why symbolization is such a difficult task for students.

Another piece of prior work that agrees with Kuchemann's and the NCTM's emphasis on variables was a piece of design work on the "Pattern Finder" window, which was a component of an algebra word problem tutor (Koedinger, Anderson, Hadley, & Mark, 1995). The Patten Finder window asked the students to articulate the arithmetic needed for 3 different values of the variable, before trying to use a variable. Here is an example interaction:

> Ann is in a lake that is 2400 yards wide. She starts out 800 yards from the dock. She rows for "m" minutes back towards the dock at a rate of 40 yards per minute. Write an expression for her distance from the dock.[4]
>
> **Tutor**:How would you calculate THE DISTANCE ANN IS FROM THE DOCK for 2 minutes?
> **Student**:800-2*40
> **Tutor**:How would you calculate THE DISTANCE ANN IS FROM THE DOCK for 3 minutes?
> **Student**:800-3*40
> **Tutor**:How would you calculate THE DISTANCE ANN IS FROM THE DOCK for 4 minutes?
> **Student**:800-4*40
> **Tutor**: Write an expression, which describes your calculations, using a variable.
> **Student**:800-m*40

The pattern finder window is based on the idea that the last step is going to be the hardest because it involves the use of the variable. This seems reasonable, since algebra is often called the "generalization of arithmetic", so the introduction of the variable should be the hard part of symbolizing.

This completes the review of prior work supporting the *Variable Hypothesis* that argues that the use (or understanding) of a variable, per se, is what makes symbolization such a hard task.

---

[4] This is not one of the problems used by the Carnegie Learning Tutor.

Prior Work that bears on Production Difficulties

Despite all the emphasis on comprehension and the use of variables, there is some research that has speculated about the potential production difficulties. However, this research did not lead to a general explanatory model nor did it provide the kind of detailed experimental support for such a model (as is presented later in this chapter). Clement and colleagues (Clement, Lochhead & Soloway (1980), Clement, 1982; Rosnick & Clement, 1982) studied student performance on the following problem where students are asked to write an expression for the following sentence "There are six times as many students as professors in the university". Students did very poorly at this task. More importantly, they argued that students' mistranslations (i.e., 68% of students errors were "6S=P") were often due to their not understanding the semantics of an equation.[5]

More recently, Koedinger & Anderson (1998) found evidence that acquiring comprehension skills is not sufficient for symbolization competence. They found that on 36% of problems that students comprehended well enough to find a numerical answer, they nevertheless failed to correctly symbolize an answer. This result suggests that, in addition to comprehension difficulties, students have difficulty in "symbolic production." This suggests an analogy to learning a foreign language. If you ask a student to translate an English sentence into French and observe that the student fails, the failure is not necessarily due to a lack of English comprehension skills, but, perhaps, is due to a lack of **production** skills for French. Similarly, students may fail in word problem solving, not because they lack English comprehension skills, but rather because they cannot **produce** algebra.

I find theoretical support[6] for this position in a study by Cedillio (2001) on using graphing calculators (that force the user to write expressions). He argued that algebra teaching should be based on the natural language acquisition work of Bruner (1983). Cedillio was interested in how students learned the grammar for algebra expression. Unfortunately, he reports no results that help to distinguish between the various hypotheses I have articulated about what makes symbolization so difficult for students. To review, those three hypotheses are:

1) Comprehension
2) Understanding of a variable
3) Production: including knowing the grammar for algebra expressions

The difficulty factors assessments presented next were designed to disambiguate between these three different hypotheses.

Introduction to the Present Studies

Our goal is to build a model of the skill of symbolization. To achieve that, a better understanding of what factors make such problems difficult was needed. What capabilities do students that are more competent have that poorer students do not? What kinds of scaffolds

---

[5] Clement et al. (1980) formulated several models to represent the way students could mistranslate these problems. For instance, one model was based on the student translating the keywords of the sentence in the same order as the sentence presents them. This also involves the student mistranslating the word "as" to mean "equal". This research led a very small prototype model (Aziz, Pain & Bna, 1995) called TAPS (for Translating Algebra Problems System), which tried to explain how students could mistranslate these problems but they were primarily interested in designing tutoring systems and how to model student's beliefs, rather then interested in the symbolization domain per se.

[6] When Paige and Simon (1966) compared symbolization to translating a sentence from English to French, they did mention that the production side included knowing the syntax and grammar of the target language. Symbolization researchers have not seriously pursued this observation.

might we provide to assist student learning? To address these questions, I performed two *difficulty factors assessments* (DFA) with Koedinger (Heffernan & Koedinger, 1997; Heffernan & Koedinger, 1998). In a difficulty factors assessment, we use theory and task analysis to generate hypotheses about the likely factors that cause student difficulties and then use these factors to systematically generate a pool of problems. By assessing performance differences on pairs of problems that vary by only one factor, we can identify what knowledge elements are needed in a model in order to adequately decompose the thinking and learning processes of students.

I will first report the results of the first DFA I performed on the effect of four different factors, and then report the second DFA that followed up on specific hypotheses that flowed from the first DFA.

Difficulty Factors Assessment #1

For the first DFA I sampled student performance on a set of 128 problems created by systematically modifying 8 core problem situations (Appendix A) along 4 binary factor dimensions. These 4 factors represent specific hypotheses about what cause symbolization difficulties and how scaffolds might ease the symbolization process. Consider the problem P0 from Table 1. This is a hard problem for ninth-grade, beginning algebra students. Only 13% of the students in the experiment (described below) answered it correctly. What makes this problem hard? I consider four different difficulty factors. Maybe what makes this problem hard is:

1. having to compose the symbolic translation of parts of the problem into a complete translation of the whole problem,
2. the presence of the distractor phrase "2400 yards wide",
3. comprehending the text well enough to translate the phases into operators and numbers, as well as knowing which numbers are matched up with which operators,
4. the presence of an algebraic variable "m" as opposed to the numeric constants students are already familiar with from arithmetic instruction.

Table 1 lists alternative problem statements for each of these factors as P1, P2, P3 and P4. In the following sections, I provide motivation for the consideration of each of these factors and illustrate them as they modify problem P0 (see Table 1).

P0: Ann is in a rowboat in a lake that is 2400 yards wide. She is 800 yards from the dock. She then rows for "m" minutes back towards the dock. Ann rows at a speed of 40 yards per minute. Write an expression for Ann's distance from the dock.
P1: A) Ann is in a rowboat in a lake that is 2400 yards wide. She is 800 yards from the dock. She then rows "y" yards back towards the dock. Write an expression for Ann's distance from the dock. B) Ann is in a rowboat in a lake that is 2400 yards wide. She then rows for "m" minutes back towards the dock. Ann rows at a speed of 40 yards per minute. Write an expression for the distance Ann has rowed.
P2: Ann is in a rowboat in a lake. She is 800 yards from the dock. She then rows for "m" minutes back towards the dock. Ann rows at a speed of 40 yards per minute. Write an expression for Ann's distance from the dock.
P3: Ann is in a rowboat in a lake that is 2400 yards wide. She is 800 yards from the dock. She then rows for "m" minutes back towards the dock. Ann rows at a speed of 40 yards per minute. Write an expression for Ann's distance from the dock. Hint 1: Ann's distance from the dock is equal to the 800 yards she started out from the dock minus the distance she has rowed in "m" minutes. Hint 2: The distance she has rowed in "m" minutes is equal to the 40 yards she rows per minute multiplied by the "m" minutes it takes her.
P4: Ann is in a rowboat in a lake that is 2400 yards wide. She is 800 yards from the dock. She then rows for **11** minutes back towards the dock. Ann rows at a speed of 40 yards per minute. Write an expression for Ann's distance from the dock.

Table 1: An example (P0) and four variants that change just one factor: (P1) decomposed, (P2) without the distractor phase (i.e., 2400 yards wide), (P3) with comprehension hints, and (P4) a concrete instance (11 instead of "m").

Factor One: Composed vs. Decomposed

Singley, Anderson & Givens (1991) reported that some students fail to solve multi-step story problems even when they can solve the individual parts that make them up. I desire to know whether or not this is simply the expected effect of having to do multiple steps each of which results in an accumulated chance of failure. Alternatively, the multi-step problem may be even harder (or easier) than the combined probability of the correct performance of the individual steps separately. Consider P1, which lists the two sub-problems of P0, which I call the *decomposed* version of P0. Of course, I would expect that solving a single part of this problem is easier than solving P0. The more interesting question is "Is solving P0 easier than solving **both** parts of P1?" (Note, I consider P1 correct only if both portions are answered correctly.)

Factor Two: Presence of Distractor Numbers

As Paige & Simon observed, less-competent symbolizers appear to sometimes rely exclusively on direct translation and do not invoke any semantic processes to recognize, for instance, that a negative board length is impossible. Tabachneck, Koedinger, & Nathan (1994) observed that novice symbolizers exhibit other kinds of shallow semantic processing, as well. In particular, students will often produce "symbol soup" by guessing at the answer using the given numbers and symbols but getting position or operations wrong. To the extent that novice symbolizers employ such a guessing strategy (perhaps as a fall-back when more specific knowledge is lacking), we should see more errors on problems that involve an extra distractor quantity (such as "2400 yards wide" in P0) than on problems that do not (such as problem P2 from Table 1).

Factor Three: Comprehension Hints

Given the attention past research has given to the role of comprehension in the symbolization process, the third factor tests a possible scaffolding technique that attempts to help students comprehend the problems more effectively. The technique involves giving the student a hint that re-expresses the problem in a form that is more amenable to direct translation to symbols. These hints are in a form that would clearly facilitate performance of a computer model such as the STUDENT program used by Paige & Simon. Consider the comprehension hints given in P3. Notice that the hints identify what mathematical operator is to be used, while the original problem-statement did not. Also note that the form of the hint is in the simple form of <Subject_Quantity> "is equal to" <Quantity1> <Operator> <Quantity2>, where <Subject_Quantity>,<Quantity1> and <Quantity2> are replaced with a verbal description of a quantity noun phrase, and <operator> is replaced by either "plus", "minus", "multiplied by" or "divided by." This simple form makes it possible for a left-to-right scan and translation of the problem to work. Also, note that these verbal recodings identify what number or variable is matched with each quantity. Since these hints identify the operation to be used, they eliminate the need for schemata or "world" knowledge, such as having to know the distance-rate-time formula.

Factor Four: Presence of Variables

As mentioned earlier, Koedinger & Anderson (1998) found that for certain classes of problems, students are better at finding a numerical answer than writing a symbolic expression (for the same problem). Koedinger & Anderson hypothesized that asking students to compute concrete instances (problems without a variable) of a general problem would facilitate symbolization of that problem. To test this hypothesis, they designed a scaffolding technique called *inductive support* and implemented it as part of an intelligent tutor.

I can illustrate the inductive support scaffolding technique with my running example P0. The scaffolding involved two questions that asked students to solve the problem if the variable were replaced with a constant (e.g., "How far is Ann from the dock in 4 minutes?") After answering these concrete arithmetic problems, students were asked to write the symbolic expression. Students using this inductive support tutor were shown to learn more than students using an alternative "textbook" tutor. The tutor’s design was adapted based on this study so that the current algebra tutor (Koedinger, Anderson, Hadley, Mark, 1995) has a "Pattern Finder" component where, rather than just answering these concrete questions, students are also asked to show how to get answers for successive small values of x, namely, 2, 3, and 4. In the example above, students are expected to answer "800 - 40 * 2", then "800 - 40 * 3" and "800 - 40 * 4". Next, they are to induce the pattern to arrive at the abstract expression "800 - 40 * x". It has come as a surprise that making this last step is not at all difficult for students; in fact, it is only the first step (writing the expression when x is 2) that gives students any substantial difficulty. I began to wonder whether this first step really is easier than the final goal of writing the abstract expression. If not, the Pattern Finder may not be such a good scaffolding technique. Thus, I added the presence of variable factor to this assessment to test whether writing a concrete expression (e.g., "800 - 40 * 11" as in P4) is in fact easier than writing an abstract expression (e.g., "800 - 40 * m "as in P1).

Procedure

Given the four binary factors there are sixteen different possible combinations of the factors. These 16 different possible combinations were crossed with 8 different cover stories (i.e., problem situations) and distributed in a Latin square design among 16 test forms that balanced for each factor. Given that students tend to perform worse on items near the end of a test, the order of various problems was systematically varied on each (e.g., the 8 composed, distractor, no hint, no variable problems were in the 8 different position on 8 different forms). However, because the cover story factor was not a variable of critical interest, the 8 cover stories appeared in the same order on each form (to do otherwise would have required many more forms). All eight cover stories had two operators implicit in the story so that the composed version required a two-operator answer. The decomposed version required two separate answers with each of those answers requiring one operator. The subjects were 79 ninth grade students (about age 14 years of age) in the first month of a low-level algebra course from an affluent suburb of Pittsburgh. Each student was randomly given one of the 16 different test forms and had 14 minutes to complete the test. After two class periods of instruction on such problems, students were again given a random form as a posttest. Since I was not interested in the experimental manipulation (it lead to a uninteresting null result), I report the results as if I had twice as many students, each taking this test. Of course, students might have learned during the experiment, but since I am not instead in their learning (Instead, I am interested in their relative performance on the different types of problems) it is safe to ignore this. Each test was then graded and no partial credit was given. A decomposed problem was considered correct only if both parts were answered correctly.

Results and Discussion

To test for effects of the four factors I performed both an item analysis and a subject analysis as recommended by Clark (1973). I performed an item analysis on students' mean performance on the 128 different problems appearing on the pre- and post-test forms. Separate item-means were computed for the pre- and post-tests. I performed a four-factor (2*2*2*2) analysis of variance (ANOVA) on the item means.

Figure 1 illustrates the relative impact of the four factors, averaged over all the students. In this figure, the legend indicates the harder version for each factor first (i.e., composed problems are harder than decomposed, distractor present problems were harder then problems where the distractor was absent, problems with variable were harder then problems with concrete instances, and finally, problems with the hints were easier than the problems without hints.) The effect of the comprehension hints appears small at best (3.1% difference in favor of hint problems) and

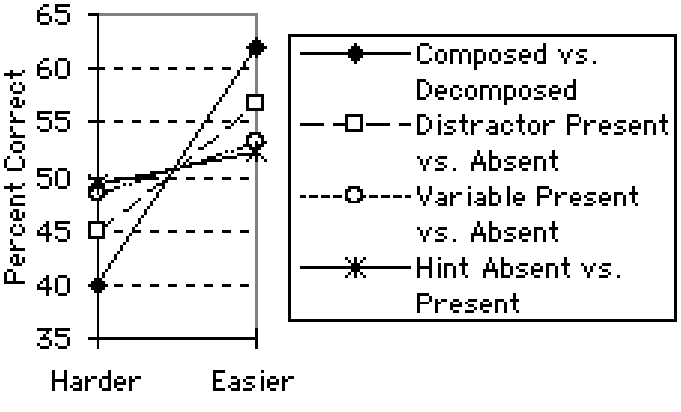


Figure 1: The average percent correct for the four factors.

this difference is not statistically significant (F(1,238)=1.127, p<.2894). Similarly, the presence of a variable is also small at best (4.5% difference in favor of no-variable problems) and not statistically significant (F(1,238)=1.531, p<.217). In contrast, the distractor effect was considerably larger (11.8% difference in favor of no-distractor problems) and statistically significant (F (1,238)=8.135, p<.0047). The composition factor had by far the largest effect (22% difference in favor of the decomposed problems), and was statistically significant (F(1,238)=37.048, p<.0001). No statistically significant interactions were found in the full ANOVA model.

To verify that these effects generalize across subjects as well as across items, I performed a subject analysis as well. I performed four repeated-measure ANOVAs with each factor as a within-subjects variable. Again, there were statistically significant effects for distractor (F(1,66)=14.018, p=.0004) and composition (F(1,66)=52.059, p=.0001) but no statistically significant effects of variables (F(1,66)=.739 p=.3932) or hints (F(1,66)=1.306, p=.2573).

The Composition Effect

These results show that a two-operator problem is harder than both of its parts presented together. I call this the *composition effect*. What skills are many students missing that prevent them from being able to deal with composed problems even though they are able to deal with the sub-problems individually? I now describe two alternative hypotheses that can account for the composition effect. The first is called the *articulating composition* hypothesis and the second is called the *combinatorial search* hypothesis.

The Articulating Composition Hypothesis

I hypothesize that the whole is harder than the sum of its parts because there is extra difficulty in putting the symbolic translations of the parts together to form a symbolic translation of the whole. I hypothesize that many students start their study of algebra with knowledge components (e.g., ACT-R production rules (Anderson 1993)) that enable them to symbolize only one-operator problems because their production rules only allow for single numerals or variables (e.g., 40 or m) to be used as arguments to the mathematical operators, as opposed to whole subexpressions (e.g., 40*m or 800-x). Such students can answer 800-x but not 800-40*m because 40*m is a subexpression and they do not know how to **substitute a subexpression into another expression**.

Anecdotal Evidence in Support of Production Difficulties

Figure 2 shows the work of a student who appears to have correctly described the mathematical process needed to solve for a value if given "h", but who failed to express that knowledge in the correct algebraic form. Instead of writing "500/(h-2)", the student indicated that first she would subtract 2 from "h" which would result in a new unknown that she again calls "h". She then indicated that 500 should be divided by this new number. She used the non-algebraic notation for division that is taught in elementary school. It seems likely that this student appears to have understood the quantitative structure of a problem (knew to subtract 2 from *h*, followed by dividing 500 by the result) but not be able to symbolize. This student's failure appears to a lack of the correct knowledge for **producing** algebraic sentences. For instance, this student might benefit from learning the correct was of expressing how to describe to do one step before another (instead of using an arrow sign, the student needs to wrap the first step with parentheses).

Bob drove 500 miles from Denver to Fargo to visit his grandmother. Normally this trip takes him "h" hours, but on Tuesday, there was good weather and he saved 2 hours. Write an expression for his driving speed.

h - 2 --> h)$\overline{500}$

Figure 2: A problem and a student's response

Figure 3 provides another example where a student demonstrates comprehension and quantitative understanding as well as an inability to correctly generate the algebraic symbols. Her answer is similar to the answer in Figure 2 in that both indicate the process that should be used to solve for an answer yet fail to output that answer in standard algebraic form. The use of the equals sign in this example appears to grow out of the way students use the equal sign as "gives" in elementary arithmetic where it is not uncommon to see students chain together steps with equal signs, as in "3*4=12-5=7" (Sfard, et al., 1993). In Figure 3, since 72-m can not be simplified the student uses a new variable "n" to stand for the result and then continues.

Sue made 72 dollars by washing 6 cars to buy holiday presents. She decided to spend "m" dollars on a present for her mom and then use the remainder to buy presents for each of her 4 sisters. She will spend the same amount on each sister. How much can she spend on each sister?

72 - m = n / 4=

Figure 3: A problem and a student's response

The Combinatorial Search (CS) Hypothesis

A second hypothesis is that the composition effect can be explained purely in terms of a combinatorial search model, in which a composed problem is harder because of the exponentially increasing number of possible sequences of arguments and operators. The large effect of distractors leads us to conclude that many students engage in some form of guessing, particularly as a fallback strategy when having difficulty. The difficulty of guessing grows with the complexity of problems, particularly as the number of possible combinations of given quantities and inferred operators grows. The composed, no-distractor problems have three quantities to choose from whereas there are only two quantities to choose from in each of the two parts of the decomposed, no-distractor problems. Thus, it may be that the composition effect is the result of this added complexity, and not the result of a missing or over-specialized skill as hypothesized in the Articulating Composition Hypothesis.

I tried a number of ways of estimating complexity depending on different assumptions. However, all of them predicted, contrary to the data, that the distractor effect should be bigger than the composition effect. I present one such estimation which has the following assumptions about how a student may guess at an answer: 1) students can pick out what numbers or variables are present in the problem and which operators will be used, 2) students know the general syntactic form of a symbolic sentence, particularly that operators need to be written between quantities, and 3) students will not use the same argument (variable or number) twice. To simplify the calculation, I ignore the difficulty of knowing when to add parentheses and assume that the operators in the problem are non-commutative so the student has to get the order of the arguments correct. Essentially, this comes down to assuming that to guess correctly, students must pick the correct order for the arguments and operators. I compare the probability of doing so for various problem types.

Let us first calculate the probability of getting the correct order for a composed problem, starting with the leftmost argument and moving right. The probability of getting the first argument correct is 1/3 since there are three possible numbers to put first. Similarly, the student picks one of the two inferred operators for the first operator slot (1/2). Then given our assumption of a non-replacement strategy, the probability of choosing the next argument correct is 1/2 since there are two remaining arguments. The final operator and arguments are then determined. So the combined probability of getting the correct answer is (1/3)(1/2)(1/2)(1/1)(1/1)=1/12.

Next, we consider the probability of guessing the correct answer for a decomposed non-distractor problem. Since there are only two arguments present, the probability of selecting the first argument is 1/2. The operator and the second argument are then both determined. Therefore, the probability of getting one part of a decomposed non-distractor problem correct is 1/2 and to get both parts correct is (1/2)(1/2)=1/4. Since 1/12 is less than 1/4, we see that this model does predict that there will be a composition effect. However, the model does not predict the relative effect of distractors, as we will now show.

Finally, consider a decomposed distractor problem. The probability of selecting the first argument is 1/3, since there are now 3 arguments present in the problem statement. The operator is determined, but the last operator is 1/2, which yields a total for one part of (1/3)(1/2)=1/6 and a total for the two parts together of (1/6)(1/6)=1/36.

In summary the SC model predicts that the distractor effect (1/36) will be larger than the predicted composition effect (1/12). However, the data shows that the composition effect is larger (22%) than the distractor effect (11%). Additionally, the composition effect was found to be statistically different from the distractor effect when we compared the means for composed, non-distractor problems with decomposed, distractor problems (F (1, 238) = 5.2, $p < .05$). I therefore favor the Articulating Composition Model.

Comprehension Hints

Next, we continue to discuss the results of this study, and in particular, I consider an explanation for the surprising absence of a statistically significant effect of the comprehension hints. After all, these hints recoded the story problem into a simpler form that is more amenable to direct translation. The hints also identified what the operators should be, which quantities to use with those operators and which order to put the operators in. However, these results are consistent with the view that the comprehension of these sentences is not that large a stumbling block, particularly when compared with the stumbling block of learning to deal with composed problems. Despite the fact that the effect of hints was not statistically significant, there is evidence that the hints did help for the decomposed problems. The trend in favor of the hint problems was much larger (a 7% difference) on the decomposed problems than on the composed problems (.01% difference) (thought this was not a statistically significant interaction). I hypothesize that the students who benefited from the hints were less able students and were most likely not to have the skills to deal with composed problems (as outlined in the Articulating Composition Hypothesis). I speculate that the hints might be more helpful if they directly addressed composition. A single "composed" hint for P3 could be:

> Hint: Ann’s distance from the dock is equal to the 800 yards she started out from the dock minus the 40 yards she rows per minute multiplied by the "m" minutes it takes her.

Variables vs. Constants

Although Koedinger & Anderson (1998) had shown that solving a concrete problem for an unknown can be easier than doing abstract symbolization (e.g., writing "800 - 40 * x"), in this study I found that concrete symbolization (e.g., writing "800 - 40 * 2") is not much easier, if at all, than abstract symbolization (the small trend in favor of concrete symbolization was not statistically significant). This result has implications for the design of the "Pattern Finder" component of the PAT algebra tutor (Anderson et al, 1995). The evidence from Koedinger & Anderson provided some support for the hypothesis that solving concrete problems aids students in symbolizing. The "Pattern Finder" is based on a further hypothesis that making this solution process more explicit through concrete symbolization would be an improved scaffold. The results of the current study put this hypothesis into question. At a minimum, they suggest that the Pattern Finder should require students to answer the concrete problem **before** doing the concrete question (e.g., first, "How far is Ann from the dock in 2 minutes?" and then "Write down how you got that answer."). Alternatively, since it appears that composing rather than abstracting is the real crux of the symbolization problem, attention should be focused on developing a scaffolding technique that directly addresses composition.

Conclusions from DFA Study 1

The large effect of the composition factor in this study, relative to the small or absent effect of comprehension hints, provides a strong case against the almost exclusive emphasis in previous research on language comprehension as the major stumbling block for students. A focus on language *comprehension* may be appropriate for the younger students learning arithmetic story problem solving. However, to address the difficulties of older students learning the new language of algebra, greater focus is needed on the language *production* skills needed to "speak algebra".

DFA #1 demonstrated a main effect of composition. The Articulating Composition Hypothesis predicts that students might be lacking knowledge of how to write composed expressions. To test this hypothesis I designed DFA#2.

Difficulty Factors Assessment #2

DFA #2 was designed to test the Articulating Composition Hypothesis that the composition effect was due to the fact that students have to learn how to write a composed expression. This is based on the idea that students have a hard time knowing that they can treat a sub-expression the same way they treat a number. Therefore, I decided to do an analysis of two factors. The first factor was composed vs. decomposed (the same as in the DFA above). The second factor compared symbolization with arithmetic evaluation.

The *articulating composition hypothesis* suggests that there should be no composition effect for arithmetic problems, but a large composition effect for symbolization problems. The reasoning behind this is that students know how to combine steps in an arithmetic problem, but they do not know how to say, in symbols, how to put two steps together. Therefore, it predicts that there should be many students who know how to do both composed arithmetic problems and decomposed symbolization problems but fail to do composed symbolization problems. The two hypotheses differ in their predictions because the *articulating composition hypothesis* predicts an

***CS*:** Sue made 72 dollars by washing cars to buy holiday presents. She decided to spend "m" dollars on a present for her mom and then use the remainder to buy presents for each of her 4 sisters. She will spend the same amount on each sister. How much can she spend on **each** sister?
***CA*:** Sue made 72 dollars by washing cars to buy holiday presents. She decided to spend 32 dollars on a present for her mom and then use the remainder to buy presents for each of her 4 sisters. She will spend the same amount on each sister. How much can she spend on **each** sister?
***DS1*:** Sue made 72 dollars by washing cars to buy holiday presents for each of her "s" sisters. She will spend the same amount on each sister. How much can she spend on each sister? ***DS2*:** Sue made 72 dollars by washing cars to buy holiday presents. She decided to spend "m" dollars on a present for her mom and then use the remainder to buy presents for her sisters. How much can she spend on her sisters?
***DA1*:** Sue made 72 dollars by washing cars to buy holiday presents for each of her 4 sisters. She will spend the same amount on each sister. How much can she spend on each sister? ***DA2*:** Sue made 72 dollars by washing cars to buy holiday presents. She decided to spend 32 dollars on a present for her mom and then use the remainder to buy presents for her sisters. How much can she spend on her sisters?

Table 2: The 4 Problem Types: Composed Symbolization (*CS*), Composed Arithmetic (*CA*), Decomposed Symbolization (*DS1* and *DS2* together), and Decomposed Arithmetic (*DA1* and *DA2*.)

interaction between these two factors, while the *generalization hypothesis* does not predict an interaction and instead predicts one broad symbolization effect.

Experimental Design and Predictions

The two factors I studied were 1) arithmetic vs. symbolization (*CA* vs. *CS* and *DS* vs. *DA*) crossed with 2) composed vs. decomposed (*CS* vs. *DS* and *CA* vs. *DA*). I illustrate all four cells of DFA#2 in Table 2.

Remember that the decomposed problem *DS* is actually one problem from the analysis point of view, but two separate problems from the student's point of view. In DFA#1 I may have even underestimated the size of the composition effect, because I had a few students who would answer "(72-m)/4" for *DA2*. These students had noticed the superficial similarity of the two problems parts of a decomposed problem and interpreted the second one to be linked to the first. To avoid such misinterpretation by students I changed a few superficial aspects of the second part of the decomposed problems including the participant's name. I also changed the numbers (making sure not to not decrease the difficulty of the arithmetic) to ensure that the results of the first part of a decomposed problem did not appear as a given in the second part of the problem. For example, *DA2* was changed to "Rebecca made 81 dollars working at the grocery store. She decided to give her mom 25 dollars for Mother's Day and put the remainder in her savings account. How much can she put in the savings account?" I found this strategy worked well and students did not think the two problems were connected.

Procedure

> Given the two binary factors that were studied, there were four different problem types: composed symbolization, composed arithmetic, decomposed symbolization and decomposed arithmetic. These 4 problem types were crossed with 8 different cover stories and distributed in a Latin square design among 4 test forms that were balanced for each factor. Therefore, each form had two problems of each type. Given that students tend to perform worse on items near the end of a test, the order of various problem types was systematically varied on each form. The cover-stories were not a variable of critical interest, and from my previous DFA I knew the performance on the cover stories would vary considerably. I used 6 of the 8 cover stories used previously and added two new cover stories to replace the two easiest cover stories from the prior study. Because I wanted to be able to compare the performance of students on cover-stories I made two versions of each form that differed only in the order of the problems. The subjects were 76 ninth grade

students in the first month of a regular-level algebra course from an urban high school in Pittsburgh. Two subjects were dropped for lack of participation in the test. Each student was randomly given one of the 8 different test forms and enough time to complete the test (about 10-30 minutes). Each test was then graded and no partial credit was given. A decomposed problem was considered correct only if both parts were answered correctly.

Analysis and Results

To test for the effect of the two factors I performed both an item analysis and a student analysis as recommend by Clark (1973). I performed a subject analysis on the students mean score for each of the four problem types. I performed a full two-factor (2*2) repeated-measures ANOVA with each factor as a within-subject variable. I found significant effects of symbolization (F(1,73)=170.8, p<.0001) and of composition (F(1,73)=22.4, p<.0001.) but not the expected interaction (F(1,73)=1.024, p>.31). To verify that these effects generalize across items as well as across subjects, I performed an item analysis on students' mean performance on the 32 different items (8 cover stories and 4 problem types) appearing on the test. I performed a full two-factor (2*2) ANOVA on the item means. Each factor was treated as a repeated measure. I found significant effects of symbolization (F(1,28)=55.476, p<.0001) and a smaller effect of composition (F(1,28)=13.26, p<.0083). Since the subject and the item analysis agree, I can be confident that both effects are real and generalize across the larger populations of both similar students and similar items. The size of the symbolization effect was much larger than the size of the composition effect. Contrary to the prediction of the *articulating composition hypothesis*, there was not a significant interaction (F(1,28)=.821, p>.395), and this will be addressed below.

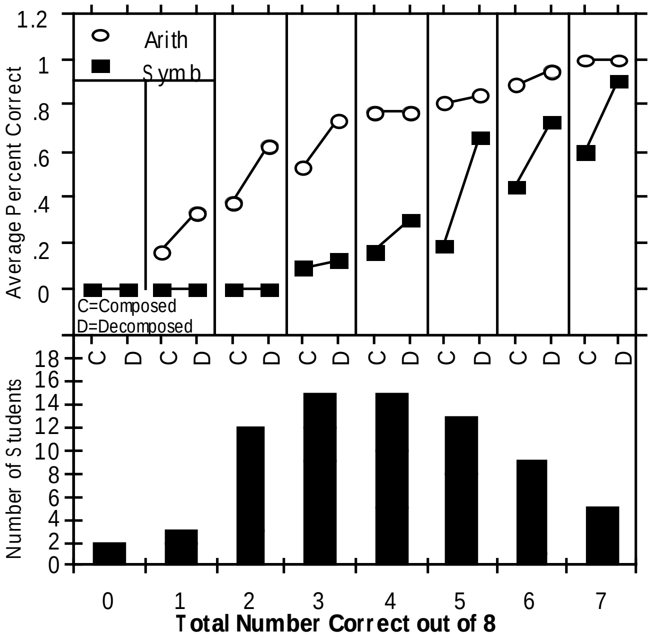


Figure 5: Lower-performing students show a composition effect for arithmetic problems, while higher performing students show a composition effect for symbolization.

I was initially surprised at the absence of the expected interaction, but upon investigation, I saw that the individual variation, as shown by the histogram in Figure 5, was very large. Figure

5[7] shows how average student performance changes as their ability level increases. The lower-performing students, with total scores of 1, 2 and 3, are all at the floor for symbolization. They also show a composition effect during arithmetic problems. This is the exact opposite interaction I was expecting to see, but there is an explanation[8] that is sensible. Students first learn to deal with the easier arithmetic problems, and the lower-performing students will show competence only in these. The fact that there is a composition effect among the arithmetic problems will be explored below. The middle-performing group (total scores of 4) show that the composition effect for arithmetic problems is going away as overall performance improves and student get close to ceiling. The high-performing students, with total scores of 5, 6 and 7, are doing equally well on both types of arithmetic problems and are improving on symbolization but primarily decomposed symbolization. In other words, the high performing students show the interaction that the *articulating composition hypothesis* predicted.

To statistically characterize the individual variation that was seen, I classified the subject population into a top-performing half and a bottom-performing half (using total score). I performed a full three-factor (2*2*2) ANOVA on the average of students' means, again treating our two difficulty factors as within-subject repeated measures, and treating the factor of "top half vs. bottom half" as a between subjects factor. Again, there were highly significant effects of symbolization (F (1,280)=172.5, p<.0001) and composition (F(1,280)=22.1, p<.0001). None of the three possible two-way interactions were close to statistical significance (P>.24 in all cases), but I did find a highly significant (F(1,280)=14.0, p<.0004) three-way interaction. Again, to verify that these effects occurred across items as well as across subjects, I did a three-factor (2*2*2) ANOVA on the item means, treating all three factors as within item repeated measure factors. Again, the only significant effects were the two main effects for symbolization (F(1,56)=50.1, p<.0002) and composition (F(1,56)=11.7, p<.012) and the three-way interaction (F(1,56)=9.4, p<.0183).

I will speculate below as to whether or not the composition effects seen in arithmetic (at the low end) and in symbolization (at the high end) are caused by a single effect or two separate effects. The *articulating composition hypothesis* suggests that they are different effects because the effect in the high end is due to students missing some knowledge of how to composed subexpressions together. I will then explore the other transitions I observed, which were not the main focus of the experiment, but nevertheless provide us with insight into the composite skills for composed symbolization.

Why are Composed Symbolization Problems Hard?

Our *articulating composition hypothesis* predicted that there would be an additional cognitive hurdle for students to be able to solve composed symbolization problems. When we look at the top-performing half of the students, we see the interaction, which suggests that composed symbolization problems are unduly difficult. However, is this average performance? Are most

---

[7] To make sure the reader understand this figure, lets look at an example. For instance, there were 3 students who got only one problem correct. For those four students, the upper potion of the graph shows how those students did on the 4 different types of problems. (Note the "C" and "D" on the upper x-axis, standing for "composed" and "decomposed", respectively.) For instance, these four students got 0% correct of the problems composed-arithmetic and the decomposed arithmetic problems correct, as indicated by the rectangular boxes. These students scored best (~30%) on the decomposed-arithmetic problems, and scored about 18% on the composed-arithmetic problems.

[8] There is a second explanation of this graph. In the main text, I argue that students have to learn to overcome composition in arithmetic problems separately from overcoming composition in symbolization problems (i.e., two different composition skills). However, my data does not rule out that what is observed in Figure 5 might be two main effects, with the tail ends being obliterated by floor and ceiling effects (i.e., a single composition skill).

of these students demonstrating the same effects? The answer is “some, but not all.” Let us define the *prerequisites* to being able to do a composed symbolization problem the ability to do both decomposed symbolization and composed arithmetic problems. Then 40% of the students who have met the prerequisites for composed symbolization, failed to solve a composed symbolization problem. When I say a student has met the prerequisites, I mean that a student was able to do at least one composed arithmetic problem, as well as one decomposed symbolization problem. There were 43 students who met this criterion, and 17 of them failed to solve a composed symbolization problem. I think that these 17 students are missing the knowledge of how to compose two symbolic expressions together.

These 17 students made a total of 34 errors on composed symbolization problems and I would like to see if these errors are consistent with the *articulating composition hypothesis*. The largest category of errors (done by eight students) is where the student wrote only the first step (e.g., one student wrote “72-m” for problem *CS*.) These errors are consistent with the *articulating composition hypothesis*, because a student that does not know how to put two steps together might simply stop once she gets to a point where she does not know how to continue. There were 3 errors of missing parentheses and one student who incorrectly used parentheses. These errors are also consistent with what one would expect from a person who can both symbolize single steps and do the arithmetic for two step word problems, but doesn’t yet know how to put two symbolizations together. Many of the other errors are not easily categorizable and do not give us much insight into what the student was thinking. About a third of the errors of students who appeared to be trying to overcome this hurdle are those that the *articulating composition hypothesis* would predict.

The Symbolization Effect Among Decomposed Problems

We now consider one of the transitions students must make before being able to deal with composed symbolization problems. This transition occurs when students learn decomposed symbolization problems. The evidence suggests that this transition is difficult. Fully one-third of the students who demonstrated competence (at least one correct) in the decomposed arithmetic problems could not get any of the decomposed symbolization problems correct (let alone the composed one). This is a substantial effect that results in an additional 108 errors occurring (152 errors on decomposed symbolization problems while only 44 errors on decomposed arithmetic problems). What is the explanation for this *symbolization gap* of 108 errors? There were many more “no answers”, increasing from 12 to 57, but these “no answer” responses do not give us insight into what is causing the difficulty. We now look at four different explanations that might account for why there are so many more errors, but cannot conclusive rule out any of them.

1. **Articulating Composition Hypothesis: Articulating is harder than computing**
2. **Generalization Hypothesis: It’s the variable!**
3. **Students using different mathematics operations: repeated addition instead of multiplication**
4. **Lack of Semantic Support**

The first hypothesis to explain the symbolization effect in decomposed problems is similar to the *articulating composition hypothesis* in that it focuses on difficulties students have with production and not with comprehension. I will introduce two types of errors that are analogous to the sorts of errors made when learning a natural language. One such error is to say words in the wrong order, for instance, writing “m-72” when “72-m” is intended. A second language production error is to use the wrong "verb". For instance, writing “72+m” when “72-m” is intended. However, just because a person has written “72+m” does not mean we can assume they meant “72-m”; we will have to compare the error rates on symbolization problems with the number of analogous errors on arithmetic problems (for instance the number of people who added 72+32 to get 104 for problem DA2 in Table 2.) It turns out that there are 7 errors of students using the wrong verb to arrive at an arithmetic answer. In contrast, for symbolization problems, there are 31 such errors. I argue that the increase in these wrong-operator errors is because students’ difficulties arise not so much in English comprehension but in symbolic production. A similar pattern for the reversal errors was observed, with no such errors occurring for arithmetic problems (nobody answers “-40” for *DS2*) while there are 12 such errors for symbolization problems. The difference in these simple articulation errors accounts for about 40% of the symbolization gap.

A second explanation is a possible consequence of the *generalization hypothesis*. Consider the 6th grade student who has never seen a variable before and is suddenly confronted with a problem with a variable for the first time. She would probably be very confused and be inhibited from using that variable in her answer. She might be much more likely to randomly guess using the more familiar numbers from the problem. Alternatively, she might answer as one student of ours did on several of the symbolization problems: “Not enough info”. This leads to the prediction that novices are inhibited from using a variable. Of the students who gave an answer, 38 answers did not contain the given variable even though in a decomposed problem there are only two numbers. This accounts for 35% of the symbolization gap. However, there is a consistency problem with this variable inhibition hypothesis. In the first DFA, I found that students’ ability to do arithmetic symbolization (e.g. answering with “72-32” rather than “40”) was not significantly different (F(1,30)=.9, p>.35) than their ability to do algebraic symbolization (e.g. answer with “72-m”). However, I do note that the trend in the data was particularly large and in the predicted direction: for decomposed problems, the absolute performance rises 58% for arithmetic symbolization problems to 68% for symbolization problems with a variable. Furthermore, the students in the current study were generally at a lower level[9] of skill where variable inhibition may be greater.

A third hypothesis is that the symbolization effect for decomposed problems is caused by the fact that students may be using back-up strategies (like repeated addition: 40+40+40) for the arithmetic problems instead of the direct arithmetic strategy (e.g. 3*40) that is necessary for symbolization. I looked at all of the responses students gave to the arithmetic problems to see if I saw any evidence that students were using any mathematics operations other than those expected, and found no evidence to support this hypothesis. At the same time, since students did not always show their work, I cannot rule this out.

A fourth explanation for the presence of a symbolization effect in decomposed problems is what I call the *semantic support hypothesis*. If a student actually has numbers to compute answers with, they have the advantage of being able to look at the answer in order to detect violations with the semantics of the problem. I discussed above that there were no reversal errors on arithmetic problems but 12 on symbolization problems. Maybe the reason reversal errors do not occur on decomposed problems is that if a student calculates “-40” she will quickly detect that a negative number of dollars to give to the sisters does not make sense. The *semantic support hypothesis* also suggests that even without doing any of the arithmetic, it might be easier for a student to figure out which operator to use if they know the relative size of the numbers given: students seem to have heuristics such as “if one number is much larger than the other, then

---

[9] On the 6 cover-stories that the two populations had in common the percentage correct, on average, for symbolized problems was significantly different (F(1,10)=7.292, p<.02) with this present group averaging a low 27.2% while the previous group averaged a higher 42.8%.

division is likely", in addition to heuristics like "always subtract the smaller number from the larger." Therefore, a student might benefit more by having numbers rather than variables even before she actually does any mathematics. In DFA#1 I failed to find a significant difference (F(1,30)=.9, p>.35) between the performance on arithmetic symbolization vs. algebraic symbolization problems (as mentioned in the introduction). There was a difference a 10% on the student's average percent correct (from 58% to 68% correct). This suggests that there might be a role for the semantic support hypothesis that helps students use heuristics based on the size of numbers as well as heuristics that check the result of arithmetic computations for semantic violations. Neither of these strategies is possible if the problem has a variable present. Further research is needed to conclusively distinguish between the merits of these explanations.

Composition Effect for Arithmetic Problems:

The third developmental transition I detected was the gap shown by students who could not do composed arithmetic problems (which I call a "composition effect for the arithmetic problems"). The gap is not large; there were 42 errors on decomposed problems while composed problems had 52. Again, I notice that 10 of the answers for composed problems were the result of doing just the first of two required steps. These students possibly stopped early because they were not reading the question carefully. There were only 5 students who were able to get a decomposed problem correct but failed to get any other type of problem, including composed arithmetic problems, correct. There were 24 students whose performance on composed arithmetic problems was worse than their performance on decomposed arithmetic problems. I assume that this composition effect is partly explained by reference to cognitive models such as LeBlanc and Russell's (1996) model that attempts to explain arithmetic word problem performance based on working memory load considerations. That is, composed problems are somewhat harder to comprehend.

An Analysis of the Errors Students Made on the Two DFAs

This chapter has, so far, has focused on whether a student can get a problem correct or not. However, in order to build a tutoring system, we want to understand what are the common errors. I performed an analysis of errors committed on the second DFA. I was interested to see if the types of errors made were consistent with my hypothesis that language production on composed problems is a major difficulty.

Procedure

I looked at thee errors made on the second DFA. Each test had 8 questions. Because of the Latin Square design, half of the questions were decomposed versions that had two sub-questions. Therefore, each student did 12 problems on a single test. This data was collected during an empirical evaluation on an unrelated instructional system. Therefore, each student did two tests, with each test version randomly chosen with replacement. Because the presence of distractors made for a much wider range of error types, I decided to focus on the half of the problems that did not have distractor numbers. Because I was interested in the composition effect for symbolization problems, I excluded those problems without variables. I was left with 136 decomposed problems and 139 composed problems (the numbers are not exactly equal because a few students did only one side of the sheet, and a few forms had errors that caused us to exclude a question). Of these problems, the decomposed problems had a percentage correct of 64%(49 errors) while the composed problems had a percentage correct of 46%(75 errors), a difference of 17.5% (26 errors) representing the size of the composition effect (for problems without distractors and also that have variables (not arithmetic). I analyzed the types of errors found and classified them into categories.

Results

Of the 75 errors on composed problems there were 5 categories of errors that were consistent with the hypothesis that the student's difficulties was one of articulation the composition of the two parts of the problem. Table 3 lists these four categories. Table 3 also lists, for reference, two other common categories that apply to both decomposed and composed problems. The first error type is missing parentheses. The ten students did made this sort of error could account for a little over a third of the composition effect (26 problems).

The second type of errors that is consistent with the composition effect is what I call a "sub" error where a student does the first step, but then stops. This could be due to the student knowing how to do the first step but then not knowing how to proceed. (It is noted that these sorts of errors could also be explained by fact that the student had the wrong goal in mind.)

The third error type is the "super" error and there were two examples of this. The student wrote the "800-m" instead of "800-40m". I classify this as a possible composition effect error because it might be that the student simply did not know how to substitute "40m" in the place where they put just "m".

The fourth error type shown is what I term *invented notations.* Table 3 shows the three examples of errors I classify as invented notations. In these, the student seems to understand the structure of the problem but did not express the answer in one correct algebraic expression.

The fifth error type occurred for two students who used parentheses in the wrong place. As in the case of the missing parentheses errors, it is possible that the students had the correct quantitative structure in mind, but simply did not know how to expresses this correctly.

	Number of Errors		
Error Name	**Decomp**	**Comp**	**Example**
Missing Parentheses	NA	10	Wrote 550/h-2 instead of 550/(h-2).
"Sub"	NA	14	Wrote "40m" instead of "800-40m".
"Super"	NA	10	Wrote "800-m" instead of "800-40m".
Invented Notations	NA	3	Wrote "62-f=t, T+62" instead of "(62-f)+62", or wrote "72-m=x/4=" instead of "(72-m)/4", or wrote "5-7 and 5*h" instead of "5h-7".
Wrong Parentheses	NA	2	Wrote "(550/h)-2" instead of "550/(h-2)".
Wrong Math Operator	17	17	Wrote "800-40/m" instead of "800-40m"
Order of Arguments	8	2	Wrote "40m-800" instead of "800-40m".

Table 3: The five different error types that are consistent with a student having a composition difficulty, plus the two error types that were prominent on one-operator problems.

Overall, the actual thirty-nine errors of the five error types of Table 3 could more than account for the size (29 errors) of the composition effect on symbolized problems. These types of errors were those one would expect if the difficulties were a knowledge-level problem, such as failing to realize that a parenthesized expression could be treated the same way a number could be treated. This analysis of errors shows that there were enough errors of the types that are consistent with the hypothesis that students were having difficulty composing the expressions together.

Study to Detect The Transfer of Composition Skill

So far, in this chapter, I have explained two difficulty factors assessments I performed. Those studies suggested the hard part of symbolization is when students have to compose expressions together, because it requires them to be able to treat an expression like a number. We can thing of this concept as learning a new rule in the grammar for algebra expressions. Therefore, as I conjectured above, instruction focused on teaching students this grammar rule should improve their performance, even if divorced from practice on word problems. Therefore, I designed an experiment to see if practice on algebraic-substitution problems would transfer to the task of symbolizing expression for word problems. The algebraic substitution problems were problems such as the following: (see Appendix B for the posttest and Appendix H for the classroom worksheet.)

> "Let X= 72-m. Let B= X/4. Write a new expression for B that composes these two steps."

These problems had no verbal problem statements! If students got better at symbolizing expressions for verbal problem statements by practicing algebraic substitution, then I can conclude that these two tasks tap skills that underlie both of these tasks.

Procedure

> The subjects were 39 students in three ninth grade regular level algebra classes. The high school was a suburban Pittsburgh classroom. I gave each student a pretest (included in the Appendix B) that included 5 symbolization problems and 5 algebraic substitution problems on Day 1. On Day 2, students were then given traditional whole class instruction on this task. The worksheet the students used during guided practice (i.e., teacher giving students time to do a few problems and then going over them) is given in Appendix H. The total time was about 45 minutes. On Day 3 student were given a posttest. The posttest was the same as the pretest. Each test was graded and no partial credit was given. The pre-post test had 5 symbolization problems as well as 5 substitution problems.

Results

I performed standard repeated measures t-tests to determine if the students showed improvements on the two skills tested. Not surprisingly, students had significant improvement on the skill they were trained on. Those means on the algebraic substitution pretest went from an average of getting .64 problems correct to 2.8 problems correct on the posttest. (p=.001, Sigma=2.36). More surprising is that students showed a small but statically significant improvement in there ability to solve symbolization problems (Figure 6). The average percents correct went from .308 to .82 problems correct (p=.0012, Sigma = .74). Since the absolute means on the pre/test and posttest were quite small I checked to see who were the students doing the learning. Surprisingly, I did not find a statistically significant correlation between the symbolization gain and the substitution gain. However, I did find that many of the students who learned the most substitution skills improved their symbolization performance considerably. Specifically, of the 19 students who gained a lot of knowledge on substitution problems (as measured by a gain of more than 3 problems from pretest to posttest) 8 of them improved their performance an average of 1.71 problems each.

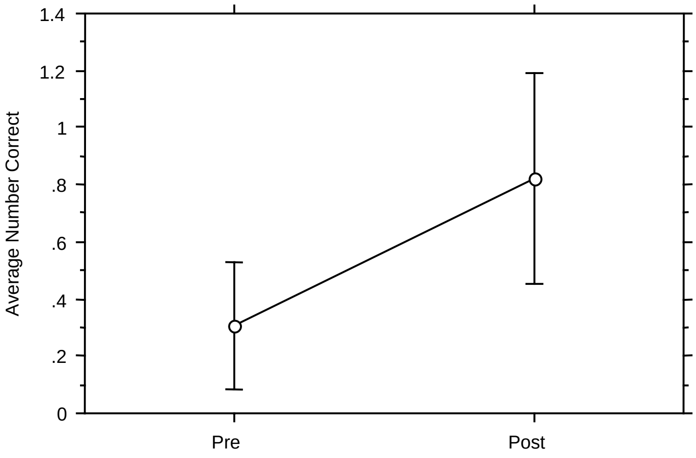


Figure 6: Number of Word-Problems Symbolized Correctly for Pre and Post Tests.

A sub-hypothesis is that maybe the students were just learning when to put in parentheses and that that is the skill they transferred. To deal with this objection, I did a second test. This time I rescored the problems so that those students' answers that had left out parentheses were counted as correct. I then looked to see if there was still transfer in the way described above. The results were that there was still difficulty left unaccounted for (Pretest means=.513, Post Test means=.872,.p=046 and sigma=.45).

Conclusion for Transfer Study

When students practice the skill of algebraic substitution, they are practicing the skills that involve them treating an expression as a quantity in and of itself. This is different than treating an expression as a procedure you can use to compute something. This practice transfers to the seemingly unrelated task of symbolization. This shows that one component skill of efficient symbolizers is the ability to treat an expression the same way they treat a number. I can state this same idea by saying that students need to learn the grammar of algebraic expressions, and in particular, that an expression can be composed of subexpressions. This is a skill that our student model needs to consider as one of the subskills for symbolization. This is possibly the strongest evidence I could present in favor of the *articulating composition hypothesis*. This result also supports the more general idea that there is great difficulty on the production side (rather than the comprehension side of translation). These students improved without practice reading algebra word problems; the only practice they had was composing expressions together.

Knowledge-Level vs. Symbol-level Explanation

Our *articulating composition hypothesis* is what Newell (1990) would have called a knowledge-level explanation as opposed to a symbol-level explanation. I hypothesize that students are missing a certain bit of knowledge (which I model as production rules). However, an alternative explanation of the additional difficulty of composed symbolization problems could be made at the symbol-level. For example, maybe both symbolization problems and composed problems put a burden on working memory, and when these two factors combine a strong interaction results. Anderson & Jeffries (1985) explained a similar result by reference to working memory. Anderson & Jeffries found that student had more difficulty time identifying the correct operator to choose to write a piece of LISP code, when the arguments to the function were more complicated. They compared student’s performance on the following two problems

(? ’(A) ’((B) (C) (D))) = (A (B) (C) (D))
(? ’((N)) ’(((O)) ((P)) ((Q)))) = ((N) ((O)) ((P)) ((Q)))

In both cases the student’s task was to figure out which LISP, function (to replace "?" above) transforms the arguments (with lists being preceded by single quotes) on the left-hand side of the equal sign into the result (list structure on the right-hand side of the equal sign). The correct answer is to use the function APPEND. The only difference between the two problems was that in the second case there is an extra set of parentheses around each element-- e.g., (A) versus ((N)). Anderson and Jeffries said that the differences between the two problems is "totally irrelevant to determining whether append is the correct function." Subjects got 80% correct for the first case and 66% for the second case. Anderson and Jeffries proposed "the error rate was increased because of the burden on maintaining the extra information [about the extra parentheses] in working memory."

However, a second alternative explanation for that difference might be that students needed to be more proficient in the grammar for LISP expressions in order to correctly interpret the extra parentheses. For example, in the second case that has the extra set of parentheses students need to use their grammar rule more often (i.e., the rule that that allows them to parse the extra set of parentheses.) Therefore, Anderson & Jeffries results could be explained by either a symbol-level (i.e., working memory) explanation or a knowledge-level explanation (i.e., increased practice with the rules of parentheses). Anderson & Jeffries predicament is similar to the one I have in this work.

One way of trying to get hold of the two types of explanation is to look at the types of error made. For instance, Payne & Squibb (1990) studies the types of error students made in algebra question solving. For instance, they compared the following two types of student errors:

$M + Nx = P \rightarrow Nx = P + M$
$M + Nx + Qx = P \rightarrow Nx + Qx = P + M$

Both of these involve the same error of failing to change the sign when moving a term across the equals sign. Payne & Squibb found that while equations that are more complicated were harder to solve, there was no increase in this specific error type. It was simply that equations that are more complicated offered more opportunities for errors. When I looked at the types of errors students made in my task (Table 3), I reported above that students made a sizable number of errors consistent with their error being explained by a knowledge-level difficulty. Additionally, Table 3 shows that the number of errors of using the "Wrong Math Operator" stayed constant. The number of errors of "Order of Arguments" actually dropped. This result is different than one of Sleeman’s (1984) results. Sleeman (1984) found that "students appear to regress under cognitive load (p. 403)" meaning that they might make an error on a harder problem that they would not make on an easier one. Unfortunately, my data is not longitudinal, so I cannot answer the question of whether students would make the same types of errors on harder problems after having stopped make that error type on easier problems. However, the increased frequency of errors on the composed vs. decomposed problems in my data, does not appear to be accounted for by the kind of "regression under cognitive load" phenomena observed by Sleeman. The errors introduced in composed problems are in new error categories, specific to composed problems, not increases in errors that can also appear in decomposed problems.

Sleeman (1984) also said "This paper claims that there are two very different types of mal-rules at large with algebra students-namely manipulative and mis-parsing mal-rules." The mis-parsing mal-rules are the sorts of errors you would expect if students were not familiar with the grammar of algebra expressions.

I also considered the possibly that a working-memory explanation might apply to **my** task. I looked for evidence by seeing if the composition effect (while doing symbolization problems) increased under a working memory load. Conveniently, I have data to make this comparison, from DFA#1. In this, I studied both composition while symbolizing as well as the effect of distractor numbers. To illustrate, I had studied problem *DS1* with and without a distractor phrase (I changed the problem to read “6 cars” rather than just “cars.”) As was mentioned earlier, a large effect for these distractor phrases was found, and it seems reasonable to speculate that this might be an effect of an increased working memory load. If the composition effect in symbolization problems is a working memory phenomenon we should expect to see the composition effect increase for distractor problems. Unfortunately, the results, shown in Figure 7, are inconclusive. There was a trend towards the predicted interaction but it is not significant (F(1,250)=1.48, p=.23). Therefore, we cannot conclusively rule on the question of whether the composition effect is a symbol level phenomenon or a knowledge level phenomenon.

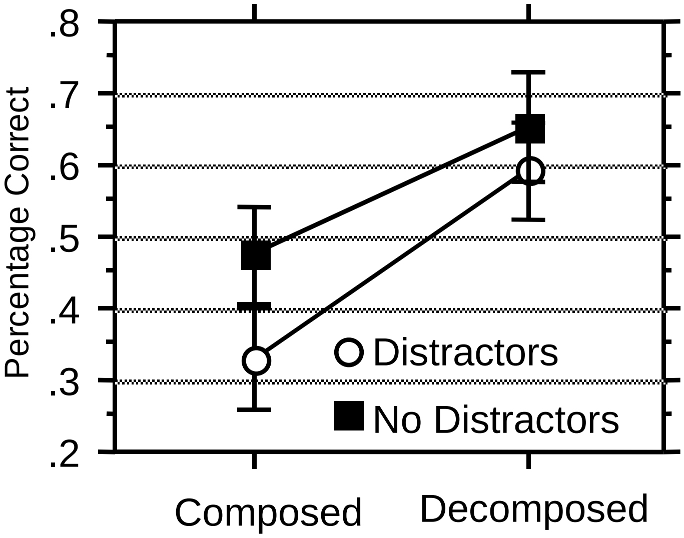


Figure 7: The composition effect gets worse in the presence of distractors. This suggested interaction (not statistically significant) is consistent with a working memory load explanation.

In a related task, Lebiere, Anderson & Reder (1994) found increased difficulty with an algebra symbol manipulation task as the working memory load increased, which might suggest the symbolization effect could be accounted for with a similar working-memory explanation. Anderson, Reder and Ritter (manuscript in preparation) found a similar result; students were more likely to make certain types of errors (such a failing to invert an operator when transforming an equation) if the problem involved decimals as opposed to whole numbers. They explained this in terms of a working memory effect; students have a harder time remembering to invert an operator if they also having to deal with the increased difficulty of using the more complex decimals. This task sounds similar to the one presented here, in that I showed that students made more symbolization errors when presented with composed as opposed to

---

[10] In an unfinished manuscript.

decomposed problems. The focus of their explanation is different; they proposed a working memory explanation (i.e., symbol-level) to explain their results.I, one the other hand, have emphasized a knowledge-level explanation for the results in my task. But, possibly there is also room to use this knowledge-level explanation (i.e., student's don't know the grammar of algebra expression) to also explain result in Anderson's domain of symbolic manipulation. For example, the following would be examples of composed and decomposed versions:

Composed Version	Decomposed Version
17=4*x+5	17=y+5 and 12=4*x

A similar composition effect might be found. This hypothetical effect could be explained with knowledge-level explanations (student don't know the grammar for algebra expression well and therefore mis-parse the expression which leads to them to first divide my 4 rather then subtracting 5 from both sides.) or symbol-level explanations (student face a higher working memory load on the composed version and therefore more likely to make an error.) Teasing apart a knowledge-level versus a symbol-level explanation is different, and a subject for future research. Possibly evidence might be found that shows clearly knowledge-level problems, such as students using shallow-rules that do not generalize.

This dissertation does not prove that there is no room for a symbol-level explanation (i.e., working memory-based explanation); rather it simply provides some evidence that is consistent with a knowledge-level explanation. This evidence is

1. The model we presented to account for the effect of distractors as well as composition cannot be made to predict the fact that the distractor effect is in fact smaller then the composition effect.
2. The fact that there are enough errors of the type consistent with the knowledge-level explanation to account for the additional number of error observed in composed problems than decomposed problems.

The fact that students get better at symbolization from instruction that focuses on the grammar or expression.

A Developmental Model

Based on the results of DFA#1 and DFA#2, I have created a developmental model of how students learn to symbolize. Contrary to common belief and the emphasis of prior cognitive science research, student' difficulty in algebra word problem solving appears less related to comprehension difficulties and more related to difficulties in producing symbolic expressions, particularly expressions that involve more than one operator. I will summarize three main transitions in this model. At the start of the developmental progression are students (two in this study) who fail to get any of the problems correct; I will ignore these two students. At the first non-trivial level, there are five subjects who showed competence in decomposed arithmetic problems (where competence is at least one of the two problems correct) but failed to show competence in any other problem type. I speculated that their poorer performance on composed problems might be due to working memory limitations and/or related to difficulties comprehending more complicated composed stories. The students at this level show evidence predicted by the *generalization hypothesis* that argued that variables make problems harder (students didn't get any problems correct that had variables). In Chapter 3, I will model this with three skills: 1) being able to retrieve the operator, 2) being able to retrieve the numbers, and 3) being able to order the augments. I have chosen not to bother modeling the additional help that students get by being able to refer to the semantics of the problem.

In going to the next level students must learn to deal with composed arithmetic problems. They most likely need to improve their reading skills and make sure they answer the question asked. Twenty-four students had made this transition to competence in composed arithmetic nevertheless, failed to be successful on any symbolization problems. In Chapter 3, I will model this as students learning a new skill of being able to retrieve a computed quantity.

The next hurdle students surmount is learning how to deal with decomposed symbolization problems. It is interesting to note that no student was competent in decomposed symbolization problems who was not also competent in composed arithmetic. This is most likely due to the fact that students see composed arithmetic problems in elementary school, but do not see variables until much later. There were 17 students at the level of showing competence (at least 1 out of 2 correct) in both composed arithmetic and decomposed symbolization, but were not successful on any of the composed symbolization problems. I presented four alternative hypotheses regarding what students need to learn to make this transition to competence on simple decomposed symbolization problems. In Chapter 3, I will model this development by adding two new skills of being able to use a variable, and being able to articulate the mathematics used to compute a quantity.

At the highest level, there were 25 students who were able to get one of the composed symbolization problems correct (only one student got both composed symbolization problems correct). In making this transition to competence in composed symbolization problems, students learn how to combine the articulation of individual steps.

Implications of the Developmental Model

I will limit myself to one instructional design suggestion that can be derived from this developmental model. In order to help student transition from competence at symbolizing one-operator problems to competence at symbolizing multiple operator problems, I recommend practice on symbolic substitution problems. On the face of it, such problems seem totally unrelated to translating word problems to symbols. However, my cognitive analysis and difficulty factors assessments have identified substantial overlap in the skills required for these apparently unrelated tasks. Furthermore, I have shown preliminary evidence that training on substitution transfers to symbolization. I will explore such a tutorial strategy in Chapter 4 and will use the name "Introduced Variable" strategy.

Chapter Conclusion

Many researchers have focused on the problems students have comprehending algebra word problems. For instance, Nathan, Kintsch, & Young (1992) "claim that symbolization is a highly reading-oriented one in which poor comprehension and an inability to access relevant long term knowledge leads to serious errors." However, this chapter presented data that shows that many students can comprehend algebra word problems well enough to evaluate an answer, but nevertheless, will fail to be able to symbolize the exact same problem, if it involves a variable. Furthermore, it is not the variable, per se, that makes symbolizing hard as suggested by Kuchemann (1981) and others. Rather the difficulty lies on the production side of the translation process; in particular, it is the articulation required when writing an expression that is hard.

This chapter also established the "composition effect" that a two-operator word-problem is harder than the some of its parts. This effect occurs in word problems that allow the student to compute a value. This increased difficulty captures any increased difficulty of the problem presentation. More interesting, there was a composition effect on problems that involved the use of a variable. Though the actual presence of the variable did not significantly increase the difficulty of the problems, I argued that the need for the student to articulate an answer lead to the increased difficulty.

This chapter ends with drawing two instructional implications that will be incorporated into the tutoring system described in the coming chapters.

Redesign of the Carnegie Learning Algebra Word-Problem Tutor

The results of this chapter suggest a redesign of the scaffolding technique used in Carnegie Learning Inc. word-problem tutor (Koedinger, Anderson, Hadley, & Mark, 1995). At the beginning of this chapter, we showed the "Pattern Finder" window used in Koedinger et al.'s tutor:

Ann is in a lake that is 2400 yards wide. She starts out 800 yards from the dock. She rows for “m” minutes back towards the dock at a rate of 40 yards per minute. Write an expression for her distance from the dock.[11]

Tutor:How would you calculate THE DISTANCE ANN IS FROM THE DOCK for 2 minutes?
Student:800-2*40
Tutor:How would you calculate THE DISTANCE ANN IS FROM THE DOCK for 3 minutes?
Student:800-3*40
Tutor:How would you calculate THE DISTANCE ANN IS FROM THE DOCK for 4 minutes?
Student:800-4*40
Tutor: Write an expression which describes your calculations, using a variable.
Student:800-m*40

Table 4. Pattern Finder dialog showing the student having answered the first 3 questions correctly.

This "Pattern Finder" scaffolding seems reasonable because the articulation of the arithmetic seems like a natural intermediate step before symbolizing with a variable. However, I showed in the first DFA that there is not a significant difference between arithmetic symbolization and algebraic symbolization so the first step of this scaffolding is as hard as the last step. Koedinger[12] has confirmed this by analyzing the performance of students on the individual steps of the pattern finder and found there is little difference between students' ability to perform any of the last three steps shown in Table 4.

My suggestion, based on the articulation composition hypotheses, is that if a student is having difficulty, first ask him (or her) to compute a value, and then follow that up with asking him articulate how he got that value. If he fails at that task, the tutor should ask him to articulate any decomposable elements (such as "40*3"). Once the student has articulated the part(s) then ask him again to articulate the whole quantity (e.g., "800-40*3"). Once that is done, then ask him to generalize his answer to the case without the variable. This is illustrated in the following example:

T1: Tutor: Hello. Please write an expression for the distance Anne has left to row.
S1: Student: Can you give me some help please!
T2: Tutor: Let me try to give you some help. We will come back to the distance Anne has left to row. Let me try to break this down for you. Please compute the distance Anne has left to row if the number of minutes rowing was 3 minutes?
S2: Student: 680
T3: Tutor: Correct. Using only 800, 40, and 3 write an expression for the distance Anne has left to row.
S3: Student: 800-120

---

[11] This is not one of the problems used by the Carnegie Learning Tutor.

[12] Personal communication.

T4: Tutor: No. But that is only because you didn't say where the 120 came from. What math did you do to get the 120?
S4: Student: 3*40
T5: Tutor: Correct. Now, lets try this question again: Using only 800, 40, and 3 write an expression for the distance Anne has left to row.
S5: Student: 800-3*40
T6: Tutor: Correct. Now use "m" instead of 3 to again express the distance Anne has left to row.
S6: Student: 800-40m
T7: Tutor: Correct.

Table 5: Proposed New Concrete Articulation Strategy

If articulation is hard, then we need to be ready to see errors at that step since that step is about as hard as having to symbolize the whole thing. Above at line S3, the student fails to fully articulate out one of the components to the problem, so the tutor asks for that component in what I call a *sub-dialog* (e.g., lines T4 and S4). Once the student answer "3*40" the tutor gets back to the main thread by asking the same questions over again (T3 and T5). I will refereed to this whole strategy as the *concrete articulation* as opposed to *inductive support* because this new name emphasizes the most difficult aspect of this symbolization while the name *inductive support* emphasizes the role of inducing a pattern from multiple instances.

A Second Instructional Scaffolding Strategy Based on my Transfer Study

Another suggested scaffolding method, which I call the "introduce variable" strategy, is shown in the following example:

Tutor1: Hello. Please write an expression for the distance Anne has left to row.
Student1: Can you give me some help please!
T2: Let me try to give you some help. We will come back to the distance Anne has left to row. Let me try to break this down for you. What is the distance rowed so far?
S2: Can you give me some help please!
T3: Let me try to give you some help. Use "40" and "m" to calculate the distance rowed so far.
S3: 40m
T4: Correct. If "B" were the distance rowed so far then what would be the distance Anne has left to row?
S4: Can you give me some help please!
T5: Let me try to give you some help. What is the distance Anne has left to row? Use "B" instead of "40*m" for the distance rowed so far. Hint: pick one of these: 800-B, B-800, 800+B, 800*B, 800/B, or B/800.
S5: 800-B
T6: Correct. Now, lets try this question again: You are correct that "800-B" is the distance Anne has left to row. Now, can you state this in terms of the numbers given in the problem?
S6: Can you give me some help please!
T7: Let me try to give you some help. What is the distance Anne has left to row? Hint: You know the answer is "800-B" but you must substitute in the value for "B". Replace "B" with "40*m". Use parentheses if needed.
S7: 800-40m
T8: Correct.

Table 6: A suggested strategy to scaffold the composing steps together, through the use of an introduced variable.

In the above example, we see that the tutor asks (T2 and T3) the student for one decomposed part (i.e., the 40*m) and then the second (T4 and T5) decomposed part (i.e., 800-B). Finally, the student is asked to compose the two parts together (T6 and T7). If articulating composition is what is hard, as I have argued, then this might be an effective strategy. Notice how the hint at line T7 explicitly instructs the student to perform the algebraic substitution of putting "40m" in place of B in "800-B". This is the same skill that was practiced in the transfer study I reported in this chapter. In that study, a statically significant effect, but small, effect was seen. The smallness of the effect might have been due to the fact that students were not told anything about how to transfer the skill to symbolization problems. This strategy can be thought of as a way of teaching students how do make this transfer.

Both the *concrete articulation strategy* and the *introduce variable strategy* will be incorporated into a new tutoring system that will be explained in the coming chapters

Chapter 2: An Experienced Human Tutor that Motivated the Tutorial Model

Chapter Overview:

In this chapter, I present some generalizations from my analysis of the transcripts of an experienced human tutor. I argue the human tutor's behavior can be roughly characterized with two processes: *dynamical scaffolding* and *tutorial strategies*. Dynamic scaffolding is a three-step process whereby the tutor 1) diagnoses the student's response, 2) possibly gives feedback, and then 3) selects a domain dependent tutorial strategy to address any error the student made. The tutorial strategies can be modeled as multi-turn dialog plans.

Motivation

Before beginning this dissertation, I was an algebra teacher. I found that my students had great difficulty with word-problems because they often made errors on the initial symbolization (i.e. translation of the problem into the language of mathematics.

One of this project's goals has been to understand how to improve the learning of algebra symbolization. Testing that understanding can be done by building better software for algebra instruction. There are many sources to draw on to figure out how best to accomplish this goal. In the last chapter, I reported on the cognitive modeling and related work (i.e. two difficulty factor assessments and a transfer study). In this chapter, I draw on a different knowledge source: a transcript of a single experienced human tutor working with a student for an hour.

Bloom (1984) reported that human tutors could leads to an effect-size of 2 sigma over ordinary classroom instruction (but this result has not been replicated). The natural question is what exactly do human tutors do that lead to these large learning gains? In this chapter, I present my hypotheses about what makes tutoring so effective. Note that, others might look at this same protocol and make different hypotheses or they might argue that our tutor is not representative. That might be. However, the ability to build a system based on one's hypotheses, and to know with certainly that learning gains were made during the use of that system, must be considered significant. I will look at this tutor through the lens of the cognitive scientist who has practiced in this domain. In the end, I hope that the reader will be convinced that the generalizations I make about this protocol seem reasonable. In addition, I hope that these generalizations will be considered worthy of testing to find out if they lead to gains in student learning.

Prior Studies of Tutors

Tutoring is reported to be a highly interactive process offering a large amount of confirmation (Merrill et al., 1995). Merrill et al. argue that the main thing human tutors do is to keep students on track and prevent them from following "garden paths" of reasoning that are unproductive and unlikely to lead to learning. Merrill et al. pointed to the large number of remarks made by tutors that helped keep students on track while learning Lisp programming. They argue that computer programming is the sort of domain that is hierarchical and can be broken into steps, and if one of those steps is wrong then everything else they do is going to be wrong. This can waste a lot of the student's time.

VanLehn et al. (1998b) found that tutorial intervention is effective usually only when the student had reached an impasse and furthermore that "different target pieces of knowledge seem to require types of different tutorial explanations in order to be learned." The CIRCSIM (Cho et al., 2000) research project has also studied tutors and found that their tutors used a somewhat consistent tutorial script (they called them "protocols").

An important and controversial question from the literature is on how much tutors diagnose student misconceptions. Brown & Burton (1978) argue that human teachers do not do enough diagnosis. Putnam (1987) reports that tutors rarely determine the exact nature of misconceptions, or attempt to do so by asking diagnostic questions. Instead, Putman argues that tutors work off a "curriculum script" that is a structured approach that is rather insensitive to students' responses, as compared to the diagnostic/remediation approach. Chi et al. (1996) concur saying "tutoring effectiveness in the form of deep understanding does not seems to arise from tutoring skills per se, such as diagnosing misunderstanding or giving didactic explanations." Graesser et al. (1999) observed untrained tutors (He cites Cohen et al. (1982) who claimed that untrained tutors can be effective) but found no evidence of "error diagnosis and remediation" or any other "sophisticated tutoring techniques" he was looking for. Lepper, Aspinwall, Mumme & Chabay (1990) reported that human tutors do some diagnosis but do not report that diagnosis to the student; rather tutors use that diagnosis to influence what scaffolding strategies to use with the student.

McArthur et al. (1990) did a sizable analysis in the algebra domain, which is particularly relevant to this work.

> *Perhaps the most important conclusion we can draw from our analysis is that the reasoning involved in tutoring is subtle and sophisticated. … First, … competent tutors possess extensive knowledge bases of techniques for defining and introducing tasks and remediating misconceptions. … The number of rules our tutors posses may approach that of expert systems …such as medical diagnosis. … [and] perhaps the most important dimension of expertise we have observed in tutoring involves planning. Not only do tutors appear to formulate and execute microplans, but also their execution of a given plan may be modified and pieces deleted or added, depending on changing events and conditions. … Our view of tutoring may be seen as a midground between a simple diagnostic-remedial model and a curriculum script view. We believe tutoring is both opportunistic (driven by current conditions and events) and also influenced by more enduring decisions such as policies and microplans.*

Implicit in this statement is that tutors were diagnosing (i.e. understanding what the student said) but had a great deal of variety in their responses. Below I will report on this question of diagnosis. McArthur et al.'s notion of micro-plans[13] has similar analogies to what VanLehn et al. (1998a) calls *knowledge construction dialogs* and what the CIRCSIM (Cho, Michael, Rovick & Evens, 1996) project calls a *tutoring protocol* or *directed line of reasoning*.

Procedure:

The protocol I collected was an hour-long protocol of an experienced human tutor working with an individual student in a coached practice session. In this case, the tutor was a female middle-school mathematics teacher (4 years of mathematics teaching experience) who had about 2 full-

---

[13] A note on terminology: I will refer to what McArthur et al. called "micro-plans", as tutorial strategies. I view a curriculum script as something that contains some elements of tutorial strategies but also an element of curriculum sequencing, which I have chosen to not define as tutorial strategy. At the end of this chapter, I will define knowledge construction dialogs to be one type of tutorial strategy.

time person-years of one-on-one tutoring experience (through both University tutoring centers and through extensive private tutoring.) This tutor charged clients 40 dollars an hour. The tutor worked with one of her seventh grade students that she had not previously tutored. The tutor was given a list of problems for the student to solve. The session was recorded on video and then transcribed with the benefit of being able to see the piece of paper the student wrote his answer on. This particular session was about one hour long and is reproduced in its entirety in Appendix C.

Results

The tutoring session was quite interactive, resulting in slightly over 400 lines. The session consisted of the tutor and student working on 17 word problems. Of these 17, 7 of them were done correctly on their first attempt. The tutor did not spend much time on these correctly answered problem (consumed only 24 of the 400+ lines). The remaining the10 problems represent the bulk of the lines in the protocol. Since most of the time the tutor and the student alternated speaking, it makes for an average of about 20 turns (defined as the student and then the tutor speaking) per problem. One problem took an exceptionally long time and stretched from line 17 to line 146. If this long problem is excluded the average number of turns to solve a problem would be slightly over ten turns per problem, which is still quite substantial. This finding is in agreement with the literature (Merrill et al., 1995) that suggests tutors give a great deal of feedback so that the student knows if he[14] is right or wrong. The tutor would give immediate confirmation if an answer was correct, but if it was wrong, she seldom told the student the answer. Instead, the tutor would generally ask a targeted question thereby giving implicit[15] (line 24, 28, 30, 32, 35, and 37) negative feedback.

The Diagnosis Question Revisited

Before I present my generalizations, I need to first address the question from the literature about the amount of diagnosis and remediation that tutors do. A lot depends upon what is meant by diagnosis. There is additional difficulty in that diagnosis is a mental operation. We can judge if it happens only by looking at the observable actions of the tutor. A simple definition of diagnosis is that the tutors can tell a correct answer from an incorrect answer. By this definition, the human tutor certainly did diagnosis, for she kept asking questions until the student solved the problem. When the student got the correct answer, the tutor said so and presented the next problem.

We could expand the definition of "diagnosis" to include the act of figuring out what sort of error the student made. Did the experienced tutor do this? She sometimes made statements (i.e. lines[16] 24, 151 254) that clearly indicate she recognized the student's error. However, more

---

[14] I will use the masculine pronoun to refer to any student. I will use the feminine pronoun to refer to the human tutor, and later, our computer tutor.

[15] Presumably, the absence of positive feedback was an indicator to the student that there was an error. When you scan the transcript, you will see very few explicitly negative feedback messages (i.e. "No"). This is consistent with Lepper et al (1987) who argue that tutors were concerned about the motivational impacts on students of negative feedback.

[16] Line 24 indicates that the tutor knew that the answer was going to involve the fraction being multiplied by d, and so focused on the error that you can't subtract just 2/3 from d. Line 151 shows that the tutor knows that the student comprehended the two steps needing to be done, but the student had not expressed the answer in a correct expression. Therefore, the tutor told the student that. Line 254 is a clear example of the tutor knowing the error was a reversal of the order of arguments, and basically told the student that, by asking which one was larger.

often, her response does not explicitly mention that she performed a diagnosis. Nevertheless, sometimes we can infer that she did diagnose by what she did **not** say. For instance, at line 242, the tutor asks a question focused on one subgoal ("m/s") that the student had gotten wrong (had said "s/m"). The tutor ignored the portion that the student got correct (the "+b" portion). At line 242, the tutor did **not** say, "You reversed the order of arguments for the 's/m'." Presumably, the tutor thought it was more conducive learning to do what she did instead (to be discussed below). This example is consistent with the hypothesis that the tutor often diagnoses to the degree that she determines what errors the student has made. However, that does not necessarily mean that the tutor will report that diagnosis to the student. In fact, the tutor rarely did so.

Did the tutor do remediation? That depends upon what we mean by *remediation*. If remediation means that the tutor says things that are specifically related to the misconception that the student had, then there is not much evidence for that. Only some of the various tutoring utterances are obviously specific to a particular error (i.e. "You did not use parentheses"). If "remediation" means that the tutor gave more problems similar to the one the student got wrong, then our tutor did not "remediate" by this definition; our tutor was given a list of problems to present. But if "remediation" means that the tutor focused on an aspect of the problem that the student did not understand and then asked questions that would be likely to help the student overcome such difficulties, then my answer is that, yes, our tutor engaged in remediation.

My overall impression of this protocol, with regard to diagnosis, is that the tutor needs to be able to understand what the student did so she knows what to focus on. However, the tutor's response is usually generic, rather than specific to the type of error the student made. It is consistent with Merrill et al. (1995) in that tutors need to do some diagnosis. This is also consistent with Lepper et al. (1990) in that the tutor did not report that diagnosis to the student, but instead appeared to use that diagnosis to decide what to focus the conversation on.

Discussion of my Generalizations

So far, I have established that our tutor was engaging in an interactive multi-turn dialog, and that she was doing some amount of diagnosis. I have argued that she did not just report that diagnosis to the student. I have yet to describe what she actually did. I consider this the most important observations of this chapter. I will first describe the two generalizations, and then illustrate them. They are:

1. Dynamic scaffolding
2. Tutorial strategies

By *dynamic scaffolding* I mean a three step process where the tutor would a) diagnosis what the student did, b) give feedback on portions that were correct, and c) most importantly, focus the dialog on the subgoals[17] of the problem that the student got wrong. Dynamic scaffolding is a domain independent strategy. For example, if a process has four steps and the tutor noticed that the student made an error on step 2, the tutor can give positive feedback on step 1 and the ask the student to do step 2 again. Once that is correct the tutor needs to decide what to ask next. Since the tutor remembers that the student could do step 3 she can skip asking for that step. Instead, the tutor can ask the original question that requires all four steps at the same time. Analogously, in this domain, if the student was supposed to say "z*(5+7/(30+g))" but instead said

---

[17] By "subgoal", I mean a portion of the problem. For instance, I view a problem whose answer is 800-40m as being composed of two subgoals. One subgoal is to figure out the 40m portion. The other subgoal is to figure out the 800-X portion. Furthermore, I view each of these goals as being decomposable into subgoals to retrieve the arguments from the problem statement, retrieve the correct operator and finally to order the arguments correctly.

"z*(5+7*(30+g))" the tutor could confirm the "30+g" step, and then ask for the step that has 7*(30+g) as an answer. Once the student got that correct, the tutor could ask for the whole thing again. This is what I mean by dynamic scaffolding.

However, maybe just asking the student to do a step over again is not beneficial? This gets us to what I call "tutorial strategies" which are domain-dependent. These tutorial strategies are multi-step plans and are similar to Mc. Arthur et al.'s "microplans". Like McArthur et al. I could see that our tutor had a large number of diverse actions she could possibly take. I will now look at an example of one of these tutorial strategies.

The "Concrete Articulation" Strategy

A common strategy, which I call the *concrete articulation* strategy, was used in 44% of applicable[18] problems (see lines 32, 242, 258 and 377). In this strategy, the tutor tried to scaffold the problem solving by asking the student to compute a concrete example as shown below.

240. STUD [reads problem] Cathy took a "m" mile bike ride. She rode at a speed of "s" miles per hour. She stopped for a "b" hour break. Write an expression for how long the trip took?
241. STUD uhm [writes "s/m+b" but should be "m/s+b"]
242. TUTOR How do you calculate the amount of time it takes you? If your, if your, if your riding at, let's make it simple. If you are riding at 20 miles per hour, OK and you go 100 miles, how many hours did that take you.
243. STUD Um 5
244. TUTOR 5 and how did you get that 5? How did you use the numbers 100 and
245. STUD 100 miles divided by miles per hour
246. TUTOR So you took the miles and divided it by the [garbled, but possibly "speed"]
247. STUD Miles divided by s plus b equals time [writes m+b]
248. TUTOR Right.

This dialog contains several interesting items. In response to the student's first attempt, the tutor seemed to recognize that the student understood that the time the trip took was equal to the amount of the "b" hour break added to the time actually spent on the bikes. Therefore, the tutor did not talk about the "b" hour break, but instead, focused on the "m/s" component. This is what I previously termed "dynamic scaffolding". This dynamic scaffolding included the step of doing the diagnosis to figure out which goals to focus on. The second step of the dynamic scaffolding was to offer positive feedback, where possible, but since the amount that was correct was only the "+b" portion, the tutor gave no positive feedback (I speculate that the "+b" was just too small a portion to give positive feedback). The third portion of the dynamic scaffolding was to focus the dialog on the subgoals that were not correctly addressed, which in this case was the "m/s" subgoal. The tutor chose for this purpose to ask the student to compute a concrete instance (line 242). I call this first step the *compute step*. After the student correctly answered with "5"(line 243) the tutor asked the student to explain where the "5" came from. The student responded that it was the "100 miles divided by the miles per hour." This is what I call the *concrete articulation step* because the student is being asked to articulate how he arrived at his answer. Once that was correctly achieved, the final step is the *generalization step*, which prompts the student to write out the answer using only the variables and numbers given in the problem.

---

[18] There were 10 problems the student got wrong. One of those was not applicable for concrete articulation since there was no variable. Of the reaming 9 problems, concrete articulation was used on 4 of those problems (often immersed with other tutorial strategies as well).

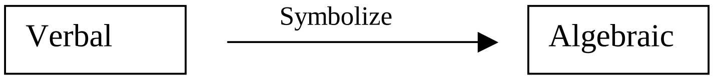
To recap, this example illustrates the domain-general tutorial plan I call *dynamic scaffolding* by where the tutor used its diagnosis to focus the conversation on the portions of the problem that the student had difficulty with. We saw that our tutor was able to diagnosis which subgoal to attack. If the student had gotten a different goal wrong, the tutor could have applied the same concrete articulation strategy. For instance, if at line 241, the student has said "m/s-b" instead of "s/m+b" it is plausible that the tutor might have said something like "How would you calculate the total trip time, if the time you were on the bikes was 3 hours and the time of the break was 1 hour?"

The domain-specific tutorial strategy that she chose to focus the dialog was the *concrete articulation strategy*. The strategy has the three steps of asking the student first to compute an instance (see lines 32, 242, 258 and 377), then articulate how that was arrived at (line 51, 57, 65), and finally, to generalize using the variables from the problem. Another example is shown below.

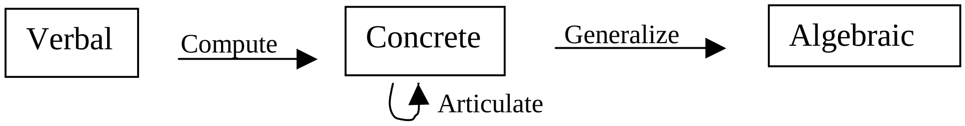
35. TUTOR Um. So if instead of "d" dollars it cost ten dollars, what would the answer be? So if it cost 10 dollars what would the answer be? Not ten dollars. Ten dollars is bad; what if it cost 9 dollars? What would the answer be?
36. STUD Three. No- it would be six dollars.
37. TUTOR It would cost six dollars. Is your expression gonna get that? Really, what are they actually asking for?

The tutors initially used 10 but then, presumably, realizing that would make the answer result in a fraction, switched to 9. The student first answered 3, (which is the amount of the discount), but then switched his response to the correct answer of six, (which is the total cost of a jacket). In this case the tutor did not immediately ask for the articulation of where the six came from, and instead asked if the student's symbolization of "(d-33%)*4=money spent" (line 34) would yield the computed value ("Is your expression gonna get that?"). This is presumably because the question asked for the price of one jacket, not of all four. The student realized that the question is asking, "How much each jacket costs" (line 3), and so got rid of the "4" at line 42. The tutor asked the articulation question at line 43 ("How would you get 6 using your expression?") The student wrote "9-33%=6" and said that the "33% is three" (line 44). So the tutor asked "how did you get that three? (line 45)". This is an example of an *embedded sub-dialog* within a dialog because one the student answers this question, the tutor would have likely ask again about where the 6 came (i.e. "Good. Now, lets return to the question I was asking you before, where did you get the 6. Use what you just told me about the where you got that three as part of your answer.") However, it turns out that student had great difficulty figuring out how to articulate that the total amount of the discount could be articulated as "9*1/3". Eventually the tutor pointed out that "9-1/3" is "8 and two thirds" (line 90). This was an interesting feedback type. Since the student already knew that his expression should yield the answer "6" this told him more than the fact that his answer was wrong. Yet, it did not tell him exactly what was wrong with it. This feedback seemed useful since the student realized that he needed to express the fact that "1/3" is of the whole number (line 93). Later, at line 96, the student realized that multiplication was needed. This problem solving continued for a long time, thereafter. In fact, the student ended up moving towards articulating it as "d*(2/3)=1/3d+d*1/3" (line 127). It did not surprise us that this was such a hard problem as indicated by it being the longest to tutor. The difficulty factors assessments (see Chapter 1) had this exact problem on it, so I knew that students found this to be a hard problem[19]. Many students wrongly thought the answer was "d-1/3". The language of the problem used the phrases "1/3 of the usual prices." I know from personal teaching experience that my students had a hard time figuring out that there is a multiplication operator needed on this sort of problem.

In preparation for the coming sections, I view what we have seen in this example through the perspective of translating between different representations. I think of the task of symbolization as a task of translating from a verbal representation (the "word-problem") to an algebraic representation.

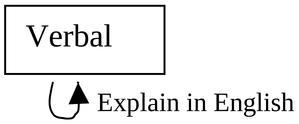


I view the concrete articulation scaffolding as inserting an intermediate representation that I will label as the "concrete" representation. It might be easier for students to think through these problems at the concrete level that does not have any variables. Indeed my data from the previous chapter suggests that the compute step is easy for student, so it make sense to first ask them to compute and then follow that up with the articulation step. This is also supported by Gluck (1999).



The "Explain in English First" and the "Translate Mathematics into English" Questions

Lets look at another example that shows two different sorts of questions the tutor would ask. In the long example given below the tutor asked the student at line 196 " How do you think you calculate the average speed?" Presumably the tutor would have liked the student to say something like "Average speed is equal to the total distance traveled divided by the total time elapsed." This sort of question, which I call a "Explain in English First" question, is one in which the tutor asked the student to work with his verbal representation of the problem.



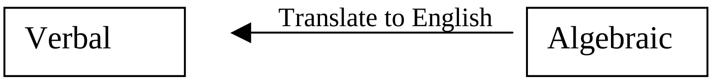
At line 197, the student responded with an explanation of a procedure, "It would be 'h' hours divided by 550 miles an hour." This answer had several errors. First, the units on 550 are "miles" not "miles per hour." Secondly, the order of the arguments was reversed. Thirdly, it was not a general explanation of how to calculate average speed. The tutor dealt with the order of arguments error quickly by asking, "So which way do you divide?" The student fixes the error

---

[19] This cover-story was the hardest cover-story, presumably because "one third off" is a very difficult wording for students to symbolize.

and the tutor seemed to ask the student to try the whole problem again. The tutor seemed to suggest that maybe he is missing something, but was not specific about what was missing (i.e. the minus 2 hours portion). That failed and the tutor came back to the question of how you calculate average speed. Before continuing, I want to point out an observation that I will use in the next chapter. This example shows that when multiple errors occur simultaneously, the tutor deals with each error separately and does not try to deal with them all in a single turn.

The tutor tried to prompt for generality, in the form shown at line 202 when she asked the student a slightly different question but with no numbers, thereby suggesting that the student describe in English how to answer the question. The student ignored this attempt so the tutor asked the student to translate the components into English (line 204, 210). Line 204 was "That's how you calculate average speed but what exactly is it? 550 represents what?" I refer to these types (e.g. lines 24, 26, 204, 210, 212, 219, 221, 223, 225, 264) of questions as "Translate from Mathematics to English" questions.



Eventually the student was coached to drop the specific quantities and say something more general at line 216. Even though the student's answer was not fully general, (using units (e.g. "hours") instead of the more descriptive and abstract phrases (e.g. "Total time")) the tutor restated it in its entirety at line 218.

		190. STUD [reads problem]Bob left at 3 P.M. and drove 550 miles from Boston to Pittsburgh to visit his grandmother. Normally this trip takes him "h" hours, but on Tuesday there was little traffic and he saved 2 hours. What was his average driving speed?	
		191. STUD ::::Well ::: Ah :::: so he saved two hours :::ahum::::::::	Student spends a long time and seems to indicate that he knows he needs to incorporate the 2 hours saved but does not know how.
Explain How to Calculate Average Speed		192. TUTOR Do you know how to calculate average driving speed?	Tutor begins working on explaining how to calculate average speed.
		193. STUD I think, but I forget	
		194. TUTOR Well average speed, as your mom drove you here did she drive the same speed the whole time.	Note: Interesting use of personalization to adapt to student
		195. STUD No	
		196. TUTOR But she did have an average speed. How do you think you calculate the average speed?	
		197. STUD It would be h hours divided by 550 miles an hour.	Student says in English a procedure that is not fully general, has arguments ordering wrong and has wrong units for 550..
	Sub-dialog focused on argument order	198. TUTOR So which way is it? It's miles PER hour. So which way do you divide?	The tutor responds to the argument ordering error with a domain-specific response for that error "Which was is it?"
		199. STUD It would be 550 divided by h [write 550/h=mph"]	Student's answer has a wrong element (should be "h-2" in place of "h")
		200.TUTOR OK so now, that's how you calculate miles per hour. So now how about for this problem? Read the problem again. Because you got the right idea. You know how to calculate average speed. But what exactly do you have to do for this trip	Tutor prompts towards incorporating the "-2" without mentioning it specifically.

		201. STUD Um. Well he saved two hours, but I don't know how that is important	
		202. TUTOR Well how do you calculate... Not for Bob but for your mom, how did you calculate what her average speed was driving to CMU this morning?	Presumably, the tutor wants to ask "Is 'h' the **total** hours?" But first has to establish that the quantity "**total** hours" is relevant so she asks the students to state something like "Average speed is equal to total distance divided by **total** time."
		203. STUD Ahm, I guess you would I would have done it 550 divided by h	Student does not give a general statement of average speed
	550 represents what?"	204. TUTOR Yeah [even though the 550 is not for his mom?] That's how you calculate average speed but what exactly is it? 550 represents what?	Asks student to translate from the algebraic representation to the verbal representation
		205. STUD Miles per hour	
		206. TUTOR No.	
		207. STUD Oh 550 miles	
		208. TUTOR Right	
		209. STUD Divided by h	
	H represents what?	210. TUTOR Which represents?	Asks student to translate from the algebraic representation to the verbal representation
		211. STUD Miles per hour	
		212. TUTOR No what does h represent?	
		213. STUD Hours	
		214. TUTOR Hours! So what are you getting? What are you dividing by what?	
		215. STUD Oh miles divided by hours.	Students gets closer to a generalization
		216. TUTOR Right TOTAL miles divided by	Tutor asks the student to fill in the blank and emphasize the "total" in miles so the student might see the importance of the "total" in "total time".
		217. STUD Total hours	

			218. TUTOR So let’s calculate it for this guy, That’s exactly the concept, TOTAL (emphasized when spoken] miles divided by TOTAL hours [writes "550/h"]	Tutor restates the English explanation for the generalization. Now that has been accomplished, the tutor moves to trying to help the student debug his answer.
Ask if answer conforms to			219. TUTOR Is that ["550/h"]what it is?	Tutor knows it is wrong.
			220. STUD Yeah	
	Ask about each partt	First	221. TUTOR Is 550 the total miles?	This is a neat combination of moves by the tutor to focus on the portion that is correct.
			222. STUD Yes	
		Second	223. TUTOR Is h his total hours?	
			224. STUD Yes	
			225. TUTOR Is it??	
			226. STUD Oh no h-2	
			227. TUTOR OK- right this again and write it correctly so that order of operations and stuff works	
			228. STUD [Writes "550/(h-2)"]	Solves Problem
Post Problem Reflections			229. TUTOR Exactly, so where did the 2 go in?	Asks about the error the student had made
			230. STUD The two hours he saved on traffic	
			231. TUTOR To calculate the total hours, so good.	
			232. TUTOR How we doing, we got lots of time. All right thinking harder. These are pretty good. Let’s try number nine. Okay	Motivational remarks
			233. STUD Okay	
			234. TUTOR [Laughs and mentions hard work]	

What qualified as an explanation for a quantitative relation, such as "Average Speed?" In the following table I detail how the tutor gradually built up to what she considered an adequate explanation. The student started with less articulate answers, such as describing a procedure. Aleven and Koedinger (2000a) have noticed similar results. They call it "procedural replay" when the student is asked to explain why something is true but the student only tells them what he did to get the answer. Therefore, I conclude that what good tutors do is help the student build up an adequate explanation.

Dimensions and fully qualified phrases	"total distance divided by total time."	(never)
Units and fully qualified phrases	"total miles divided by total hours.	(line 218)
Just units	"miles divided by hours"	(line 205)
Procedure with units	"'h' hours divided by 550 miles per hour"	(line 197)
Procedure for an instance	"550 divided by h"	(line 199)
Algebraic symbolization	"550/h"	(line 199)

This excerpt also reveals some other interesting aspects of tutoring. From line 219-225, the tutor used an interesting (some might call Socratic) technique, which can be viewed as being part of a more global plan. The tutor knew that the student had written a wrong answer. Specifically, the tutor made what I called a "super" error type; the student's answer was largely correct expect for the fact that a sub-expression (i.e. "h-2") was simplified into one of its components (i.e. just "h"). The tutor established that the student knew that "Average speed is equal to total miles divided by total hours." Therefore, the tutor asks if the students and was correct, but the student doesn't notice the contradiction, so the tutor asks about each sub-quantity, and again the student didn't notice the mistake until the tutor said "Is it?" (line 225).

The reader might have imagined that the tutor would first just ask a question that focused on the total hours (i.e. "Is "h" the total hours?") which was eventually what happened at line 223. Instead, the tutor first got the student to explain the top-level goal structure. Maybe this is because the tutor wanted the student to realize that the "total" hours is what is needed and that the tutor wanted the student to realize that the "total" is the key idea to getting a correct solution. I could model this behavior with the following rule. If a student makes an error of type *super*, we first ask the student to explain how to achieve the top-level goal in English, and then follow that up by asking if each part (like 221,223) of the students answer meets the criterion that the student had previously explained.

This suggest a slightly different sort of dynamic scaffolding strategy when the student has made a certain type of domain specific reasoning since the scaffold depends upon this sort of error arising. On the other hand, it might have been better if the tutor had simply asked, "Is 'h' the total hours?" at line 200.

An important point to make is that it is easier to see what the tutor said, then why she said it. A case in point occurs after line 197, discussed above. There were any errors with this answer and there were also many things the tutor could have done (i.e. use the concrete articulation strategy as we saw in line 240-248 on a similar case where the student reversed the order of arguments on a division problem). But instead the tutor opted to ask "So which way is it?"(line 198). A natural question, is how are these two situation different? McArthur et al. pointed out that tutors "are capable of taking a variety of different events and conditions into account when selecting from their diverse array of techniques. … The antecedent conditions of [their] 'if-then' rules are often nontrivial." It could be that because the student made more then one error (as in this case line 197) the tutor thought it was helpful to quickly eliminate some of the errors, and thus chose the simpler pedagogical move. Alternatively, it could have been that the tutor used the concrete articulation strategy on the later problem because it was the second time this sort of error appeared. On the other hand, it could have been that the tutor was purposely choosing to vary the tutorial strategy picked for variety sake. Then again, maybe the decision was random? Because of this ambiguity, my model will have an easier time modeling the actions of the tutor, but less accuracy in modeling the conditions that determine which actions to take.

Another interesting example of the use of the "Explain in English" strategy occurs on a later problem that also involves calculating speed. The student has written "(10+A)*600" (line 305) when it should have been "600/(10+A)". The tutor again asked the student to define average speed (line 306, 308) and then at 310 give the hint "It has to do with total. The word total has to there." The tutor then referred the student back to the previous problem, which appeared to be quite successful. Yet another example of the "Explain in English" strategy appeared on line 384 when the tutor asked the student "How is rate defined?"

The "Introduce a Variable" Strategy

Another common occurrence (4 times on 44% of the applicable problems) was for the student to introduce a variable to stand for a quantity. Variables were introduced at lines 158, 174, 178, 300,and 335. In all of these cases, the tutor did not prompt this but the tutor did work with the student's introduced variable.

The first time a variable was introduced was on a problem where the student was supposed to say "(72-32)/s". The student was told that he was not supposed to do any arithmetic and instead articulate out the mathematics that he would do. Nevertheless, the student writes "72-32=40". The student then made a classic "production" error by continuing to add onto the expression until it stated "72-32=40/s". Of course, it was easy for the tutor to see what was meant, but this was a miss use of the equal sign because 72-32 does not equal 40/s. In elementary school, this sort of notation never caused any trouble, because it was understood that the "equals" symbol simply meant "results in" (Sfard and Linchevski, 1993). However, with algebra the equals sign has a great deal more meaning. The student made this sort of error on line 158 and 160. Interestingly, at line 158 the student introduced the variable "a" to stand for the "40". The tutor said " Right, instead of 'a', let's just use this. Pretend". This is speculation, but I presume the tutor thought:

> "Okay, if he wants to use 'a' for that quantity, that is fine. Lets get him to say that the amount he can spend on each sister is 'a/s' and then afterward I can tutor him on substituting back in the expression of 'a' with '72-32'"

Interestingly, the student appeared to get more adept at the introduction and use of a variable and did so three more time. I now look at those instances.

The next opportunity where the student used an introduced variable was on line 174; the student introduced 'a' to stand for '972/5'. Next, the student did the separate subgoal of b/5. Then the student mistakenly set them equals. After much thought he eventually expressed the third step with 'a+c' where it is reasonable to assume that he was thinking that 'c' would stand for 'b/5.' The student lacked confidence ("I don't understand this one part") presumably with the addition operation.

The next two examples show the student successfully introducing a variable to stand for subexpressions. In both cases the student followed by using them to express the problem's goal. Once, that was done the tutor prompted for the student to do the substitution step. The student seemed to be getting more proficient at doing this.

At line 300, the student introduced the variable "a" to stand for a "300/20" which represented "the time it takes to get to Grandmother's house when going 20 miles an hour". He then went on to use "a" to express the final goal at line 319, at which point the tutor prompted him for the substitution step. Interestingly, this problem also shows another parallel use of *place holding*: using a concrete instance to hold a place. In this same problem, the student used 10 to stand for "the time it takes going 30 miles an hour." The introduced variable and the "10" served the same roles. They both allowed the student to forget about the internal workings of that subgoal and concentrate on how that particular part was used in the rest of the problem. This might be one reason why the concrete articulation strategy, discuss above, is successful. Reducing the working memory load seems like a useful tactic. Another reason it might help to introduce a variable or to compute a concrete instance, is that it encourages coming up with a name for that portion of the problem, and doing so might strengthen the mental picture that allows the student to see the hierarchical nature of a problem. This might be an important cognitive step in building up the quantitative structure in ones head.

At line 303, the student continued to use "A" while struggling to figure out which operator to use to calculate average speed. Using the introduce variable made it easier for the tutor to ignore the low level details and conduct this "sub-dialog"(lines 303-318) on what was the correct mathematics operation to use (I use the term *sub-dialog* to refer to a point where the tutor seemed to have a productive digression. Usually it involved the tutor focusing on a subgoal that needed to be solved. In this case, it was focusing on the mathematics operator needed. Once that was cleared up, the tutor could get back to the bigger picture. I think that a hierarchical view of this dialog is a good first approximation to what humans do. This will get more attention in the coming chapters.)

So far, we have seen three problems with introduced variables. In the first example, we saw a variable was used to represent a concrete instance that had already been computed. The next two examples had introduced variable that stood for a one-operator quantity that had not been computed. Now, we look at a fourth instance and this time the variable was used to stand for an expression, which was, itself, hierarchical (i.e. had two operators). This might indicate the student was getting more comfortable with this technique of problem simplification. Consider the strategy that is used between lines 327 to 351. The student initially expressed confusion about a particular quantity (line 327, "What’s her commission?"). The tutor let him know that this is a subgoal ("That’s a good question?"). (Notice how the tutor was encouraging the student to work on this problem in a bottom-up fashion, a point I will come back to.) So the student proceeded to calculate the commission and then introduced a new variable, "c", to stand for the calculated commission (line 333 "c=(200-x)/4)"). The tutor confirms that that was correct (line 334). Then the student used that variable to express the goal of the problem (line 336 "200+c*4") but made a small error by saying "4" instead of "h". The tutor prompts with "This month?" which causes a small sub-dialog focused on the subgoal of determining what represents the number of cars sold this month. After that short sub-dialog was the prompt for doing the substitution (line 345 and 347).

I must digress for a moment to reinforce a running theme. The tutor’s utterance of "This month?" is an excellent example of what I have been talking about. The tutor diagnoses the problem (i.e. used ""4" instead of "h") but did not tell him that explicitly (i.e. did not tell him "Use ’h’ instead of ’4’"). Instead, the tutor gave a short prompt on that particular subgoal. This let the student focus on one particular aspect of the problem, while not telling the student what to do to fix the problem. This tutor engaged in these important steps: 1) diagnosis to determine what goal was not achieved, and 2) scaffold the problem by 3) asking a question related to that goal by asking a targeted question directing attention towards a subgoal.

A general observation to make of these examples is that knowing how to do the substitution step is not always easy. The student certainly seems to have the hang of it by the end, but this is a hard skill to get good at. This is the skill my transfer study taught students. I encountered an interesting example a student who introduced variables correctly but then did not know how to put them together. I collected this example while I was tutoring a student over a teletype in real time. It is reproduced verbatim in the Appendix E. The student was supposed to answer 5g+7(30-g). Instead, the student said

"30 - b = n   n * 7 = m   b * 5 = o   m + o = y"

The point is that the student initially knew how to express a complicated four-operator problem as four decomposed problems, which amazed me. I thought it would be easy to proceeded from there. However, I was wrong. Only with much prompting, lasting over 20 minutes was I able to get the student to correctly compose them all together. This demonstrates that there is a lot of difficulty hidden inside of what I have called "production" skills.

The Reflective Follow-up Questions and Generalization

Our tutor also encouraged the student to sometimes generalize and reflect at the end of a problem. In line 246 when the tutor restates the student correct answer but does so in general terms is an example of this. We have already seen an example above where the tutor got the student to learn the general procedure for calculating elapsed time, by coaching the student to generalize from a single calculation of a particular elapsed time. Another example of generalizations was on line 401 when the tutor confirmed an answer and stated it in general terms. Another example of post-problem reflection occurred on line 229 when the student gave a correct answer but the tutor encouraged him to reflect on it by challenging his answer. A very similar situation occurred on lines 252-268 when the tutor got the student to write the correct answer but spent a long time afterwards making sure the student understood why.

Where do these Strategies come from?

An important question is "Where do these scripts come from?" They could come from general tutorial strategies such as those outlined by VanMarcke and Vedelaar (1995) (see also VanMarcke, 1998). Alternatively, they could come from an analysis of the domain. I hypothesize that a good tutor is not just one who is an expert in the domain and also knows some general tutorial strategies. I think that a good tutor probably needs extensive experience tutoring that particular domain to come up with what Shulman (1986) calls "pedagogical content knowledge." This sort of knowledge includes good questions to ask, as well as an understanding of common errors students make.

Who has the Initiative?

Is it the student or the tutor that takes most of the initiative? This is of particular interest because I desire to build a system capable of having a dialog with students; I must understand what student behavior are likely. If a primary characteristic of tutoring in my domain is that students need to ask many questions of the tutor, then I need to build a system that can respond. However, if my observations of tutoring in this domain reveal that the tutor has the initiative most of the time, then I need to build a system that is able to simulate that behavior.

Most work in tutoring has generally found that the tutor is taking the initiative most of the time. Research shows that tutors typically set most of the agenda, introduce 95% of the new topics and ask 80% of the questions (Graesser and Person, 1999). Our tutoring protocol was also mostly tutor-initiated. Most things our tutor said were questions, which is consistent with Lepper et al.'s (1990) finding that 80-90% of tutors' remarks were questions. For instance, lines 20-35 show our tutor just asking a series of questions of the student. Not only did our tutor ask a great deal of questions, but the student did not ask many questions and the few he asked were shallow and arguably not what I think of as initiatives. For instance, if we look at just the first 100 lines there were only three instances of the student asking the tutor a question. The first on line 18 was asking a clarification about the problem statement. The second and third instances (line 85 "This way?" and line 98 "I think. Would it be that way?") were both examples where the student was simply asking the tutor for feedback and expressing doubt.

Although it might be nice if students were more active in their own learning, my goal was to build a system that could respond to students the way they usually behave, which is, unfortunately, quite passive. I felt that I would get the biggest returns from my investment by largely ignoring the problem of student initiative. I feel that a reasonable first step was to build a tutor-initiated system.

Motivation

Next, I consider if Lepper et al.'s (1997) claim that tutors pay attention to motivational goals was supported by the evidence. An inspection of the transcript reveals that very few comments seem to relate to motivation. The few comments seem to be at the beginning or end of a problem like when our tutor said (line 11) that she was looking for a harder problem since the student got the first three problems correct. Also at line 403 our tutor said "I can see you are getting tired. This is our last problem." Nevertheless, in general, our tutor did not do much in the way of explicit motivational remarks. Even when the student dropped his pencil (in exasperation?) and holds his head (line 116), our tutor makes no "motivational" remarks. Nevertheless, our tutor indicated after the session that motivational concerns were important, so possibly her implicit theory was that the student's awareness of his own learning was what will lead to motivational gains. Lepper et al. also reported that when instructional goals come in conflict with goals to increase student motivation, the goals related to motivation take precedence. This protocol showed little evidence of explicit concern for the motivation state of the student. However, the fact that the tutor so rarely gave explicitly negative feedback and instead relied on implicit negative feedback might be because the tutor was concerned about the student's motivation.

Categorization of Tutorial Utterances: The List of Operators from the Thesis Proposal

I chose to model the tutorial utterances at a high level. Appendix D shows the categorization I had of the tutor's utterance. I refer to these categories as the tutorial "operators" as they represent the major choices that the tutor had to choose between when deciding how to assist a student. In Appendix D I give each operator a name and number (in no particular order) of the form Op*x*, a brief description, and point out some real examples and cite them by line number as they appear in Appendix C. In Appendix D, for illustrative purposes, I offer an invented canonical example using the same sample problems. These categories for tutorial utterances were viewed as design goals for the system I built. In Chapter 4 I will come back to these design goals and see which one were met. In the next section, I will discuss my generalizations

Chapter Conclusion

I hypothesize two types of tutorial interaction will be particularly effective and correlated with learning. These two hypotheses all fall under the single hypothesis that good tutors “ask more and tell less” (the phase “ask more and tell less” is from the CIRCLE Center's grant proposal by VanLehn et al., 1998b). Specifically, I need dynamic scaffolding and tutorial strategies.

By "dynamic scaffolding", I refer to three steps. The first is to diagnosis that last students answer. Secondly, try to give useful feedback. Thirdly, for the portions of the answer that are wrong, focus the dialog on those portions using what I call tutorial strategies. Just using dynamic scaffolding would mean that you would simply ask for any subgoal that the student got wrong (For instance, our tutor could have asked "Write an expression for the time actually on bikes?" at line 242, rather then use the concrete articulation strategy. This would meet my criterion of dynamically scaffolding the portions the student got wrong.) However, there is more to tutoring then just dynamic scaffolding; good tutoring includes pedagogical content knowledge including multiple-step tutorial strategies.

One strategy is the "Concrete Articulation" strategy that is composed of three steps. There is also the "Explain in English" strategy that asks the student to first explain a quantitative relation and then follow that up by writing the algebra expression for the that goal. Another strategy is the "Translate the Mathematics to English" strategy. A final strategy "Introduce Variable" was that a variable could be introduced to scaffold the writing of an expression into two steps: first write an expression using the variable, and then to substitution in what the variable was standing for. I refer to these as my four general strategies.

Four General Tutorial Strategies or Knowledge Construction Dialogs
Concrete Articulation
Explain in English
Translate Mathematics into English
Introduce a Variable

One thing the four strategies have in common is that they are general, in the sense that they can be used to respond to any type of error. On the other hand, a KRD is specific to a particular class of errors (e.g., a parenthesis error).

Error Type	What the Tutor Said	Line
"Super" Error like 550/h instead of 550/(h-2)	"Explain in English" "how do you calculate average speed?"	200
Wrong Operator	"Explain in English" "What is the definition of average speed?"	306
Super Error like d-33% instead of d-1/3*d	Concrete Articulation	35
Arg. Order for Division	Concrete Articulation	242

Therefore, I refer those four strategies as "general" strategies, for the tutor can use any of them in response to any type of error. To emphasize that these strategies can help in a variety of different circumstances, I will follow VanLehn et al. (1998b) and call my specific models of these strategies *Knowledge Construction Dialogs (KCDs).* The next chapter will show specifically how I modeled these strategies as KCDs.

As opposed to general strategies, sometimes our tutor responded in a way that is very specifically connected to the type of error committed. The following table shows 4 different examples, each of which you can almost tell what type error was made just from what the tutor said. This was not you case in your general strategies. These responses are "tactical" in that the tutor only used them in response to certain specific situations. To distinguish these different types of strategies from the 4 KCDs, I will call these responses *Knowledge Remediation Dialogs* (KCDs). This name highlights that the tutor is responding with a specific remediation strategy to a specific error type.

Error Type	**What the Tutor Said**	**Line**
Arg. Order for Division	"So which way is it?"	198
Arg. Order for subtraction	"So which is larger?"	252,374
Parentheses error	"What is being divided by 's'? Because order of operation says you divide before you subtract."	161
Parenthesis error	"You are just multiplying "A" times 600 but I suspect you want to do the whole thing?"	306

I suggest one heuristic to use in a model of tutoring: the tutor should select KRDs in favor of KCD, because the first has a more specific fit to the student's problem. In the next chapter, I will show how I have turned these observations of our tutor into a system that models some of the aspects I have highlighted in this chapter. It should be said that our model is a simplification of the experienced tutor's behavior. The two key components I use are dynamic scaffolding and tutorial strategies (including both KCDs and KRDs). I will also use the heuristic to favor KRDs over KCDs. Given that I do not know which KCDs are most appropriate (or effective) for a given situation, our model will pick arbitrarily from the 4 KCDs I have modeled.

Chapter 3: The Architecture of Ms. Lindquist

Chapter Overview

This chapter will examine the general architecture we have designed that is an enhancement to model-tracing tutors. We will illustrate this architecture with specific reference to "Ms. Lindquist", a system we have built inside of this architectural framework. This chapter will focus on the architecture (i.e., domain independent aspect), while the next chapter will discuss more details about Ms. Lindquist and her behavior (i.e., specific to the symbolization domain). This chapter is divided into two sections: the student model and tutorial model.

The Need for Change

McArthur et al. (1990) criticized Anderson's et al. (1985) model-tracing ITS and model-tracing in general "because each incorrect rule is paired with a particular tutorial action (typically a stored message), every student who takes a given step gets the same message, regardless of how many times the same error has been made or how many other errors have been made. … Anderson's tutor is tactical, driven by local student errors (p. 200)." They go on to argue for the need for a more strategic tutor. Ohlsson (1986) agreed and said that in model-tracing ITS "there is no adaptation to the current cognitive state of the learner other than the classification of his last step as an instance of a particular type of error." The system described in this chapter address these criticisms and we will come back to them at the end of this chapter.

Architecture Overview

Ms. Lindquist is a coached practice system that has a curriculum broken into sections of similar difficulty problems as shown below.

Section Description	**Example Problem**
1) One-Operator Problems.	The carnival committee is dividing up the whole field for carnival booths. If there are "x" booths and the area of the field is 400 square yards, write an expression for the area each booth gets.
2) One-Operator involving distance, rate and time.	Rose Mary needed to drive to Houston, Texas that was "x" miles away. If she planned on driving 55 miles per hour, how long is the ride going to take her?
3) Two-Operator linear forms.	Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for "m" minutes and stops to rest. How far is she from the dock now?
4) Two-Operator with some involving division and parenthesis.	You go on a bike ride. You ride at a speed of "s" miles per hour. You bike "m" miles. You take a "h" hour break. Write an expression for how long the trip took.
5) Three and Four Operator Problems	Debbie has two jobs over the summer. At one job, she bags groceries at Giant Eagle and gets paid 5 dollars an hour. At the other job she delivers newspapers and gets paid 7 dollars an hour. She works a total of 30 Hours a week. She works "g" hours bagging groceries. Write an expression for the total amount she earns a week.

A student is presented with a problem that they are asked to symbolize as an algebra expression for a given quantity expressed as an English phrase (i.e., "Write an expression for the distance Ann rowed."). If a student gets the problem correct, the system checks to see if the student has

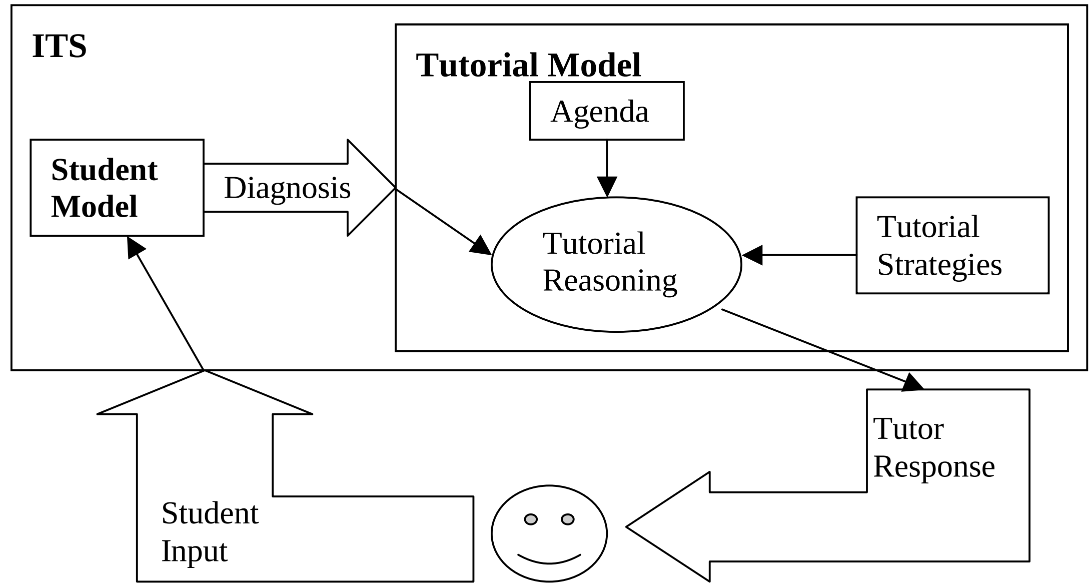


Figure 1: The student model and the tutorial model are separate.

reached the mastery criterion for that section. We have used a simple mastery criterion of getting some fixed number of problems correct (on their first attempt) in a row. Depending upon the section, the number of problems the student needs to get correct is 3, 4 or 5. If the student does not get the problem correct Ms. Lindquist engages them in a dialog that is designed to try to help them learn the skills needed to be successful. Once that dialog is complete, the students are given a new problem. If they have done every problem in a section but have not reached mastery, then this is noted by the system and the student is pushed onto the next section. Next, we will explore the student model.

Separation of Student Model from Tutorial Model

In the previous chapter, I argued that the first thing the tutor needs to be able to do is to diagnose a student's utterance. This is handled by the *student model*. Then the *tutorial model* is used to decide what you do next (ask a new question, give a hint, confirm an answer, etc.). Once the tutorial model has responded, the student's response is solicited. Figure 1 shows this schematically. We will discuss each element below starting with the student model.

The Architecture of the Student Model

Anderson & Pelletier (1991) defined the *model-tracing* approach to building intelligent tutoring systems. The new architecture presented in this dissertation does not modify the way a *student model*[20] (i.e. the cognitive model) is implemented. We use the same production system as

---

[20] I use the term student model to refer to what is sometimes called either a cognitive, domain or expert model. Some others use the term student model to refer to just the set of probabilities representing the systems belief that a student knows a particular skill, (this is sometimes called the overlay model). I use them term student model to differentiate it from the tutorial model. The tutorial model is concerned with modeling the behavior of the human tutor (i.e., what to say to the student) while the student model is concerned with modeling student thinking, which includes both the cognitive model as well as the overlay model. However, it should be noted that this thesis is not

Anderson & Pelletier, called Tertl. It is a simplification of the ACT (Anderson, 1993) theory of cognition. The Tertl production system is written in lisp. A production system is a group of if-then rules operating on a set of what are called *working memory elements (wmes)*. We use these rules to model the cognitive steps a student could use to solve a problem. Our student model has 68 production rules[21]. We will now move onto how the working memory elements are initially configured for each problem.

Simple Problem Encoding Representing Initial Working Memory

In Chapter 1, we showed that the main difficulty students face is not comprehension of the word problem. Therefore, we are not modeling the comprehension process. Instead, we assume that a quantitative network is in place as a result of comprehension. We will now describe how that quantitative network is represented. To tutor on a new problem, the system author has to encode the relevant information. Consider the following problem.

> Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for "m" minutes and stops to rest. How far is she from the dock now?

Note that our system does not parse this text. Instead, the problem author must encode the essential information. The author must specify an English phrase for each quantity (e.g., "the distance rowed so far"), the units of that quantity (e.g., "yards")", the dimension of that quantity (e.g., "distance"), and the algebraic expression for that quantity (e.g., "40*m"). Additionally, for each quantity that is a variable, the author must specify a good concrete instance to use in the place of that variable. This number (e.g., "3" for "m") is used for the *concrete articulation* strategy (see previous chapter). For the above problem, here is the information that is encoded in a problem file.

Verbal Phrase	**Dimension**	**Units**	**Symbolization**	**Concrete Instance**
The distance Anne has left to row	Distance	Yards	800-40m	
The distance rowed so far	Distance	Yards	40*m	
Her speed rowing	Speed	Yards per minute	40	
The number of minutes rowing	Time	Minutes	m	3
The distance she started from the dock	Distance	Yards	800	

Table 1: An example of the information needed to encode a problem.

Each row in the table represents a quantity in the problem. The problem author needs to encode a verbal phrase, dimension, units and symbolization. We see that the quantity represented by "The number of minutes rowing" has two symbolizations (i.e., "m" and "40"). The first is the variable presented in the problem statement. The second is a good number to use as a concrete instance as part of the concrete articulation strategy. A good concrete instance is usually a small number, but it must also be reasonable. For instance, "3" is not a reasonable number for the

---

concerned with the overlay model. In fact, we have not even turned on this feature in the system we have built. Nevertheless, this architecture allows for the tutorial model to use the information from an overlay model (e.g., decide which tutorial strategy to use based upon the system's belief they know something.)

[21] A production rule can make calls our to lisp functions, so the complexity of a system is not well judged by a count of the number of rules.

speed of a jet in miles per hour since jets fly much faster. A good instance is a number that makes the rest of the mathematics easy, which includes avoiding fractions in the answer. Finally, a good instance is one that will not be result in confusion of any of the other quantities. We try to have each quantity represented by a unique concrete value. For instance, we would not use "20" for "m" as both "the distance rowed so far" and "the distance Ann has left to row" would have the same value of 800.

The units and dimension information listed in the above table are not used in the student model. This information is used only in the tutorial model so that the tutor speaks coherently (e.g., refers for "7 miles per hour" instead of simply "7").

All of this information is encoded into working memory elements as shown by the following graphic in what I refer to as a *quantitative network* (see Tabachneck, Koedinger and Nathan, 1995, and Hall, Kibler, Wenger and Truzaw, 1989).

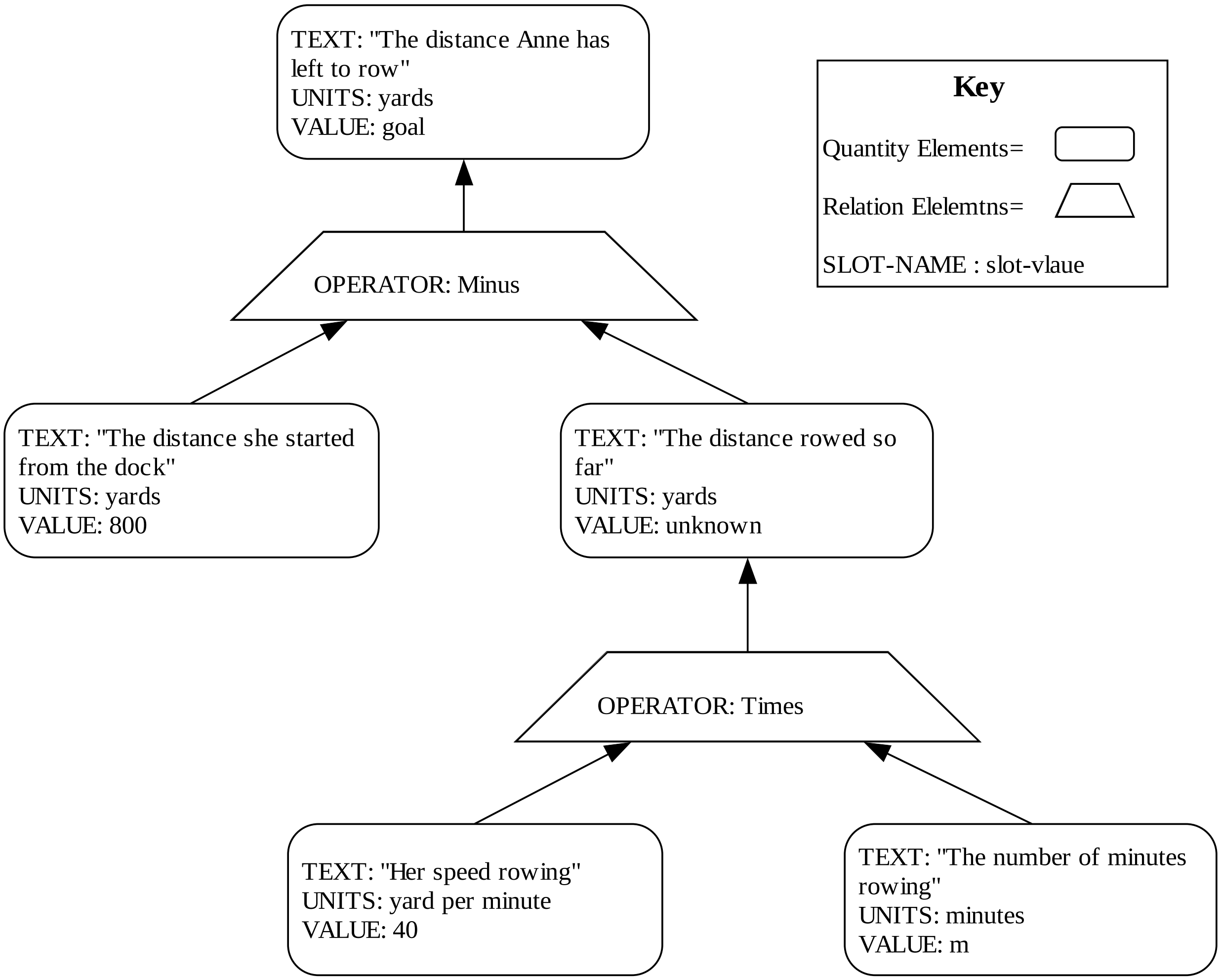


Figure 2: A graphical depiction of working memory

This quantitative network shows an example involving 5 quantity working memory elements and 2 relation working memory elements. Each wme has a type that defines a list slot names (e.g., "TEXT", "UNITS", and "VALUE"). In Figure 2, we see two different types of working memory elements (e.g., quantities and relations). Ms. Lindquist uses this set of working memory elements to be able to understand a student's attempt at the problem. For instance, we can see in Figure 1, that Ms. Lindquist "knows" that the quantity for the "distance rowed so far" has an unknown value and is involved in two relations that could possibly be used to find the value for that quantity. As it turn out, Ms. Lindquist can figure out that the "distance rowed so far" can be computed by using the operator "times" and the two quantities whose text are "speed of rowing" and the "time rowing." This knowledge is represented in production rules.

Production rules

In the ACT-R Theory of Cognition (Anderson, 1993), a production rule is used to represent a cognitive step. For example, if your problem domain was mutli-step column arithmetic your cognitive model to solve such a problems could include productions that 1) choose a column to work on, 2) process a column, and 3) write out answers depending upon whether a carry is needed or not. Production rules are written in the TERT1 production system as IF-THEN rules. The productions can have variables so that their can be many different ways a production can apply. More information on productions, including examples of production rules, can be found in Anderson (1993).

Ms. Lindquist Production Rules

Ms. Lindquist's has about 70 production rules for representing how students solve symbolization and related problems. These productions can be placed into a few categories:

1) Productions that do a goal-directed search of the working memory. These productions operate on the quantitative network.

2) Productions about symbolizing numbers, literals and subexpressions. Once the model knows a relation it wants to use, it needs to pull out of the associated quantities the actual numbers, variables or subexpressions (i.e., "40*m").

3) Productions that determine the operator to use for a subexpression.

4) Productions that determine the order to express the arguments in.

5) Productions that put parentheses around an expression.

6) Productions that compute an answer, or articulate the mathematics used to compute an answer.

These productions get "chained" together to mimic correct, or incorrect, problem-solving steps.

An Example of a Production Rule

We will take as an example how our model knows when to add parentheses around an expression[22]. This following production rule is used to add parentheses around an expression if it is the second argument (e.g., "7*(30-g)"). There is a separate rule for adding parentheses if the subexpression is one the right (e.g., "(30-g)*7"). Here is the rule with the variables Q, Q2, Op1, Op2, A1,A2,A3 and A4 as variables.

---

[22] For efficiency and software engineering purposes our model actually accomplishes the same thing but with a series of productions. However, for purposes of elucidating what a production rule does, we have chosen to abstract out the essentials.

```
If goal is to express Q in algebra
        And Q is found by Q1 Op1 Q2
        And Q1 is expressed as A1
        And Q2 is expressed as A2
        And A2 is found my A3 Op2 A4
        And Op1 has higher precedence then Op2
Then
        Set a goal to write A1
        And write Op1
        And write "("
        And set goal to write A2
        And write ")"
```

So if we were solving the following problem,

> Debbie has two jobs over the summer. At one job, she bags groceries at Giant Eagle and is paid 5 dollars an hour. At the other job, she delivers newspapers and is paid 7 dollars an hour. She works a total of 30 Hours a week. She works "g" hours bagging groceries. Write an expression for the total amount she earns a week.

we could instantiate this rule with the following variables.

Q ="the pay from the newspaper"
Q1="the pay rate for delivering papers"
A1=7
Op1="*"
Q2="the hours delivering newspapers"
A2=30-g
A3="30"
Op2= "-"
A4="g"

This production would generate "7*(30-g)".

Model-tracing

The production system has rules that operate on wmes like these and can be used to understand a student’s input using an algorithm called "Model-tracing" (Corbett, Koedinger and Pelletier, 1995). Model-tracing is a plan recognition technique. The model-tracing algorithm is given three inputs;

1. The state of working memory: represented by a group of working memory elements.
2. A set of productions. Each production represents a cognitive step, which may, or may not, have observable actions.
3. The student’s input that we wish to "trace".

The first two inputs are collectively referred to as the *student model*. The model-tracing algorithm has two outputs.

- A Boolean value indicating if the student’s input was "traced".
    - If the input was traced, the algorithms output is a set of interpretations. Each interpretation is a list of productions that are chained together. Each production represents a different set of steps that could have resulted in the student’s action.

If the model-tracing algorithm finds such a sequence, we can say that the system "traced" or *understood* the student's input. For instance, supposed we wanted to trace a student symbolization a word-problem that had the answer of "(72-m)/4" it would take 33 productions. To trace a correct answer of "5g+7(30-g)" requires 65 productions. Traces for both of these are given in Appendix F.

If the model-tracing algorithm fails to find a sequence of productions that could have generated the student's response, we say that the student's response was *uninterpretable*. If a response is *understood* that means that, the model-tracing algorithm found a list of productions that could generate the student's response. If the algorithm reports multiple different interpretations to a student's action, one of them is picked arbitrarily, which appears to work fine for this domain as I do not know of any responses by students that generate more then one reasonable interpretation. Other research systems have focused work on this problem. There is one additional caveat, which is that the system will favor interpretations that have no buggy rules. The next section will explain what buggy rules are.

Buggy Rules

We model the common errors that students make with a set of "buggy" productions. For example, it is a common error for a student to forgot parenthesis. We showed above an example of a rule that knows when to put parentheses in. Our model also has a buggy rule that leaves them out. The following is an example.

```
If goal is to express Q in algebra
        And Q is found by Q1 Op1 Q2
        And Q1 is expressed as A1
        And Q2 is expressed as A2
        And A2 is found my A3 Op2 A4
        And Op1 has higher precedence then Op2
Then
        Set a goal to write A1
        And write Op1
        And set goal to write A2
```

This rule could be used to model the student that simply ignores operator precedence and does not use parentheses.

From our data, we compiled a list of student errors and analyzed what were the common errors. In Chapter 1, I represented data showing that these error categories could be used to give an interpretation for over 75% of the errors that students made. We illustrate the error categories with an example error, in the context of a problem that has a correct answer of "5g+7(30-g)".

- Missing parentheses (e.g., "5g+7*30-g")
- Wrong operator (e.g., "5g-7(30-g)")
- Wrong order of arguments (e.g., "5g+7(g-30)")
- Missing a component (e.g., "5g+7g" or "g+7(30-g)" or "5g+30-g")
- Omission: correct for a sub-expression (e.g., "7(30-g)" or "5g")

These "buggy" productions are used to allow us to make sense of a student's input even if she has made several incorrect steps. We do not want a computer system that cannot understand a student if he gives an answer that has parts that are completely correct and parts that are wrong.

We want the system to be able to understand as much as possible of what a student says and be able to give positive feedback even when the overall answer to a question might be incorrect.

There are many things that Ms. Lindquist does not "understand" (or "trace"). Examples include (again in the context of the tutor asking the student to symbolize a problem whose answer is "5g+7(30-g)"):

- Using numbers not in the problem (e.g., "656")
- Confusing quantities (e.g., “7g+5(30-g)”)
- Writing an equation instead of an expression (e.g., "7*(30-g)=210-7g")
- Writing an English sentence ("I don't understand- help me.")

Currently, any answer that is not "traced" is considered *uninterpretable* and the Tutorial Model is informed of that and acts accordingly (see below). Note that some of these errors might be worth getting Ms. Lindquist to be able to understand. For instance, the bug about "Confusing quantities" is a likely candidate. If we decided that this error was common enough for use to trace, we could add a "buggy" production to the student model.

So far, we have been talking about having the student model "trace" certain errors and that can be useful to error localization so that the tutor could focus on what is causing the difficulty and not on the stuff that the student's already knows how to do. For instance, if the student says "5g+7(g-30)" instead of "5g+7(30-g)" we would like a system that understands the parts that are correct (e.g., everything but the "30-g") and give positive feedback on the correct portions and then focuses only on the incorrect portion. This brings us to the question of how the feedback is generated

Traditional Model-Tracing Feedback

There are two ways traditional model-tracing tutors give feedback;

1. **Buggy feedback** based on a diagnoses error
2. **Hints** toward a correct action.

First, we look at buggy feedback. Each rule in the student model that represents a common error is marked as a "buggy" rule and contains a template used to generate a feedback message. Consider the example of the buggy production we showed above that neglected to add parenthesis when needed. A traditional model-tracing tutor could have the following buggy template on the right hand side of the production rule:

Buggy Template: "Around the ’A2’ you should put parentheses."

Remember that A2 is a variable in the production rule that represents an expression. So if the student was suppose to say "5g+7*(30-g)" but instead said "5g+7*30-g", this buggy rule would generate the following buggy feedback "Around the '30-g' you should put parentheses."

The second form of feedback from a model-tracing tutor are hints. Hints are usually only given when requested by the student. When a student asks for a hint on a given question, the tutor runs the production system to generate a correct answer. The result of this is a list (sometimes called a "chain") of productions that represents the cognitive step the student should take to arrive at the next step. This chain is used to generate text for the series of hints the tutor will offer the student. Specifically, each production in the chain can have a text template it uses to generate its portion of a hint message. More specifically, the hint messages of each production in the chain are concatenated to form each hint.

To illustrate this let us suppose our model what just two productions as shown below. We show the hint text that is attached to each production.

```
Production Step1-choose-object
        Hint text (     "What quantity should you start with?"
        ""
      "Select <object>")
Production Step2-choose-operation
        Hint text (     ""
                        "What operation should you do to that quantity?"
                        "and then do the <operation> to it.")
```

These two productions would generate the following hint chain, and they would be displayed one at a time. Notice how the text is concatenated together to form each hint.

A Chain of Hints
" What quantity should you start with?"
"What operation should you do to that quantity?"
"Select 2*(4-x) and then do the distribute operation to it.")

There are several disadvantages to generating feedback, (both hints and buggy feedback) in this way. First and foremost, is that we would rather ask a question than tell the student something whenever possible. We note that two of the hints above are phased as question, and that is typical of other model-tracing tutors. The system asks this question of the student but does not allow the student to answer it. We hypothesize, but will not prove in this thesis, that student learning will be enhanced by having the computer ask new questions, rather than just tell the student information. We therefore propose that whenever the student is having difficulty, the tutor should try to break down the problem and ask the student questions about the goals that the student did not accomplish correctly. .

Additionally, from a software engineering point of view, generating feedback in this way makes it very hard to separate tutoring reasoning from the diagnosis provided by the student model. Moore (1996) has also makes objections to the feedback generated from model tracing tutors.

The Problem of Multiple Errors on the Same Turn

Intelligent tutors have always had to deal with the problem of what to say in response to a student making several errors, all at the same time. For instance, the work of Moore et al. (1993, 1996), was an attempt to make a system more coherent when trying to communicate multiple different ideas. The PROUST and CHIRON systems (Sack & Soloway, 1992) were intelligent tutoring systems that detected multiple errors in Pascal programs, but Sack & Soloway recognized that the system could be confusing when spitting out what amounted to separate error messages.

Traditional model-tracing tutors have not been able to model-trace an input that involves the use of more than one buggy rule. This means that if the student makes more than one error[23] the system would not understand your answer. Because I wanted to be able to identify multiple

---

[23] This assumes that each buggy rule is written for a single error. However, it would be possible to add a different buggy rule for each different student input you wanted to have a buggy feedback message for. This would easily result in an exponential explosion in the number of rules you need to write.

errors occurring at the same time, I have modified the traditional model-tracing approach. Specifically, I first run the model-tracing algorithm but allow no buggy productions. If that succeeds, we know the answer was totally correct. If that fails, I run the model-tracing algorithm again but this time allow one buggy rule to be used. If that succeeds, we know there is only one error. If that fails, we run the algorithm again but this time allowing two buggy rules. We can continue this pattern until it takes too long to trace the input. For our system, we use a cut off of allowing 3 buggy rules. We then pass the last interpretation, (or lack of an interpretation) to the tutorial model to determine what to say next (which will be subject of the next section). This approach guarantees that we will find an interpretation of the student's answer using the least number of buggy rules possible.

Keep in mind that if a student has made errors on two subgoals, we need to generate a plan that deals with each of the errors separately. This is what the human tutor did in Chapter 2 (as shown after line 197). These questions and others will be addressed in the upcoming section describing the tutorial model.

The Architecture of the Tutorial Model

Up to this point, we have described how the student model is used to provide a diagnosis for each student utterance. That diagnosis is fed, as input, into the tutorial model. The tutorial model that we will now describe, was designed to be able to:

- Carry on a coherent dialog, which entails being able to have sub-dialogs that ask new questions rather then simply giving hints.
- Recognize pieces of a correct answer and give positive feedback on those portions while at the same time plan how to tutor the incorrect pieces.
- Support multiple different multi-step tutorial strategies.
- Have feedback that is sometimes specific to a type of error if we have a good pedagogical response (i.e., KRD), otherwise, use more general strategies (i.e., KCD).
- Give reflective follow-up questions, particularly after evidence of weak understanding by the student.

The Behavior of the Tutorial Model

Ms. Lindquist usually behaves as follows. She starts each problem by asking the student to answer the problem. Thereafter, the student model will diagnose the latest student response, and pass that diagnosis over to the tutorial model to plan what action to take. The tutorial model has what we call an agenda. The agenda is used to store the current tutorial plan. The agenda always starts out with one question. That question is always a symbolization question. At various points, the tutor might decide to ask a new question. We will now take a look at an example, but first, a caveat; this chapter is meant to discuss the architecture in general. The next chapter is meant to discuss an example of this architecture (i.e., the Ms. Lindquist tutor). Therefore, you will find fuller information on all the question types and strategies that the Ms. Lindquist tutor uses in Chapter 4. Nevertheless, its very hard to talk about an architecture without being able to refer to an example, so I will give several example, but the reader is asked to hold question about the particular questions and strategies until the next chapter.

An Example to illustrate how the agenda works

We will illustrate the agenda mechanism with the following dialog taken from a log file of a high school student who was tutored by Ms. Lindquist under the conditions described in Chapter 5. In keeping with the last chapter, we will always refer to a student with the masculine pronoun and the tutor with the feminine pronoun. This example took about 5 minutes. Each entry in the log is date and time-stamped and also gives the elapsed time in the last set of square brackets; for instance at line 2 the computer took zero seconds to generate the question, while at line 3 the student took 76 seconds to generate "(y*3)". Each question then gives the question's type (indicated with "qtype="). For this example, the question types are either QSYMB for symbolization, QCOMPUTE for arithmetic, QEXPLAIN for the articulation step, or QGENERALIZE for the generalization question (see Chapter 4 for details on these questions). Additionally, each question is tagged with a "goal" that gives one of the possible correct ways to answer the question (e.g., at line 2 the correct answer is an expression that is equivalent to "y+3*y"). The student's responses are listed in bold.

1. <Problem_Statement>You are mixing paint. You start with "y" liters of yellow paint and then you must add 3 times as much red as you did yellow. Write an expression for the amount of paint you used.</Problem_Statement>
2. <Tutor time="[7/24/2000][19:1:42][0]" qtype="QSYMB" goal="y+3*y">Hello. Please write an expression for the total amount of paint you used. </Tutor>
3. <Student time="[7/24/2000][19:2:58][76]">**(y*3)**</Student>
4. <Tutor time="[7/24/2000][19:2:59][0]" qtype="QCOMPUTE" goal="20"> Hmm. No. We will come back to the total amount of paint you used. Let me try to break this down for you. Please compute the total amount of paint you used if the amount of yellow paint was 5 liters? </Tutor>
5. <Student time="[7/24/2000][19:3:24][26]">**5*3**</Student>
6. <Tutor time="[7/24/2000][19:3:24][0]" qtype="QCOMPUTE" goal="15"> Hmm. No. Let me ask you a different question. Please compute the amount of red paint you used if the amount of yellow paint was 5 liters? </Tutor>
7. <Student time="[7/24/2000][19:3:47][22]">**15**</Student>
8. <Tutor time="[7/24/2000][19:3:47][0]" qtype="QCOMPUTE" goal="20">Correct. Now, lets try this question again: Please compute the total amount of paint you used if the amount of yellow paint was 5 liters? </Tutor>
9. <Student time="[7/24/2000][19:4:3][16]">**15**</Student>
10. <Tutor time="[7/24/2000][19:4:3][0]" qtype="QCOMPUTE" goal="20"> Hmm. No. Try again please. If the amount of yellow paint was 5 liters instead of "y" what would the total amount of paint you used be?</Tutor>
11. <Student time="[7/24/2000][19:4:34][31]">**20**</Student>
12. <Tutor time="[7/24/2000][19:4:34][0]" qtype="QEXPLAIN" goal="5+3*5">Correct. Using only 3, and 5 write an expression for the total amount of paint you used.</Tutor>
13. <Student time="[7/24/2000][19:5:19][45]">**(3*5)+5**</Student>
14. <Tutor time="[7/24/2000][19:5:19][0]" qtype="QGENERALIZE " goal="y+3*y">Correct. Now use "y" instead of 5 to again express the total amount of paint you used.</Tutor>
15. <Student time="[7/24/2000][19:6:3][44]">**(3*y)+5**</Student>
16. <Tutor time="[7/24/2000][19:6:3][0]" qtype="QGENERALIZE " goal="y+3*y">It looks like you didn't understand what I was asking you. You should use the variables from the problem. For instance, use y instead of 5. Try this question again: Now use "y" instead of 5 to again express the total amount of paint you used.</Tutor>
17. <Student time="[7/24/2000][19:6:36][34]">**(3*y)+Y**</Student>
18. <Tutor time="[7/24/2000][19:6:36][0]" qtype="NIL" goal="NIL">Correct. A new problem is needed </Tutor>

We will use this example to give a quick overview of the system focusing on the tutorial agenda structure. We then will come back in subsequent sections to explore the system in more detail. In the above dialog, we see that the student initially said "(3*y)". The tutor chose to use the *concrete articulation* strategy arbitrarily among its four KCDs. This strategy involves three steps

Qcompute: Total Paint: 20
QExplain: Total Paint: 5+3*5
QGeneralize: Total Paint: y+3*y

The questions on top are asked before the questions on the bottom. The student went on to answer the question at line 5 incorrectly with "5*3". Because the tutor knew how to break down that question, it did so. This involved adding an extra computational question to the stack so that the agenda looked like the following.

Qcompute: Red Paint: 15
Qcompute: Total Paint: 20
QExplain: Total Paint: 5+3*5
QGeneralize: Total Paint: y+3*y

At line 7, the student got the top-level question correct so the tutor popped that question off of the top of the agenda. The student is presented again with the same question from line 4. The agenda looks like the following.

Qcompute: Total Paint: 20
QExplain: Total Paint: 5+3*5
QGeneralize: Total Paint: y+3*y

At line 9, the student gets the question wrong, but since we have already done one KCD on this step, the tutor instead gives a hint. A hint only changes how the top-level question is phrased. It is not viewed as a new question. In this case, the first level hint simply asks the student to try again. If the student had made another error, the hint message would get more explicit about what the student should do. However, it turns out that this hint is enough to get the student to answer the question correctly at line 11. Then the tutor presents the articulation question. The agenda looks like the following.

QExplain: Total Paint: 5+3*5
QGeneralize: Total Paint: y+3*y

The student gets the next two questions correct (line 13 and 15) and each time they are popped off the agenda. After line 17, the tutor's agenda is empty and a completely new problem is selected.

We now explore the algorithm's behavior more generally. The following is a picture (Figure 3) depicting the behavior of the algorithm.

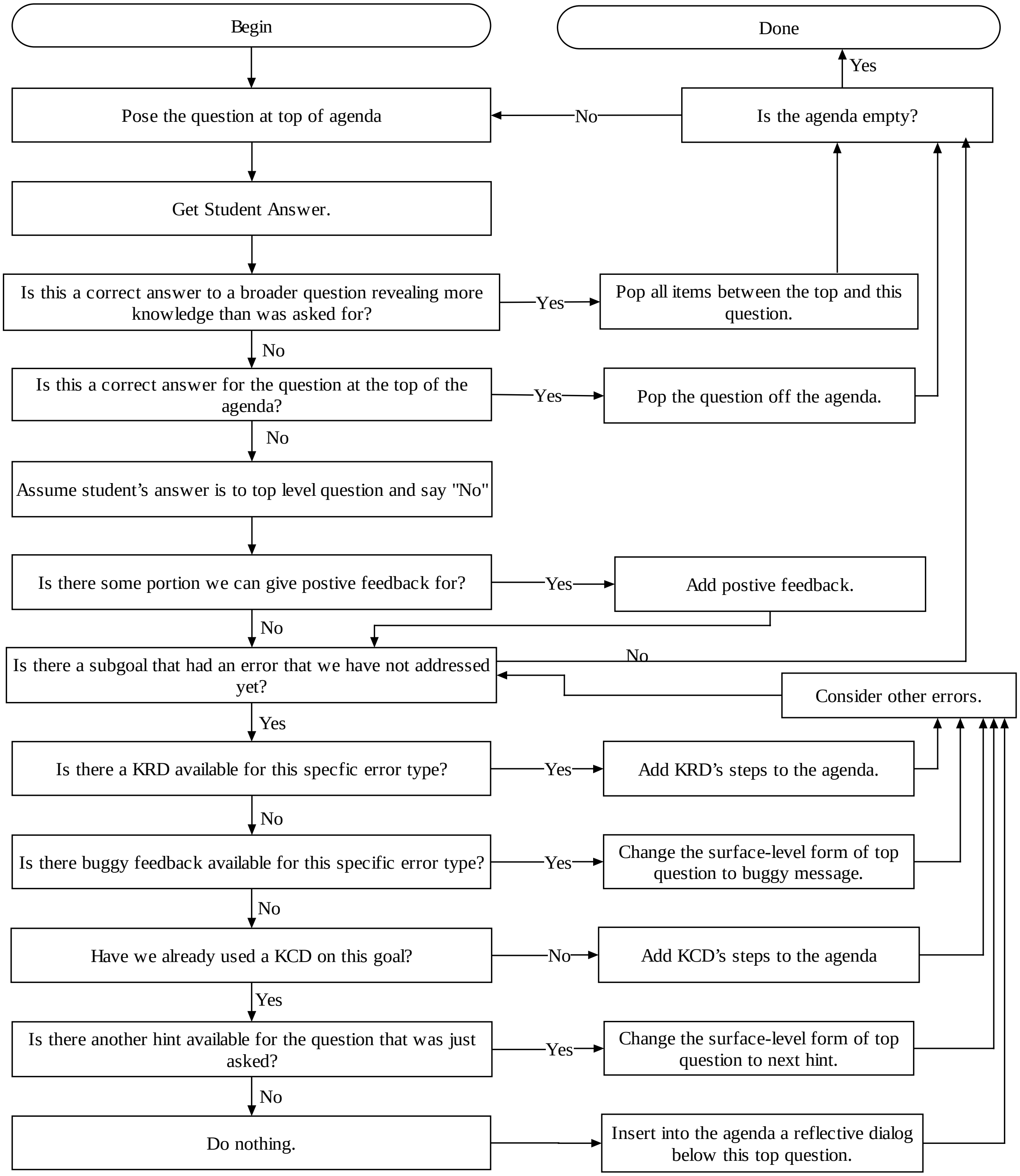

Begin
Done
Yes
Pose the question at top of agenda
No
Is the agenda empty?
Get Student Answer.
Is this a correct answer to a broader question revealing more knowledge than was asked for?
Yes
Pop all items between the top and this question.
No
Is this a correct answer for the question at the top of the agenda?
Yes
Pop the question off the agenda.
No
Assume student's answer is to top level question and say "No"
Is there some portion we can give postive feedback for?
Yes
Add postive feedback.
No
Is there a subgoal that had an error that we have not addressed yet?
No
Consider other errors.
Yes
Is there a KRD available for this specfic error type?
Yes
Add KRD's steps to the agenda.
No
Is there buggy feedback available for this specific error type?
Yes
Change the surface-level form of top question to buggy message.
No
Have we already used a KCD on this goal?
No
Add KCD's steps to the agenda
Yes
Is there another hint available for the question that was just asked?
Yes
Change the surface-level form of top question to next hint.
No
Do nothing.
Insert into the agenda a reflective dialog below this top question.

Figure 3: A flow diagram depiction the normal behavior.

In order to provide an overview, I will briefly describe the steps shown in the figure. The next sections will go into greater detail. First, the tutor presents a new problem and asks the student the first question. The system then diagnoses if the student has correctly answered more than he was asked for. Assuming that is not the case, the system then determines if the answer is correct or not. Assuming it is incorrect, the system tries to give positive feedback. Next, the system must decide what to do for each subgoal that was diagnosed as having an error. The system first considers using a knowledge remediation dialog specific to the error type encountered. Next, the system considers responding with buggy feedback, which changes the form of the question, without changing what is asked for. Following that, the system checks to see if the question can be broken down by using a knowledge construction dialog. If that also is not applicable, we give the next available hint. Note that a hint simply changes the surface-level form of the question asked but does not change what is being asked for. If none of these conditions apply we repeat the last question asked, which will be the most explicitly hint message. If the system gets to its bottom-out hint, there is a good chance the student does not have a good understanding. Therefore, the system can ask a reflective follow-up question

Types of Diagnosis

As described above in the student model section, the diagnosis will fall into one of three categories.

1) Correct for the question the tutor just asked.
2) The students answer appears to be an answer to a previous question- This is a special case and will dealt with below.
3) Wrong answer: This has two sub-categories
   a) We trace the input which means we know which buggy productions were used to trace the student's input
   b) Uninterpretable

The Tutorial Model tries to find the most precise pedagogical response to the student model's diagnosis

So how does the tutor respond to the diagnosis? The easiest case is if the student says something correct in which case the tutor confirms the answer and asks the next question on the agenda or goes on to the next problem if the agenda is empty. We will now deal with the case in which the student said something wrong but nevertheless we could trace the input. There are four possible responses.

1. Use a Knowledge Remediation Dialog (KRD)
2. Give Buggy Feedback
3. Use a Knowledge Construction Dialog (KCD)
4. Give a Hint

The first two types are responses that apply only if the student made a certain type of error. The following table gives an example of each. One of the KRDs we have operating happens if the student commits an error of omission. An error of omission is defined to be the student correctly symbolizing only a piece of the problem. For example, suppose the student was supposed to say "800-40m" but instead said "40*m" the tutor would use the one step KRD that ask the student to identify what the "40*m" represents. Note, that a KCD is a multi-step plan that can be used to tutor any sub-expression that has at least one operator and they will each be described in a section below. Finally, our system will give a hint if the other three possibilities have been exhausted. These four types of response are summarized in the following table. The first column contains responses that simply tell the students some information, without changing the question. The first row in the table has tutorial responses that always apply, while the responses in the second row apply only in response to certain conditions.

	Two types of Tutor Reponses	
	Tell the student something	**Ask a New Question(s)**
Always Applies	Model-tracing Hint at the expression level. "You need to use 40 and ’m’ to find the distance rowed."	Use one of the four KCDs such as the three-step *Concrete Articulation* Strategy
Applies selectively	Buggy Feedback in Model-tracing e.g., You are missing parentheses around the “30-g”.	Use a KRD such as "Your answer of ’40m’ is part of the answer. What does ’40m’ represent in English?"

In summary, a guiding principle in this tutor's architecture is that when deciding between alternative possible responses, the tutor will choose to say the response that matches the student’s response most specifically.

What to do if the student model cannot interpret the student’s answer?

Most of the time, if the student says something that is uninterpretable, we treat that as if the student made an error on each subgoal. There is one exception to this rule and it only applies to using the concrete articulation strategies, which will be addressed in the next section.

We have Four General Pedagogical Responses (KCDs) That Always Apply

The four general strategies we have implemented are listed below. Each of them involves asking a series of questions. Examples of each will be provided in the follow chapter.

1) Concrete Articulation: Scaffold from a concrete instance by asking three questions
   a) Compute for a concrete instance.
   b) Articulate the mathematics operations needed to do that computation.
   c) Generalize that articulation to the case when a variable is present.
2) Explain Verbal: Proceed in a bottom-up manner asking students to
   a) Identify a sub-goal to achieve[24]
   b) Explain in English (using a pull down menu) how they would achieve the goal
   c) Follow that up by symbolizing that subgoal
3) Worked Example: Tell the student the answer. Then, in a bottom-up manner, ask the following questions to see if they understand the example.
   a) Explain the components of each sub-expression.
   b) Explain what each subgoal represents
4) Decomposition & Substitution: For a one-operator expression, simply ask the student to symbolize it. For expression with more than one operator, introduce a variable to stand for any subexpression and ask the student:
   a) To symbolize the expression with the introduced variables
   b) Then re-express the same quantity, but substituting in the correct expression of the introduced variable.

All of these strategies attack the incorrect goals in a bottom-up fashion, meaning they deal with the subexpression (e.g., "40*m") before dealing with the expressions containing them (e.g., "800-40*m"). For example, suppose the goal it to symbolize "5g+7*(30-g)" and the student answered with "5g+7/(30-g)", the tutor would confirm the completely correct portions (i.e., the "5*g" and

---

[24] I initially implemented this step, but we turned it off before our controlled experiment, thinking this step might not be worthwhile. I now speculate that we should have left it on because there might be benefits to letting students pick the subgoal to address rather then letting the tutor pick. This is a question for future work. All the examples for the rest of this dissertation omit this step and instead the tutor picks a subgoal to ask students to explain.

the "30-g") and push onto the agenda goals to deal with the "7*(30-g)" subexpression. Since the top-level subexpression was done correctly (i.e., the adding the portions representing both the "5g" and the "7*(30-g)" together), the system would not add goals to the agenda for the top-level subexpression (i.e., "5*g+7*(30-g)"). If, on the other hand, the answer had been "5*g-7/(30-g)" (i.e., two errors) then the tutor would also add goals to deal with the top-level subexpression. To summarize, the system generally addresses subexpressions that the student made an error on. In addition, the system addresses the errors by dealing with sub-expression before dealing with containing expressions (i.e., bottom-up).

There is one exception to dealing with the goals in a bottom-up manner and it applies only when the system is using the concrete articulation strategy. If the concrete articulation strategy is used on a problem, *and also* the student's answer is uninterruptible, the tutor will act as if the only goal that was not achieved was the top-level goal. This means that the system will push compute, articulate and generalization questions for the top-level goal only. If the student has trouble, the tutor can further break down those goals. This allows the tutor to "dive-down" on a sub-expression as needed[25].

How do we pick which tutorial response to use?

Each of these four strategies can be generally applied to any subexpression the student did not get correct. Currently, one of these general strategies is arbitrarily (at random) set by the system ahead of time. We imagine in the future we might have learned good meta-tutorial rules that suggest which strategy to use, on which problems, or in response to what types of errors were made. In fact, Arroyo et al. (2000) have started to develop this sort of valuable knowledge. One could even imagine learning what types of students, with what types of errors, learn fastest when presented with different sorts of scaffolding strategies. Learning this is beyond the scope of this thesis as it requires a great deal of data and is left as a goal of future work.

Issues Relevant to Building A Dialog System: Lower Level Issues of Text Generation:

So far, we have been primarily concerned with figuring out how to break problems down for students, but there are also other important issues that have to be dealt with if a dialog system is going to be effective. We illustrate some of the issues with an example from the experiment we describe in a coming chapter. The example also illustrates a danger inherent to systems that ask new questions; sometime the student might misunderstand what a question is asking for. The system designer needs to try to get the system to recognize such misunderstanding and respond appropriately.

1. `<Problem_Statement>Latisha, Kendra and Tonya worked all summer mowing lawns. Tonya made "t" dollars and Latisha made twice as much as Tonya. Kendra worked the hardest and made three times as much as both of the other girls combined. Write an expression for the amount of money Kendra made mowing lawns.</Problem_Statement>`
2. `<Tutor time="[7/24/2000][18:54:53][1]" qtype="QSYMB" goal="(t*2+t)*3">Hello. Please write an expression for the amount Kendra made. </Tutor>`
3. `<Student time="[7/24/2000][18:56:23][90]">`**`t*2*3`**`</Student>`
4. `<Tutor time="[7/24/2000][18:56:23][0]" qtype="QCOMPUTE" goal="24"> No. But, "t*2" is correct for the amount Latisha made. We will come back to the amount Kendra made. Let me try to break this down for you. Please compute the combined amount of Latisha and Tonya if the amount Tonya made was 8 dollars? </Tutor>`
5. `<Student time="[7/24/2000][18:57:25][62]">`**`24`**`</Student>`
6. `<Tutor time="[7/24/2000][18:57:25][0]" qtype="QEXPLAIN" goal="8*2+8">Correct. Using only 2, and 8 write an expression for the combined amount of Latisha and Tonya.</Tutor>`
7. `<Student time="[7/24/2000][18:58:9][45]">`**`t*2*8`**`</Student>`

---

[25] See lines 4 and 6 from the most recent example presented for an example of this "diving-down" behavior.

```
8.  <Tutor time="[7/24/2000][18:58:10][0]" qtype="QEXPLAIN" goal="8*2+8">It looks like you didn't
    understand what I was asking you.  For this question, you should not use variables: For
    instance, use 8 here instead of "t".  Try this question again: Using only 2, and 8 write an
    expression for the combined amount of Latisha and Tonya.</Tutor>
9.  <Student time="[7/24/2000][18:58:33][23]">2*8</Student>
10. <Tutor  time="[7/24/2000][18:58:33][0]"  qtype="QEXPLAIN"  goal="8*2+8">  No.[sic  programming
    error results in saying 'No. No.'] No. Your answer of "2*8" equals "16", but you wanted to
    write an expression that equaled "24".  Using only 2, and 8 write an expression for the
    combined amount of Latisha and Tonya. Hint: Use 8*2 for the amount Latisha made and 8 for the
    amount Tonya made. </Tutor>
11. <Student time="[7/24/2000][18:59:16][43]">8*2+8</Student>
12. <Tutor time="[7/24/2000][18:59:16][0]" qtype="QGENERALIZE" goal="t*2+t">Correct. Now use "t"
    instead of 8 to again express the combined amount of Latisha and Tonya.</Tutor>
13. <Student time="[7/24/2000][18:59:39][23]">t*2+8</Student>
14. <Tutor  time="[7/24/2000][18:59:39][0]"  qtype="QGENERALIZE"  goal="t*2+t">It  looks  like  you
    didn't understand what I was asking you.  You should use the variables from the problem.  For
    instance, use t instead of 8.  Try this question again: Now use "t" instead of 8 to again
    express the combined amount of Latisha and Tonya.</Tutor>
15. <Student time="[7/24/2000][19:0:10][32]">t*2+8</Student>
16. <Tutor  time="[7/24/2000][19:0:10][0]"  qtype="QGENERALIZE  "  goal="t*2+t">It  looks  like  you
    didn't understand what I was asking you.  You should use the variables from the problem.  For
    instance, use t instead of 8.  Try this question again: Now use "t" instead of 8 to again
    express the combined amount of Latisha and Tonya.</Tutor>
17. <Student time="[7/24/2000][19:0:22][13]">t*2+t</Student>
18. <Tutor  time="[7/24/2000][19:0:22][0]"  qtype="QSYMBWRAPUP"  goal="(t*2+t)*3">Correct.    Now,
    lets  try  this  question  again:   Please  write  an  expression  for  the  amount  Kendra  made.
    </Tutor>
19. <Student time="[7/24/2000][19:1:1][38]">t*2+t*3</Student>
20. <Tutor  time="[7/24/2000][19:1:1][0]"  qtype="QSYMBWRAPUP"  goal="(t*2+t)*3">  No.  You  need  to
    put parentheses around the "t*2+t".  Try again. Please write an expression for the amount
    Kendra made. </Tutor>
21. <Student time="[7/24/2000][19:1:42][41]">(t*2+t)*3</Student>
22. <Tutor time="[7/24/2000][19:1:42][0]" qtype="NIL" goal="NIL">Correct. A new problem is needed
    </Tutor>
```

This example illustrates positive feedback, dialog coherence moves, dealing with misunderstanding of the questions, and negative feedback that does not just tell the student the answer. We now point out each of these.

The first is positive feedback. The first response the tutor makes shows the tutor saying that the answer is wrong, but providing positive feedback for a portion of the answer. Ms. Lindquist also interpreted the "*3" as correct, but being applied to "2t" instead of the quantity "t+2t", which is what Ms. Lindquist was expecting. Therefore, Ms. Lindquist chose to scaffold the goal of "`the combined amount of Latisha and Tonya.`" Remember that Ms. Lindquist has four strategies at her disposal she can use to scaffold any subexpression. She arbitrarily picked to use the concrete articulation strategy on this subgoal. The concrete articulation strategy involves adding to the tutorial agenda three questions, on top of the question that is already on the agenda. The agenda now looks like this:

QCOMPUTE: with t=8 to get an answer of 24
QEXPLAIN: 8+2*8
QGENERALIZE: t+2*t
QSYMB: (t+2t)*3

Second, our early user testing revealed that we had to pay attention to the fact that students needed cues to realize that the tutor was asking a different question. Our system uses phrases like "Let me try to break this down for you"(line 4) and "Lets try this question again"(line 18) as appropriate to maintain overall coherence of the dialog. An example of this is shown (at line 4) by the tutor, when Ms. Lindquist says, "`We will come back to the` amount Kendra made. Let me try to break this down for you." Ms. Lindquist is warning the student that a new question is coming. How does Ms. Lindquist know when to put in a cue? She can tell, by observing the structure of the dialog. Therefore, if she just decided to add a new sub-dialog, she can be sure to add the appropriate conversational cues. These bits of text and generated from templates. As we said above, Ms. Lindquist chose to ask a compute question. The student answers the question correctly with "24". But the next question asks the student to articulate the mathematics that was used to get that "24", and the student does not understand that so Ms. Lindquist said (line 8) "It looks like you didn't understand what I was asking you. For this question, you should not use variables: For instance, use 8 here instead of "t". The student followed the direction of the tutor not to use a variable in his answer, but still was unable to explain how he got the 24. This is not very surprising given the results we saw from the DFAs that showed that articulation is substantially harder than arithmetic.

Next (at line 10), Ms Lindquist gave constructive negative feedback with "No. Your answer of "2*8" equals "16", but you wanted to write an expression that equaled "24". Ms. Lindquist also followed that with a hint. The student was then able to get the correct answer and was then asked to generalize the answer, but this time the student didn't understand and used the concrete number that introduced to help in the compute question, so Ms. Lindquist says (line 14) " It looks like you didn't understand what I was asking you. You should use the variables from the problem. For instance, use t instead of 8." This misunderstanding took two attempts before it was cleared up.

This example ends with the student making an error of not using parentheses. Remember, our heuristic is that the tutor chooses the response that has the most precise pedagogical fit. In this case there is a pedagogical response to this precise error and Ms. Lindquist says (line 20) "You need to put parenthesis around the 't*2+t'". This example has shown several examples of the behavior of Ms. Lindquist that have to do with lower-level issues of text generation to maintain dialog coherence.

Examples of noticing the student appeared to answer a question other than the top question on the agenda

Above we mentioned that sometimes the tutorial model's diagnosis would be that the student's answer appears to be answering a different question, or possibly answering in a way that reveals more knowledge than he was asked to provide. Our goal is that the tutor should not be pedantic. If the student's answer reveals more knowledge than was asked, we should confirm the student's response and move on to the next question. For instance, in the following problem the student appears not to notice that at line 4 a new question was asked. The student quick response time also seems to indicate that the student was trying to quickly correct his answer, not realizing the system asked a new question. The student did not get the top level-question correct, so the system did not give him credit, but on line 7 the student does, and the system gives him credit.

```
1. <Problem_Statement>A plumber not only charges $30 per hour but she also charges $15
   just to come to your house.  Write an expression for how much you have to pay if
   this plumber works for "h" hours.</Problem_Statement>
2. <Tutor time="[7/18/2000][8:48:19][0]" qtype="QSYMB" goal="15+30*h">Hello.  Please
   write an expression for the total you pay the plumber. </Tutor>
3. <Student time="[7/18/2000][8:49:0][41]">30+15*h</Student>
```

```
4. <Tutor time="[7/18/2000][8:49:0][0]" qtype="QSUB_QUAN_STRICTLY" goal="30*h"> Hmm.
   No.   We will come back to the total you pay the plumber.  Let me try to break
   this down for you. What is the amount she makes without the house call
   charge?</Tutor>
5. <Student time="[7/18/2000][8:49:10][10]">30+15/h</Student>
6. <Tutor time="[7/18/2000][8:49:11][0]" qtype="QSUB_QUAN_STRICTLY" goal="30*h"> Hmm.
   No.  Use "30" and "h" to calculate the amount she makes without the house call
   charge.</Tutor>
7. <Student time="[7/18/2000][8:49:31][21]">30*h+15</Student>
8. <Tutor time="[7/18/2000][8:49:31][0]" qtype="NIL" goal="NIL">Correct. A new problem
   is needed </Tutor>
9.
```

Though it is true that the student can always move on if he can type the answer to the very first symbolization question, the algorithm is more subtle than that. Consider the following example[26].

```
1. <Problem_Statement>Debbie has two jobs over the summer.  At one job she bags groceries at
   Giant Eagle and gets paid 5 dollars an hour. At the other job she delivers newspapers and
   gets paid 7 dollars an hour.  She works a total of 30 Hours a week.  She works "g" hours
   bagging groceries.   Write an expression for the total amount she earns a
   week.</Problem_Statement>
2. <Tutor time="[10/10/2000][18:29:55][0]" qtype="QSYMB" goal="5*g+7*(30-g)">Hello.   Please
   write an expression for the total pay per week. </Tutor>
3. <Student time="[10/10/2000][18:35:37][343]">(30-g*7)+(g*5)</Student>
4. <Tutor time="[10/10/2000][18:35:38][0]" qtype="QSUB_QUAN_NOT_STRICTLY" goal="7*B"> No. But,
   "30-g" is correct for the hours delivering newspapers, and "g*5" is correct for the pay from
   the grocery.  We will come back to the total pay per week.  Let me try to break this down for
   you.  If "B" was the hours delivering newspapers then what would be the pay from the
   newspaper? </Tutor>
5. <Student time="[10/10/2000][18:37:25][108]">30-g(*7)</Student>
6. <Tutor time="[10/10/2000][18:37:25][0]" qtype="QSUB_QUAN_NOT_STRICTLY" goal="7*B">It looks
   like you didn't understand what I was asking you.  That is not a valid math expression.  Try
   this question again: If "B" was the hours delivering newspapers then what would be the pay
   from the newspaper? </Tutor>
7. <Student time="[10/10/2000][18:37:58][33]">7*(30-g)</Student>
8. <Tutor time="[10/10/2000][18:37:58][0]" qtype="QSYMBWRAPUP" goal="5*g+7*(30-g)">Correct.
   Now, lets try this question again: Please write an expression for the total pay per week.
   </Tutor>
9. <Student time="[10/10/2000][18:38:20][21]">7*(30-g)+(g5)</Student>
10. <Tutor time="[10/10/2000][18:38:20][0]" qtype="NIL" goal="NIL">Correct. A new problem is
   needed </Tutor>
```

Notice the positive feedback (line 4) that is part of our dynamic scaffolding. The student has an error only of forgetting parentheses around the "30-g". So the tutor adds two questions to the agenda making the agenda look like the following.

QSUB_QUAN_NOT_STRICTLY: 7*b
QSUBSTITUTE: 7*(30-g)
QSYMB: 5*g+7*(30-g)

The student then hits a buggy rule by type "30-g(*7)" (line 5) and it told his answer is not a valid mathematics expression. We can see that this answer is trying to skip over (consciously or unconsciously) the question asking for "7*b". At line 7, the student answers and the system registers that the student answered a question lower down on the agenda, gives credit, and pops two questions off the agenda.

---

[26] This example is from a student using our web site.

How Can The Tutorial Model be Extended?

One goal of mine was to have an architecture that could easily be extended. I believe I have accomplished that. Generally speaking, adding a new KRD (or KCD) means that you need to do the following four steps:

1) Write the tutorial rule that adds the new question(s).
2) Write the student model rule(s) that allow you to trace this answer for each new question asked.
3) Write a tutorial rule that generates the response for each question type added.
4) If you are interested in tracing certain types of errors, you need to add buggy rules to the student model to generate those error types.

These four steps are illustrated in Appendix J by showing the few rules needed a simple KRD.

Discussion: Distinctions between Problem Solving Goals and Tutorial Pedagogical Goals.

There is an important distinction between the subgoal that the student model uses to model problem solving actions and the sorts of goals (i.e., related to questions) that the tutorial model uses. The tutorial model will sometimes ask questions that look just like they are problem-solving goals (i.e., "Write an expression for the distance rowed"). However, often the questions the tutorial model asks are not necessarily steps along the problem solving solution path (i.e., "Explain how to calculate average speed."), but nevertheless the tutor asks them because the tutor's goal is future performance and not current performance. If the tutor's goal was just to assist current performance then the tutor could just tell the student the answer, which would certainly speed things up. But, of course, the tutors goal is to do what is possible to encourage learning and sometime that involves discussing something that might seem like a diversion.

Consider the example of asking the student to "Explain how to find average speed in English?" This is not strictly on the solution path. In one sense, the student has to figure this out in order to do the mathematics. But, in fact, we have reason to believe that this task is harder than the original mathematics question. (See Chapter 2 where the student had great difficulty when this was asked on lines 202 and 306 and also Aleven and Koedinger (2000a) that show that asking the student for an explanation is often harder than doing the mathematics.) This is why we say that there is a difference between problem solving goals and tutorial goals.

We can also view this in terms of the tutorial model, which has what Shulman (1986) called pedagogical content knowledge. Our tutoring system has both a student (i.e., "domain") model that provides for problem solving expertise, as well as a tutorial model that has some general tutorial strategies (i.e., dynamic scaffolding), but a crucial element is also pedagogical content knowledge, encapsulated in our knowledge construction dialogs.

Are these strategies just problems solving goals at a finer grain size? No. If we look at the compute question used in the concrete articulation strategy, we see that to answer that question requires students to use their procedures for arithmetic facts, (e.g., 5+2=7) but you do not have to know your arithmetic facts to symbolize a problem. Therefore, these strategies can involve elements that are not part of the problem solving process.

What is the theory as to why these strategies might be effective? Our answer is twofold. Having students construct knowledge probably plays a part. In addition, students can learn by the transfer of the underlying similar skills. Doing a computation problem practices some underlying skills that are the same in our student model as those for doing a symbolization problem. Therefore, the theory is that these strategies can help by allowing students to practice the same skills on easier problems. One possible reason these questions are easier is that they may be more amenable to errors-checking strategies. One reason symbolization is harder than computing is that it involves two new skills.

Chapter Conclusions

Our goals in designing the tutorial model that was more like human tutors were the following. We wanted a new system that could incorporate the advances of model-tracing tutors but that was also able to

1. Carry on a coherent dialog, which entails being able to have sub-dialogs where we ask new questions and don't just give hints.
2. Recognize pieces of a correct answer and give positive feedback on those portions while at the same time tutor the student on the incorrect pieces of their answer.
3. Support multiple different multi-step tutorial strategies.
4. Have feedback that is sometimes specific to a type of error if we have a good pedagogical response, otherwise, use strategies that are more general.
5. Give reflective follow up questions, particularly after evidence of weak understanding by the student.[27]

At the beginning of this chapter, we stated some criticisms of model-tracing tutors that I would like to repeat and revisit. McArthur et al. (1990) criticized Anderson's et al. (1985) model-tracing ITS and model-tracing in general "because each incorrect rule is paired with a particular tutorial action (typically a stored message), every student who takes a given step gets the same message, regardless of how many times the same error has been made or how many other errors have been made. … Anderson's tutor is tactical, driven by local student errors (p. 200)." They go on to argue for the need for a more strategic tutor.

Ms. Lindquist addresses this criticism. Ms. Lindquist's model of tutorial reasoning is both strategic (i.e., has general multi-step plans that can be used to breakdown problems) and tactical (i.e., can recognize and respond to "teachable moments.") The use of the agenda data structure to provide for the dialog management, gives Ms. Lindquist a strategic character. If the student makes a mistake, she can have a sub-dialog to deal with that mistake. If Ms. Lindquist can come up with a tutorial strategy (either a knowledge remediation dialog or buggy feedback) specific to the error the student made it do so. Otherwise, she will try to respond with a knowledge construction dialog.

Ms Lindquist also address the criticism form Ohlsson who said that in model-tracing ITS "there is no adaptation to the current cognitive state of the learner other than the classification of his last step as an instance of a particular type of error." Ms. Lindquist's responds differently according to what has previous happened in the dialog. The first time the student makes an error on a given question, the student will usually get some sort of tutorial plan. After that plan has been executed (or as part of that plan) the tutor will come back to the same question, but now the question might be phrased differently (e.g., consider the difference between "Please write an expression for the total amount of paint you used" vs. "Now use 'y' instead of 5 to again express

---

[27] I have not discussed reflective follow-up questions because I disabled them for the experiment in Chapter 5. The idea is a straightforward one. The if the students shows poor understanding the system can add a reflective question underneath the current question on the agenda. For example, once the student has received the most specific hint, the system can add a question that once the students complete the current question (i.e., guesses the correct answer or types in the answer presented in the hint) he will be asked.

the total amount of paint you used." The second response occurs after the student has already computed a concrete value and also articulated the arithmetic.)

I argue that Ms. Lindquist is more like a human tutor as defined by Merrill et al. (1995) who said "student-tutor dialogues were centered much more around student-initiated events, as they attempted to actively understand new instructional material and solve problems, [rather] than around tutorial presentation of material and subsequent querying of student understanding." Merrill et al. went on to say that "microanalysis of student-tutorial interactions in problem-solving situations suggests that tutors do more than simply re-teach a correct procedure component when students encounter impasses or errors. Our tutors focused on guiding the error repair process rather than communicating their guesses about what student’s misconception." Ms. Lindquist meets these criticisms as well by having dialogs focus on the student errors. She also does not reteach a procedure; rather she tries to have the students construct the knowledge form himself.

In the next chapter, we will explore more of the behavior of this system and see how it compares to the experienced human tutor we observed. In addition, we will compare it to other existing system.

Chapter 4: Ms. Lindquist's Tutorial Strategies and Comparisons

Chapter Overview

In the previous chapter we focused on the general architecture that Ms. Lindquist is built on and only briefly mentioned the specific dialogs (KCDs and KRDs) that she can use with students. This chapter focuses on the domain specific aspects of Ms. Lindquist, while the previous chapter focused on the domain independent aspects. This chapter provides a detailed description of the behavior of Ms. Lindquist and a qualitative evaluation of the system as a model of human tutoring. This qualitative evaluation comes in the form of a case-by-case review of the tutorial strategies exhibited by Ms. Lindquist and how these strategies are exhibited by the experienced human tutor. In my thesis proposal, I extracted 26 different tutorial operators that the experienced human tutor used and here I discuss which are modeled in Ms. Lindquist and which are not. This chapter also compares Ms. Lindquist's tutorial dialog architecture with the traditional model tracing architecture.

Prior Work

The title of this dissertation refers to the fact that model-tracing tutors have "forgotten the tutor". Model-tracing tutors have largely ignored modeling the tutor. However, other intelligent tutoring system, prior to model-tracing, had "forgotten the student", to the degree that they were lacking rich and accurate models of student behavior. The approach of Ms. Lindquist is to capture effective aspects of both.. By Ms Lindquist is not just a combination of a tutorial model with a student model. Ms. Lindquist's tutorial model contained many aspect of PCK (i.e., tutorial strategies). I will argue that no other exiting systems provide the level of integration between the student model and tutorial model.

In reviewing the literature, I will point out how well other systems have met the following criterion.

1. Being a dialog-based tutor.
2. More then a demo system, but actually used by students.
3. Is a systems that have a rich domain-specific model of student thinking and development that underlies tutorial decision making.

This third criterion is often associated with *model-tracing tutors* (Anderson, Boyle & Reiser, 1985, see also Wolfe, 1988 and Shute, 1995 for reviews). I will point out these aspects in systems, where appropriate. There have been no systems

In reviewing the literature leading up to Ms. Lindquist, I will first discuss those systems that are in the algebra symbolization (as opposed to algebra symbol-manipulation) domain.

Prior Work on Algebra Symbolization

There have been no systems focused on algebra symbolization that have met all three of these criterions. The ANIMATE system (Nathan, Kintsch, & Young, 1992) came out of the out of Kintsch and colleague's research on comprehending word problems described in Chapter 1. ANIMATE was a tool students could use to describe equations for word problems, which could then be "animated." The system did not tell the students when they made a mistake; instead, the student observed the behavior of the animation to figure what was wrong. The system was not dialog-based tutor.

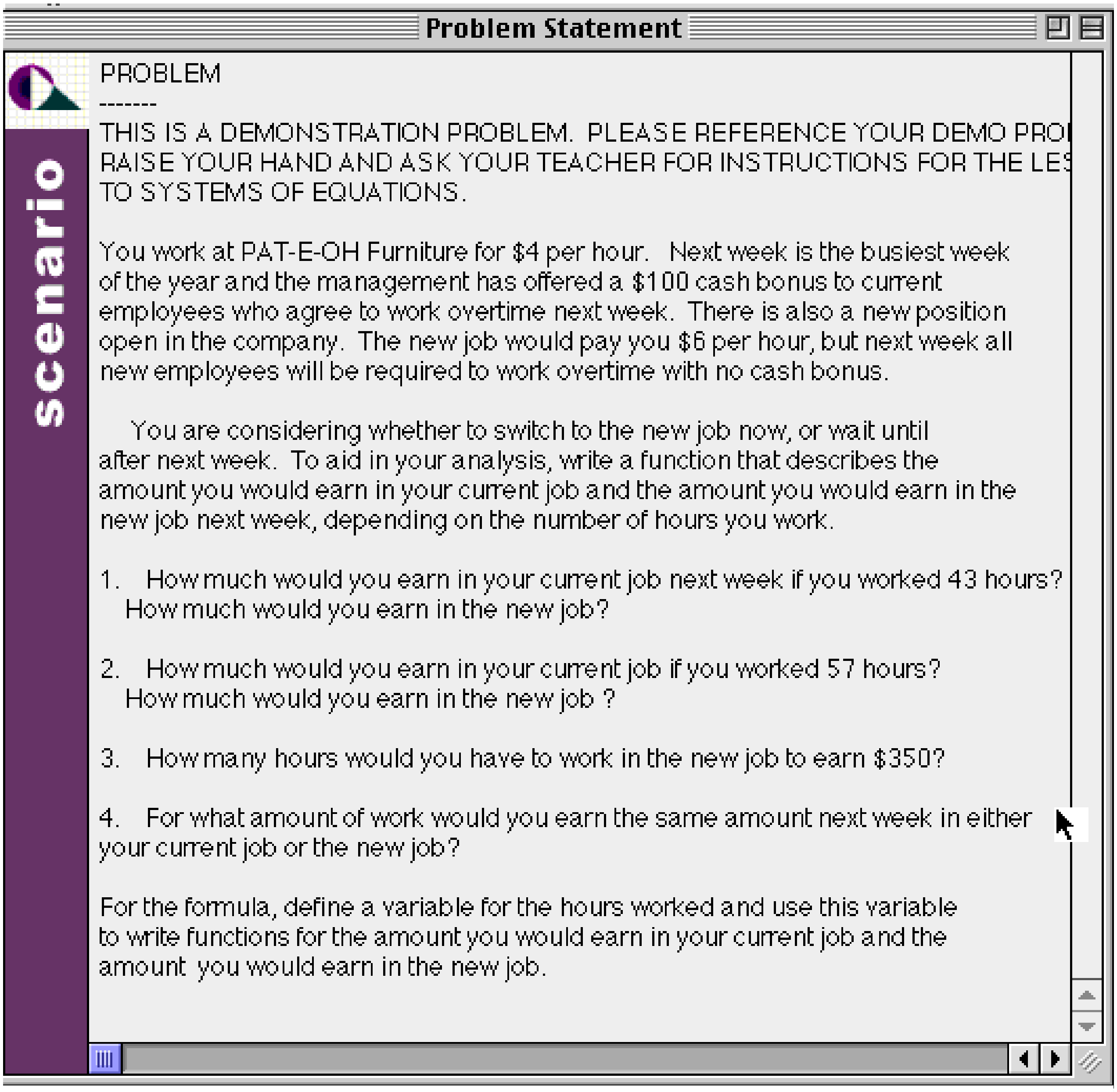
Sleeman (1982) studied student's errors in the realm of algebra equation solving and then produced a tutoring system based on this analysis. He observed that "when the problem is 'hard', the student makes error with rules that he previously succeeded (p. 198)." This is similar to the composition effect we explored in Chapter 1. However, Sleeman's system did not have a tutorial model, nor was it for the domain of symbolization.

Aziz, Pain & Brna (1995) built a "prototype" system called TAPS (Translating Algebra Problems System) that looked a how students would mistranslate the "student/professors" problem that Clement et al. (1980) studied. The researchers were particularly interested in modeling student's persistence in maintaining their beliefs even when presented evidence to the contrary. TAPS was neither a dialog-based nor a model-tracing tutor.

The tutor produced at Carnegie Mellon University is important to this dissertation and will be addressed in its own section.

The Carnegie Learning Tutor

Researchers at Carnegie Mellon University (Koedinger, Anderson, Hadley & Mark, 1995; Anderson, Corbett, Koedinger & Pelletier, 1995) have built the most widely used algebra tutoring system. In fact, it is the most widely used intelligent tutor, with over 50,000 students using it this year. Koedinger and his colleagues have spent over 7 years building and then refining their algebra word problem tutor, which is now marketed under the name "Cognitive Tutor". It also has a rich model of student thinking. The software teaches various skills in algebra that Ms. Lindquist does not address (i.e., graphing and equation solving). The two most relevant windows related to symbolizations are shown in Figure 1 and Figure 2. Figure 1 is a statement of a word problem, which poses multiple questions for the student to answer in the worksheet window (Figure 2).



PROBLEM

THIS IS A DEMONSTRATION PROBLEM. PLEASE REFERENCE YOUR DEMO PRO
RAISE YOUR HAND AND ASK YOUR TEACHER FOR INSTRUCTIONS FOR THE LE
TO SYSTEMS OF EQUATIONS.

You work at PAT-E-OH Furniture for $4 per hour. Next week is the busiest week of the year and the management has offered a $100 cash bonus to current employees who agree to work overtime next week. There is also a new position open in the company. The new job would pay you $6 per hour, but next week all new employees will be required to work overtime with no cash bonus.

You are considering whether to switch to the new job now, or wait until after next week. To aid in your analysis, write a function that describes the amount you would earn in your current job and the amount you would earn in the new job next week, depending on the number of hours you work.

1. How much would you earn in your current job next week if you worked 43 hours? How much would you earn in the new job?

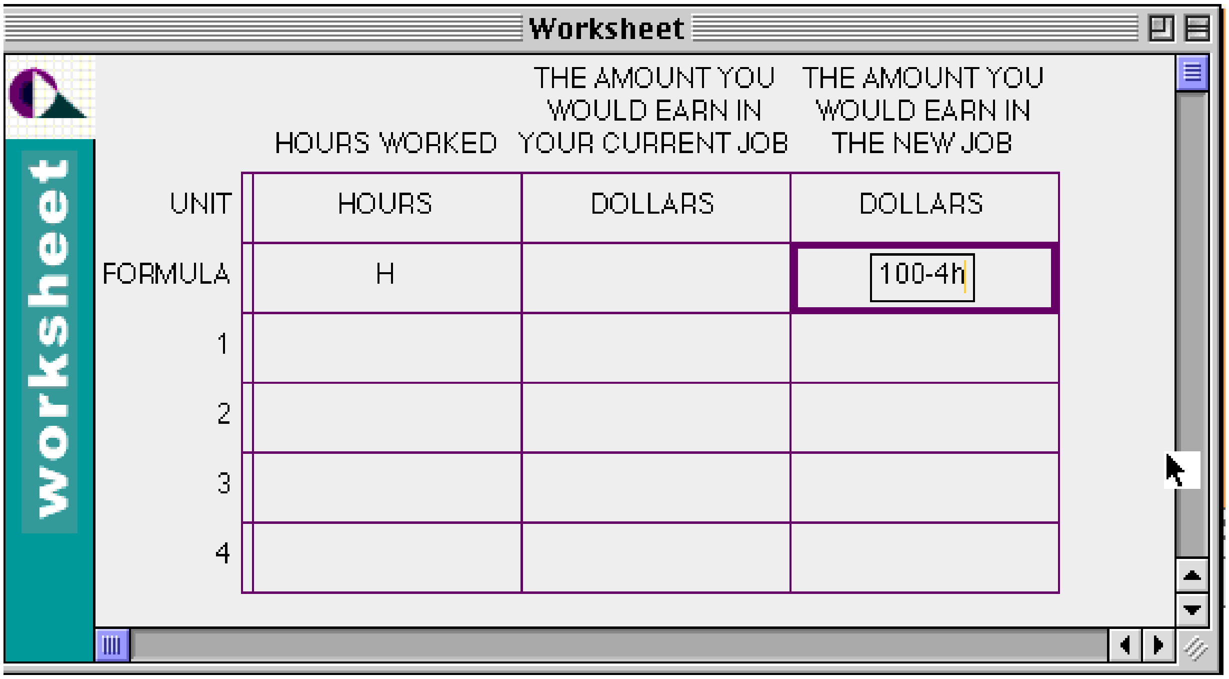
2. How much would you earn in your current job if you worked 57 hours? How much would you earn in the new job ?

3. How many hours would you have to work in the new job to earn $350?

4. For what amount of work would you earn the same amount next week in either your current job or the new job?

For the formula, define a variable for the hours worked and use this variable to write functions for the amount you would earn in your current job and the amount you would earn in the new job.

Figure 1: The Problem Statement window from the Carnegie Learning Tutor.



	HOURS WORKED	THE AMOUNT YOU WOULD EARN IN YOUR CURRENT JOB	THE AMOUNT YOU WOULD EARN IN THE NEW JOB
UNIT	HOURS	DOLLARS	DOLLARS
FORMULA	H		100-4h
1			
2			
3			
4			

Figure 2: The worksheet window from the Carnegie Learning tutor. The student has already filled in the column headings and the units, and is working on the formula row. The student has just entered "100-4h" but has not yet hit the return key.

We can see that the system reifies many of the steps of the problem including identifying the names for the quantities, identifying units, defining a variable, computing a few instances and symbolizing an expression. The portion of the interface that Ms. Lindquist is concerned with is the formula row. Figure 2 shows that the student is in the middle of attempting to answer this cell and has typed in "100-4*h" but has not yet hit return. The correct answer is 100+4*h. This is composed of the "bonus" (i.e., 100 dollars) plus the normal pay (i.e., "the money earned in your current job"). When the student hits return, he will get the a buggy message (see the first row in Table 1.)

	Error Types	The buggy message generated in response to those errors	Made-up possible response by the student.
1	100-4*h	Does the money earned in your current job increase for decrease as the number of hours worked increases?	It increases.
2	4*h 10+4*h	How many dollars do you start with when you calculate the money earned in your current job?	100 dollars
3	100-h 100+h 100+3*h	How much does the money earned in your current job change for each hour worked?	Goes up 4 dollars for every hour
4	4+100*h	Which number should be the slope and which number should be the intercept in your formula?	The 4 dollars an hour would be the slope.
5	100/(4*h) 4-100*h 100+4+h 100+4/h	Try using the pattern finder.	

Table 1: A list of the buggy messages generated for 5 different types of wrong answers. The first 4 error types ask questions that it would be nice if the student were allowed to answer. The fifth error type is a catchall error category.

Notice how the first four buggy messages are asking questions of the student. They seem like very reasonable questions that a tutor would ask a student. The last column in Table 1 shows responses that the questions seem to be asking for. Unfortunately, those are only rhetorical questions, for the student is not allowed to answer them, as such, and is only allowed to try to answer the original question again. (This is a problem the Ms. Lindquist architecture solves by asking embedding such questions into a dialog.)

When the system does not understand a student's response, the then system suggests that the student switch to a different tool, called the pattern finder (which was shown in Table 2 at the end of Chapter 1). Examples of errors that generate this response are shown in the last row of Table 1. Table 2 shows the hint sequence for this same symbolization question.

Text of Hint	**Comment**
Enter an expression to calculate the money earned in your current job using the hours worked.	General goal orientation
First, consider the initial value of the money earned in your current job. Next, consider how the money earned in your current job will change for each hour.	Consider the initial and the change
Write an expression that means the same thing as the value of the money earned in your current job plus the change in the money earned in your current job for each hour times the hours worked.	Word equation
Write an expression that means the same thing as 100+4 times the number of hours worked.	Partial Hint
Enter 4.00H + 100.00.	Bottom Out Hint

Table 2: The list of hints provide to students upon request.

The first hint (in Table 2) is to state the question the student is supposed to be answering. The second hint suggests that the student first figure out the initial (i.e., 100 dollars) value and then figure out the amount of change (i.e., 4 dollars) per hour (i.e., 'h' hours). The subsequent hints get increasingly explicit about what the student is supposed to do.

In conclusion, I believe that the Carnegie Tutor could be improved by replacing rhetorical questions with a dialog component. A second way the system could be improved would be if there were more than one tutorial strategy available. A third improvement would be if the scaffolding was more dynamic, so that a student did not always have to fill out a complete worksheet for every problem. Next, I review the relevant prior work on intelligent tutoring systems.

Prior Work on Intelligent Tutors

As an early intelligent tutoring system, Carbonell's (1970) SCHOLAR pointed the way to building Socratic dialog systems, but it was not was not tested in classrooms. SCHOLAR's ability to plan multi-turn dialog acts was limited.

Clancy's GUIDON (1983) was an early ITS that had both domain rules and tutoring rules but was not a model-tracing ITS. It is also not clear how accurately GUIDON's model is to approximating real students and tutors. GUIDON seems to have been closer to the "demonstration" category of systems, then the sort of system actually used by real students.

The CIRCSIM-Tutor project (see Cho, Michael, Rovick, and Evens, 2000, and Freedman & Evens, 1996) has done a great deal of research in building dialog based intelligent tutors systems. Their tutoring system, while not a model-tracing tutor, engages the student in multi-step dialogs based upon two experienced tutors. In CIRCSIM-Tutor, the dialog planning was done within the APE framework (Freedman, 2000). Freedman's approach, while developed independently, is quite similar to my approach in that is a production system that is focused on having a hierarchal view of the dialog. Dialog systems can have natural language generation and understanding components, as well as dialog planning components. The CIRCSIM-Tutor project has investigated several aspects of natural language understanding, including dealing with student initiatives. The Ms. Lindquist project is more focused on dialog planning and generally ignores the natural language understanding and generation problems that have occupied the CIRSCIM-Tutor project.

Freedman's APE framework is now incorporated into the Atlas-Andes (VanLehn, et al., 2000) project, which attempts to augment a traditional model-tracing tutor with knowledge construction dialogs. VanLehn et al. (2000) say that Atlas-Andes uses natural-language generation technology originally developed for the CIRCSIM tutor (Freedman & Evans, 1996), the LC-FLEX parser (Rose & Lavie, in press) and the COCONUT model of collaborative dialog (DiEugenio, Jordan, Thomas & Moore, in press). While working on problem solving, if a student is having difficulty with a particular principle, Atlas-Andes can select a KCD for the student to work on. These KCDs set-up a new situation (i.e., asking you to suppose that you were traveling in an elevator, or holding a rock in your hand, when the original situation was about a block on an inclined plane). This highlights one difference between Andes and Ms. Lindquist; Ms Lindquist's dialogs are all embedded within the problem-solving context the student is currently working on. The risk is that dialogs that set up a new context might be viewed as extra work, when in reality, the dialogs are selected to deal with the particular needs of the student. Another difference is that Ms. Lindquist model-traces all the answers given by students whereas Atlas-Andes does not.

Auto-tutor (Graesser et al., 1999) is a system that has a "talking head" that is connected to text-to-speech system. Auto-tutor asks students questions about computer hardware and the student types a sentence in reply. Auto-tutor uses latent semantic analysis to determine if a student's utterance is correct. That makes for a much different sort of student modeling, then model-tracing tutors. The most impressive aspects of Auto-tutor are its talking head and natural language understanding components. The Auto-tutor developers (Graesser et al.,1999) de-emphasize dialog planning based on the claim that novice human tutors do not use sophisticated strategies, but nevertheless, can be effective. Auto-tutor does have multiple tutorial strategies (i.e., "Ask a fill-in-the-blank question" or "Give negative feedback."), but these strategies are not multi-step plans. However, work is being done on a new "Dialogue Advancer Network" (Person & Graesser 1999) to increase the sophisticated of its dialog planning. Auto-tutor's tutorial strategies incorporate general pedagogical knowledge, but are lacking in more specific and powerful pedagogical content knowledge that is critical to more effective instruction.

Core, Moore, and Zinn (2000) propose a model that incorporates both student modeling and tutorial modeling in the context of a reactive planner but that system is in the planning stages.

Aleven and Koedinger (2000a & 2000b) have built a geometry tutor in the traditional model-tracing framework but have added a requirement for students to explain each of their problem-solving steps. An enhanced version allows natural language understanding of these explanations (Popescu & Koedinger, 2000). The new system uses the same parser (Rose & Lavie, in press) as the Atlas systems described above. Its uses Loom (MacGregor, 1991) for knowledge representation. The system's goal is to use traditional buggy feedback to help students refine their explanations. Many of the hints and buggy messages ask new questions, but they are only rhetorical. Unlike Ms. Lindquist, the system is not prepared to accept a student answer to such questions and then engage in further dialog to help students reason about how this answer helps them with the problem-solving step they are stuck on. I believe such dialog enhances student learning.

An important issue that all intelligent tutoring systems have faced is what to do if the student makes more then one error at a time. Many systems treat multiple errors as part of a generic “anything else” category but some systems have attacked this difficult diagnosis problem and provide can assistance specific to different kinds of multiple error situations. For another example, the PROUST and CHIRON systems (Sack & Soloway, 1992) were intelligent tutoring systems that detected multiple errors in Pascal programs, but Sack & Soloway recognized that the system could be confusing when displaying for students, what amounted to, separate error messages.

The MENO-TUTOR (Woolf & McDonald, 1984) attempted to model the discourse strategies of human tutors. It included modeling tutorial tactics, strategies and more general pedagogical states. The MENO-TUTOR is one of the few intelligent tutoring systems to include a significant pedagogical component, but unlike Ms. Lindquist, it did not emphasize, nor model, pedagogical content knowledge.

The Generic Tutoring Environment (GTE) is an authoring environment by Van Marcke (1998). Murray (1998) has said few researchers in intelligent tutoring systems have modeled complex tutorial strategies in such depth as Van Marcke. Murray adds that "Van Marcke describes both a formalism for representing instructional expertise, and a large database of encoded instructional knowledge." Unfortunately, GTE is not a model-tracing tutor, nor does it shows how a model-tracing component could be integrated into his architecture. Van Marcke emphasis on domain independence completely ignores the importance of pedagogical content knowledge.

The 4 demonstrations systems built by Rickel, Ganeshan, Lesh, Rich & Sidner, (2000) are interesting due to the incorporation of an explicit theory of dialog (Grosz & Sidner, 1986) but their pedagogical content knowledge is very weak.

There has also been relevant prior work on comparing human tutors and intelligent tutoring systems. Merrill, Reiser, Ranney and Trafton (1992) argued that human tutors were more subtle and flexible then model-tracing tutors, but nevertheless, the two were more similar than normally thought. However, they argued that one area for improvement was in how the computer assisted the error recovery process. They said "In general, human tutors manage to assist student while having them do more of the error recovery process" as compared to model - tracing tutors. Ms. Lindquist's dialogs are designed to do that by allowing the student to do more of the error repair themselves, by asking students targeted questions.

Prior Work Relevant to the Particular Strategies and Dialog Capabilities of Ms. Lindquist

Ms. Lindquist has four different tutorial strategies, two of which are direct implications of my own work and two others that are also supported by other research.

1. The **"Concrete Articulation"** Strategy: This strategy grew out of our difficulty factors assessment (DFA) research presented in Chapter 1.
2. The **"Explain in English First"** Strategy: This strategy, inspired by our experienced human tutor, finds theoretical support in the work on subgoal-reification by Corbett & Anderson (1995). They provide evidence that a tutor enhanced to first ask students to state their goals before pursing them improves student learning beyond a tutor that does not. Others have recommended or experimented with subgoal reification within tutoring systems or in instruction more generally (e.g., Collins & Brown, 1985; Catrambone, 1998).
3. The **"Introduce Variable"** Strategy: I invented this strategy based on the results of our DFA work. I then also found that this strategy appeared spontaneously in tutorial

dialogs, though was generated by the students rather than encouraged by the tutor (as was discussed in the "Introduced Variable" section of Chapter 2).

4. The "**Convert the problem into an Example to Explain**" strategy: This strategy was not based on either the DFA work or the experienced tutor. Instead, this strategy was inspired by the prior work of others. For instance, Zhu & Simon (1987) suggest a similar strategy of presenting students with worked out examples. See also Renkl (1997), VanLehn (1998), Sweller & Cooper (1985) for related studies of student use of worked examples. Catrambone's (1986) also suggests that learning from examples can be improved by asking questions that focus on the goal structures of the domain. Also, to the degree that this strategy focuses on asking students to explain the quantitative structure in English (though using a structured editor), rather then simply focusing on problem solving, the literature related to self-explanation is also relevant. For instance, Chi (1996), Conati et al. (1997), Mwangi & Sweller's (1998), and Aleven, Koedinger, & Cross (1999) have both proposed and in some cases demonstrated that asking students to explain can be beneficial. Most concretely of all, this strategy also gets at the goal of practicing translation from the algebraic representation to the verbal representation, which Corbett et al. (2000) has shown to be effective for learning symbolization.

Before we compare Ms. Lindquist's individual strategies with those of the experienced human tutor, we will review Ms. Lindquist's ability to dynamically scaffold a problem.

Dynamic Scaffolding without any Tutorial Strategies

In chapter 2, I argued that our experienced human tutor used dynamic scaffolding and tutorial strategies. What if we built a system that used jut dynamic scaffolding? What would that look like? I defined dynamic scaffolding as a three step process that:

1. Diagnosed the student's response
2. Gave positive feedback on the correct portions
3. Focused the dialog on the portions that were incorrect

Normally, the third step was implanted with tutorial strategies, but if we got rid of those, then the system would ask questions for each of the subexpression that the student did incorrectly. Here is a made up example:

> **Tutor:** Hello. Please write an expression for <u>the total pay per week</u>.
>
> **Student:** 5g+7/(30-g)
>
> **Tutor:** No. But, "5*g" is correct for the pay from the grocery, and "30-g" is correct for the hours delivering newspapers. We will come back to the total pay per week. Let me try to break this down for you. Write an expression for <u>the pay from the newspaper</u>?
>
> **Student:**7*(30-g)
>
> **Tutor:** Correct. Now, lets try this question again: Please write an expression for the total pay per week.

The important thing to note is that the tutor focused on the 7*(30-g) subexpression, because that was done incorrectly. If the tutor also used a tutorial strategy, the tutor would have added the 2 or 3 steps of that tutorial strategy (e.g., the compute, articulate and generalize steps). Next, we examine the tutorial strategies Ms. Lindquist uses.

Dynamic Scaffolding with Tutorial Strategies

We start by reviewing the four general tutorial strategies (KCDs) that Ms. Lindquist is capable of using. Ms Lindquist's behavior is quite rich and cannot be fully explained by a single example. For each strategy, I will show other examples to demonstrate the particular features of Ms. Lindquist, such as sub-dialogs, positive feedback, negative feedback, buggy messages, etc. In order to prevent the reader from too much context switching, the first example in each section will be with the same "bike-trip" problem. For each example, I will illustrate what the crucial *agenda* data structure looks like at important moments. Each section will close by showing the hint messages Ms. Lindquist uses for each question used in the strategy.

Strategy 1: "Concrete Articulation"

We now look back to an example from the experienced tutor's protocol discussed in Chapter 2 and compare the behavior of Ms. Lindquist on a similar set of simulated student inputs.

Experienced Human Tutor **Exact Transcript**	**The Ms. Lindquist Tutor** **Similar simulated input.**
240. Student: [reads the "bike-trip" problem] Cathy took a "m" mile bike ride. She rode at a speed of "s" miles per hour. She stopped for a "b" hour break. Write an expression for how long the trip took.	T1: [The "bike-trip" problem was been presented] Please write an expression for the total time the trip took.
241. S: uhm [Writes "s/m+b"]	S1: s/m+b
242. Tutor: How do you calculate the amount of time it takes you? If you're, if you're, if you're riding at, let's make it simple. If you are riding at 20 miles per hour, OK, and you go 100 miles, how many hours did that take you?	T2: No. Let me ask you an easier question. Please compute the time actually on the bikes if the speed was 20 miles per hour, and the distance ridden was 100 miles?
243. S: Umh 5	S2: 5
244. T: 5. And how did you get that 5? How did you use the numbers 100 and ?[trails off]	T3: Correct. What math did you do to get that five?
245. S: 100 miles divided by miles per hour.	S3: 100/20
246. T: So you took the miles and divided it by the [garbled, but probably "speed"]	T4: Correct. Now write your answer of 100/20 using the variables given in the problem (i.e., put in 'm' and 's')?
247. S: Miles divided by 's' plus 'b' equals time [writes m/s+b]	S4: b+m/s
248. T: Right.	T5: Correct.

Table 3: An experienced tutor's transcript and Ms. Lindquist's attempt to model it.

The similarities are strong and are found at two important levels. The first similarity is that they both engage in what we have called "dynamic scaffolding" by diagnosing where the student made an error and then asking questions focused on that goal. Both the human and the computer ignore the portion that was done correctly (i.e., the adding of the "b" hour break to the time actually on the bikes) and instead focuses on the subgoal that were not done correctly (i.e., the "s/m" portion that should have been "m/s".) The second similarity is that both tutors used what we have called the concrete articulation strategy. This strategy involves three steps that ask the student 1) to compute a value for a concrete instance, then 2) to articulate what mathematics they did, and then finally 3) to generalize that using the variables from the problem. Notice that the question at T4 is looking for the student to say "m/s" but since the student's answer reveals more knowledge than was expected, the system responds intelligently (popping multiple goals from the agenda). Overall, there is a good match between Ms. Lindquist and the experienced human tutor.

Ms Lindquist can do much more then just replicating this dialog. Next, we will look at 3 examples of Ms. Lindquist working with real students. The first example shows the same bike-ride problem but with different student input. The second example shows that, not only, can Ms. Lindquist break down the initial symbolization question, but she can also break down the new questions she asks students. This second example also shows embedded negative feedback as well as a hint. The third example is the longest and shows three embedded sub-dialogs, including ways to break down a QCOMPUTE question. This last example also shows a more extensive use of the agenda data structure as well as an example of how Ms Lindquist can gracefully deal with a student who did not understand what he was being asked to do. Given that human tutors are often unsure of what the student is saying, its important to have a system robust enough that is can say something intelligent and useful; the last two examples show that Ms. Lindquist can be helpful even if the tutor doesn't initially understand the student's answer (their because the student asks for help or because the student says something un-interpretable).

Here is the first example, showing a real student[28] working on the same problem[29], but with different student inputs.

1. <Problem_Statement>You go on a bike ride. You ride at a speed of "s" miles per hour. You bike "m" miles. You take a "h" hour break. Write an expression for how long the trip took. </Problem_Statement>
2. <Tutor time="[12/7/2000][19:32:20][0]" qtype="QSYMB" goal="h+m/s">Hello. Please write an expression for the total time the trip took. </Tutor>
3. <Student time="[12/7/2000][19:33:20][60]">**s*m+h**</Student>
4. <Tutor time="[12/7/2000][19:33:20][0]" qtype="QCOMPUTE" goal="5"> No. We will come back to the total time the trip took. Let me try to break this down for you. Please compute the time actually on bikes if the speed ridden was "20" miles per hour, and "the distance ridden" was "100" miles? </Tutor>
5. <Student time="[12/7/2000][19:33:37][17]">**5**</Student>
6. <Tutor time="[12/7/2000][19:33:37][0]" qtype="QEXPLAIN" goal="100/20">Correct. Using only 100, and 20 write an expression for the time actually on bikes.</Tutor>
7. <Student time="[12/7/2000][19:33:49][12]">**100/20**</Student>
8. <Tutor time="[12/7/2000][19:33:49][0]" qtype="QGENERALIZE" goal="m/s">Correct. Now, say that again but this time use the variables from the problem. </Tutor>
9. <Student time="[12/7/2000][19:34:3][14]">**m/s**</Student>
10. <Tutor time="[12/7/2000][19:34:3][0]" qtype="QSYMBWRAPUP" goal="h+m/s">Correct. Now, lets try this question again: Please write an expression for the total time the trip took. </Tutor>
11. <Student time="[12/7/2000][19:34:15][12]">**m/s+h**</Student>

---

[28] This is from a real student using Ms. Lindquist on the web site. This student identified herself as a seventh-grader taking pre-algebra and that, "My math teacher told my class that this would be a good site to help us."

[29] You can notice slight wording differences from the previous example including the use of "h" instead of "b" for the time of the break. This is due to minor changes in the version of the software.

```
12. <Tutor time="[12/7/2000][19:34:15][0]" qtype="NIL" goal="NIL">Correct. A new
    problem is needed </Tutor>
```

This student had a similar difficulty in trying to figure out how to calculate the time actually on bikes, so Ms. Lindquist again focused on that portion. In this example, we see that the student succeeded in doing the compute (line 5), articulate (which the log files record as "QEXPLAIN" question types as in line 7) and generalize (line 9) questions correctly.

In this second real example, we will see Ms Lindquist working with a student[30] on a different problem. Note the embedded sub-dialog (line 10), as well as the embedded negative feedback (line 8) and the hints at the end of lines 8 and 12.

```
1.  <Problem_Statement>Jane is "j" years old. Peg is "p" years old. Mary is Peg's
    age minus Tom's age. Tom is 3 years younger than Jane. Peg is "x" years older
    than Mark. Write an expression for Mary's age. </Problem_Statement>
2.  <Tutor time="[7/18/2000][9:22:58][0]" qtype="QSYMB" goal="p-(j-3)">Hello.
    Please write an expression for Mary's age. </Tutor>
3.  <Student time="[7/18/2000][9:25:21][143]"> Can you give me some help
    please![31]</Student>[This means the student hit the help button.]
4.  <Tutor time="[7/18/2000][9:25:21][0]" qtype="QCOMPUTE" goal="23"> Let me try to
    give you some help. We will come back to Mary's age. Let me try to break this
    down for you. Please compute Mary's age if Jane's age was "20" years, and
    "Peg's age" was "40" years? </Tutor>
5.  <Student time="[7/18/2000][9:28:10][169]">23</Student>
6.  <Tutor time="[7/18/2000][9:28:10][0]" qtype="QEXPLAIN" goal="40-(20-
    3)">Correct. Using only 40, 20, and 3 write an expression for Mary's
    age.</Tutor>
7.  <Student time="[7/18/2000][9:30:16][126]">(20-3)-40</Student>
8.  <Tutor time="[7/18/2000][9:30:16][0]" qtype="QEXPLAIN" goal="40-(20-3)"> No.
    Your answer of "(20-3)-40" equals "-23", but you wanted to write an expression
    that equaled "23". Using only 40, 20, and 3 write an expression for Mary's
    age. Hint: Use 40 for Peg's age and 20-3 for Tom's age. </Tutor>
9.  <Student time="[7/18/2000][9:31:4][47]">(40-20)+3</Student>
10. <Tutor time="[7/18/2000][9:31:4][0]" qtype="QEXPLAIN" goal="20-3"> Hmm. No.
    Let me ask you a different question. Using only 20, and 3 write an expression
    for Tom's age.</Tutor>
11. <Student time="[7/18/2000][9:31:19][15]">20-3</Student>
12. <Tutor time="[7/18/2000][9:31:19][0]" qtype="QEXPLAIN" goal="40-(20-
    3)">Correct. Now, lets try this question again: Using only 40, 20, and 3
    write an expression for Mary's age. Hint: Use 40 for Peg's age and 20-3 for
    Tom's age. </Tutor>
13. <Student time="[7/18/2000][9:33:36][138]">40-(20-3) </Student>
14. <Tutor time="[7/18/2000][9:33:37][0]" qtype="QGENERALIZE " goal="p-(j-
    3)">Correct. Now, say that again but this time use the varaibles from the
    problem. </Tutor>
15. <Student time="[7/18/2000][9:34:2][25]">p-(j-3)</Student>
16. <Tutor time="[7/18/2000][9:34:2][0]" qtype="NIL" goal="NIL">Correct. A new
    problem is needed </Tutor>
```

In this example, the student knows how to compute a concrete instance (line 5), but has trouble articulating a correct answer on lines 7. Even though the answer was wrong, the system understood[32] it. Therefore, the system gave a specific type of buggy feedback that is particular to an articulation question (i.e., "Your answer of ’(20-3)-40’ equals ’-23’, but you wanted to write an

---

[30] This dialog was collected as part of the experiment described in Chapter 5.

[31] When a student hits the "help" button, they get virtually the same thing they would have gotten as if they had said something uninterpretable. The one difference is that the tutor does not say "No" and instead says, "Let me try to give you some help." Therefore, this help button is there mainly to not make the student feel bad if they need help.

[32] The system prefixes any response with "Hmm" to indicate it did not understand (i.e., model-trace) the input. In this case I mean that the system knew that the student got the "20-3" portion correct.

expression that equaled ’23”.) The student tried again at line 9 but again showed articulation difficulties, and this time in a way the tutor could not trace (as indicated by the "Hmm") so there was no specific response possible to this specific answer; therefore, the system broke down this step of the KCD and asked the student to explain how to get Tom’s age. Once this question was asked the student got the rest of the question on the agenda correct (line 11, 13 and 15).

In this third and final example, [33] we see a new type of sub-dialog. The last example showed that Ms Lindquist could break down a QEXPLAIN question with a sub-dialog. This example shows two instances of the tutor breaking down a QCOMPUTE question (lines 6 and 12). Line 8 also shows an interesting example of the tutor realizing that the student did not understand the directions well enough (to write a concrete instance as opposed to using variables), and therefore told the student and aspect that made his answer wrong. Also note the way the system inserts text to ensure dialog coherence. She marks when a new question is being asked (line 4, 6 and 12), as well as when the student is going to be asked a question for a second time, but there has been an intervene dialog that could cause the student to forget that he has already seen this question (lines 10 and 14).

1. <Problem_Statement>Michael Jordan was checking to see how fast he could run. Michael checked his time running the "x" meters down the length of a basketball court and found that it took him 5 seconds. The then checked his time running around the bases of a baseball diamond, that has a total distance of "y" meters and it took him 30 seconds. He knew he was slower running around the bases because he had to turn corners and he just wasn’t having as much fun as on the basketball court! Write an expression to tell Michael how much slower he is running around the bases then on the basketball court in meters per second.</Problem_Statement>
2. <Tutor time="[7/24/2000][19:29:43][1]" qtype="QSYMB" goal="x/5-y/30">Hello. Please write an expression for how much slower he is on the baseball diamond. </Tutor>
3. <Student time="[7/24/2000][19:30:42][60]">**[Student hit return]**</Student>
4. <Tutor time="[7/24/2000][19:30:43][0]" qtype="QCOMPUTE" goal="3"> Hmm. No. We will come back to how much slower he is on the baseball diamond. Let me try to break this down for you. Please compute how much slower he is on the baseball diamond if the distance around a baseball diamond was "120" meters, and "the length of a basketball court" was "35" meters? </Tutor>
5. <Student time="[7/24/2000][19:30:44][2]">**[Student Hit return again]**</Student>
6. <Tutor time="[7/24/2000][19:30:44][0]" qtype="QCOMPUTE" goal="7"> Hmm. No. Let me ask you a different question. Please compute his speed running on the basketball court if the length of a basketball court was 35 meters? </Tutor>
7. <Student time="[7/24/2000][19:31:28][43]">**x+y** *[supposed to be 35/5=7]* </Student>
8. <Tutor time="[7/24/2000][19:31:28][0]" qtype="QCOMPUTE" goal="35/5">It looks like you didn’t understand what I was asking you. For this question, you should not use variables: For instance, use 120 here instead of "y". Try this question again: Please compute his speed running on the basketball court if the length of a basketball court was 35 meters? </Tutor>
9. <Student time="[7/24/2000][19:32:56][88]">**7**</Student>
10. <Tutor time="[7/24/2000][19:32:56][0]" qtype="QCOMPUTE" goal="3">Correct. Now, lets try this question again: Please compute how much slower he is on the baseball diamond if the distance around a baseball diamond was "120" meters, and "the length of a basketball court" was "35" meters? </Tutor>
11. <Student time="[7/24/2000][19:33:32][37]">**120/35**</Student>
12. <Tutor time="[7/24/2000][19:33:32][0]" qtype="QCOMPUTE" goal="120/30"> Hmm. No. Let me ask you a different question. Please compute his speed running around the baseball diamond if the distance around a baseball diamond was 120 meters? </Tutor>
13. <Student time="[7/24/2000][19:37:11][218]">**4**</Student>

---

[33] Collected under the conditions described in Chapter 5.

```
14. <Tutor time="[7/24/2000][19:37:11][0]" qtype="QCOMPUTE" goal="3">Correct.  Now,
    lets try this question again:  Please compute how much slower he is on the
    baseball diamond if  the distance around a baseball diamond was "120" meters,
    and "the length of a basketball court" was "35" meters?  </Tutor>
15. <Student time="[7/24/2000][19:37:30][19]">3</Student>
16. <Tutor     time="[7/24/2000][19:37:30][0]"     qtype="QEXPLAIN"     goal="35/5-
    120/30">Correct. Using only 35, 5, 120, and 30 write an expression for how much
    slower he is on the baseball diamond.</Tutor>
17. <Student time="[7/24/2000][19:38:14][43]">35/5-120/30</Student>
18. <Tutor   time="[7/24/2000][19:38:14][0]"   qtype="QGENERALIZE   "   goal="x/5-
    y/30">Correct. Now, say that again but this time use the variables from the
    problem. </Tutor>
19. <Student time="[7/24/2000][19:38:53][39]">x/5-y/30</Student>
20. <Tutor  time="[7/24/2000][19:38:53][0]"  qtype="NIL"  goal="NIL">Correct.  A  new
    problem is needed </Tutor>
```

In this example, the student hits the return key and Ms Lindquist breaks the problem down using the concrete articulation strategy. The student thinks for 60 seconds and then hits return. Because Ms. Lindquist was not sure what portion of the problem is causing the difficulty she decided to focus on the top-level goal and changes the agenda to look like the following: (We use "How much slower he is" as shorthand for "the differences in speed between the basketball court and the baseball diamond.")

QCOMPUTE: Quantity= "How much slower he is": Correct Answer="3"
QEXPLAIN: Quantity="How much slower he is": Correct Answer="35/5-120/30"
QGENEALIZE: Quantity="How much slower he is": Correct answer="x/5-y/30"

But, the student immediately makes another "error" (this time the student hits the return key after just 2 seconds) when asked to compute a concrete instance. Since Ms. Lindquist knows how to break down this step, she does so (line 6) by asking the student to compute one of the sub-quantities (i.e., the speed on the basketball court). At this point, she could have also pushed onto the agenda the question to ask about the speed on the baseball diamonds as well, but we made the design choice not to do this; instead, Ms. Lindquist holds in reserve this question and can still use it if she needs to later. The agenda now looks like this:

QCOMPUTE: Quantity= "Speed on the basketball court": Correct Answer="7"(i.e., 35/5)
QCOMPUTE: Quantity= "How much slower he is": Correct Answer="3"
QEXPLAIN: Quantity="How much slower he is": Correct Answer="35/5-120/30"
QGENEALIZE: Quantity="How much slower he is": Correct answer="x/5-y/30"

The student answers with "x+y", but the system knows that he should be using numbers and not variables to answer this question. This is an example of buggy feedback since the agenda was unchanged but the student was told something about his answer. At line 9, the student correctly figures out his speed on the basketball court so the system pops the top question off of the agenda and gives the student another chance to state the difference in his speeds. To this the student says ’120/35’, which is wrong. The question that we said the tutor held in reserve is now added to the agenda, and the student must compute his speed on the baseball diamond. The agenda now looks like this:

QCOMPUTE: Quantity= "Speed on the baseball diamond": Correct Answer="4"(i.e., 120/30)
QCOMPUTE: Quantity= "How much slower he is": Correct Answer="3"
QEXPLAIN: Quantity="How much slower he is": Correct Answer="35/5-120/30"
QGENEALIZE: Quantity="How much slower he is": Correct answer="x/5-y/30"

From here on out the student does well and answers each of the remaining questions correctly, and the system pops each one off the agenda.

To summarize, we see that the system is capable of combining buggy feedback (line 8) with KCDs to provide embedded sub-dialogs (lines 4, 6 and 12). I argue that these embedded sub-dialogs are similar to the way the experienced human tutor would break problems down when the student was having trouble.

The final aspect of the concrete articulation strategy that we have not discussed is the hints given. Through out this chapter we will present the hints instantiated with the problem we first started with. The following are the hints for the three steps in the concrete articulation strategy. We present two versions, one for each of the main quantities to be computed. Notice that the first hint is the initial representation of the question. The last hint needs to make sure it gives enough information to make sure the student will not be stuck. We have decided to use a multiple-choice question as our last hint to try to deter students from being lazy. You can notice that the hints are written with knowledge of the fact that Ms. Lindquist will try to break down problems if possible. This is demonstrated by the fact that the 3rd hint for the "total time the trip took" refers to the fact that the time actually on bikes is 5 hours. It is guaranteed that the student will already have figured out that the hint can confidently state that to be the case.

Also notice that the generalization hints are written to make them appear more like natural dialog with the use of the pronoun "that" (i.e., "Say that again …") to refer to an expression that the system just told the student were correct (because a Generalization question always follows after a student correctly articulated an answer).

Compute Question	
The time actually on bikes	**The total time the trip took**
Please compute the time actually on bikes if the speed ridden was "20" miles per hour, and "the distance ridden" was "100" miles?	Please compute the total time the trip took if the speed ridden was "20" miles per hour, the distance ridden was "100" miles, and "the hours for break" was "2" hours?
<Repeat question and add> Hint: Use the fact that the distance ridden is 100 miles and the speed ridden is 20 miles per hour.	If the speed ridden was "20" miles per hour instead of "s", the distance ridden was "100" miles instead of "m", and the hours for break was "2" hours instead of "h" what would the total time the trip took be?
<Repeat question and add>Hint: It is one of these 6 choices: "100-20", "20-100", "100+20", "100*20", "100/20", or "20/100".	<Repeat question and add> Hint: Use the fact that the hours for break is 2 hours and the time actually on bikes is 100/20=5 hours.
	<Repeat question and add> Hint: It is one of these 6 choices: "2-5", "5-2", "2+5", "2*5", "2/5", or "5/2".
Explain (or Articulation) Question	
The time actually on bikes	**The total time the trip took**
Using only 100, and 20 write an expression for the time actually on bikes.	Using only 2, 100, and 20 write an expression for the total time the trip took.
<Repeat question and add>Hint: Use 100 for the distance ridden and 20 for the speed ridden.	Hint: Use 2 for the hours for break and 100/20 for the time actually on bikes.
<Repeat question and add>Pick one of these 6 choices: "100-20", "20-100", "100+20", "100*20", "100/20", or "20/100"	<Repeat Question and add> Hint: Use 2 for the hours for break and 100/20 for the time actually on bikes. Pick one of these 6 choices: "2-100/20", "100/20-2", "2+100/20", "2*100/20", "2/(100/20)", or "100/20/2"
Generalize Question	
The time actually on bikes	**The total time the trip took**
Now, say that again but this time use the variables from the problem.	Now, say that again but this time use the variables from the problem.
Rewrite 100/20 but in place of 20 put in "s", and in place of 100 put in "m"?	Rewrite 2+100/20 but in place of 20 put in "s", in place of 100 put in "m", and in place of 2 put in "h"?[sic]

In this section, we saw some rich tutoring behavior that we have argued maps well to the concrete articulation strategy that our experience human tutor used. That includes both dynamic scaffolding, hinting, positive and negative feedback, and embedded sub-dialogs. We will now look at some of the other tutorial strategies.

Strategy 2: "Explain in English First"

The "Explain in English First" strategy has two steps.

- First explain in English (using pull down menus) how they would achieve the goal
- Follow that up by symbolizing that subgoal

We use dynamic scaffolding to identify which of the goals need assistance, and then ask the two questions for each sub-goal. When there is more than one error, we have chosen to proceed in a bottom up manner.

This first example shows how this strategy plays out on the familiar bike-trip problem. We will follow that with a second example that also shows a related knowledge remediation strategy for a particular error type.

This first example[34] shows an instance of a student making an error only at the top-level goal (i.e., using "-" instead of "+").

```
1) <Problem_Statement> You go on a bike ride.  You ride at a speed of "s" miles per
   hour.  You bike "m" miles.  You take a "h" hour break.  Write an expression for how
   long the trip took. </Problem_Statement>
2) <Tutor time="[12/5/2000][8:10:13][13]" qtype="QSYMB" goal="h+m/s">Hello.  Please
   write an expression for the total time the trip took. </Tutor>
3) <Student time="[12/5/2000][8:11:18][65]">m/s-h</Student>
4) <Tutor time="[12/5/2000][8:11:18][0]" qtype="QEXPLAIN_VERBAL" goal="h+m/s"> No.
   But, "m/s" is correct for the time actually on bikes.  We will come back to the
   total time the trip took.  Let me try to break this down for you. Can you explain
   how you would find the total time the trip took?</Tutor>
5) <Student time="[12/5/2000][8:13:9][111]">the total time the trip took is equal to
   the time actually on bikes plus the hours for break.</Student>
6) <Tutor          time="[12/5/2000][8:13:9][0]"          qtype="Q_SYMB_AFTER_VERBAL"
   goal="h+m/s">Correct.  Now, follow that up by writing an expression for the total
   time the trip took</Tutor>
7) <Student time="[12/5/2000][8:13:28][19]">m/s+h</Student>
8) <Tutor time="[12/5/2000][8:13:28][0]" qtype="NIL" goal="NIL">Correct. A new problem
   is needed </Tutor>
```

We see at line 3 the student correctly symbolize the time actually on bikes but added the hours for the break. Ms. Lindquist responds by confirming the portion that was done correct and then asking the student to explain in English the top-level goal (i.e., the total time the trip took.) For this purpose, the student is presented with a fill-in-the-blank question that begins "The total time the trip took is equal to". This question is followed by three choice-menus. The first and second choice menus contained the five phases describing quantities from the problem, as listed here:

The phrases in the 1st and 3rd Menus
The actual time on bikes
The distance ridden
The hours for break
The time actually on bikes
The total time the trip took

The second choice menu contained these operation descriptors.

The phrases in the 2nd menu
Plus
Minus
Times
Divided by

At line 5 we see that the student correctly explained how to compute the total time, so Ms. Lindquist then asked him to say the same thing but in symbols, which the student did correctly.

Here is a second example[35] of the "Explain in English" KCD that also shows a KRD (line 4) triggered by a certain type of error.

```
1) <Problem_Statement>Michael starts a business selling lemonade.  He spent 35 dollars
   buying supplies like lemons, pitchers, and cups.  He sells a 16 ounce glass of
   lemonade for 2 dollars.  If he sells "g" glasses of lemonade, how much profit will
   he end up making?</Problem_Statement>
2) <Tutor time="[7/18/2000][9:13:53][0]" qtype="QSYMB" goal="2*g-35">Hello.  Please
   write an expression for the profit he makes. </Tutor>
3) <Student time="[7/18/2000][9:14:37][44]">2*g</Student>
```

---

[34] This example is with a student who said he (or she) was a sixth grader who learned about Ms. Lindquist from "A Google search".

[35] Collected under the conditions described in Chapter 5.

```
4)  <Tutor time="[7/18/2000][9:14:37][0]" qtype="QREPRESENT" goal="the amount of money
    he collects"> No. That is part of the answer. Tell me this about your answer. What
    does 2*g represent in words?</Tutor>
5)  <Student time="[7/18/2000][9:15:7][30]">the amount of money he collects</Student>
6)  <Tutor  time="[7/18/2000][9:15:7][0]"  qtype="QSYMB"  goal="2*g-35">Correct.   Now,
    lets try this question again:  Please write an expression for the profit he makes.
    </Tutor>
7)  <Student time="[7/18/2000][9:15:22][15]">2*g+g</Student>
8)  <Tutor time="[7/18/2000][9:15:22][0]" qtype="QEXPLAIN_VERBAL" goal="2*g"> Hmm.  No.
    We will come back to the profit he makes.  Let me try to break this down for you.
    Can you explain how you would find the amount of money he collects?</Tutor>
9)  <Student time="[7/18/2000][9:17:2][100]">the amount of money he collects is equal
    to  the  price  for  a  glass  of  lemonade  times  the  number  of  glasses  he
    sells.</Student>
10) <Tutor          time="[7/18/2000][9:17:2][0]"          qtype="Q_SYMB_AFTER_VERBAL"
    goal="2*g">Correct. Now, follow that up by writing an expression for the amount of
    money he collects</Tutor>
11) <Student time="[7/18/2000][9:17:25][23]">g*2-35</Student>
12) <Tutor time="[7/18/2000][9:17:25][0]" qtype="NIL" goal="NIL">Correct. A new problem
    is needed </Tutor>
```

The student starts out with a typical production error (that we called a "Sub" error type in Chapter 1) common for students who do not have much practice at writing composed expression. However, it is also possible that the student simply failed to understand that his goal was to determine the total profit. Therefore, Ms. Lindquist used a knowledge remediation dialog that is specific to this type of error. Specifically, Ms. Lindquist asked "`That is part of the answer. Tell me this about your answer. What does 2*g represent in words?`" This is similar to the occasions when the experienced tutor asked the student to translate from the algebra back into English (lines 24,26, 204, 210, 212, 264). The student is presented with a choice menu to select the appropriate noun phase from a list of 5 noun phases that are occur in that problem. The student correctly chooses "`the amount of money he collects.`" Then the student is asked to retry the original problem but fails. Ms. Lindquist does not understand (as indicated by first saying "Hmm") the student's response of "2g+g" and therefore, next asks the student to explain in English how to achieve both subgoals. First, Ms. Lindquist asks the student to explain in English how to compute the "the amount of money he collects". This is very similar to what the experienced human does at line 196, asking the student to explain how to find average speed. Ms. Lindquist presents the student with a sentence and the three multiple choices slots. Ms. Lindquist's agenda now (i.e., at line 8) looks like the following:

Qexplain: answer=" `the amount of money he collects is equal to the price for a glass of lemonade times the number of glasses he sells."`
Qsymbolize: answer="2g"
Qexplain: answer=" `The profit he makes is equal to the amount of money he collects minus the amount he spends on supplies."`
Qsymbolize: answer="2g-35"

The student then explains (line 9), on his first attempt, how to find "`The amount of money he collects`". The student is then presented with a prompt that asks the student to write an algebraic expression for the amount of money collected. Instead of writing the correct answer of "2g", the student writes "2g-35". Since this answer is correct, Ms. Lindquist removes the intervening questions she was planning to ask next (explain "the profit he makes"). She then confirms the answer and then, realizing there are no other questions on the agenda, moves onto the next problem.

The final aspects to discuss for this strategy are the hints for these two questions.

"Explain in English First" Question	
The time actually on bikes	**The total time the trip took**
Can you explain how you would find the time actually on bikes?.	Can you explain how you would find the total time the trip took?
Figure out some way of using both the distance ridden and the speed ridden.	Figure out some way of using both the hours for break and the time actually on bikes.
The answer is that the time actually on bikes is equal to the distance ridden divided by the speed ridden.	The answer is that the total time the trip took is equal to the hours for break plus the time actually on bikes.
Q_SYMB_AFTER_VERBAL **Symbolize after an "Explain In English First" Question**	
The time actually on bikes	**The total time the trip took**
Now, follow that up by writing an expression for the time actually on bikes	Now, follow that up by writing an expression for the total time the trip took
You already told me that the time actually on bikes is equal to the distance ridden divided by the speed ridden. Can you say that using the symbols from the problem statement?	You already told me that the total time the trip took is equal to the hours for break plus the time actually on bikes. Can you say that using the symbols from the problem statement?
One answer is m/s. Type that.	One answer is h+m/s. Type that.

Strategy 3: "Introduce a New Variable"

A third strategy that Ms. Lindquist uses also appears to match rather well what the experienced human tutor protocol shows. We again will start with an example[36] of a student doing the same bike-trip problem.

```
1)  <Problem_Statement>You go on a bike ride.  You ride at a speed of "s" miles per
    hour.  You bike "m" miles.  You take a "h" hour break.  Write an expression for how
    long the trip took. </Problem_Statement>
2)  <Tutor time="[12/6/2000][4:43:8][1]" qtype="QSYMB" goal="h+m/s">Hello.   Please
    write an expression for the total time the trip took. </Tutor>
3)  <Student time="[12/6/2000][4:45:10][122]">m/(s+h)</Student>
4)  <Tutor time="[12/6/2000][4:45:10][0]" qtype="QSUB_QUAN_STRICTLY" goal="m/s"> Hmm.
    No.   We will come back to the total time the trip took.  Let me try to break this
    down for you. What is the time actually on bikes?37</Tutor>
5)  <Student time="[12/6/2000][4:46:15][65]">m/s</Student>
6)  <Tutor        time="[12/6/2000][4:46:15][0]"        qtype="QSUB_QUAN_NOT_STRICTLY"
    goal="h+B">Correct.  If "B" was the time actually on bikes then what would be the
    total time the trip took? </Tutor>
7)  <Student time="[12/6/2000][4:46:53][38]">B+h</Student>
8)  <Tutor  time="[12/6/2000][4:46:53][0]"  qtype="QSUBSTITUTE"  goal="h+m/s">Correct.
    Now, lets try this question again:  You are correct that "h+B" is the total time
    the trip took.  Now, can you state this in terms of the numbers given in the
    problem?</Tutor>
9)  <Student time="[12/6/2000][4:47:37][44]">m/s+h</Student>
10) <Tutor time="[12/6/2000][4:47:37][0]" qtype="NIL" goal="NIL">Correct. A new problem
    is needed </Tutor>
```

The idea behind this strategy is that if you want a student to symbolize an expression (i.e., "m/s+h") and that expression involves more than one operation (Since "m/s+h" has both division and addition) you can break these down into two steps. However, before the first step, you need

---

[36] This example comes from the web site. The student identified himself or herself as being in sixth grade and who heard about Ms. Lindquist through the "Netscape search engine."

[37] Notice that this question generated using just dynamic scaffolding and has no tutorial strategy associated with this. An object of future work will be to figure out the best way of combine these strategies. One idea is to use "Concrete Articulation" strategy for expression that involves just one operator, and to use the "Introduced Variable" strategy for all large (one than one operator) expression components.

to make sure that the components (i.e., "m/s" and "h") have been symbolized (we could ask the student "What represents the time of the break?" but we don't bother asking of expressions that have just one literal). The two steps are then as follows.

3. Introduce variables to stand for any components (line 6)
4. Then ask the student to do the symbolic substitution to remove the introduced variables (line 8)

This procedure is similar to what your experienced human tutor did as we argued in Chapter 2. Consider the following fragment from the experienced human protocol.

327. STUD [reads problem] A car salesperson is paid a base salary of $200 per month plus the additional amount of money in commissions for each car she sells. She sold four cars last month and received "x" dollars that month. If she sells "h" cars this month, write an expression showing how much she earns this month. What's her commission?
328. TUTOR Good question.
329. STUD Oh I have to find this out don't I
330. TUTOR I think so
331. STUD Um x [minus] that
332. STUD It doesn't say how many cars. Ah 4 cars last month so it would be x-200 divided by 4
333. [he puts in the parenthesis only after writing "x-200/4"]
334. TUTOR OK
335. [he has written (x-200)/4 = commissions" and then adds a "c" before the commission apparently to indicate that commission's will have the variable "c" stand for it]
336. STUD Equals commissions or "c". So this month would be 200+c*4 = salary
337. [writes 200+c*4=salary]
338. TUTOR This month?
339. STUD Yeah
340. TUTOR Read it again.
341. STUD This month, or she sold h cars
342. [scratches out the 4 and replace it by "h"]
343. TUTOR You want to skip a line so you aren't messy
344. [starts writing "200+c*h" and tutor interrupts after the "c" to say
345. TUTOR OK now is C something that was given to you?
346. STUD No
347. TUTOR Where did it come from? What numbers
348. STUD So have to do this [scratches out the "200+c*h" he just wrote]
349. TUTOR You have to rewrite the WHOLE expression.
350. [writes 200+(x-200)/4*h=salary]
351. TUTOR Good

The student, without any prompting, introduces a variable ("c") to stand for the commission. He writes "c=(x-200)/4" which is correct. The student then says that the final answer is "200+c*4" but the correct answer would be "200+c*h". The tutor prompts the student with, "This month?" and then tells him to read the problem again. The tutor's comments indicate that the tutor understood and followed the student's use of this variable since the tutor knew how to debug the student error by asking "this month?" that focused on the one error in the expression. The student attempts it again and gets it correct with "200+c*h". The tutor then prompts for the substitution step (line 345) by reminding the student that the "C" was a variable he introduced.

Our final example is of a real student from our experiment reported in Chapter 5. Take note to the dynamic scaffolding and the positive feedback at line 4, followed by the second step in the strategy at line 6.

1.<Problem_Statement>Adam, Stan and Doug spent the day picking strawberries. Adam picked twice as much as Stan and Doug picked 7 more kilograms than Adam. Suppose Stan picked "s" kilograms of strawberries. Stan and Doug cooperated to make a pie. How many kilograms of strawberries did Stan and Doug have together?</Problem_Statement>

```
2.<Tutor time="[7/20/2000][9:16:5][0]" qtype="QSYMB" goal="s+(2s+7)">Hello.  Please
   write an expression for the combined number of kilograms for Stan and Doug.
   </Tutor>
3.<Student time="[7/20/2000][9:19:25][201]">7+(2*s)</Student>
4.<Tutor time="[7/20/2000][9:19:25][0]" qtype="QSUB_QUAN_NOT_STRICTLY" goal="s+C">
   No. But, "7+(2*s)" is correct for the number of kilograms Doug picked.  We will
   come back to the combined number of kilograms for Stan and Doug.  Let me try to
   break this down for you.  If "C" was the number of kilograms Doug picked then
   what would be the combined number of kilograms for Stan and Doug? </Tutor>
5.<Student time="[7/20/2000][9:20:42][77]">c+s</Student>
6.<Tutor time="[7/20/2000][9:20:42][0]" qtype="QSUBSTITUTE"
   goal="s+(2s+7)">Correct.  Now, lets try this question again:  You are correct
   that "s+C" is the combined number of kilograms for Stan and Doug.  Now, can you
   state this in terms of the numbers given in the problem?</Tutor>
7.<Student time="[7/20/2000][9:21:26][44]">7+(2*s)+s</Student>
8.<Tutor time="[7/20/2000][9:21:26][0]" qtype="NIL" goal="NIL">Correct.  A new
   problem is needed </Tutor>
```

The hints for first step in this strategy are as follows.

Introduce a variable to stand for a sub-quantity	
The time actually on bikes	**The total time the trip took**
<This step does not apply to one-operator problems since there are no variables to introduce>	If "B" was the time actually on bikes then what would be the total time the trip took?
	What is the total time the trip took? Use "B" instead of "m/s" for the time actually on bikes. Hint: pick one of these: h-B, B-h, h+B, h*B, h/B, or B/h.
The 2nd Step in Introduce Variable Strategy	
The time actually on bikes	**The total time the trip took**
Use "m" and "s" to calculate the time actually on bikes.	Now, lets try this question again: You are correct that "h+B" is the total time the trip took. Now, can you state this in terms of the numbers given in the problem?
What is the time actually on bikes? Hint: pick one of these: m-s, s-m, m+s, m*s, m/s, or s/m.	What is the total time the trip took? Hint: You know the answer is "h+B" but you must substitute in the value for "B". Replace "B" with "m/s". Use parentheses if needed.
	Type h+m/s.

Strategy 4: "Convert the Problem into an Example to Explain"

The strategy we are now about to present did not follow from either our cognitive studies or from the experienced human tutor, but rather from the prior work of other researchers. Zhu and Simon (1987) have argued that a good way to teach a problem-solving skill is through looking at an example integrated with problem-solving. This suggests that if a student is having difficulty, we can turn the problem into an example by telling the student the answer and then asking him to explain the steps. We have taken Catrambone's (1986) suggestion that learning from examples can be improved by asking questions that focus on the goal structures of the domain (see also Renkl (1997) for using worked example.) Chi (1996) and Conati et al. (1997) have both proposed asking students to explain is beneficial. This strategy also gets at the goal of practicing translation from the algebraic representation to the verbal representation, which Corbett et al. (2000) have shown to be effective for learning symbolization. The following is an example with a simulated student.

```
1) <Problem_Statement>You go on a bike ride.  You ride at a speed of "s" miles
   per hour.  You bike "m" miles.  You take a "h" hour break.  Write an
   expression for how long the trip took. </Problem_Statement>
```

2) <Tutor time="[12/14/2000][14:20:38][1]" qtype="QSYMB" goal="h+m/s">Hello. Please write an expression for the total time the trip took. </Tutor>
3) <Student time="[12/14/2000][14:20:52][14]">**m*s-b**</Student>
4) <Tutor time="[12/14/2000][14:20:52][0]" qtype="QEXAMPLE_EXPLAIN" goal="m/s"> Hmm. No. The answer is actually h+m/s. Let me ask you some questions to help you understand it. Can you explain where the m/s came from?</Tutor>
5) <Student time="[12/14/2000][14:21:8][16]">**m/s is equal to the distance ridden divided by the speed ridden.**</Student>
6) <Tutor time="[12/14/2000][14:21:8][0]" qtype="Q_WORKED_REPRESENT" goal="the time actually on bikes">Correct. What does m/s represent?</Tutor>
7) <Student time="[12/14/2000][14:21:18][10]">**the time actually on bikes**</Student>
8) <Tutor time="[12/14/2000][14:21:46][0]" qtype="QEXAMPLE_EXPLAIN" goal="h+m/s"> Hmm. No. Can you explain where the h+m/s came from? Hint: You need to use the hours for break and the time actually on bikes.</Tutor>
9) <Student time="[12/14/2000][14:22:4][18]">**h+m/s is equal to the hours for break plus the time actually on bikes.**</Student>
10) <Tutor time="[12/14/2000][14:22:4][0]" qtype="Q_WORKED_REPRESENT" goal="the total time the trip took">Correct. What does h+m/s represent?</Tutor>
11) <Student time="[12/14/2000][14:22:17][13]">**the total time the trip took** </Student>
12) <Tutor time="[12/14/2000][14:22:17][0]" qtype="NIL" goal="NIL">Correct. A new problem is needed </Tutor>

The reader might be asking if the experienced tutor we observed used this strategy. If you look at the human tutor at one level the answer is "no" since the tutor never told the student what the correct answer was. However, on another level, the tutor did use some similar types of questions as the example above used. In particular, the experienced tutor would ask the student to translate his answers into English (i.e., line 204 "550 represent what?")

The hints for this strategy are as follows:

Translate a mathematics relation to English	
The time actually on bikes	**The total time the trip took**
Can you explain where the m/s came from?	Can you explain where the h+m/s came from?
<repeat question and add> Hint: You need to use the distance ridden and the speed ridden.	<repeat question and add> Hint: You need to use the hours for break and the time actually on bikes.
The answer is the distance ridden divided by the speed ridden.	The answer is the hours for break plus the time actually on bikes.
Translate a mathematics quantity	
The time actually on bikes	**The total time the trip took**
What does m/s represent?	What does h+m/s represent?
<There are no hints since there are usually only a few choice to pick from.>	

While implementing this strategy, we learned that it is very easy to add strategies (or at least this one) to Ms. Lindquist. In fact, it took only about 2-3 hours to add all the productions needed to implements this strategy (See Appendix J for more information). We look forward to testing the extensibility of this architecture by attempting to add other strategies that might be further away in the design space.

Strategy 5: "Cut to the Chase"

Ms. Lindquist is also capable of using what we call the cut-to-the-chase strategy. It is, maybe, a corruption of the word "strategy" since this strategy is simply to tell the student the answer. Not surprisingly, our human tutor never did this. This strategy has the one advantage that it takes less time and thus frees students to do other problems.

Next, I will discuss the 26 different tutorial operators that the experienced human tutor used from my thesis proposal.

Modeling the Tutorial Operators of the Experienced Human Tutor

After I transcribed the protocol of the experienced human tutor, I created a list of "operators" (or tutorial moves) that the human tutor used. These 26 operators were generalizations of the behavior we observed that our experienced tutor used. I took this list of operators as design goals for the building of a system that was capable of using similar "moves". I presented these design goals at my thesis proposal in April 1998. Appendix D shows that exact same list.

Fourteen of these goals I have achieved (Table 4). Six of them I have not (Table 6). An addition, six of them (Table 5) I have made possible, but I have not implemented them. The groups of operators that have been achieved are mainly focused on the tutorial strategies that I have implemented, as well as implementing what I now call dynamic scaffolding. The six operators that were not achieved have different reasons for their failures, and are given in Table 6. The six operators that were made possible, but not implemented, are what I now call *knowledge remediation dialogs,* because they focused on a particular type of error the student could make. I gave such operators lower priority, as they do not occur as often *knowledge construction dialogs.*

In summary, a substantial number of the operators I identified from my human tutor were achieved. Next, I will present a comparison between Ms. Lindquist and traditional model tracing tutors.

Operators that were achieved.
Op1: Concrete Instantiation Analogy
Op3: Encourage the student to generalize.
Op4: Reflection after the correct answer.
Op5: Reflection on, or challenge of, a correct answer if the tutor suspects guessing.
Op7: Requesting that the student recall information by either questioning and or hinting.[38]
Op9: Ask the student to identify the name of a quantity represented by a symbol (or expression). Include follow up clarification if the student is not specific enough. [39]
Op10: Ask a student to identify the symbol (or expression) that represents a quantity.[40]
Op12: Stating the general quantitative relationships in words. This is like generalization, but can occur anywhere in the problem.[41]
Op13: If the student might have forgotten what work he has already accomplished then remind the student what steps they have already completed.[42]
Op14: Positive Feedback on parts that are correct.
Op15: Simple Feedback on an identifiable bug category.
Op16: Ask the student to figure out what sub-goal to set.[43]
Op24: Levels of Specificity
Op25: Be able to differentiate a close answer from a very wrong answer, from an unintelligible one.[44]

Table 4: Operators that were achieved.

Operators that were made possible, but not implemented
Op11: Correct a bug by referring to the implicit semantics about the relative size of numbers.
Op17: Socratic Technique showing a contradiction from a student's error.
Op18: Order of Operations Sub-Dialogue
Op19: Teach students how to do unit analysis.
Op20: Coach the student to realize distractor numbers are not needed.
Op21: Slips and other mistakes that tutors do not dwell on.

Table 5: Operators that were made possible but not implemented.

---

[38] Ms. Lindquist asks questions like this, but it should be noted that we do not model the retrieval of definitions any differently then the way we model the execution of any skill.

[39] This is the KRD for error of omission

[40] This is used in the "Explain in English" strategy as well as the "Worked example" strategy.

[41] This is used in the "Explain in English".

[42] Several of our hints do this.

[43] We can do this but chose not to as explained in the "Explain in English" section.

[44] Right now, we differentiate between an unintelligible response and a wrong response. We indicate this difference with the use of "hmm" for unintelligible response. It is a small change to expand this to include saying something different according the number of errors detected (i.e., "No, But that is close. You have made one mistake.")

Operator	Comments
Op2: Tutor makes reference to dialogue history.	Ms. Lindquist does not keep around a great deal of context, so it cannot pull out of memory a similar example that the student did and make reference to that. However, Ms. Lindquist does pay attention to dialogue coherence. She uses phrases like "No, lets try this question again."
Op6: "Feedforward" (given before the student makes an anticipated mistake)	What would it take to generate such a move? We could be keeping track of the probability that a student will answer a question correctly, and if that is very low then we could use "feedforward". We have not done so because we have not bothered to keep track of probabilities of correctness for each skill. Ms. Lindquist's architecture does not prevent this at all; we just have chosen to focus on aspects that we thought would provide a bigger pedagogical benefit.
Op8: Challenge the student's answer	It would be interesting and possibly to use the probability that the student guesses as way to determining when to challenge the student doing so even sometimes when it is correct.
Op22: Tutor focuses attention on a previous answer that was more correct	We do not model the repair process of how a student takes a wrong answer and changes it to try to repair it. Consequently, our tutor does not say anything like "Getting closer". Since our system already does count the number of errors in an answer, it would be possible to add a tutorial rule that would provide this sort of feedback.
Op23: Tutor comments on the repair the student attempted	See above Op 22. A full treatment of this type of feedback would really require us to model the repair process itself. At present each time the student give an answer we assume that the student did all the processes again to get that answer, but it seems reasonable that when a student is told something is wrong they hypothesize what aspect is wrong, and if we were to detect the change they made we might come up with a good pedagogical response.
Op26: Engage the student to try to diagnosis what the student was thinking	We do not ask such opened questions that are very difficult to understand. We are thinking of adding such natural language understanding capabilities but that will be a very big project. Consequently we try to achieve some of the same objectives with more targeted questions but it might be that such opened questions are a key way that human tutors differ from computer tutors and thus a possible reasons why they are so much more effective.

Table 6: Operators that were not achieved, and the reason for each.

Comparison of Ms. Lindquist Architecture and the Traditional Model Tracing Architecture

Similarities

Ms. Lindquist's architecture is an extension of the traditional Andersonian model-tracing architecture and, therefore, has many things in common with it. Both architectures are designed for coached problem-solving; they are both based on the learning theory that student learn how to solve problems by doing. These systems invite a student to solve a problem and progress through a curriculum using masterly learning. Both architectures are based around a cognitive model of student thinking, using production to model cognitive steps. Both architectures use the model-tracing algorithm to perform plan-recognition on the student's answer. Both systems use buggy rules to model misconnections and common errors. Both architectures use the idea of providing a series of hints.

Differences

However, the two architectures are quite different. The architectures of both are quite different. Figure 3 is a depiction of the model-tracing architecture that the reader can compare to the depiction of Ms. Lindquist's architecture (also shown in Figure 2 of Chapter 3). Probably the most important additions of the Ms. Lindquist architecture are the tutorial strategies (i.e., the KCDs and KRDs). This and other differences are summarized in Table 8, and each will be explored in more detail.

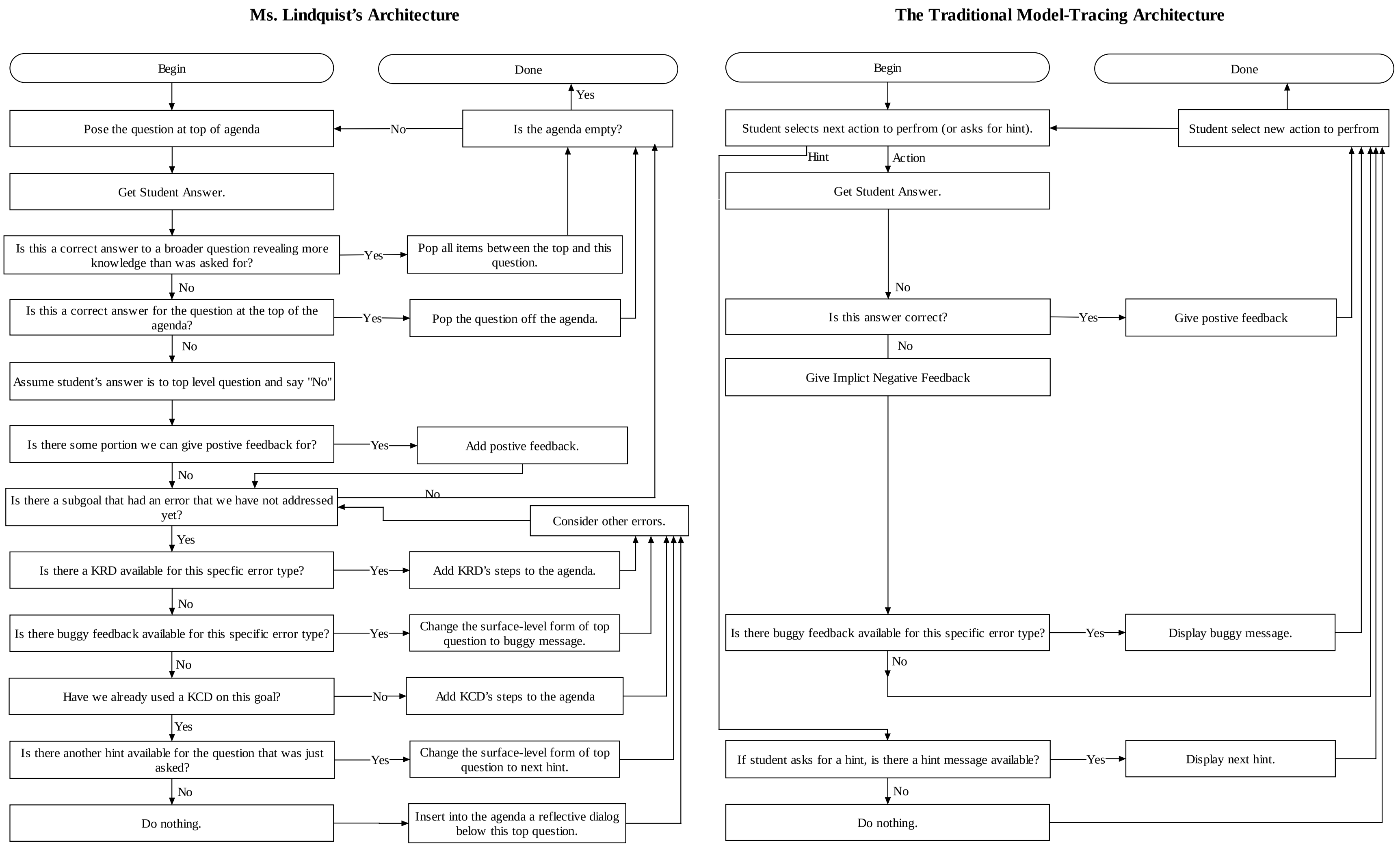

Ms. Lindquist's Architecture
Begin
Done
Yes
Pose the question at top of agenda
No
Is the agenda empty?
Get Student Answer.
Is this a correct answer to a broader question revealing more knowledge than was asked for?
Yes
Pop all items between the top and this question.
No
Is this a correct answer for the question at the top of the agenda?
Yes
Pop the question off the agenda.
No
Assume student's answer is to top level question and say "No"
Is there some portion we can give postive feedback for?
Yes
Add postive feedback.
No
Is there a subgoal that had an error that we have not addressed yet?
No
Consider other errors.
Yes
Is there a KRD available for this specfic error type?
Yes
Add KRD's steps to the agenda.
No
Is there buggy feedback available for this specific error type?
Yes
Change the surface-level form of top question to buggy message.
No
Have we already used a KCD on this goal?
No
Add KCD's steps to the agenda
Yes
Is there another hint available for the question that was just asked?
Yes
Change the surface-level form of top question to next hint.
No
Do nothing.
Insert into the agenda a reflective dialog below this top question.
The Traditional Model-Tracing Architecture
Begin
Done
Student selects next action to perfrom (or asks for hint).
Student select new action to perfrom
Hint
Action
Get Student Answer.
No
Is this answer correct?
Yes
Give postive feedback
No
Give Implict Negative Feedback
Is there buggy feedback available for this specific error type?
Yes
Display buggy message.
No
If student asks for a hint, is there a hint message available?
Yes
Display next hint.
No
Do nothing.

Figure 3:A comparison between Ms. Lindquist's architecture and the traditional model-tracing architecture

	Traditional Model Tracing Architecture	**Ms. Lindquist's Architecture**
1	Provide buggy-feedback or hints to a step.	If the step can be broken down, choose a KCD to scaffold that step. Uses hints only if it is not possible to break down a step.
2	Only one strategy made explicit by the reified interface.	Allows multiple different strategies. Also allows for those strategies to be multi-step.
3	Allows only one error in an input.	Deals with inputs with multiple errors.
4	Student needs to learn a new interface.	Tutor asks questions in natural language.
5	An all-or nothing scaffolding technique.	Just-in-time scaffolding. First, ask the "main" question, followed by provided scaffolding on just the goals missed.
6	No model of dialog: ignores all context (or nearly so).	Has an agenda that holds context information.

Table 8: A comparison of differences between the traditional model-tracing architecture, and Ms. Lindquist's architecture.

The first difference in Table 8, is that Ms. Lindquist has what I have termed *dynamic scaffolding*. This means that the tutor diagnosis the student's answers and gives positive feedback on the correct steps, followed by focusing the dialog on the steps that had errors. "Focusing the dialog on a step" could mean something as simple as just asking a single question about a particular step, but a potentially better approach is to have what we have called *knowledge construction dialogs (KCD).* These KCDs are multi-step plans to help students. These plans are stored on the *tutorial agenda*, which has no analog in the traditional model tracing architecture. The individual steps of a KCD can themselves lead to sub-dialogs. However, traditional model-tracing tutors do not have an ability to ask a new question; instead, they must rely on giving a buggy-feedback messages or letting the student ask for a hint. Therefore, a major difference is that Ms. Lindquist's architecture allows for a much more sophisticated dialog between the student and the tutor (One can plausibly argue that model tracing tutors do not really have a dialog with student since the tutor never asks the student a new question.)

The second difference in Table 8 is that model-tracing tutors make only one strategy explicit by reifying certain aspect in the interface. VanLehn, Freedman & et al. (2000) say that "Model-tracing tutors are sometimes criticized for allowing only one problem solving strategy." VanLehn et al. cites Reiser, Kimberg, Lovett, & Ranney's (1992) criticism of Anderson and Corbett's lisp tutor because it forced students to enter code top-down. Another example of a single scaffolding strategy is the interface of the Carnegie Learning tutor. That tutor was designed to scaffold problem solving by first doing several concrete instances, but there are other ways to scaffold symbolization. For instance, Ms Lindquist provides the "Explain in English" strategy as well as the concrete articulation strategy (the later is more like the Carnegie Learning tutor.) These two strategies are quite different, yet Ms. Lindquist's architecture makes it easy to have different tutorial strategies, and to uses them at any time. In contrast, model-tracing tutors have a single strategy, and are tough to change because the pedagogical responses are embedded inside of the student model.

The third difference has to do with how the system deals with more then one buggy rule applying at a time. All systems built at Carnegie Mellon University have only been able to interpret a student's incorrect answer if their response could be matched to a single buggy feedback rule. I viewed this a serious limitation, and therefore designed Ms. Lindquist 's architecture so that she can handle responses by students that requires multiple different buggy rules in order to trace them. This seems more important for problems that are more complicated, such as the ones Ms. Lindquist was designed for (e.g., four operator problems with answers like "5*g+7*(30-g)"). Being able to deal with multiple errors not only involves changes in the student model and model tracing, but also in the tutorial model, since the tutorial model needs to be able to respond intelligently to more then one error occurring at a time.

The fourth difference is that Ms. Lindquist is designed with natural language dialog in mind, while traditional model-tracing systems encourage the author to use graphical user-interface elements to force the student to display most of their reasoning. For instance, the student might be asked to define variables (Gertner & VanLehn, 2000), label columns (Koedinger et al., 1995), or specify a goal tree (Koedinger & Anderson, 1993; Singley, 1990; Reiser, Beekelaar, Tyle, & Merrill, 1991). Many ITS designers claim that asking students for this information increases student learning; some have found evidence to support this (e.g., Merrill & Reiser, 1994; Singley, 1990). However, one reason to think there might be an advantage to natural language is that students already understand language. Therefore, students do not have to learn a new-user interface for each strategy that the tutoring system employs[45].

The fifth difference is that Ms. Lindquist's architecture can be used to provide for a more natural fading technique, since the tutor starts out by asking for the top-level question. If we compare Ms. Lindquist with the Carnegie Learning tutor, we notice that that system asks a student to fill in many questions per problem (For the worksheet shown in Figure 2, there are 21 questions in 3 columns with 7 questions per column, resulting in 21 questions.) It might be a poor use of student's time to ask this many questions per problem. It might be better to use Ms. Lindquist to first let the student try to answer the problem without assistance. If the student fails, then ask just a few questions focused on the aspects the student got wrong. Ms. Lindquist's ability to engage in sub-dialogs means that the author of the intelligent tutoring system, doesn't have to ask the student a bunch of preliminarily questions and instead can choose to ask those questions, only if the student is having trouble. This might make better use of student's time.

The sixth difference is that traditional model-tracing tutors tend to respond to the same errors in the same manner, regardless of the state of the dialog. This is due to the fact that model-tracing tutors do not model dialog at all. On the other hand, Ms. Lindquist has an agenda that keeps dialog information around, so that the system responds differently depending upon that context. For instance, the first time the student makes an error on a particular question he might get a KCD. Suppose that when that KCD is finished the student is asked the question again but fails; the student will get a hint rather then do the KCD over again.

It is worth noting that these innovations, listed above, are not random improvements; they are related to one another. For instance, once you decided that you want a system that can detect multiple errors simultaneously, then it makes sense to design a way to give feedback for different errors in some way (i.e., a dialog system that can ask about the first error followed by asking

---

[45] However, it should be noted that since natural language is intractable, in the general case, Ms. Lindquist gets around many of the difficult questions (e.g., uses pull-down menus). Even so, I still say evidence that students had to learn this new interface, but the hope is that it is easier to learn the way the system asks questions, then a complicated graphical-user interface.

about the second error). In addition, once you have decided that you want a system that can ask new questions, you need a system that can deal with dialog intelligently. That, also, naturally leads to asking a series of questions that are all related to a single goal (i.e., a KCD).

Chapter Conclusions

In this chapter, we have seen that Ms. Lindquist's behavior maps well onto some of the features we observed in our human tutor. For instance, we saw the concrete articulation strategy mapped quite well. We also compared Ms. Lindquist to a traditional model-tracing tutor and noted salient point where Ms. Lindquist has improved upon traditional model-tracing tutors. In the next chapter, I will present an empirical evaluation of Ms. Lindquist.

Chapter 5: A Formative Analysis Comparing Ms. Lindquist with a Computer Aided Instruction Control

Chapter Overview

In this chapter, I report on a formative evaluation of Ms. Lindquist, a computer tutor designed to be more like a human tutor by engaging the student in a dialog. Our goal in this study was to detect any benefits of Ms. Lindquist's dialogs as compared to a control condition, representing a classical Computer Aided Instructional (CAI) approach. This study answers the question "If you keep the number of problems fixed, do students learn more by going though a dialog with Ms. Lindquist." This study does not answer the question "It is worth the extra time it takes?"

Independent Variable: Type of feedback after student errors

Control Condition: A traditional Computer-Aide Instructional (CAI) approach: Let the student try to answer the problem on his or her own. If he or she answers incorrectly provide the student with the correct answer and then move to the next problem.

Experimental Condition: A dialog-enabled Intelligent Tutoring System approach: This system had the following salient features (described in the previous chapters):

- Dynamic subgoal scaffolding used to break problems down into steps the student answered incorrectly, while also providing positive feedback on the subgoals that the student accomplished correctly. The scaffolding depends upon the student model diagnosis of the student's previous answer.
- For the subgoals that need scaffolding, the tutor provides multi-step tutorial strategies including: 1) Concrete Articulation, 2) Explain in English First, 3) Introduce a Variable to represent a quantity, and 4) Tell the student the answer and then ask them to explain why it is correct.

Dependent Variables: Learning Measure

The outcome measure of interest was student learning as measured by a posttest.[46]

---

[46] I was also interested in how long it took students to reach mastery, measured both in terms of total time and the number of problems it took. Unfortunately, most of my students did all of the problems in a section before reaching mastery, which basically turned the experiment into one where the number of problems was controlled. Not surprisingly, students took two to three times longer if you engaged them in a dialog, than if you just simply told them the answer. Therefore, I will not be able to compare the conditions in terms of the amount time needed to reach mastery. I am still able to compare the conditions to see which did better when the number of problems completed is held constant (The average number of problems done by the two groups was equal.) Consequently, the conclusions from this study will be formative in nature.

Procedure

I conducted a randomized controlled experiment to compare my two conditions. Twenty high-school students participated. These students were participating in a full-day, month long summer enrichment program at a nearby university. They used the software as part of their regular mathematics classroom time. There were seven students going into the tenth grade, eleven students going into the eleventh grade, and two students going into the twelfth grade.

Students were given two paper-and-pencil assessments during their normal classroom period that took approximately 40 minutes to complete. I used the assessments to block the subjects into two groups of equal ability level. The first assessment was an eleven-item assessment on symbolization (see Appendix G.) I gave this same assessment at the post-test. The second assessment was a 21-item measure of general mathematical knowledge. This assessment was used only as a pretest. It was given to help block the students into two groups with more precision. When creating the two blocked groups, I blocked first by using the more heavily weighted symbolization assessment. I used the general mathematics knowledge assessments as a secondary sorting key. Three students who were not present the day of the assessments were randomly assigned to a condition (two in one condition, one to the other).

In order to occupy my students' time roughly equally, I decided that all of them would see both conditions. The curriculum was split roughly in half. Half of the students were to proceed through the curriculum in the control condition, while the other half were in the experimental condition. After completing a six-item, embedded (the student did it on the computer) mid-test, the students switched to the other condition. Both the curriculum and the mastery-learning algorithm used to proceed through the curriculum were the same for both conditions. While in the computer lab, students worked independently. Two mathematics teachers (including myself) were present to help students but the teachers sat idle most of the time.

Students used the software during five class periods each lasting about 45 minutes. Students were given additional time if needed to finish the curriculum. All but 4 students finished the curriculum.

As described in the previous chapters, the particular strategy Ms. Lindquist uses to tutor a subgoal is chosen in advance from among the 4 strategies. I decided not to switch the strategy a student was receiving while in the middle on a section. I did this because I wanted the students to become familiar enough with a strategy so that they could begin learning from its use. The following table illustrates how I made those assignments across the 20 subjects. Each column represents a student and indicates the type of interaction the system provided.

	Students broken into two groups																			
	Exp-Con										Con-Exp									
Section 1	1	1	1	1	1	1	1	1	1	1	c	c	c	c	c	c	c	c	c	c
Section 2	1	1	1	1	1	1	1	1	1	1	c	c	c	c	c	c	c	c	c	c
Section 3	4	4	4	2	2	2	3	3	3	3	c	c	c	c	c	c	c	c	c	c
	Six Item Mid-Test																			
Section 4	c	c	c	c	c	c	c	c	c	c	3	3	3	4	4	2	2	1	1	1
Section 5	c	c	c	c	c	c	c	c	c	c	2	1	4	1	3	1	3	2	3	4
	A Different Six-Item Post-Test																			

Key: 1) Concrete Articulation, 2) Explain in English First, 3) Introduce a Variable to stand for a quantity, and 4) Tell the student the answer and then ask them to explain why it is correct. The control condition is indicated by "c".

Of the two equal ability groups I called one the "Exp-Con" group since they received the experimental condition on the first half of the curriculum, (section 1-3) and the control condition on the second half (sections 4-5). The "Con-Exp" group experienced the conditions in the opposite order.

The first ten students listed (i.e. the first ten columns) received the concrete articulation strategy (indicated by the "1") on the first two sections of the curriculum. I did this because I thought this strategy probably worked better then any of the other strategies on one-operator problems. Then for the third section, each of these ten students received one to the three remaining strategies. Finally, all ten of these students were in the control ("c") condition for the last two sections. The other ten students received the control condition first and then some combination of the experimental conditions.

The curriculum I used was divided up into the five sections (see the first figure in chapter three). Each section had a different number of problems and its own mastery criterion. The sections that had more problems had a higher mastery criterion.

Section Description	**Number of Problems**	**Mastery Criterion**
1) One-Operator Problems.	9	4
2) One-Operator involving distance, rate and time.	12	4
3) Two-Operator linear forms.	16	5
4) Two-Operator with some involving division and parenthesis.	13	4
5) Three and Four Operator Problems	17	5

Within a section, the problems were randomly ordered to prevent students moving in synch with the student sitting next to them. Above I mentioned the six-item mid-test between section 3 and 4. For those problems, students received the control condition feedback. All the students were given those six problems in the same order, to better insure reliability across subjects. At the end of section 5, the students received a different six-item test on the computer. Finally, the students completed the same 11-item symbolization test, on paper, that had been administered to check for overall learning. Note, because 4 students did not finish the curriculum and thereby did not get to the six-

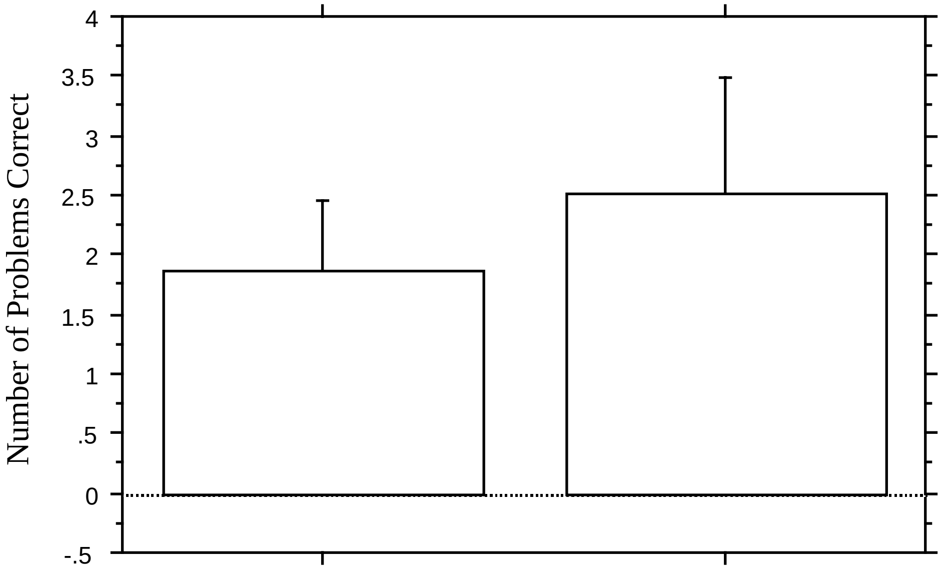


Figure 1: Overall, students improved their symbolization abilities (by close to one problem) as measured by the number of problems they got correct on the repeated 11-item symbolization paper-and-pencil test.

item post-test given on the computer, I added these same six-items to the student's paper and pencil post test.[47]

Results

As I mentioned above, students took 2-3 times longer in the experimental condition (First half of the curriculm:91 minutes vs. 43 minutes, Second half: 2 hours 29 minutes vs. 37 minutes). The first result I report is that students demonstrated learning, regardless of condition.

First, I report a sanity check that, overall, students learned something. To check for overall learning I compared the performance of students on the paper-and-pencil eleven-item test that was given at the start and the end of the study. I did a repeated measures analysis of variance (F(1,16)=2.195,p=.16). The three subjects with no pre-test were thrown out for this analysis. Overall, students gained by about one-half of a problem (pre=1.9,post=2.5).

To test for differential learning rates by condition, I performed a 2 factor ANOVA on the students' test scores. One factor of the model was taking into account whether the student received the experimental or the control condition on the sections directly preceding the test. The second factor was the test (the mid-test after section 3 vs. the posttest after section 5). I found that there was a statistically significant difference (F(1,37)=3.4,p=.07) between the number of problems correct on the mid-test (1.7) and the posttest (.95). Since the post-test items were harder, this is not surprising. I found a close to marginally statistically significant difference (F(1,37)=2.6,p=.12) in the number

---

[47] All four of those students completed the fourth section and got partially through the 5th section. In particular, the four students got to problems 3, 6, 6 and 14 respectively.

of problems correct between the experimental and the control conditions. The experimental condition averaged 1.65 problems correct per test, while the control group averaged only one problem correct. Taking the standard deviation into account, I found a respectable effect size of .76 between the experimental and the control conditions (see Appendix I for a discussion of how to interpret effect size measures.)

The following figure shows that the group "Exp-Con" did better on the mid-test than compared to the "Con-Exp" group. However, after the subjects completed the mid-test and were switched to the second condition, a reversal happened and the "Con-Exp" group did better. This argues that the experimental condition (i.e., Ms Lindquist) was more effective at promoting learning. Note, that this experimental design probably underestimated the impact of Ms. Lindquist, since the students who were performing worse at the mid-test were able to overcome this deficit to do better then the other group.

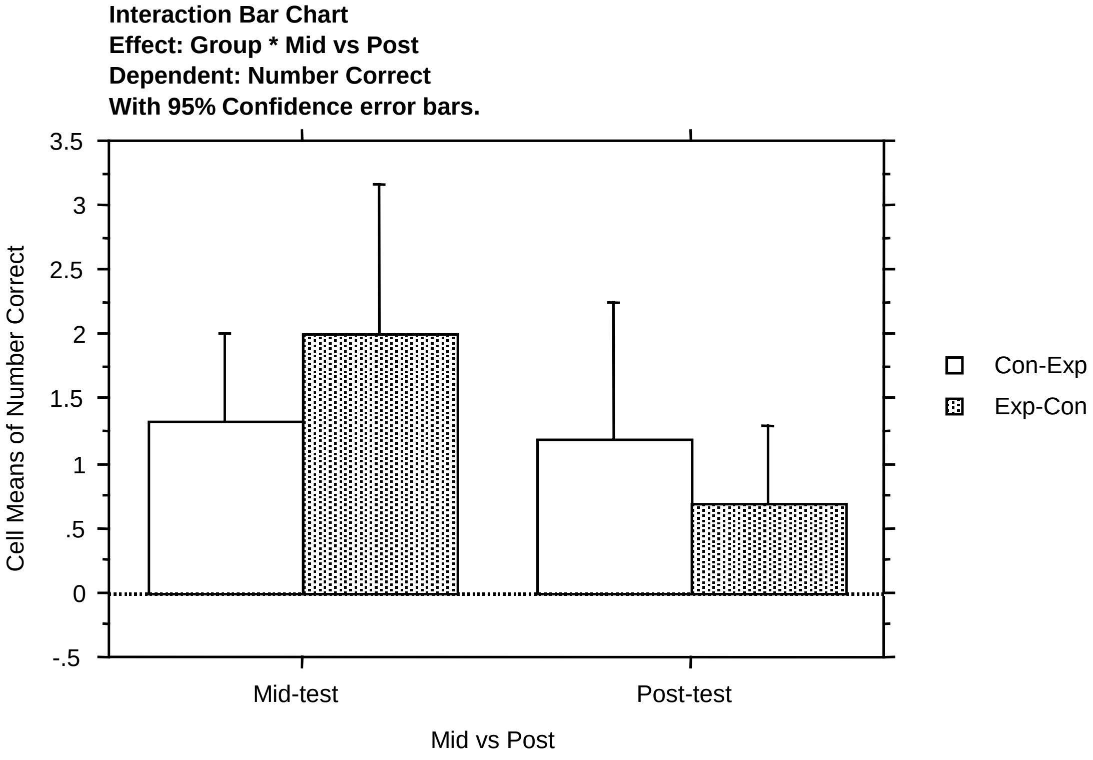


Figure 2: Student did better on the embedded tests, after having just received the experimental treatment.

When is the Learning Happening?

Since mastery-learning was used this means that some students would graduate from a section before doing every problem. However, since these problems were hard and the mastery criterion was high, (4 or 5 correct in a row, depending upon the section) most students did not graduate ("graduate" means reach the mastery criterion before running out of problems in that section) from most of the sections. This is evidenced by the fact that on average, student did 34.5 of the 37 problems in the first half of the curriculum. This average of 34.5 was the exact same for the two groups. Therefore, we do not see any evidence of the experimental group being promoted to the next section any faster.

On average, the students were getting slightly over 13 problems correct on the problems they did. I found no evidence that students were getting more problems correct in either condition (F(1,18)=.236, p=.63, experimental=13.2, control =14.0). This was similar in the second half of the curriculum.

Discussion

What accounts for the fact that the dialogs that Ms. Lindquist had with students lead to more learning? We take an in-depth look at a single student to see what is going on in these dialogs.

A Extended Look at the Performance of a Single Student

I shall refer to this student with the masculine pronoun. This particular student received the experimental condition on the first three sections and the control condition on the last two sections. In particular, as part of the experimental condition he received *concrete articulation* scaffolding on first two sections of the curriculum. Here is an example form the first section showing the concrete articulation strategy doing what it was designed to do. The student initially fails to do a correct symbolization, but does the arithmetic step correctly. Nevertheless, and consistent with our theory, getting the articulation correct takes three attempts for the student. The generalization step then takes one attempt and is easy once the articulation has been accomplished.

```
1.  <Problem_Statement>Mary goes to store with "m" dollars. She spent all
    her money to buy 5 bottles of her favorite soda. Write an expression for
    how much each bottle costs.</Problem_Statement>

2.  <Tutor time="[7/11/2000][9:26:50][0]" qtype="QSYMB" goal="m/5">Hello.
    Please write an expression for the price of one bottle. </Tutor>
3.  <Student time="[7/11/2000][9:28:28][98]">m-5</Student>

4.  <Tutor time="[7/11/2000][9:28:29][0]" qtype="QCOMPUTE" goal="20"> No.
    We will come back to the price of one bottle. Let me try to break this
    down for you. Please compute the price of one bottle if the amount of
    money she went to the store with was 100 dollars? </Tutor>
5.  <Student time="[7/11/2000][9:28:48][19]">20</Student>

6.  <Tutor time="[7/11/2000][9:28:48][0]" qtype="QEXPLAIN"
    goal="100/5">Correct. Using only 100, and 5 write an expression for the
    price of one bottle.</Tutor>
7.  <Student time="[7/11/2000][9:29:0][12]">100-5</Student>

8.  <Tutor time="[7/11/2000][9:29:1][0]" qtype="QEXPLAIN" goal="100/5"> No.
    No. Your answer of "100-5" equals "95", but you wanted to write an
    expression that equaled "20". Using only 100, and 5 write an expression
    for the price of one bottle. Hint: Use 100 for the amount of money she
    went to the store with and 5 for the number of bottles. </Tutor>
9.  <Student time="[7/11/2000][9:29:30][29]">100*5</Student>

10. <Tutor time="[7/11/2000][9:29:30][0]" qtype="QEXPLAIN" goal="100/5"> No.
    No. Your answer of "100*5" equals "500", but you wanted to write an
    expression that equaled "20". Using only 100, and 5 write an expression
    for the price of one bottle. Hint: Use 100 for the amount of money she
    went to the store with and 5 for the number of bottles. Pick one of
    these 6 choices : "100-5", "5-100", "100+5", "100*5", "100/5", or
    "5/100" </Tutor>
11. <Student time="[7/11/2000][9:30:11][41]">100/5</Student>
```

```
12. <Tutor      time="[7/11/2000][9:30:11][0]"      qtype="QGENERALIZE      "
    goal="m/5">Correct. Now use "m" instead of 100 to again express the price
    of one bottle.</Tutor>
13. <Student time="[7/11/2000][9:30:19][8]">m/5</Student>

14. <Tutor  time="[7/11/2000][9:30:19][0]"  qtype="NIL"  goal="NIL">Correct.  A
    new problem is needed </Tutor>
```

The first section had nine problems. The student did all nine, getting only two correct, and consequently, failed to reach the mastery criterion of four correct in a row, before running out of problems. Remember that the concrete articulation strategy has only three steps; arithmetic, articulation and generalization. Both our experimental DFA results and our theory suggest that arithmetic steps should be easier then the student's initial attempts at the problems. I looked for evidence of this but it was complicated since the student might jump to the articulation step. I found that of the seven times he was asked an arithmetic question, he twice answered with articulations, getting only one of them correct. Consistent with our hypothesis, he had an easy time with the arithmetic problems getting all five correct. Articulations were also harder, as predicted. He got three of seven articulations correct (this number includes the two, aforementioned, times he tried to articulate without first answering the arithmetic question). The example dialog presented above is an example in which the student initially used the wrong operator (i.e. "m-5"), then correctly got the computation question (i.e. "20"). However, when asked to articulate the answer, he used the same wrong operator he used initially (i.e. the subtraction in place of division in his answer of "100-5"). This exact pattern appeared on the next problem as well. Finally, agreeing with theory, the generalization step was easier than the articulation step, with only one error out of seven attempts.

When I looked at the sorts of errors that were made I found that, overwhelmingly, the errors were of using the wrong operator. Of the twelve errors that occurred on all of the questions in this section combined, nine of them were due to the student using the wrong operator[48]. This should not be that surprising since the word problems in this section are simple and give only two quantities to use, so it is not very surprising that the main difficulty is knowing which operator to use. This is consistent with the error analysis I presented from our DFA data where I found that student made many errors due to picking the wrong operator. Unfortunately, this student did not reach mastery and was pushed onto the next section.

I then looked at how he did on the second section and I found similar results.
The second section was composed of more one-operator problems, all of which were about calculating distance, speed or elapsed time. This is a notoriously difficult task for students. This student got four correct out of twelve, including the last two in the section, possibly suggesting evidence of learning. In this section, the student was much more

---

[48] Also consistent with my DFA data but in no way statistically reliable were the types of operators they confused. In particular, getting division problems were the hardest. The wrong operations were misused as follows: when the student was supposed to use division, he used subtraction three times and multiplication twice. When the student was supposed to use multiplication, he used addition once and division twice. Finally, when the student was supposed to use subtraction, he used multiplication once. Note that of the nine problems in the section, one was an addition problem, three were subtraction problems, two were multiplication problems, and two were division problems. Also, note that the student could make more then one error on a problem. The two division problems appeared to be the hardest accounting for five of the nine errors.

likely to try to answer the arithmetic scaffolding question with an articulation; only three of eight gave an arithmetic value as expected when asked to compute a value. Of those three, only one of them was done correctly. This speaks to the increased difficulty of these problems.

The following example is from this second section and is a problem on which the student struggled quite a bit.

1. <Problem_Statement>Amy walks at "s" miles per hour. This weekend she took a 12 mile walk. Write an expression for how long the walk lasted.</Problem_Statement>

2. <Tutor time="[7/12/2000][9:31:54][0]" qtype="QSYMB" goal="12/s">Hello. Please write an expression for the amount of time her walk lasted. </Tutor>
3. <Student time="[7/12/2000][9:32:8][14]">**s*12**</Student>

4. <Tutor time="[7/12/2000][9:32:8][0]" qtype="QCOMPUTE" goal="3"> No. We will come back to the amount of time her walk lasted. Let me try to break this down for you. Please compute the amount of time her walk lasted if Amy's walking speed was 4 miles per hour? </Tutor>
5. <Student time="[7/12/2000][9:33:20][72]">**4**</Student>

6. <Tutor time="[7/12/2000][9:33:21][0]" qtype="QCOMPUTE" goal="3"> Hmm. No. Try again please. Please compute the amount of time her walk lasted if Amy's walking speed was 4 miles per hour? Hint: Use the fact that the distance Amy walks is 12 miles and Amy's walking speed is 4 miles per hour. </Tutor>
7. <Student time="[7/12/2000][9:33:44][24]">**4/12**</Student>

8. <Tutor time="[7/12/2000][9:33:44][0]" qtype="QCOMPUTE" goal="3"> Hmm. No. Try again please. Please compute the amount of time her walk lasted if Amy's walking speed was 4 miles per hour? Hint: It is one of these 6 choices: "12-4", "4-12", "12+4", "12*4", "12/4", or "4/12".</Tutor>
9. <Student time="[7/12/2000][9:33:53][9]">**4/12**</Student>
10. <Tutor time="[7/12/2000][9:33:44][0]" qtype="QCOMPUTE" goal="3"> Hmm. No. Try again please. Please compute the amount of time her walk lasted if Amy's walking speed was 4 miles per hour? Hint: It is one of these 6 choices: "12-4", "4-12", "12+4", "12*4", "12/4", or "4/12".</Tutor>
11. (Since Ms. Lindquist repeats her self until the student guess correctly. We omit her text.)
12. <Student time="[7/12/2000][9:34:2][9]">**12*4**</Student>
13. <Student time="[7/12/2000][9:34:11][8]">**12-4**</Student>
14. <Student time="[7/12/2000][9:34:19][9]">**12+4**</Student>
15. <Student time="[7/12/2000][9:34:26][7]">**4-12**</Student>
16. <Student time="[7/12/2000][9:34:41][15]">**12-4**</Student>
17. <Student time="[7/12/2000][9:34:49][7]">**4-12**</Student>
18. <Student time="[7/12/2000][9:34:55][6]">**12*4**</Student>
19. <Student time="[7/12/2000][9:35:5][10]">**12+4**</Student>
20. <Student time="[7/12/2000][9:35:16][11]">**12/4**</Student>

21. <Tutor time="[7/12/2000][9:35:16][0]" qtype="QGENERALIZE " goal="12/s">Correct. Now use "s" instead of 4 to again express the amount of time her walk lasted.</Tutor>
22. <Student time="[7/12/2000][9:35:27][11]">**s/4**</Student>

23. <Tutor time="[7/12/2000][9:35:27][0]" qtype="QGENERALIZE " goal="12/s">It looks like you didn't understand what I was asking you. You should use the variables from the problem. For instance, use s instead of 4. Try this question again: Now use "s" instead of 4 to again express the amount of time her walk lasted.</Tutor>

The students initial attempt at the arithmetic step was simply to answer with "4" to which Ms. Lindquist gave the hint "Use the fact that the distance Amy walks is 12 miles and Amy’s walking speed is 4 miles per hour." This seems to have gotten the student to get closer for then the next student utterance was ’4/12’ but this is wrong due to the wrong order of the arguments. If the student had done the division, he might have noticed his error since he might know that his answer should be larger then one hour. However, he does not and the tutor gives the student the most explicitly hint that presents the problem as a multiple-choice problem. Thereafter the student guesses answers, each time guessing articulations (and presumably not doing the arithmetic to check to see if the numbers make sense). Eventually, persistence pays off and the student guesses the correct answer. Since Ms. Lindquist had reflective follow-up questions turned off for this data collection, the student did not get any such questions, but this presumably would have been a moment where such questions might have been effective. Note, those four reflective follow-up questions were triggered because the student reaching the last hint in a sequence. Once the student guesses correctly, Ms. Lindquist could have followed that up with "Let me check your understanding. What does your answer represent in words?" whereupon the student would be presented with the same interface used in the Explain strategy with the three pull down menus. Having such reflection questions would be consistent with what our experienced tutor did (line 169-170, 189, 229-234, 254-268, and 324). Sometimes the human tutor would ask the student questions even after they got the answer correct. Using such a reflective follow-up question might have made it more likely for this student to understand that you calculate elapsed time by dividing the distance by the speed. As it turns out the students does not finish that problem that day because the class period ends. Because the student did not finish the problem that day, he got a chance to start over. However, he failed to remember how to correctly solve the problem and it took him four more attempts as shown below. It is probably no coincidence that this same student had difficulty doing the, normally easy, generalization question. Clearly, this is an important area for future improvements.

1. <Problem_Statement>Amy walks at "s" miles per hour. This weekend she took a 12 mile walk. Write an expression for how long the walk lasted.</Problem_Statement>

2. <Tutor time="[7/13/2000][8:34:40][0]" qtype="QSYMB" goal="12/s">Hello. Please write an expression for the amount of time her walk lasted. </Tutor>
3. <Student time="[7/13/2000][8:48:0][800]">**s*12**</Student>

4. <Tutor time="[7/13/2000][8:48:0][0]" qtype="QCOMPUTE" goal="3"> No. We will come back to the amount of time her walk lasted. Let me try to break this down for you. Please compute the amount of time her walk lasted if Amy’s walking speed was 4 miles per hour? </Tutor>
5. <Student time="[7/13/2000][8:49:19][78]">**8**</Student>

6. <Tutor time="[7/13/2000][8:49:19][0]" qtype="QCOMPUTE" goal="3"> Hmm. No. Try again please. Please compute the amount of time her walk lasted if Amy’s walking speed was 4 miles per hour? Hint: Use the fact that the distance Amy walks is 12 miles and Amy’s walking speed is 4 miles per hour. </Tutor>
7. <Student time="[7/13/2000][8:51:3][105]">**4/12**</Student>

8. <Tutor time="[7/13/2000][8:51:3][0]" qtype="QCOMPUTE" goal="3"> Hmm. No. Try again please. Please compute the amount of time her walk lasted if

```
    Amy's walking speed was 4 miles per hour?  Hint: It is one of these 6
    choices: "12-4", "4-12", "12+4", "12*4", "12/4", or "4/12".</Tutor>
9.  <Student time="[7/13/2000][8:51:15][12]">12/4</Student>

10. <Tutor time="[7/13/2000][8:51:15][0]" qtype="QGENERALIZE " 
    goal="12/s">Correct. Now use "s" instead of 4 to again express the amount
    of time her walk lasted.</Tutor>
11. <Student time="[7/13/2000][8:51:32][17]">s/12</Student>

12. <Tutor time="[7/13/2000][8:51:32][0]" qtype="QGENERALIZE " goal="12/s">
    No. Rewrite 12/4 but in place of "4" put s. </Tutor>
13. <Student time="[7/13/2000][8:51:45][13]">12/s</Student>

14. <Tutor time="[7/13/2000][8:51:45][0]" qtype="NIL" goal="NIL">Correct.  A
    new problem is needed </Tutor>
```

We now get back to our reporting on how this student did on this second section. When I looked at the first time the student gave an articulation, I found that he got two of eight correct (The eight include some instances where he did not first correctly compute an answer). Given that I found that it was easier to start with the arithmetic step first, it was unfortunate that the student did not do so, since he might have been more successful. Finally, I looked again at the generalization success rate. This student's results were similar to my DFA results and the results for this student on the first section: generalization is easy (six of eight correct) once you have gotten the articulation correct.

I then looked at the sort of errors being made on the student's first attempt at a problem. I found, once again, many errors involving the wrong operator (6 on 8) and all of these confused multiplication with division or vice versa. Only one of the eight errors was due to the wrong order of the arguments.

The third section introduced two-operator problems (none of which required parentheses). There were 16 problems and the student only got four correct. This demonstrates how hard it is for student to symbolize even simple linear form equations. This topic consumes a large proportion of typical algebra classes and still the TIMSS results show that on a similar problem, 8th grade students get only 50% correct. The errors that the student made are not so easy to categorize. The only error types from our categorization that appeared more then once were "sub" errors and those appeared only three times. The rest of the attempts had more then one error. Of those 11 other errors, nine of them involved at least one wrong operator error. I think it is reasonable to assume that these sorts of errors often are the result of student's having no good idea about how to proceed. Instead, the student might apply an approach I call "quantity and operator soup", which is simply to randomly put quantities together with operators (see also Tabachneck, Koedinger, & Nathan, 1994).

This student received the *introduced variable strategy* for this section. This next passage is a look at the last problem the student did in this section:

```
1.  <Problem_Statement>Amy wanted to try parachuting.  She jumped out of a
    plane at 5,000 feet.  She did a free-fall for "s" seconds at an average
    speed of 214 feet per second before pulling the cord to open her
    parachute.   How far was Amy from the ground when she pulled the
    cord?</Problem_Statement>

2.  <Tutor time="[7/19/2000][9:3:59][0]" qtype="QSYMB" goal="5000-
    214*s">Hello.  Please write an expression for the distance Amy is from
    the ground. </Tutor>
```

3. <Student time="[7/19/2000][9:5:54][116]">**5000-s-214**</Student>

4. <Tutor time="[7/19/2000][9:5:54][0]" qtype="QSUB_QUAN_STRICTLY" goal="214*s"> No. We will come back to the distance Amy is from the ground. Let me try to break this down for you. What is the distance of Amy’s free-fall?</Tutor>
5. <Student time="[7/19/2000][9:6:19][24]">**214-s**</Student>

6. <Tutor time="[7/19/2000][9:6:19][0]" qtype="QSUB_QUAN_STRICTLY" goal="214*s"> No. Use "214" and "s" to calculate the distance of Amy’s free-fall.</Tutor>
7. <Student time="[7/19/2000][9:6:31][12]">**214*s**</Student>

8. <Tutor time="[7/19/2000][9:6:31][0]" qtype="QSUB_QUAN_NOT_STRICTLY" goal="5000-B">Correct. If "B" was the distance of Amy’s free-fall then what would be the distance Amy is from the ground? </Tutor>
9. <Student time="[7/19/2000][9:6:46][14]">**b/5000**</Student>

10. <Tutor time="[7/19/2000][9:6:46][0]" qtype="QSUB_QUAN_NOT_STRICTLY" goal="5000-B"> Hmm. No. What is the distance Amy is from the ground? Use "B" instead of "214*s" for the distance of Amy’s free-fall. Hint: pick one of these: 5000-B, B-5000, 5000+B, 5000*B, 5000/B, or B/5000.</Tutor>
11. <Student time="[7/19/2000][9:6:54][8]">**5000/b**</Student>

12. <Tutor time="[7/19/2000][9:6:54][0]" qtype="QSUB_QUAN_NOT_STRICTLY" goal="5000-B"> Hmm. No. What is the distance Amy is from the ground? Use "B" instead of "214*s" for the distance of Amy’s free-fall. Hint: pick one of these: 5000-B, B-5000, 5000+B, 5000*B, 5000/B, or B/5000.</Tutor>
13. <Student time="[7/19/2000][9:7:2][8]">**5000*b**</Student>

14. <Tutor time="[7/19/2000][9:7:2][0]" qtype="QSUB_QUAN_NOT_STRICTLY" goal="5000-B"> Hmm. No. What is the distance Amy is from the ground? Use "B" instead of "214*s" for the distance of Amy’s free-fall. Hint: pick one of these: 5000-B, B-5000, 5000+B, 5000*B, 5000/B, or B/5000.</Tutor>
15. <Student time="[7/19/2000][9:7:8][6]">**5000-b**</Student>

16. <Tutor time="[7/19/2000][9:7:8][0]" qtype="QSUBSTITUTE" goal="5000-214*s">Correct. Now, lets try this question again: You are correct that "5000-B" is the distance Amy is from the ground. Now, can you state this in terms of the numbers given in the problem?</Tutor>
17. <Student time="[7/19/2000][9:7:39][30]">**5000-214*s**</Student>

18. <Tutor time="[7/19/2000][9:7:39][0]" qtype="NIL" goal="NIL">Correct. A new problem is needed </Tutor>

In this dialog Ms. Lindquist interpreted the student’s first answer as having a "wrong operator" error because the student used subtraction instead of multiplication to connect the "s" and the "214". Ms. Lindquist decided to scaffold this problem by first focusing on the embedded subexpression and therefore asked the student to try to answer just that portion. The student answer of "214-s" was also wrong.

This sort of error is representative of this student’s work in this section. Remember that there were 16 problems of which the student got 4 correct on the first attempt. Ms. Lindquist scaffolded the embedded subexpression on nine different problems including the one shown. Of those nine times, only twice did the student correctly symbolize the embedded subexpression immediately. Of the other seven problems, there were three types of errors that occurred. There were two instances when the student's only error was using the wrong operator (as was done on this problem). There were three instances when the student answered with just the y-intercept value, (which is all three cases was a concrete value as opposed to a variable). Finally, twice the student used the correct operator as well as one of the correct arguments but used the wrong concrete value (in both cases it was the y-intercept instead of the slope; e.g. "800m" instead of "40m" and "2h" instead of "3h").

To continue with the example dialog, the student was given another chance to symbolize the embedded expression and this time succeeded in getting it correct, so Ms. Lindquist asked the next question which was `"If "B" was the distance of Amy's free-fall then what would be the distance Amy is from the ground?"`

The student took several turns before answering this correctly with "5000-b". This student then put it all together on the first try. One might think that once being told that both "5000-b" is correct and that "b" was equal to "214*s" it would be easy to say that the answer was "5000-214*s." However, the student was only able to get this sort of question correct (on the first attempt) three out of the nine times this situation arose. This result is consistent with our claim that a major difficulty students have with composed symbolization problems can be explained with our "foreign language" hypothesis. Even when they have gotten the two steps of an algebra problem solved separately, putting the two steps together is a difficult step. This suggest that this strategy would be more productive if students first had practice with some decontextualized symbolic substitution problems like what I had students do in the transfer study reported in Chapter 1.

To continue the reporting of this student's progress, the student next did the 6-item mid-test which he got all six items incorrect.

After the mid-test, the subject switched from the experimental to the control condition and was given cut-to-the-chase on the remaining problems. Unsurprisingly, since the student did not do very well in the previous sections, during the next two sections, this student got only one of problem correct. Perhaps most surprising of all was that this student did not give up and take the easy way out by hitting return and seeing what the answer was. As I reported above, the cut to the chase group still took as long to read the problems and make their first attempt as the experimental group. This student was no exception, often spending minutes to read the problems and make an attempt. Often times his attempts were quite close.

Finally, this student took the six-item computer posttest. This student got one correct. On another item, he only forgot parenthesis. On a third item, he made a relatively simple error of confusing two quantities (said `"2m+3s"` instead of `"3*m+2*s")`.

Chapter Conclusion

This chapter has shown evidence if you control for the number of problems, the dialogs that Ms. Lindquist has with student, can lead to increases in student learning as measured by a post-test.

Chapter 6: Conclusions and Future Work

I will end this dissertation by summarizing the components that are clearly articulated in Ms. Lindquist's architecture so we can view them in terms of the future work that this dissertation enables.

Ms Lindquist has multiple enhancements to traditional model tracing tutors (see Figure 1) including being able to deal with more then one error at a time, dialog coherence "moves" and embedded feedback. However, the two aspects that are likely to provide the biggest learning gains are what I have called dynamic scaffolding and tutorial strategies. These tutorial strategies come in two forms: knowledge construction dialogs (KCDs) and knowledge remediation dialogs (KRDs). Ms. Lindquist has only one KRD implemented, but I identified several other KRDs that the experienced human tutor used that could be added in the future. I will first discuss dynamic scaffolding, since the knowledge construction dialogs are layered on top of dynamic scaffolding.

Ms. Lindquist's Architecture

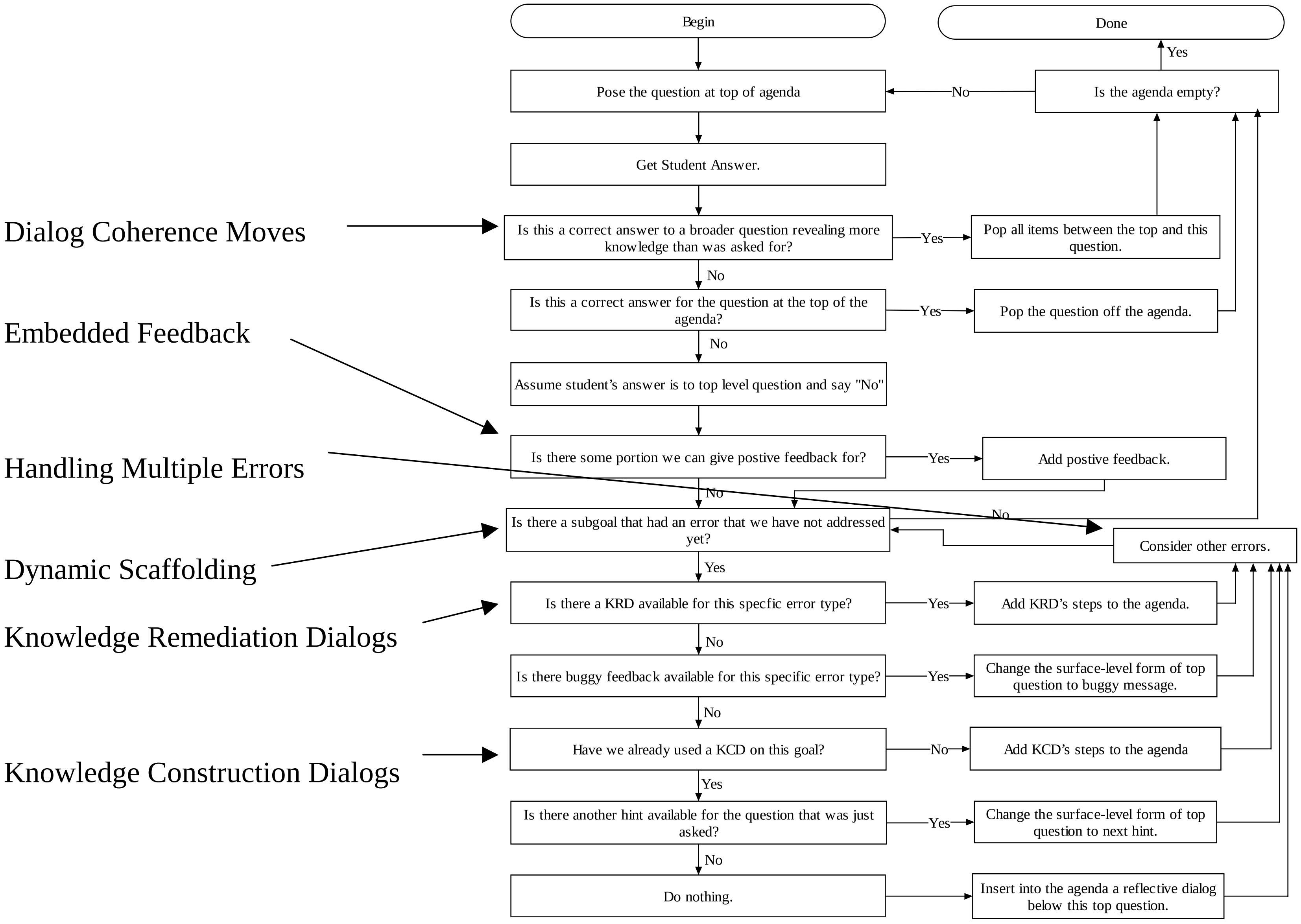

Begin
Done
Yes
Pose the question at top of agenda
No
Is the agenda empty?
Get Student Answer.
Dialog Coherence Moves
Is this a correct answer to a broader question revealing more knowledge than was asked for?
Yes
Pop all items between the top and this question.
No
Is this a correct answer for the question at the top of the agenda?
Yes
Pop the question off the agenda.
Embedded Feedback
No
Assume student's answer is to top level question and say "No"
Is there some portion we can give postive feedback for?
Yes
Add postive feedback.
Handling Multiple Errors
No
No
Is there a subgoal that had an error that we have not addressed yet?
Consider other errors.
Dynamic Scaffolding
Yes
Is there a KRD available for this specfic error type?
Yes
Add KRD's steps to the agenda.
Knowledge Remediation Dialogs
No
Is there buggy feedback available for this specific error type?
Yes
Change the surface-level form of top question to buggy message.
No
Have we already used a KCD on this goal?
No
Add KCD's steps to the agenda
Knowledge Construction Dialogs
Yes
Is there another hint available for the question that was just asked?
Yes
Change the surface-level form of top question to next hint.
No
Do nothing.
Insert into the agenda a reflective dialog below this top question.

Figure 1: Aspect of Ms. Lindquist that could be "lesioned".

Dynamic Scaffolding

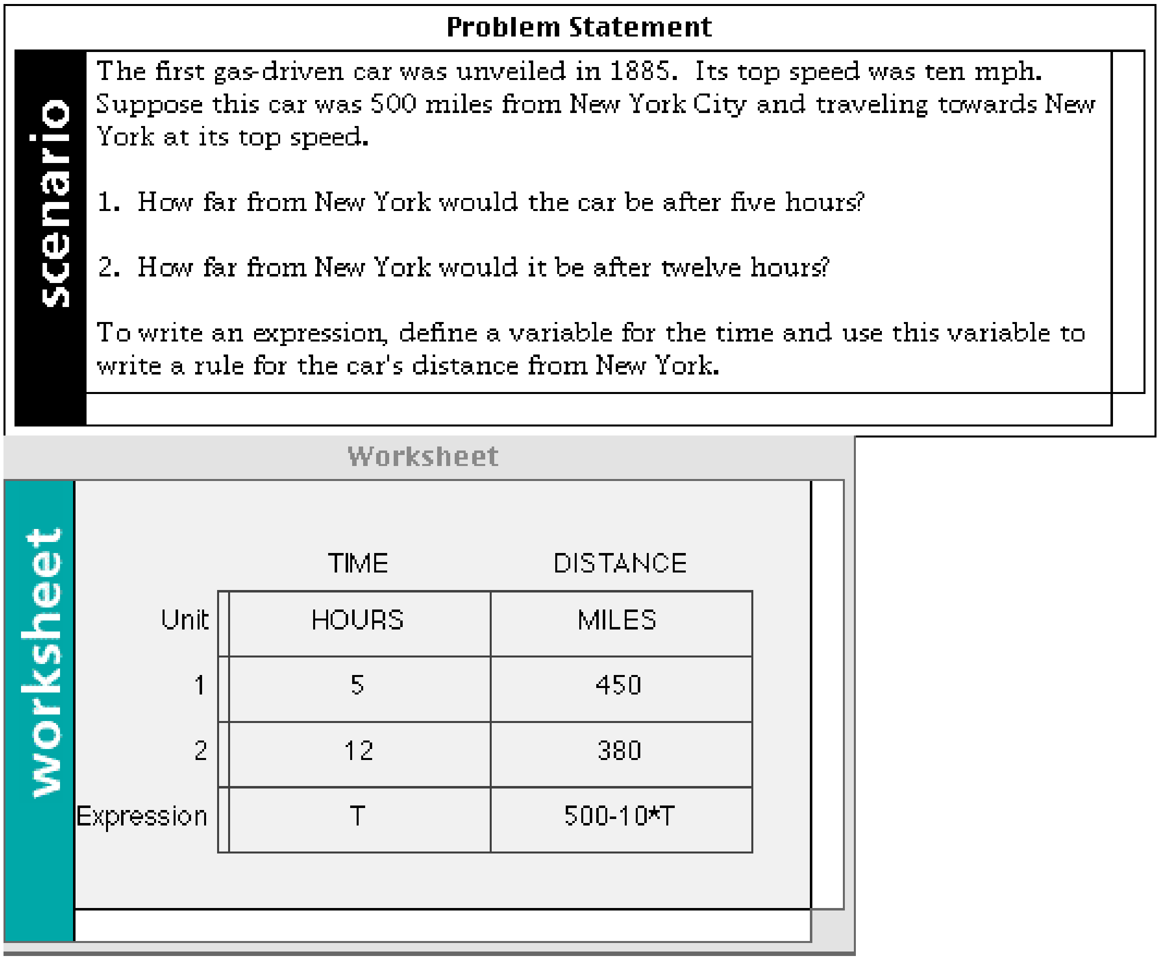
Ms. Lindquist has what I have termed dynamic scaffolding. I have been using the term dynamic scaffolding to encompass two aspects. The first aspect is that the scaffolding appears only after an error, rather than "pre" scaffolding that the student is presented with ahead of time. A second aspect is that scaffolding is context sensitive to the step the student made an error on. I will illustrate each of these aspects in the 3 types of scaffolding listed below:

1. Static & Pre-Scaffolding
2. Static & Post-Scaffolding
3. Dynamic Scaffolding

I will now discuss each one in turn and provide an example.

Static & Pre-Scaffolding

Figure 2 shows the Carnegie Learning Inc. tutor (Koedinger & Anderson, 1998), which is an example of scaffolding that is both "pre" and static. First, it is "pre"-scaffolding, because students are expected to do the scaffolding steps, (e.g., labeling the columns, identifying units, and computing instances) before attempting to write the expression in the bottom, right hand corner. It is "static" in that there are **fixed** elements that are always asked regardless of whether the systems believes that the student needs such assistance. Next, I present an example that has scaffolding that is still static, but that appears after the student has made an error (thus "post" as opposed to "pre").



Problem Statement

The first gas-driven car was unveiled in 1885. Its top speed was ten mph. Suppose this car was 500 miles from New York City and traveling towards New York at its top speed.

1. How far from New York would the car be after five hours?

2. How far from New York would it be after twelve hours?

To write an expression, define a variable for the time and use this variable to write a rule for the car's distance from New York.

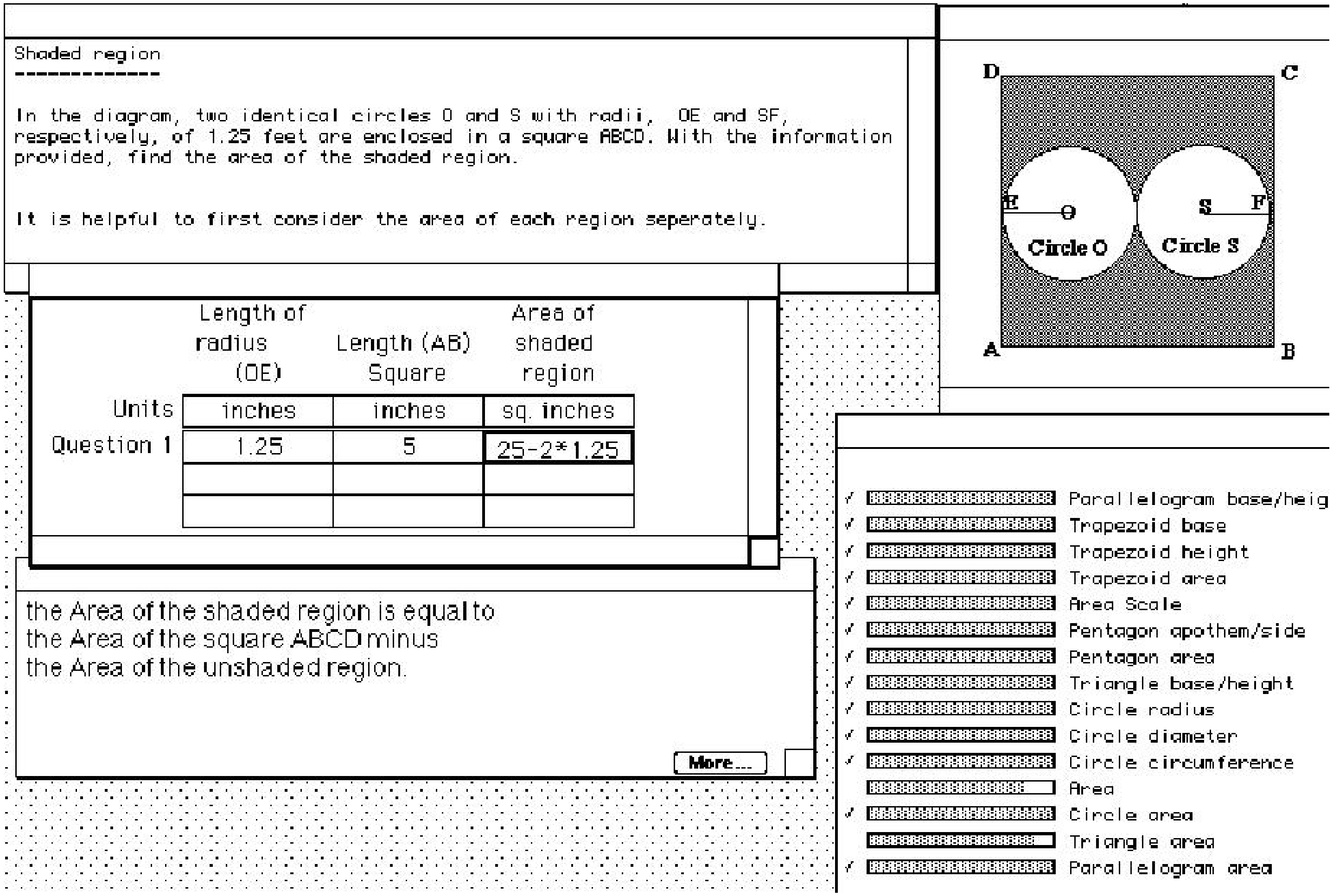
Worksheet

	TIME	DISTANCE
Unit	HOURS	MILES
1	5	450
2	12	380
Expression	T	500-10*T

Figure 2: The Carnegie Learning tutor that is an example of scaffolding that is both static and done ahead of time (i.e., "pre"). The student needs to do nine scaffolding steps before writing the final expression.

Static & Post-Scaffolding

An example of scaffolding that occurs after an error is shown in Figure 3. This is a screen shot of the PACT Geometry Tutor (Koedinger, Snyder & Aleven, 1998), and it shows an example of a student attempting to calculate the area of the shaded region shown in the figure. The student types in "25-2*2.125" when the correct answer is "25-2*(π*1.25^2)". The student made two errors, leaving out the squaring of the radius (1.25) as well as leaving out the π (i.e., used the radius instead of the area of a circle). Because two errors were made, the system failed to "trace" the student's answer, resulting in no buggy messages being available. When the student asked for a hint, the first hint the student received is shown in Figure 3. The hint told the student to calculate the area of the shaded region by subtracting the area of the unshaded region from the area of the square. This message was not helpful, since the student's answer already revealed that he knew how to do that.



Shaded region

In the diagram, two identical circles O and S with radii, OE and SF, respectively, of 1.25 feet are enclosed in a square ABCD. With the information provided, find the area of the shaded region.

It is helpful to first consider the area of each region seperately.

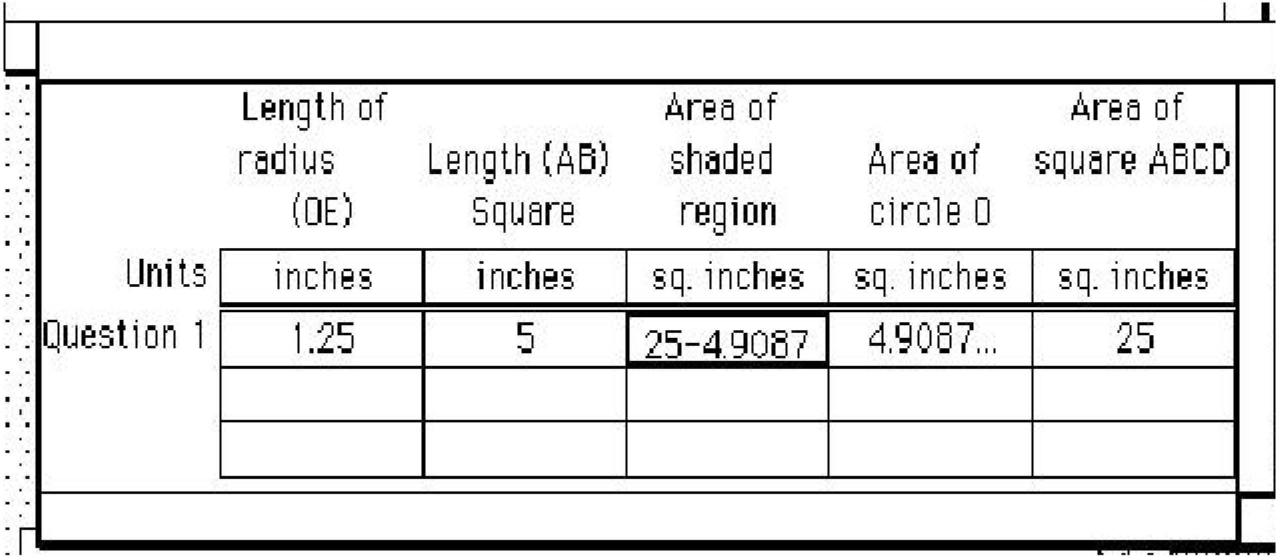
	Length of radius (OE)	Length (AB) Square	Area of shaded region
Units	inches	inches	sq. inches
Question 1	1.25	5	25-2*1.25

the Area of the shaded region is equal to
the Area of the square ABCD minus
the Area of the unshaded region.

✓ Parallelogram base/heig
✓ Trapezoid base
✓ Trapezoid height
✓ Trapezoid area
✓ Area Scale
✓ Pentagon apothem/side
✓ Pentagon area
✓ Triangle base/height
✓ Circle radius
✓ Circle diameter
✓ Circle circumference
Area
✓ Circle area
Triangle area
✓ Parallelogram area

Figure 3: The PACT Geometry Tutor showing a hint that is NOT context sensitive to the location of the error that the student made.

Therefore, the student asked for a second hint. The second hint read "You can add two columns from the table menu to compute the Area of the square ABCD and the Area of the unshaded region first." The student then used a menu to add two columns to the table, as shown in Figure 4. (The student decided add a column for the area of the circle as well as the area of the square, which was slightly different then what the hint suggested.) This suggestion to add columns is a type of scaffolding, which occurs after the student makes an error. However, the scaffolding in still not what I term *dynamic* because the student was presented with questions that were not sensitive to the error the student made. For instance, it suggested to the student to calculate the area of the square, when the student had already done that correctly. I will use the term *error localization* for behavior that is sensitive to where an error is made. The example just presented is an example of a system that did not do error localization.



	Length of radius (OE)	Length (AB) Square	Area of shaded region	Area of circle O	Area of square ABCD
Units	inches	inches	sq. inches	sq. inches	sq. inches
Question 1	1.25	5	25-4.9087	4.9087...	25

Figure 4: The table shown in Figure 3, after the student has added two columns.

If this system had asked the student to add a column **only** for the portion(s) that the student made an error on (i.e., the area of the circle) then I would call it *dynamic* scaffolding. The next section will present such an example.

Dynamic Scaffolding

What I term *dynamic* scaffolding has two important elements. Its occurs **after** an error has been made (i.e., "post".) Additionally, it does *error localization* in that it focuses the dialog only on the portions of the problem that the student did not do correctly. An example Ms. Lindquist doing dynamic scaffolding is shown below:

1. Tutor: *Debbie has two jobs over the summer. At one job, she bags groceries at Giant Eagle and gets paid 5 dollars an hour. At the other job she delivers newspapers and gets paid 7 dollars an hour. She works a total of 30 Hours a week. She works "g" hours bagging groceries. Write an expression for the total amount she earns a week.*
2. Student: **(30-g*7)+(g*5) [Should have been "(30-g)*7+g*5"]**
3. Tutor: No. But, "30-g" is correct for the hours delivering newspapers, and "g*5" is correct for the pay from the grocery. We will come back to the total pay per week. Let me try to break this down for you. What is the pay from the newspaper?

In this example of dynamic scaffolding, the key elements are that the tutor breaks down the problem **after** the student has made an error, as well as the fact that the system did the **error localization** by asking a question focused on the subgoal that was done incorrectly (i.e. the pay from the newspaper, which is (30-g)*7.)[49] Additionally, the positive feedback on the correct portions (i.e., "g*5" and "30-g") can be thought of as more *error localization,* which focused the student on the error, by telling him what portions do not contain the error.

Using a Tutorial Strategy

I now consider the two different types of tutorial strategies that can be combined with dynamic scaffolding. Future work is needed to determine the relative benefits of these two different dialog types.

Dynamic Scaffolding with a Knowledge Construction Dialog

The previous example we saw that dynamic scaffolding involved asking a new question focused on a subgoal that the student made an error on. In this next example, I will show that you can combine dynamic scaffolding with a knowledge construction dialog, that instead of asking a single question (related to a subgoal) the tutor uses a two-step plan (related to the same subgoal) to scaffold the student's rectifying of the error.

1. Tutor: *Debbie has two jobs over the summer. At one job, she bags groceries at Giant Eagle and gets paid 5 dollars an hour. At the other job, she delivers newspapers and gets paid 7 dollars an hour. She works a total of 30 Hours a week. She works "g" hours bagging groceries. Write an expression for the total amount she earns a week.*
2. Student: **(30-g*7)+(g*5) [Should have been "(30-g)*7+g*5"]**
3. Tutor: No. But, "30-g" is correct for the hours delivering newspapers, and "g*5" is correct for the pay from the grocery. We will come back to the total pay per week. Let me try to break this down for you. If "B" was the hours delivering newspapers then what would be the pay from the newspaper?
4. Student: **B*7**
5. Tutor: Correct. Now, lets try this question again: You are correct that "B*7 " is the pay from the newspapers. Now, can you state this in terms of the numbers given in the problem?
6. Student:**(30-g)*7**
7. Tutor: Correct. Now, lets try this question again: Write an expression for the total amount she earns in a week

This example shows that the tutor attacked the subgoal that the student made an error on with a two-step (lines 3 and 5) strategy (the "Introduce a Variable" strategy discussed in Chapter 4.)

Dynamic Scaffolding with a Knowledge Remediation Dialog

The only difference between a construction dialog and a remediation dialog is that a remediation dialog is used for a particular sort of error type. I will now present an example of a remediation dialog (not actually implemented by Ms. Lindquist). I will do so using the same problem and the same error we have just seen.

1. Tutor: *Debbie has two jobs … (the same as line 1 above)*
2. Student: **(30-g*7)+(g*5) [Should have been "(30-g)*7+g*5"]**

---

[49] Note, that in this example the tutor assumes that the student made an error of leaving out parentheses, rather than of thinking that "g" should be multiplied by 7 before subtracting the resulting quantity from 30.

3. Tutor: No. But, "30-g" is correct for the hours delivering newspapers, and "g*5" is correct for the pay from the grocery. We will come back to the total pay per week. Let me try to break this down for you. In your answer, what is the 7 dollars an hours being multiplied by? The number of hours bagging groceries or the number of hours delivering newspapers?
4. Student: **the hours delivering newspapers**
5. Tutor: Correct. Now consider the order of operations. Which do you do first, multiplication or subtraction?
6. Student: **multiplication**
7. Tutor: Correct. Now use that information to state the pay from the newspaper.
8. Student: **(30-g)*7**
9. Tutor: Correct. Now, lets try this question again: Write an expression for the pay from the

Notice that in this example, the questions the tutor asks are appropriate for a missing parentheses error. This example combines the error localization of dynamic scaffolding with the additional context-sensitive aspect of responding to the particular *error type* the student committed. Ms Lindquist does not currently have many remediation dialogs. The other remediation dialogs, outlined in the Chapter 4, point to other remediation dialogs that could be implemented.

Discussion:

Now that I have summarized some of the main features of Ms. Lindquist, I propose a few "lesion" studies as the subject of future work. Because intelligent tutoring systems are so costly to produce, it is important for intelligent tutoring system designers to know where the "biggest bang for the buck" is in order to decide what aspects to build into intelligent tutoring systems.

The first revolves around the question of when is it better to present students with scaffolding before they make an error, and when is it better to present them with dynamic scaffolding (after they make an error). Dynamic scaffolding can potentially save time for the student if they would get the problem correct without the scaffolding. On the other hand, if you know that student will make an error, it might be better to present them with the scaffolding ahead of time. Ms. Lindquist can be used to easily study this question, by creating a static scaffolding version to compare the to the current version that has dynamic scaffolding. Ms. Lindquist's architecture allows for a clear study of this use. Presumably, when students are first gaining competence in a skill, they could use static scaffolding, and as they start gaining competence, this scaffolding should be dynamic (i.e. it fades away).

A second issue is "How important is error localization in tutoring?" Maybe error localization is what is important in tutoring, and researchers should not waste too much time on building sophisticated dialog systems when all that is needed is error localization. More likely, it is that error localization is one of the key components, but is not sufficient by itself. Again, Ms. Lindquist can be used to study this by lesioning the components that allows Ms. Lindquist to ask questions focused on individual subgoals.

A third issue is "How important is being able to ask new questions of students?" Traditional model tracing tutors ask only rhetorical questions. A system that has just dynamic scaffolding will ask a new question for each subgoal that is done incorrectly. A system that has knowledge construction (or remediation) dialogs will ask 2-3 questions for each step. Again, a lesion study could be done by comparing just dynamic scaffolding with dynamic scaffolding combined with knowledge construction dialogs. Where is the biggest "bang for the buck"?

A forth issue is "How important are tutoring responses that are sensitive to not only where the error was made (i.e., error localization) but also the type of error made?" This is what knowledge remediation dialogs do. Because Ms. Lindquist currently has only has knowledge remediation dialog currently implemented, it could be that that it is missing the most useful benefit of being able to engage the students in a dialog. This thesis did show several other knowledge remediation dialogs that could be implemented, as well as showing how easy it is to do so. Future work could investigate the impact of such strategies on student learning.

Other Future Work

An object of future work is to run a study similar to the one presented in Chapter 5, but which controls for time. Since Ms. Lindquist's dialogs take additional time, I would expect the control condition to do more problems. Such a study would determine if the benefits of Ms. Lindquist are worth the additional time it takes.

Another aspect of future work is to compare Ms. Lindquist to the existing model tracing sold by Carnegie Learning, which has already been shown to be effective. However, there are multiple differences between the two systems, and not all of those differences are scientifically interesting.[50]

Another object of future study is how best to combine the tutorial strategies. It might be that some strategies are better as some times of problems compared to other strategies. For instance, maybe the "introduce variable" strategy works well on large problems and the "concrete articulation" strategy works well on small problems. One idea would be to use the "concrete articulation" strategy on one-operator sub-expression, and use the "introduce variable" strategy on sub-expression that contain more then one operator.

Future Work on the "Explain First" Strategy

Future work could also be done to provide more ways of breaking down problems. Consider that if a student makes an error on a symbolization question, the system has ways of breaking those problems down into simpler steps. In addition, with regards to the concrete articulation strategy, the system knows how to break down the QCOMPUTE and QEXPLAIN questions. Similarly, it would be nice to have ways to assists students on the explain question used in the "Explain in English" strategy. Given that this dissertation argues that, we should "ask rather then tell", I would like to be able to breakdown these questions further. There are several ways we could break these questions up into simple questions. For instance, we could do what our experienced human tutor did and get the student to first articulate the answer using just units (i.e., "Average speed is equal to miles divided by hours") and then --only after that was done-- get them to use the full noun phrases. This proposal seems to take us in the direction of adding natural language understanding, which would be a large project. However, there are also some approaches that we could use to avoid this hard problem. For instance, another idea is to ask for one piece at a time as in the following:

1. "What is one of the quantities you need to use to explain X"
2. "What is another quantity you need to use?"
3. "How can you combine these quantities together to find X".

Because Ms. Lindquist's architecture is easily extendable, these strategies would be easy to add.

---

[50] For example, the Carnegie Learning tutor helps students with solving equations, and graphing, none of which is covered by Ms. Lindquist. Additionally, Ms. Lindquist's curriculum has problems that are more complicated, and those are the same problems where dynamic scaffolding might prove to be most effective at saving time.

Dissertation Conclusion

I would like to end with a brief review of this dissertation's contributions. In Chapter 1, I argued that contrary to prior research, the main difficulty of symbolizing is not comprehension, or the presence of variables, but instead is due to the difficulty of producing symbols in the language of algebra. I hypothesized that symbolizing is like learning a foreign language where the difficulty is in learning how to produce symbols in the new language (i.e., algebra). This hypothesis was supported with two difficulty factors assessments, an analysis of the errors students made, and a successful transfer study. The transfer study showed that students could get better at symbolizing by practicing algebraic substitution. I argued that this result could be explained by realizing that these two skills were both using the grammar for algebraic expressions. In particular students were learning that you could treat an expression that same where you treated a number.

The cognitive science research of Chapter 1, was used in several ways in designing the Ms. Lindquist tutoring system. First, there is the incorporation of the *decomposition & substitution* tutorial strategy that is based on the transfer study in Chapter 1. This study argued that one reason students do poorly at symbolization is that they do not know the grammar for expressions, particularly when it comes to composing expressions together. The *decomposition & substitution* tutorial isolates the composition step (i.e., the substitution step) in the hope of improving student learning.

Interestingly, this grammar explanation can also be used unify a prior result by Koedinger & Nathan's (in peer review). This dissertation argued that students need to know the grammar of expressions in order to **produce** in the language of algebra, but student also need to know the grammar of expressions in order to **comprehend** expressions, as shown by Koedinger & Nathan. They compared a student's ability to do the problems shown in Table 1. Student did much better on both story problems and "word equations" than compared to equations. At least a portion of student's difficulties can be explained by the hypothesis that students did not know well the grammar for algebra expressions. Evidence in support of this is found in two places. First, students had a much harder time even getting started on the equation problems. There were three times as many students who left the equation problems blank as on the other two problem types. I take this as evidence that students had a harder time beginning the equation problems because they did not understand how to parse the equations. Secondly, this explanation is supported by the fact that there were many order of operation errors on the equation problems, but no such errors on the other two problems types, presumably because students were better at understanding the grammar (i.e. English grammar as opposed to the grammar for algebra expressions) in which the other problems were written in.

Problem Types	Example Problem
Story Problem	When Ted got home from his waiter job, he took the $81.90 he earned that day and subtracted the $66 he received in tips. Then he divided the remaining money by the 6 hours he worked and found When Ted got home from his waiter job, he took the $81.90 he earned that day and subtracted the $66 he received in tips. Then he divided the remaining money by the 6 hours he worked and found his hourly wage. How much per hour does Ted make?
"Word equation"	Starting with 81.90, if I subtract 66 and then divide by 6, I get a number. What is it?
Equation	Solve for x: $(81.90 – 66) / 6 = x$

Table 1: Koedinger and Nathan's Problem Types

A second way in which the cognitive science work influenced the tutoring system is in the "Convert the Problem into an Example to Explain" Strategy. This strategy is partially based on Corbett, McLaughlin, Scarpinatto, & Hadley (2000) counter-intuitive result that students can learn to symbolize by practicing translating algebraic expressions back into a verbal representation, as shown in Table 2. I argue that students improved at symbolizing when students had to interpret a component of an expression, because they are practicing the grammar rules for algebra expressions.

> Suppose you have $10 to spend on refreshments at a movie theater. A box of popcorn costs $2.00 at the snack bar and a beverage costs $1.50. You can use the following inequality to represents the different snack combinations you can afford.
>
> $$2x+1.5y<=10$$
>
> An example question: What does the term $2x$ represent in the situation?
> Example answer: *The total you spend on popcorn.*

Table 2: A problem (Corbett et al., 2000) where students practice how to translate expressions back into English.

A final way in which the cognitive science work influenced the tutoring system is in relation to the *concrete articulation* strategy. Since I showed that a major difficulty students have with symbolization problems is in articulating the steps they used to compute a quantity, this tutorial strategy isolates that step so that it can be addressed in more detail.

The insights from Chapter 1 were also helpful in Chapter 2, where I presented a model of tutoring, including dynamic scaffolding and tutorial strategies. This model was implemented in architecture, described in Chapter 3, which expanded the model-tracing paradigm by adding a tutorial model. In Chapter 4, I showed that this model was able to produce some of the features we observed in an experienced human tutor. Finally, in Chapter 5, I showed that this model could lead to increases in student learning.

References

Aleven, V., and Koedinger, K. R., (2000a) The need for tutorial dialog to support self-explanation. In the Proceedings of the AAAI 2000 Fall Symposium, Building Dialog Systems for Tutorial Applications. Technical Report FS-00-01. AAAI Press. Menlo Park, CA.

Aleven, V., and Koedinger, K. R (2000b). Limitations of student control: Do students know when they need help? In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, ITS 2000, edited by G. Gauthier, C. Frasson, and K. VanLehn. Berlin: Springer Verlag.

Anderson, J. R. (1993). *Rules of the Mind*. Hillsdale, NJ: Erlbaum.

Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring systems. *Science,* 228, 456-462.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995) Cognitive tutors: lessons learned. *The Journal of the Learning Sciences*, 4 (2), 167-207.

Anderson, J. R. & Jeffries (1985). Novice LISP errors: Undetected losses of information from working memory. *Human-Computer Interaction,* 22, 403-423

Anderson, J. R. & Pelletier, R. (1991) A developmental system for model-tracing tutors. In Lawrence Birnbaum (Eds.) *The International Conference on the Learning Sciences.* Association for the Advancement of Computing in Education. Charlottesville, Virginia (pp. 1-8).

Anderson, J. R., Reder, L. M., & Ritter, S. (manuscript in preparation). Algebraic slips: a working memory explanation.

Arroyo, I, Beck, J., Woolf, B., Beal, C., Schultz, K., (2000) Gender and cognitive differences in learning with respect to hint interactivity and symbolism. In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, ITS 2000, edited by G. Gauthier, C. Frasson, and K. VanLehn. Berlin: Springer Verlag. Lecture Notes in Computer Science no. 1839

Aziz, N. A., Pain, H., & Bna, P. (1995) Modeling and mending student's misconceptions in translating algebra word problems using a belief revision system in TAPS. In *The Proceedings of the World Conference on Artificial Intelligence in Education* (pp. 107-114.)

Bassok, M. (1990). "Transfer of domain-specific problem-solving procedures." *Journal of Experimental Psychology: Learning, Memory, & Cognition 16*(3): 522-533.

Beaton, A. E., Mullis, I. V. S., Martin, M. O., Gonzalez, E. J., Kelly, D. L. & Smith, T. A. (1996) *Mathematics Achievement in the Middle School Years: Third International Science and Mathematics Study(TIMMS)*. International Association for the Evaluation of Educational Achievement. Boston College.

Bloom, B. S., (1984) The 2-sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher* June , (pp. 4-16.)

Bobrow, D. G. (1968). Natural language input for a computer problem-solving system. In *Semantic Information Processing*. M. L. Minsky. Cambridge, MIT Press.

Brown, R., Dolciani, M, & Sorgenfrey, R., & Cole, W. (1990) *Algebra: Structure and Method.* Houghton Mifflin: Boston.

Bruner, J. (1983). *Child's Talk.* New York: Norton.

Carbonell, J. R. (1970) AI in CAI: Artificial-intelligence approach to computer assistance instruction. *IEEE Transactions on Man-Machine Systems 11(4)*:190-202

Cardelle-Elawar, M. (1992). Effects of Teaching Metacognitive Skills to Students with Low Mathematics Ability. *Teaching and Teacher Education 8*(2): 109-21.

Carpenter, T. P., et al. & National Council of Teachers of Mathematics Inc. Reston VA. (1981). *Results from the Second Mathematics Assessment of the National Assessment of Educational Progress*. Virginia**:** 167.

Carpenter, T. P. and J. M. Moser (1982). The development of addition and subtraction problem-solving skills. In *Addition and subtraction : a cognitive perspective*. T. P. Carpenter, J. M. Moser and T. A. Romberg. Hillsdale, N.J., L. Erlbaum Associates**:** 9-24.

Carpenter, T. P., Moser, J. M. & Romberg, T. A. (1982). *Addition and subtraction : a cognitive perspective*. Hillsdale, N.J., L. Erlbaum Associates.

Catrambone, R. (1996). Transferring and Modifying Terms in Equations. In Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society Hillsdale, NJ: Erlbaum. (pp. 301-305).

Catrambone, R. (1998). The subgoal learning model: Creating better examples so that students can solve novel problems. *Journal of Experimental Psychology: General*, 127 (4), 355-376.

Cedillo, T. (2001). Towards an algebra acquisition support system: A study based on using graphing calculators in the classroom. Received manuscript from the author who stated that this paper is scheduled to be published by April 2001 in *Mathematical Thinking and Learning*, Erlbaum.

Chi., M. T. H. (1996) "Constructing Self-Explanations and Scaffolded Explanations in Tutoring" Applied Cognitive Psychology, Vol 10,S33-S49.

Cho, B., Michael, J., Rovick, A., Evens, M. (2000) The analysis of multiple tutoring protocols. Appeared in Intelligent Tutoring Systems: 5th International Conference (Eds Gauthier, Frasson VanLehn) Springer, Lecture Notes in Computer Science no. 1839, pp. 212-221.

Clancey, W. J., (1982) Tutoring rules for guiding a case method dialog. In D. Sleeman & J. S. Brown (Eds.) *Intelligent Tutoring Systems* London: Academic Press. (pp. 201-226.)

Clement, J. (1982). Algebra Word problem solutions: Though process underlying a common misconceptions. *Journal for Research in Mathematics Education*, 13, 16-30.

Clement, J., Lochead, J. & Soloway, E. (1980). Positive effects of computer programming on students' understanding of variables and equations. National Conference of the Association for Computing Machinery.

Collins, A., & Brown, J. S. (1985). The computer as a tool for learning through reflection. In H. Mandl and Lesgold (Eds.), *Learning issues for intelligent tutoring systems*. New York: Springer.

Collis, K. F. (1975a). *A Study of Concrete and Formal Operations in School Mathematics: A Piagetian Viewpoint.* ACER Research Series 95. Australian Council for Educational Research, Victoria, Australia.

Collis, K. F. (1975b). *Cognitive Development and Mathematics Learning.* PMEW Paper, Chelsea College, University of London.

Conati, C., Larkin, J. and VanLehn, K. (1997) A computer framework to support self-explanation. In : du Bolay, B. and Mizoguchi, R.(Eds.) Proceedings of AI-ED 97 World Conference on Artificial Intelligence in Education. Vol.39, pp. 279-276, Amsterdam: IO Press.

Cooney, J. B. & Swanson, H. L. (1990). Individual differences in memory for mathematical story problems: Memory span and problem perception. *Journal of Educational Psychology 82*(3): 570-577.

Corbett, A. T., and Anderson, J. R., (1995) Knowledge decomposition and subgoal reification in the ACT programming tutor. in *Proceedings of Artificial Intelligence in Education* (pp. 469-476)

Corbett, A. T., McLaughlin, M., Scarpinatto, C., & Hadley, W. (2000) Analyzing and generating mathematical models: an algebra II cognitive tutor design study. In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, ITS 2000, edited by G. Gauthier, C. Frasson, and K. VanLehn. Berlin: Springer Verlag. Lecture Notes in Computer Science no. 1839.

Core, M. G., Moore, J. D., and Zinn, C.. (2000) Supporting constructive learning with a feedback planner. In the *Proceedings of the AAAI 2000 Fall Symposium, Building Dialog Systems for Tutorial Applications*. Technical Report FS-00-01. AAAI Press.Menlo Park, CA. Cummins, D. D., Kintsch, W., Reusser, K. & Weimer, R. (1988). The role of understanding in solving word problems. *Cognitive Psychology*, 20, (pp. 405-438.)

Cummins Dellarosa, D. (1991). Children's interpretations of arithmetic word problems. *Cognition & Instruction 8*(3): 261-289.

Cummins, D. D., Kintsch, W., Reusser, K. & Weimer, R. (1988). The role of understanding in solving word problems. *Cognitive Psychology*, 20, 405-438.

del Solato, T. & du Boulay, B. (1995) Implementation of motivational tactics in tutoring systems. *Journal of Artificial Intelligence in Education* 6(4) (pp. 337-378.)

DiEugenio, B, Jordan, P. W., Thomason, R. H., & Moore, J. D. (in press) The agreement process: An empirical investigation of human-human computer-mediated dialogues. *International Journal of Human-Computer Studies.*

Freedman, R. (2000 to appear) Using a reactive planner as the basis for a dialogue agent. In *Proceedings of the Thirteenth Florida Artificial Intelligence Research*

Freeman, R. & Evens, M. W. (1996) Generating and revising hierarchical multi-turn text plans in an ITS. In C. Frasson, G. Gautheir and A. Lesgold (Eds.), *Intelligent Tutoring Systems: Proceedings of the 1996 Conference* (pp. 632-640). Berlin: Springer.

Gertner, A. S., & VanLehn, K. (2000). Andes: A coached problem solving eveirmrment for physics. In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, ITS 2000, edited by G. Gauthier, C. Frasson, and K. VanLehn. Berlin: Springer Verlag.

Gluck, K. (1999) Eye movements and algebra tutoring. Ph.D. dissertation. Psychology Department, Carnegie Mellon University.

Graesser, A.C., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R., & the TRG (1999). AutoTutor: A simulation of a human tutor. *Journal of Cognitive Systems Research*, 1, 35-51.

Grosz, B. J. & Sidner, C. L. (1986) Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175-204.

Hall, R., Kibler, D., Wenger, E. & Truxaw, C. (1989) Exploring the episodic structure of algebra story problem solving. *Cognition and Instruction* 6(3) (pp. 223-283.)

Heffernan, N. T., & Koedinger, K. R.(1997) The composition effect in symbolizing: the role of symbol production versus text comprehension. In *Proceeding of the Nineteenth Annual Conference of the Cognitive Science Society* (pp. 307-312). Hillsdale, NJ: Lawrence Erlbaum Associates.

Heffernan, N. T., & Koedinger, K. R. (1998) A developmental model for algebra symbolization: The results of a difficulty factors assessment. *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, (pp. 484-489) Hillsdale, NJ: Lawrence Erlbaum Associates.

Hegarty, M., Mayer, R. E. & Monk, C. A. (1995). Comprehension of arithmetic word problems: A comparison of successful and unsuccessful problem solvers. *Journal of Educational Psychology 87*(1): 18-32.

Hinsley, D. A., Hayes, J. R. & Simon, H. A. (1977). From words to equations: meaning and represntation in algebra word problems. In *Cognitive Process in Comprehension*. M. A. Just and P. A. Carpenter. Hillsdale, NJ, Erlbaum**:** 89-106.

Hoz, R. & Harel, G. (1990). Higher order knowledge involved in the solution of algebra speed word problems. *Journal of Structural Learning 10*(4): 305-328.

Katz, S., Lesgold, A., Hughes, E., Peters, D., Eggan, G., Gordin, M., & Greenberg, L. (1998). Sherlock 2: An intelligent tutoring system built on the LRDC framework (pp. 227-258). In C. P. Bloom & R. B. Loftin (Eds.), *Facilitating the development and use of interactive learning environments*. Mahwah, NJ: Erlbaum.

Judd, T. P. & Bilsky, L. H. (1989) Comprehension and memory in the solution of verbal arithmetic problems by mentally retarded and nonretarded individuals. *Journal of Educational Psychology* 18(4), 541-546.

Kintsch, W. (1977). On comprehending stories. In *Cognitive Process in Comprehension*. M. A. Just and P. A. Carpenter. Hillsdale, NJ, Erlbaum**:** 33-62.

Kintsch, W. (1991). A theory of discourse comprehension: Implications for a tutor for word algebra problems. In *Learning and instruction: European research in an international context*. M. Carretero, M. L. Pope and et al. Oxford, England UK, Pergamon Press. 3**:** 235-253.

Kintsch, W. (1998). *Comprehension : a paradigm for cognition*. Cambridge, U.K. ; New York, Cambridge University Press.

Kintsch, W. & Greeno, J. G. (1985). Understanding and solving word arithmetic problems. *Psychological Review 92*(1): 109-129.

Koedinger, K. R., & Anderson, J. R. (1998). Illustrating principled design: The early evolution of a cognitive tutor for algebra symbolization. In *Interactive Learning Environments*, 5, 161-180.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1995). Intelligent tutoring goes to school in the big city. In *Proceedings of the 7th World Conference on Artificial Intelligence in Education*, (pp. 421-428). Charlottesville, VA: Association for the Advancement of Computing in Education.

Koedinger, K. R., & MacLaren, B. (1997). Implicit strategies and errors in an improved model of early algebra problem solving. In *Proceedings of the Nineteenth Annual Meeting of the Cognitive Science Society* (pp. 382-7). Mahwah, NJ: Erlbaum.

Koedinger, K. R. & Nathan, M. J. (submitted to). The real story behind story problems: Effects of representations on quantitative reasoning. Submitted to peer review.

Koedinger, K. R., Snyder, J., & Aleven, V. (1998). PACT Geometry Cognitive Tutor. Software created at the Pittsburgh Advanced Cognitive Tutor (PACT) Center. Carnegie Mellon University. Pittsburgh, PA.

Kuchemann, D. (1981) Algebra. In Hart (Ed.) *Children understanding of Mathematics 11 - 16*. John Murray.

Lebiere, C., Anderson, J. R., & Reder, L. M. (1994). Error Modeling in the ACT-R production system. In *Proceedings of the 16th Annual Conference of the Cognitive Science Society*, pp. 555-559. Erlbaum.

LeBlanc, M. D., & Weber-Russell, S. (1996). Text integration and mathematical connections: a computer model of arithmetic word problem solving. *Cognitive Science* 20,357-407.

Lepper, M. R., Aspinwall, L. G., Mumme, D. l., & Chabay, R. W. (1990). Self-perception and social-perception process in tutoring: Subtle social control strategies of expert tutors. In (J. M. Olson, & M. P. Zanna Eds.), *Self-inference processes: The Ontario Symposium*. Hillsdale, NJ: Erlbaum. (pp. 217-237).

Lepper, M. R., Drake, M. F., O'Donnell-Johnson, T. (1997) "Scaffolding techniques of expert human tutors. In K. Hogan & M. Pressley (Eds.) *Scaffolding Student Learning: Instructional approaches and Issues*. Cambridge MA: Brookline Books (pp. 108-144.)

Lepper, M. R., & Malone, T. W. (1987) Intrinsic motivation and instructional effectiveness in computer-based education. in R. E. Snow & M. J. Farr (Eds.) *Aptitude, learning, and instruction: III.* Cognitive and affective process analyses. Hillsdale, N. J.: Erlbaum.

Lewis, A. B. & Mayer, R. E. (1987). Students miscomprehension of relational statements in arithmetic word problems. *Journal of Educational Psychology, 79(4)*, 363-317.

MacGregor, R. (1991) Using a description classifier to enhance deductive inference. In *Proceedings of the Seventh IEEE Conference on AI Applications*, Miami, 141-147

Mayer, R. E. (1981). Frequency Norms and Structural Analysis of Algebra Story Problems in Families, Categories, and Templates. *Instructional Science 10*, 135-175.

Mayer, R. E. (1982). Memory for algebra story problems. *Journal of Educational Psychology 74*(2): 199-216.

Mayer, R. E., Lewis, A. B. & Hegarty, M. (1992). Mathematical misunderstandings: Qualitative reasoning about quantitative problems. In *The nature and origins of mathematical skills*. J. I. D. Campbell and et al. Amsterdam, Netherlands, North-Holland**:** 137-153.

McArthur, D., Stasz, C., & Zmuidzinas, M. (1990) Tutoring techniques in algebra. *Cognition and Instruction*. 7 (pp. 197-244.)

Merrill, D. C., & Reiser, B, J. (1994). Scaffolding effective problem solving strategies in interactive learning environments. In A. R. A. K. Eiselt (Ed.), *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* (pp. 629-634). Hillsdale, NJ: Earlbaum.

Merrill, D. C., Reiser, B. J, Merrill, S. K., & Landes, S. (1995) Tutoring: guided learning by doing. *Cognition and Instruction, 13(3)* (pp. 315-372.)

Merrill, D. C., Reiser. B. J., Ranney, M. and Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *Journal of the Learning Sciences 2*(3): 277-305.

Moore, J. D. (1993) What makes Human Explanations effective? In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society* Hillsdale. Hillsdale, NJ: Erlbaum. (pp. 131-136).

Moore, J. D. (1996) Discourse generation for instructional applications: Making computer-based tutors more like humans. *Journal of Artificial Intelligence in Education, 7(2),* 118-124

Moore, J. D. & Mittal, V. O. (1996) Dynamically generated follow-up questions. *IEEE Computer*, 75-86.

Murray, T. (1998). Authoring instructional expertise in knowledge based tutors. *Instructional Science 26*: 263-280.

Mwangi, W. & Sweller, J. (1998). Learning to solve compare word problems: The effect of example format and generating self-explanations. *Cognition & Instruction 16*(2): 173-199.

Nathan, M. J., Kintsch, W. & Young, E. (1992). A theory of algebra-word-problem comprehension and its implications for the design of learning environments. *Cognition & Instruction 9*(4): 329-389.

Nathan, M. J. & Koedinger, K. R. (2000a). An investigation of teachers' beliefs of students' algebra development. *Cognition & Instruction 18*(2): 209-237.

Nathan, M. J. & Koedinger, K. R. (2000b). Teachers' and researchers' beliefs about the development of algebraic reasoning. *Journal for Research in Mathematics Education 31*(2): 168-190.

National Council of Teachers of Mathematics (2000) *Principles and Standards for School Mathematics* NCTM: Reston, VA.

Ohlsson, S. (1986) Some principles for intelligent tutoring. *Instructional Science, 17*, 281-307.

Paige, J. M. & Simon, H.(1966). Cognitive process in solving algebra word problems. in B. Kleinmentz (Ed.*) Problem Solving: Research, method and theory*. New York: John Wiley & Sons. Republished in H. A. Simon (Ed.) *Models of Thought.* New Haven, Yale University Press, 1979.

Passolunghi, M. C., Cornoldi, C. & De Liberto, S. (1999). Working memory and intrusions of irrelevant information in a group of specific poor problem solvers. *Memory & Cognition 27*(5): 779-790.

Payne, S. J., & Squibb, H. R. (1990) Algebra mal-rules and cognitive accounts for error. *Cognitive Science, 14*, 445-481.

Pelletier, Ray. (1995) The Goal Driven Production System Tertl and its Abstract Machine. Masters Thesis as part of the Computational Linguistics Program at Carnegie Mellon University. Note: The Computational Linguistics Program was incorporated in the Language Technology Institute.

Person, N. K., & Graesser, A. (1999) Evolution of discourse in cross-age tutoring. In O'Donnell and King (Eds.) *Cognitive perspectives on peer learning.* (pp. 69-86). Mahwah, NJ: Erlbaum.

Piaget, J. (1970) *Genetic Epistemology.* Norton: New York.

Popescu, O., V., and Koedinger, K. R., (2000) Towards understating geometry explanations In the Proceedings of the AAAI 2000 Fall Symposium, Building Dialog Systems for Tutorial Applications. Technical Report FS-00-01. AAAI Press. Menlo Park, CA.

Putnam, R. T., (1987) Structuring and adjusting content for students: A study of live and simulated tutoring of addition. *American Educational Research Journal, 24.* 13-48.

Rapp, C. W. (1986) ALGEBRA READER: an expert algebra world problem reader. Technical Report #86-30-6. Dept. of Computer Science. Oregon State University.

Reed, S. K. (1999). *Word problems : research and curriculum reform*. Mahwah, N.J., Lawrence Erlbaum Associates.

Reed, S. K. & Ettinger, M. (1987). Usefulness of tables for solving word problems. *Cognition & Instruction 4*(1): 43-58.

Reusser, K. (1989). From text to situation to equation: Cognitive simulation of understanding and solving mathematical word problems. In *Learning and Instruction*. H. Mandl, E. DeCorte, N. Bennett and H. F. Friedrich. Oxford, Pergamon Press**:** 477-498.

Reusser, K. (1996). From cognitive modeling to the design of pedagogical tools. In *International perspectives on the design of technology-supported learning environments*. S. D. C. Vosniadou, Erik et al. Mahwah, NJ, Lawrence Erlbaum Associates.

Reiser, B. J., Beekelaar, R., Tyle, A., & Merrill, D. C. (1991). Gil: Scaffolding learning to program with reasoning-congruent representations. In L. Birnbaum (Ed.), *The International Conference of the Learning Sciences: Proceedings of the 1991 Conference* (pp. 382-388). Charlottesville, NC: Association for the Advancement of Computers in Education.

Reiser, B. J., Kimberg, D. Y., Lovett, M. C. & Ranney, M. (1992). Knowledge representation and explanation in GIL, an intelligent tutor for programming. In *Computer-assisted instruction and intelligent tutoring systems: Shared goals and complementary approaches*. J. H. Larkin, R. W. Chabay and et al. Hillsdale, NJ, USA, Lawrence Erlbaum Associates**:** 111-149.

Renkl, A. (1997) Learning from worked examples: A study of individual differences. *Cognitive Science, 21(1),* 1-30.

Rickel, J., Ganeshan, R., Lesh, N., Rich, C. & Sidner, C. L. (2000). Task-oriented tutorial dialogue: issues and agents. AAAI Fall Symposoum on Building Dialogue Systems for Tutorial Applications, Cape Cod, MA, AAAI Press.

Riley, M. S. and Greeno, J. G. (1988). Developmental analysis of understanding language about quantities and of solving problems. *Cognition and Instruction,* 5(1), 49-101.

Rose, C. P., & Lavie, A. (in press). Balancing robustness and efficiency in unification augmented context-free parsers for large practical applications. In J. C. Junqua & G. V. Noord (Eds.), *Robustness in Language and Speech Technology* : Kluwer Academic Press

Rosnick, P. & Clement, J. (1980). Learning without understanding: The effects of tutoring strategies on algebra misconceptions. *Journal of Mathematical Behavior,* 3(1), 3-27.

Roth, W.-M. (1996). Where is the context in contextual word problems?: Mathematical practices and products in Grade 8 students' answers to story problems. *Cognition & Instruction 14*(4): 487-527.

Sack, W. & Soloway, E. (1992). From PROUST to CHIRON: Intelligent Tutoring System Design as Iterative Engineering Intermediate Results are Important! (pp. 239-277), In *Computer Assisted Instruction and Intelligent Tutoring Systems: Shared Issues and Complementary Approaches*. In J.H. Larkin & R. W. Chabay (Eds.) Hillsdale, NJ: Erlbaum.

Sakamoto, M. (1999). The evaluation of a computerized tutoring program for arithmetic word problem solving. *Psychologia: An International Journal of Psychology in the Orient 42*(1): 40-50.

Schwarz, B. B., Nathan, M. J. & Resnick, L. B. (1996). Acquisition of meaning for arithmetic structures with the Planner. In *International perspectives on the design of technology-supported learning environments*. S. E. D. C. Vosniadou, Erik et al. Mahwah, NJ, Lawrence Erlbaum Associates.

Sfard, A., & Linchevski, L. (1993). The gain and the pitfalls of reification- the case of algebra. *Educational Studies in Mathematics,* 00: 1-38.

Shulman, L. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher, 15,* 4-14.

Shute, V. J. (1995) Intelligent tutoring systems: past, present, and future. In D. Jonassen (Ed), *Handbook of Research on Educational Communications and Technology*. Scholastic Publications.

Singley, M. K. (1990). Reification of goal structures in a calculus tutor: Effects on problem solving performance. *Interactive Learning Environments, 1*, 102-123.

Sleeman, D. (1982). Assessing aspects of competence in basic algebra. In Sleeman and Burton (Eds) *Intelligent Tutoring Systems*, Academic Press, London.

Sleeman, D. (1984) An attempt to understand student's understanding of basic algebra. *Cognitive Science, 8*, 387-412.

Stern, E. (1993). What makes certain arithmetic word problems involving the comparison of sets so difficult for children. *Journal of Education Psychology,* 85(1),7-23.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12, 257-285.

Sweller, J. & Cooper, G.. (1985) Use of worked examples as a substitute for problem solving in algebra learning. *Cognition and Instruction*, 2, 58-89

van Dijk, T. A. (1977). Semantic macro-structures and knowledge frames in discourse comprehsion. In *Cognitive Process in Comprehension*. M. A. Just and P. A. Carpenter. Hillsdale, NJ, Erlbaum**:** 3-32.

van Dijk, T. A. & Kintsch, W. (1983). *Strategies of discourse comprehension*. New York, Academic Press

VanLehn, K. (1998). Analogy events: How examples are used during problem solving. *Cognitive Science, 22* (3), 347-388

VanLehn, K, Anderson, J., Ashley, K., Chi. M., Corbett, A., . Koedinger, K., Lesgold, A., Levin, L., Moore, M., and Pollack, M., (1998a) NSF Grant 9720359. *CIRCLE: Center for Interdisciplinary Research on Constructive Learning Environments.* NSF Learning and Intelligent Systems Center. January, 1998 to January, 2003.

VanLehn, K. Freedman, R., Jordan, P., Murray, C., Osan, R., Ringenberg, M., Rose, C., Schulze, K., Shelby, R., Treacy, D., Weinstein, A. and Wintersgill, M. (2000) Fading and deepening: The next steps for Andes and other model-tracing tutors. In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, ITS 2000, edited by G. Gauthier, C. Frasson, and K. VanLehn. Berlin: Springer Verlag.

VanLehn, K., Siler, S., Murray, C. & Bagget, W. (1998b) What makes a tutorial event effective? In: M. A. Gernsbacher & S. Derry (Eds.) *Proceedings of the Twenty-first Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Erlbaum. pp. 1084-1089

Van Marcke, K. (1998) GTE: An epistemological approach to instructional modeling. *Instructional Science* 26, 147-191.

Van Marcke, K. & Vedelaar, H. (1995) Learner adaptivity in generic instructional strategies. In the proceedings of the *Artificial Intelligence in Education* conference, Washington, August, pp.323-333

Woolf, B (1988) Intelligent tutoring systems: a survey. in H. Schrobe and AAAI (Eds.) *Exploring Artificial Intelligence*. Morgan Kaufmann (pp. 1-43.)

Woolf, B & McDonald, D. (1984). Building a computer-tutor: design issues. IEEE Computers. September, 61-73.

Zhu, X. & Simon, H. A. (1987). Learning mathematics from examples and by doing. *Cognition & Instruction 4*(3): 137-166.

Appendices

Appendix A: The Cover Stories (Problem Situations) For DFA #1

John and his wife Beth have been saving for 7 months to give the 5 children presents for the Holidays. John has saved 972 dollars for presents and Beth has saved "b" dollars. They give each child the same amount. Write an expression for how much each child gets.

Ms. Lindquist, a math teacher, teaches 5 classes. Ms. Lindquist teaches 62 girls. Ms. Lindquist teaches "f" fewer boys than girls. Write an expression for how many students Ms. Lindquist teaches.

Sue made 72 dollars by washing 6 cars to buy Holiday presents. She decided to spend "m" dollars on a present for her Mom and then use the remainder to buy presents for each of her 4 sisters. She will spend the same amount on each sister. How much can she spend on **each** sister?

Mary opened a new music store. She got CDs delivered on her first day. She got 5 truck loads of CDs delivered. Each truck that arrived dropped off 12 boxes. Each box she received had "c" CDs. She sold CDs for 12 dollars each. How many CD were delivered that first day?

Bob left at 3 P.M. and drove 550 miles from Boston to Pittsburgh to visit his grandmother. Normally this trip takes him "h" hours, but on Tuesday there was little traffic and he saved 2 hours. What was his average driving speed?

Ann is in a rowboat in a lake. She is 800 yards from the dock. She then rows for "m" minutes back towards the dock. Ann rows at a speed of 40 yards per minute. Write an expression for Ann's distance from the dock.

Mike starts a job at McDonald's that will pay him 5 dollars an hour. Mike gets dropped of by his parents at the start of his shift but he takes a 10 minute taxi ride home that costs him 7 dollars. Mike works a "h" hour shift. After taking into account his taxi ride, write an expression for how much he makes in one night?

Mark went to the store to buy 4 jackets that cost "d" dollars each. When he got there the store was having a sale of 1/3 off the usual prices. How much did each jacket cost him?

Appendix B: Test for Transfer Study

Name: ____________________ Teacher :____________________
Period :____________

Your answer will be answer expression like “4+3x”, not a number like “15”. Put a circle around your answer.

1) Ann is in a rowboat in a lake. She starts out 800 yards from the dock. She then rows part of the way back toward the dock for "m" minutes at a speed of 40 yards per minute. She then rests. How far is Ann from the dock now?

2) Ms. Lindquist is a math teacher. Ms. Lindquist teaches 62 girls. Ms. Lindquist teaches "f" fewer boys than girls. Write an expression for how many students Ms. Lindquist teaches.

3) John is a shopkeeper and this week he sold 12 oranges and “a” apples. John also has many children that come to his store to buy candy and he sold 3 times as many Snickers bars as he sold pieces of fruit. How many Snickers bars did he sell?

4) Sue made 72 dollars by washing cars to buy Holiday presents. She decided to spend "m" dollars on a present for her Mom and then use the remainder to buy presents for each of her 4 sisters. She will spend the same amount on each sister. How much can she spend on each sister?

5) Debbie has two jobs over the summer. At one job she bags groceries at Giant Eagle and gets paid 5 dollars an hour. At the other job she delivers newspapers and gets paid 7 dollars an hour. She works a total of 30 hours a week. She works "b" hours bagging groceries. Write an expression for the total amount she earns in a week.

Composing Skill:
Example

```
      Step 1:     21 - 6 = 15
      Step 2:     15 / 3  = 5
      Write the composed expression   __________.
      When you put the two steps together you get that the answer is
(21-6)/3

1)    Step 1:     10 - 3 = 7
      Step 2:     4 * 7  = 28
      Write the composed expression   ___________
      Hint: Show how to get 28 from the numbers 10, 3 and 4.

2)    Step 1:     x + 3 = A
      Step 2:     5 / A
      Write the composed expression   ___________

3)    Step 1:     f - g = X
      Step 2:     X / X
      Write the composed expression   ___________

4)    Step 1:     5 * 3   = X
      Step 2:     3 * 4   = Y
      Step 3:     X + Y   = 27
      Write the composed expression   ___________
```

5) Step 1: $x + y = A$
Step 2: $5 / A = B$
Step 3: $A - B$
Write the composed expression ___________

Appendix C: A Protocol of an Experienced Tutor

In the following transcription the student’s written remarks are indicted in brackets. The tutor is an experienced current middle school mathematics teacher. The student is a seventh grade male student who is a student in the tutor’s classroom. The student had a list of problems in front of him and each problem’s text is reprinted (underlined) when the student reads the problem. The student had a blank sheet of paper on which he wrote his answers. Generally, his paper included only attempts at symbolization with a few accompanying words possibly indicating the units. The session lasted approximately one hour. The session consisted of 17 problems, 8 of which the student answered correctly on the first try. This transcription was made from a video tape of the session. Pauses are indicated with colons and one colon indicates about one half of a second pause.

2. TUTOR [Opening remarks and asks student to read out aloud and begin]
3. STUD [Reads problem] Mike starts a job at McDonald's that will pay him 5 dollars an hour. Mike gets dropped off by his parents at the start of his shift. Mike works a "h" hour shift. Write an expression for how much he makes in one night?
4. [writes “h*5=how much he makes”]
5. TUTOR That’s' right number. #2

6. STUD [reads problem] Mary opened a new music store. She got CDs delivered on her first day. She got 5 truck loads of CDs delivered. Each truck that arrived dropped off 12 boxes. Each box she received had "c" CDs. She sold CDs for 11 dollars each. How many CDs were delivered that first day?
7. [writes “5*12*c=# of CD’s”]
8. TUTOR OK

9. STUD [reads problem] Ann is in a rowboat in a lake that is 2400 yards wide. She is 800 yards from the dock. She then rows for "m" minutes back towards the dock. Ann rows at a speed of 40 yards per minute. Write an expression for Ann's distance from the dock.
10. [writes “800-11*40”]
11. TUTOR OK I know there are some harder ones in here. :::::
12. TUTOR Read out aloud so we know what's wrong.

14. STUD [reads problem] Ms. Lindquist is a math teacher. Ms. Lindquist teaches 62 girls. Ms. Lindquist teaches "f" fewer boys than girls. Write an expression for how many students Ms. Lindquist teaches.
15. [writes “62+(62-f)= # of students”]
16. TUTOR Good. All right. Number of students. Cruising!

17. STUD [reads problem] Mark went to the store to buy 4 jackets that cost "d" dollars each. When he got there the store was having a sale of 1/3 off the usual prices. How much did each jacket cost him?::::::
18. STUD Are the jackets all the same price?
19. TUTOR Yup. Each.:::::
[writes "(d-2/3)*4="]:::::::::::
20. TUTOR What's that?[points at the subtraction sign] Is that a subtraction?
21. STUD That's two thirds
22. TUTOR You mean that subtraction?
23. STUD Yeah, that’s a subtraction of two thirds.
24. TUTOR OK. Can you.. Is two thirds.. What does two-thirds represent?
25. STUD 33 percent
26. TUTOR Right.. Or.. I mean. Not, Not, what is it in percent but what does it represent?
27. STUD 2/3 of the whole price.
28. TUTOR Is that what you are subtracting?
29. STUD Yeah
30. TUTOR Are you subtracting two thirds of the whole price?
31. STUD Yes
32. TUTOR What if.. What if the jacket costs 9 dollars, How much would you be subtracting?
33. STUD Wait hold on ::: Oh that would be wrong.
34. [crosses out previous line and writes "(d-33%)*4 = money spent"]:::::::::::::::::::::::::::::
35. TUTOR Um. So if instead of "d" dollars it cost ten dollars, what would the answer be? So if it cost 10 dollars what would the answer be? Not ten dollars. Ten dollars is bad; what if it cost 9 dollars? What would the answer be.
36. STUD Three. No- it would be six dollars.
37. TUTOR It would cost six dollars. Is your expression gonna get that? Really, what are they actually asking for?
38. STUD Um.. How much did each jacket cost?
39. TUTOR Right.
40. STUD Then this would be...
41. TUTOR That's not necessary [Tutor points to the "4"]. So would your expression work?
42. STUD [crosses out the incorrect “4” from his answer of “(d-33%)*4”] Yeah, I think so
43. TUTOR Try it. How would you get 6 using your expression. Because your expression should give me that answer that you know it’s right.

44. STUD It would be nine minus 33% equals ::::[writes "9-33%=6"]:::: 33% is three, yeah, so the number would be 6.

45. TUTOR So how did you get that three?

46. STUD Well one third of nine is three.

47. TUTOR Right. You have to tell your reader that. That's what your expression is all about. It’s telling the person to do that. You haven't told the person to do that. That you.. You know instinctively to do it but you haven't told the person to do that

48. STUD So I should right 33% percent as one third? [writes "33%=1/3"]

49. TUTOR Well.. But are you just gonna subtract one third or are you gonna subtract.33333 Is that what you are subtracting?

50. STUD Well I am only subtracting like :::

51. TUTOR Yeah! No! - so what are you really subtracting. How did you calculate that?

52. STUD Well. :::::::::::::::::: Well if the whole thing is nine.

53. TUTOR Ah uhm

54. STUD And making 33% plus 33% ,33%, equal 100%

55. TUTOR Equals one hundred PERCENT

56. STUD Uhm. Yeah, 33% could equal three

57. TUTOR How did you calculate that? ::: Just doing that little piece, what would you do to get that three. What did you do in your head?

58. STUD I transfer 33 to 1/3 and

59. TUTOR [interrupts] Right- oh- you could have. And you never really needed the 33, you could have just used the third.

60. STUD [continuing] and divided nine by three which equals three,

61. TUTOR Which equals three.

62. STUD But I am not using one I am using two

63. TUTOR Right

64. STUD equals 6 [wrote 9/3=3*2=6]

65. TUTOR So, So you could.. So you did. Is that how you got your six? Oh did you get your six this[points to the expression "9-33%"] way. This is different. Here you are using subtraction and here you multiplied by two. So you need to pick one way or the other and write a complete expression

66. STUD OK.

67. TUTOR Because either way is fine

68. TUTOR We could even right both. Let’s write both. Let’s write an expression this way [points at “9/3=3*2=6”] and write an expression this way[points at “9-33%”].

69. STUD OK.

70. TUTOR But you have got to tell your reader to do this thing [points at “9/3=3*2=6”], because that's the thing you forgot to tell your reader.

71. STUD OK. ::::

72. TUTOR The person who is gonna use your formula, because you are making a formula.

73. TUTOR Because if somebody else came along, and used this they would go nine- minus 33%, or nine minus point 333. What is nine minus 1/3?

74. STUD Nine minus one third is 2/3

75. TUTOR No. Nine minus one third is eight and two thirds. Right?

76. STUD Well, one third.... So I have to mention that minus, Um, one third, of whole number

77. TUTOR Of what whole number

79. STUD Of the whole

80. TUTOR Right which is?

81. STUD [wrote "9-1/3 of whole=2/3"]

82. TUTOR Yeah, exactly, which is nine. Which actually is what in the original problem?

83. STUD Um, "D"

84. TUTOR "D", So how would we write the expression this way?

85. STUD This way?

86. TUTOR Yup, so skip a line...with using d. now write the expression

87. STUD Minus ::::::::[wrote “d-1/3=2/3="]:: equal :::2/3 :::which equals :::I don't know

88. TUTOR Nine minus one third. Does that give you the answer

89. STUD well no

90. TUTOR No, because that gives you eight and two thirds

91. STUD Yeah

92. TUTOR What do you have to do. Because if you have to do it you have to right it down for your expression?

93. STUD Say, it’s of the whole number

94. TUTOR of the whole number. You need to write that. Let’s get rid of that by rewriting d minus :::::: and of ::: and of means what

95. [writes "d-1/3 of whole number"]

96. STUD Times

97. TUTOR Times! So you can use one of your minus, plus [operators I guess]:::: So now let’s rewrite this one. This one looks great, but let’s rewrite it using just letters and times symbols instead of the words. "D minus"

98. STUD I think it.. Would it be that way?

99. [writes "d-1/3"]

100. TUTOR Ahhum, But don't forget the, That's only this part. That's one third of the whole number. And then you got to go. Don't forget this part [points at the "d-" I assume] "d" minus

101. STUD Where does that go in? [that I presume refers to the "d-"]

102. TUTOR Where do you think it goes in?

103. STUD I would just disregard that part

104. TUTOR Well what is this gonna give you[points at d*1/3 I think] Let’s say d is nine, what's this gonna give you

105. STUD Um. 6

106. TUTOR One third of nine?

107. STUD No, it’s gonna give me three

108. TUTOR It’s gonna give you three, Which you know is not the right answer. Right? What do you want to do with that three?

109. STUD Times the other two

110. TUTOR Oh- OK

111. STUD I means d times 2 equals 6 [writes =d*2=6]

112. TUTOR Well what's d? D is nine.

113. STUD Oh yeah

114. TUTOR You don't want to multiple d times two, what do you want to multiple by times two

115. STUD Nine...if you are using...

116. [drops pencil in exasperation and holds head]

117. TUTOR Well let’s go back to the third. This third, When you right 33% that's just the percent, that's not the quantity. You want to subtract the quantity, not the percent, or the fraction. You don't want to subtract a third you want to subtract a third of the whole number, or a third times the whole number, SO instead of writing just 33% or just one third, you want to right THAT[points at 1/3d], because that's the quantity not just the fraction, because the fraction just tells you what part, it doesn't tell you exactly how much. A third could be a third of two thousand, it could be a third of seven. So let try. So you want to take your number and subtract 33% but not 33%, you want to subtract 33% OF[emphasized] the whole number. So how would you write that.

118. TUTOR That's the

119. STUD Yeah

120. TUTOR You did this

121. TUTOR That's 33% percent. This is a good part of your expression. So how would we write the whole expression? there are two different ways. And you have played around with both of them. Let’s stick with this one for now. This one you take d, your whole amount, and what do you want to subtract. What exactly do you want to subtract?

122. STUD I want to subtract, Actually I want to add one third to it

123. TUTOR You want to add one third? Oh, You want to take a third and then another third. and add the two together, or multiple by two

124. STUD Add one third of the whole number

125. TUTOR OK- So write that down. So you want to take

126. STUD I want to take the 1/3 of d.

127. [writes 1*3d +1/3d "]

128. TUTOR Ahhum. Good. And you got the one third of d not just one

129. STUD Times, I mean plus, another third of d

130. TUTOR Good, and that's one way to right the expression

131. STUD which equals two-thirds

132. [adds "2/3d=?" to the line with 1/3d+1/3d]

133. TUTOR Right

134. STUD times d

135. TUTOR Right

136. STUD equals

137. TUTOR Good, and that is one way to write the expression and that's a perfectly good way. If you know a third is coming off, you know two thirds is left.

138. STUD Yeah

139. TUTOR You could have also used this way, which you originally started. You could have done "D" minus a third of d, because that also gives you two-thirds of d. But you can't do, You would have to say "d minus 1/3 of 3, you can't just say d minus a third because you can't just subtract a third. A third of what? So you have to have the "of what" part. Either of those expressions are great! [tutor writes "d-1/3d"]

140. TUTOR See it?

141. STUD Yeah

142. TUTOR This[points at d-1/3 presumably] is where you fell apart, because you wanted to subtract a third, but you just can't subtract a third but you can't just subtract a third of something, you have to subtract a third OF something, and that “of” triggers multiplication. See it

143. STUD Yeah, that's how I would have done it in mind, because I understand what I mean

144. TUTOR Right, Exactly, but when you are writing an expression you are writing it for any old person who comes and uses that formula. Right? Or , more specifically, but in your case, you will probably be writing this for a computer. If you are writing this for a computer your computer has to.. you have to be VERY specific for your computer, or it’s not gonna do what you want it to do. Right?

145. STUD Yeah

146. TUTOR OK, let’s try number 6

147. STUD [reads problem] Sue made 72 dollars by washing 6 cars to buy holiday presents. She decided to spend 32 dollars on a present for her Mom and then use the remainder to buy presents for each of her s sisters. She will spend the same amount on each sister. How much can she spend on each sister?
148. STUD Now uhm :::::: times thirty two equals :::::: 40 writes "72-32=40 "]
149. TUTOR Remember that we are trying to write an expression, we are not trying to do any work.
150. STUD Yeah, divided by "s" equals money spent on my sisters [continues on writing a single line by adding the "/s" onto the line above to get "72-32=40/s=money spent on sisters"]
151. TUTOR Now write one big expression that shows everything, without having , without having done any work
152. STUD Can I just use like numbers or letters,
153. TUTOR Yeah, WELL use the numbers, but here. You actually.. Remember how I talk about all the time, this[?] is not really equal to this. Are these two things really equal? [points to the "72-32" and the "40/s"]
154. STUD Ah NO
155. TUTOR [repeats his no] Kind of yucky, so let's down here right one expression, without doing any math. Pretend you forgot how to subtract. Can you right an expression with doing any of the subtraction, division, or multiplying?
156. STUD Yeah, I think so
157. TUTOR That shows the whole thing?
158. STUD :::::::[writes 72-32=a/Stud] ::::::
159. TUTOR Right, instead of 'a', let's just use this. Pretend
160. STUD Oh I get it :::: it will be 72 minus 32 equals something minus thirty two divided by s.
[writes 72-32=72-32/s = money spent"]
161. TUTOR so really you don't want to put the equal sign, you just want to write that Now!, What is being divided by 's'? because order of operation says you divide before you subtract. So what is really being divided by 's'?
162. STUD Ah [writes parenthesis]
163. TUTOR right, now you wanted to write this equals, because you want to go one step at a time. Bad habit, because then you get equal signs between things that aren't equal. So here
164. STUD So here I really don't need this [??]
165. TUTOR Right. You really want this [??], and then we do...then we work down, to show that everything above it is equal to [inaudible]
166. STUD It's just because I can't write on this [student crosses out "72-32=" that was part of the "72-32=72-32/4"
167. TUTOR So you see what I mean?
168. STUD Yeah
169. TUTOR [talks over the student] Yeah, that's like forgetting to capitalize at the beginning of the sentence. It's just yucky. OK Laughs
170. TUTOR What if someone said it wasn't thirty-two dollars, this doesn't give me the right answer. I want to spend 30 dollars on my mother. Now you can change it easily. Here, it not as easy to change, because you don't know where the 40 came from. OK, number 7what does number 7 says

171. STUD [reads problem] John and his wife Beth have been saving to give the 5 children presents for the holidays. John has saved 972 dollars for presents and Beth has saved "b" dollars. They give each child the same amount. Write an expression for how much each child gets.
172. STUD The same amount between them, or the each amount between just one person.
173. TUTOR It doesn't matter which way.. If each person gives them the same amount, there gonna get the same amount at the end, right?
174. STUD Ah, yes. So it would be 972 divided by 5 equals = a [wrote "972/5"]
175. TUTOR Just leave, because well make one big expression.:::::
176. STUD "b divided by 5" wrote onto 972/5 = b/5"
177. TUTOR So how much does each kid get?
178. STUD Ah. :::::::They get :::::::They get a+c? I don't understand this one part
179. TUTOR Hoe much are they gonna take in?
180. STUD They are gonna take in this divided by 5 and this divided by 5
181. TUTOR AND [Emphasis apparent in the audio presumably is signal the addition operation]
182. STUD Ah Plus, So it would be :::::::::writes "(972/5)+(b/5):::
183. TUTOR Next one. Do you need those parenthesis there?
184. STUD Not really
185. TUTOR Why not?
186. STUD Ah, because division get done first left to right
187. TUTOR Right so those aren't necessary [(b/5)+(972/5)]this one here, those were. You need to tell your person you got
188. STUD Because you have to minus the two numbers first
189. TUTOR right, you got to do the subtraction first

190. STUD [reads problem]Bob left at 3 P.M. and drove 550 miles from Boston to Pittsburgh to visit his grandmother. Normally this trip takes him "h" hours, but on Tuesday there was little traffic and he saved 2 hours. What was his average driving speed?
191. STUD ::::Well ::: Ah :::: so he saved two hours :::ahum::::::::
192. TUTOR Do you know how to calculate average driving speed?
193. STUD I think, but I forget
194. TUTOR Well average speed, as your mom drove you here did she drive the same speed the whole time.
195. STUD No

196. TUTOR But she did have an average speed. How do you think you calculate the average speed?

197. STUD It would be h hours divided by 550 miles an hour.

198. TUTOR So which way is it? It's miles PER hour. So which way do you divide?

199. STUD It would be 550 divided by h

200. [write 550/h=mph"] TUTOR OK so now, that's how you calculate miles per hour. So now how about for this problem? Read the problem again. Because you got the right idea. You know how to calculate average speed. But what exactly do you have to do for this trip

201. STUD Um. Well he saved two hours, but I don't know how that is important

202. TUTOR Well how do you calculate... Not for Bob but for your mom, how did you calculate what her average speed was driving to CMU this morning?

203. STUD Ahm, I guess you would I would have done it 550 divided by h

204. TUTOR yeah [even though the 550 is not for his mom?] That's how you calculate average speed but what exactly is it? 550 represents what?

205. STUD Miles per hour

206. TUTOR No.

207. STUD Oh 550 miles

208. TUTOR Right

209. STUD Divided by h

210. TUTOR Which represents?

211. STUD Miles per hour

212. TUTOR No what does h represent?

213. STUD Hours

214. TUTOR Hours! So what are you getting? What are you dividing by what?

215. STUD Oh miles divided by hours.

216. TUTOR Right TOTAL miles divided by

217. STUD Total hours

218. TUTOR So let's calculate it for this guy, That's exactly the concept, TOTAL miles divided by TOTAL hours [writes "550/h"]

219. TUTOR Is that what it is?

220. STUD Yeah

221. TUTOR Is 550 the total miles? [neat!]

222. STUD Yes

223. TUTOR Is h his total hours?

224. STUD Yes

225. TUTOR Is it??

226. STUD Oh no h-2

227. TUTOR OK- right this again and write it correctly so that order of operations and stuff works

228. STUD [Writes "550/(h-2)"]

229. TUTOR Exactly, so where did the 2 go in?

230. STUD The two hours he saved on traffic

231. TUTOR To calculate the total hours, so good.

232. TUTOR How we doing, we got lots of time. All right thinking harder. These are pretty good. Let's try number nine. Okay

233. STUD Okay

234. TUTOR [Laughs and mentions hard work]

235. STUD [read problem]Julie was trying to raise money to help fight Cancer. She got 7students to each donate "s" dollars and "t" teachers to each donate 10dollars. Write an expression for how much she collected?

236. TUTOR Number 10, or no number 9

237. STUD [writes 7*s +t*10=money to fight cancer"]

239. TUTOR Good. Next problem

240. STUD [reads problem] Cathy took a "m" mile bike ride. She rode at a speed of "s" miles per hour. She stopped for a "b" hour break. Write an expression for how long the trip took?

241. STUD uhm :::::::::::::::::::::: writes "s/m+b":::::::::::::::::::::::::::::::::

242. TUTOR How do you calculate the amount of time it takes you? If your, if your, if your riding at, let's make it simple. If you are riding at 20 miles per hour, OK and you go 100 miles, how many hours did that take you?

243. STUD Um 5

244. TUTOR 5 and how did you get that 5? How did you use the numbers 100 and

245. STUD 100 miles divided by miles per hour

246. TUTOR So you took the miles and divided it by the [garbeled, probably "speed"]

247. STUD Miles divided by s plus b equals time [writes m/b+t]

248. TUTOR Right, OK, whenever I get these.. did you see how I had to stop and think? I have stop and think for these to? so I always remember to stop and think, which way do I have to divide, because I know I have to divide, which way? OK? So you have to figure out which that is? OK number 11

249. STUD [reads problem] Debbie has two jobs over the summer. At one job she bags groceries at Giant Eagle and gets paid 5 dollars an hour. At the other job she delivers newspapers and gets paid 7 dollars an hour. She works a total of 30 hours a week. She works "b" hours bagging groceries. Write an expression for the total amount she earns a week.
250. STUD ::::::::::::::::::::::::::::::::::writes b*5+(30-b)*7:::::::::::::::::::::::::::::::::::::::
251. TUTOR Beautiful, excellent, good work That one was tough

252. STUD [reads problem]Michael starts a business selling lemonade. He buys 35 dollars worth of supplies including lemons, pitchers, cups and advertising. He sells a 16 ounce glass of lemonade for 2 dollars. If he sells "g" glass of lemonade, how much profit will he end up making
253. STUD ::::: 2 dollars for each cup:::35::: minus 2 * g [writes "35-2*g"]
254. TUTOR Which number do you want to be bigger? Which number is gonna be bigger? 35 or 2*g?
255. STUD Ah.. ::: well let’s see what it’s gonna be Else he loses money and [writes "2g-35"]
256. TUTOR So this would calculate how much money he lost? Now if you do this, will this calculate how much money he lost?
257. STUD No
258. TUTOR What if he sold 5 glasses what's the answer gonna be.
259. STUD Ten
260. TUTOR That would be ten. What's ten minus 35?
261. STUD Oh I got it., money made, then 35
262. TUTOR No let look at this I think you can. This should work for both, because what's the answer? If he sold 5 glass what would the answer be?
263. STUD Um - minus
264. TUTOR What does that represent?
265. STUD Well you could just take off the minus and it would be how much money he lost
266. TUTOR Exactly. So that minus just represents "lost". So this works for both of them
267. [writes 10-35= -25"]
268. TUTOR So the math even at negative number, those negative numbers allow it to tell us all the answer, even with one expression, that's why those negative numbers are so cool.

269. STUD [reads next problem] A jacket that normally cost d dollar goes on sale for 2/3 of its original price. How much does the jacket cost on sale?
270. STUD We already did this, but :: two thirds time d
271. [writes 2/3*d]
272. TUTOR And the only thing I am gonna tell you is that when you write it like that you have to put parenthesis, because otherwise it looks like 3 times d or you could write it like this
273. STUD Actually I think...
274. TUTOR which is the same thing
275. TUTOR point 66666 which is 66%, but this way is always better.

276. STUD [reads problem] Rebecca makes a "h" hour car trip. For 3 of those hours it was raining and Rebecca drives at 40 miles per hour. The rest of the time it was sunny and she drove 55 miles per hour. Write an expression for the total distance Rebecca drives.
277. STUD h-3 times 55 :::: would be ::::: plus ::::::::: 3 times 40   [writes "(h-3)*55mph + 3*40mph"]
278. TUTOR Right and usually with the expression we don't put the units in

279. STUD [reads problem] John drove 300 miles to grandmother's at 30 miles per hour. He drove back at 20 miles per hour. He drove a total of 600 miles. What was his average speed?
280. STUD 600-30+20 divided by ::::::::::::: two :::::::: no this parts wrong ::: writes 600-[(30+20)/2] and then scratches out the 600-"
281. TUTOR Right
282. TUTOR That [points at (30+20)/2"] looks great but it doesn't work. OK You would think it would, you are just averaging, but it doesn't work. What did we define average speed as earlier?
283. STUD Um
284. TUTOR Had the words total in it. Had to do with totals.
285. STUD Ah total : Ah total miles plus 50, plus total miles per hour
286. TUTOR No, no. To calculate miles per hour what do we need?
287. STUD Miles and hours
288. TUTOR Miles and hours. We need TOTAL miles and what else?
289. STUD Total hours
290. TUTOR Exactly
291. STUD So:::
292. TUTOR So for his complete trip
293. STUD So seeing it’s 600, it has to be half, 300 miles could be the first half
294. [writes "300"]
295. TUTOR You have to deal with this idea of total
296. STUD I am gonna figure out the hours for each half of the trip and then add them together.
297. TUTOR Exactly
298. STUD Uhm ::::::::::::::::::
299. TUTOR You can, pick an easy problem and figure it out

300. STUD :::::::::"writes divided by 30=10 hours 300/20=A"::
301. TUTOR Remember that you don't need to do the calculations because I am gonna ask you at the very end to actually just write the expression without doing the math. OK.
302. STUD So it would be 10 plus A? mumbles
303. [writes "10+A" and then adds in front of it "600*"] Is that how you calculate miles per hour
304. STUD Ah no
305. STUD Mumbles
306. TUTOR That is just the exact same thing you had before, except you are that you are just multiplying "A" times 600 but I suspect you want to do the whole thing? [Student writes "(10+A) * 600"] How did we calculate. Once again what is the definition of average speed? Average miles per hour?
307. STUD Um
308. TUTOR Tell me out loud. Definition of average speed
309. STUD :::
310. TUTOR It has to do with total. The word total has to be there.
311. STUD :::
312. TUTOR How did we do it over here?[points to previous problem where speed was calculated]
313. STUD 550 total miles per hour total?
314. STUD Ah, total miles
315. TUTOR OK
316. STUD Then h-2 total hours
317. TUTOR Right, so we took, total miles and did what?
318. STUD Divided Ah
319. TUTOR Divided it by total hours writes ["600/(10+A)"]OK beautiful. You weren't given 10 or given 'A.'. What were you given? So the person that is reading this doesn't have any arithmetic already done.
320. STUD 600 divided by 30, no it's 300 divided by 30..write 300/30
321. TUTOR Good that's the tens; plus three hundred divided by 20 adds "+300/20"
322. TUTOR OK
323. STUD ::::600 divided by that then divides by putting the 600/ in front of it. for a final expression of "600/(300/30+300/20)"
324. TUTOR Yup, and after next year you won't right it with that division
326. TUTOR OK the next one is a good one.

327. STUD [reads problem] A car salesperson is paid a base salary of $200 per month plus the additional amount of money in commissions for each car she sells. She sold four cars last month and received "x" dollars that month. If she sells "h" cars this month, write an expression showing how much she earns this month. What's her commission?
328. TUTOR Good question.
329. STUD Oh I have to find this out don't I
330. TUTOR I think so
331. STUD Um x [minus] that
332. STUD It doesn't say how many cars. Ah 4 cars last month so it would be x-200 divided by 4
333. [he puts in the parenthesis only after writing "x-200/4"]
334. TUTOR OK
335. [he has written (x-200)/4 = commissions" and then adds a "c" before the commission apparently to indicate that commission's will have the variable "c" stand for it]
336. STUD Equals commissions or "c". So this month would be 200+c*4 = salary
337. [writes 200+c*4=salary]
338. TUTOR This month?
339. STUD Yeah
340. TUTOR Read it again.
341. STUD This month, or she sold h cars
342. [scratches out the 4 and replace it by "h"]
343. TUTOR You want to skip a line so you aren't messy
344. [starts writing "200+c*h" and tutor interrupts after the "c" to say
345. TUTOR OK now is C something that was given to you?
346. STUD No
347. TUTOR Where did it come from? What numbers
348. STUD So have to do this [scratches out the "200+c*h" he just wrote]
349. TUTOR You have to rewrite the WHOLE expression.
350. [writes 200+(x-200)/4*h=salary]
351. TUTOR Good
352. TUTOR Let's see if that last problem is any harder

353. STUD [read problem] A candle that has been burning for 2 minutes is 8 inches long. Three minutes later the candle is "x" inches long. Assume that the candle will burn at this same rate. Write an expression for the height of the candle after the candle has been burning for "m" minutes.
354. STUD So assuming that for each minute it's x inches long it's a little tricky
355. TUTOR Ahum
356. STUD :::::::: x inches divided by 8inches equals
357. [writes "xinches/8 inches=rate of burning " {does this quantity name the rate or the amount burnt?}]

358. STUD rate no it would be minus, [changes the minus 8 to divided by 8"] inches is rate of burning

359. TUTOR Inches minus inches is gonna give you answer in what units?

360. STUD Inches

361. TUTOR Inches, is that a rate unit?

362. STUD It would be one inch per minute

363. TUTOR Yes, that’s what you want. so the per stands for what operation?

364. STUD divided by. so I was right!

365. [starts to change the minus back to a dividing sign]

366. TUTOR No, no, no, but wait a second, you want inches divided by inches?

367. STUD No

368. TUTOR That's gonna give you

369. STUD

370. TUTOR You are on the right track because you want inches per [[left a sentence completion]

371. STUD Yes, I was think If I divided by minus x inches by minus 8 inches it would be the inches per minute

372. TUTOR No. what does that give you?

373. STUD That would be x-8

374. TUTOR Which is longer, x or 8?

375. STUD x

376. TUTOR [Read the problem again]

377. STUD "assuming the candle has been burning for 2 minutes is eight inches long. Three minutes later the candle is x inches long."

378. STUD Oh three minutes later. so it would be five minutes "Assume that the candle will burn at the same rate. Write an expression for the height of the candle after the candle has been burning for "m" minutes. " :::::

379. TUTOR Now you got a good start. I like what you did but what does it give you?

380. STUD It give me uh um the number of inches the candle has burned in the time of three minutes

381. TUTOR Right! So will that help you find a rate?

382. [To the line that now reads "8 inches-x inches=rate of burning" he adds "in 3 minutes" which is improper]

383. STUD Yeah

384. TUTOR Yeah because rate is what? How is rate defined?

385. STUD Its’ ::: I don't know

386. TUTOR You just told me

387. STUD It’s the time or something like that

388. TUTOR It’s got a "per" in it

389. STUD Yeah

390. TUTOR So what is it for this situation?

391. STUD The amount of candle that has burned in so many minutes

392. TUTOR Right. So and candle burned is measured in what in this problem?

393. STUD Inches

394. TUTOR Inches. it could be measured in grams of wax or something. But in this problem it is measured in inches. So we want our rate to be what? Inches [left time for completion]

395. STUD um. wait, if it’s going less it will go x

397. TUTOR Yeah-- that you can go ahead and fix, so that's gonna give you inches. How are we gonna have to fix that to give us the rate?

398. STUD And we can divide eight minus 8 by three to give us one minute

399. TUTOR That’s gonna give us the rate. Exactly!

400. STUD Eight

401. TUTOR The unit rate!

402. STUD Eight divided by oh eight. [on a new line he writes 8inches/" and stops] I mean to say x times 8.[changes the line to say "x*8inches/"] Oh that's wrong.

403. TUTOR I can see you are getting tired, this is our last problem.

404. STUD Eight minus x inches :: divided by [writes (8-x inches)/3=rate in 1 minute of burning"]

405. TUTOR Right

406. STUD Three equals rate for one minute mumbles

407. TUTOR OK so what's the actual question?

408. STUD m times (8-x) inches divided by three [writes "m*(8-x)/3"]

409. [Note: this is the amount that has burned not the end height of the candle but the tutor accepted the student’s answer.]

410. TUTOR Because rate times time gives you inches,

411. TUTOR Beautiful! Those are hard. Those are good ones.

End of transcript

Appendix D: A list of Observed Tutorial Operators

What follows is a list of tutorial operators that might be incorporated into the model of tutorial reasoning I propose to build. I have attempted to define types of tutorial interaction. I note any use of these operators in Appendix C and cite them by line number. The categories are not mutually exclusive. To make this list easier to browse I have provided an idealized example in the context of the following problem:

> Ann is in a lake that is 2400 yards wide. She starts out 800 yards from the dock. She rows for "m" back towards the dock at a rate of 40 yards per minute. Write an expression for her distance from the dock.

Op1: Concrete Instantiation Analogy

The first step is an instantiation request and is found in lines 32, 104, and 258. The second and third steps are both requests for the student to articulate how they got an answer. An example where that was tough for the student is at lines 45, 51, 56, and 65. The last step is a request to show the two evaluation steps in a single expression and is found in lines 70, 82 and 151.

Tutor:That is not right. Let me ask you an easier question. Suppose the number of minutes she had been rowing was "3" instead of "m". What would the distance to the dock be?
Student:680
Tutor:Correct. Can you show the math for how you got that?
Student:800-120
Tutor:That is partially correct. How did you get the 120?
Student40*3
Tutor: Good. Now write how you got the 680 again, but this time use 40*3 in place of 120.

Op2: Tutor makes reference to dialogue history

On line 282 the tutor asks the student to recall a fact that they used several problems previously on line 216. Line 306 shows the tutor using "Once again, what is..." (Moore (1996) addresses issues related to generating natural language with the natural human quality of being sensitive to what has been previously said in the dialog.) On line 312 the tutor points to a previous problem to encourage the student to use that as an example.

Tutor:You have made the mistake of calculating speed using multiplication several times today; try to remember how you calculated speed on that problem about the Indy 500.

Op3: Encourage the student to generalize

On line 183 the tutor challenges the correct but unnecessary use of parentheses. Line 401 shows the tutor confirm a correct response and at the same time trying to introduce vocabulary. Line 246 is an example where, after the student states how to compute an instance, the tutor states the procedure in general terms. Line 170 is an example where the tutor suggests that a generalization is possible but doesn't ask the student to do the generalization.

Tutor:Good. Now what if the speed wasn't 40 but was "s." What's the answer now?

Op4: Reflection after the correct answer

On line 229 the student is challenged on how he used one of the numbers given in the problem, which he had mentioned he did not know how to use (earlier on line 201) so the tutor came back to reinforce. On line 169-170 the tutor adds additional comments after the student has already arrived at the correct answer.

Tutor:Good. You didn't make the mistake of using division for the fractional relation as you have a few times today. Try to remember that on future problems.

Op5: Reflection on, or Challenge of, a correct answer if the tutor suspects guessing

On line 253 and 255 the student quickly corrects his initial wrong answer with the other easy to guess alternative, so the tutor spends considerable time (lines 255-268) challenging his understanding of that answer, presumably due to a suspicion that he doesn't understand why his answer is correct.

Tutor:How do you know the answer is 800-40*m and not 800+40*m?

Op6: Feedforward (given before the student makes an anticipated mistake)

On line 227 the tutor anticipates a missing parentheses mistake and possibly chooses to help the student avoid that problem because of the rather lengthy dialog the tutor and tutee are engaged in.

Tutor:Now, during the last few problems you have made several mistakes on problems that ask you to figure out the amount of a fractional discount. This upcoming problem is another of that type. Try to remember what operator you needed on problems like this (like the problem about selling T-shirts).

Op7: Requesting that the student recall information by either questioning and/or hinting

Line 306-310 shows the tutor asking the student to recall a definition, and then when the student fails, switching to hinting by providing part of the definition "The word total has to be there?" Line 384 & 282 are recall requests both followed by line 388 and 284 which are hints towards that information. Line 192 is a similar request for recall.

Tutor:How do you calculate distance traveled when given a speed and time?(question form)
Tutor:How do you calculate distance. Remember d=r*t. (hint form)

Op8: Challenge the student's answer

Line 219, 221, 223 and 225 challenge parts of the students answer, rather than say it is wrong. Line 221 even challenges part that is correct. Line 75 and 106 are other examples of this very general style of delivering what is usually negative feedback. Line 242 begins with a rhetorical question about the procedure a student uses, that is presumably asked to focus the student's attention on the part of the answer that is wrong, and not to actually elicit a response as shown by the fact the tutor immediately asks a follow up question.

Tutor:Are you sure about that plus?

Op9: Ask the student to identify the name of a quantity represented by a symbol (or expression). Include follow up clarification if the student is not specific enough.

Line 210 and 212 are examples. This does not just apply to numbers given in the problem. On line 24-27 the student has apparently used "2/3" to stand for "two-thirds of the cost of a shirt" and the tutor asks the student to identify what the "2/3" represents?

Tutor:Your answer of 2400+40*m is not correct. But the 40*m is part of the answer. Before we go on, can you please tell me in words what the 40*m represents?
Student:The start distance
Tutor:From where?
Student:the dock
Tutor: Good.

Op10: Ask a student to identify the symbol(or expression) that represents a quantity (vice versa of above)

On line 82 the tutor asks the student to recall from the problem statement the symbol used to represent a quantity.

Tutor:What is the speed of the rowing?

or

Tutor:In your expression "800+40*m" what part represents the distance rowed after "m" minutes?

Op11: Correct a bug by referring to the implicit semantics about the relative size of numbers

Line 374 and 252 both show this technique of correcting a bug for the wrong ordering of subtraction arguments by asking "Which is larger?"

> Tutor:Your answer of 40m-800 is wrong. Which one is supposed to be larger?

Op12: Stating the general quantitative relationships in words. This is like generalization, but can occur anywhere in the problem

Line 216-218 and 246 are examples and line 282 is a request for such a statement.

> Tutor:So the distance she has left to row is equal to the distance she starts from the dock minus the distance she has rowed back towards the dock. Now that we have that figured out, Let's go back to the original question of how far she has left to row

Op13: If the student might have forgotten what work he has already accomplished then remind the student what steps they have already completed

Line 407 is an example of the student having completed a subgoal of finding the slope and then being reminded to think about what the original question asked for.

> Tutor:Good. You now have told me that the amount she made at bagging groceries was 5*h and the amount she made delivering newspapers was (30-h)*7. What is the total amount she earned at both jobs?

Op14: Positive Feedback on parts that are correct

Lines 315, 334 and 326 are just a few of the many examples explicit positive feedback in the form of "OK". Lepper et al. (1997) argued that a primary characteristic of good tutors was their indirect style including negative feedback and positive feedback. The implicit positive feedback on line 183 is an example where the positive feedback for the answer was simply in the form of indicating the student should move on to the next problem. Line 362 seems to show an extreme example where the tutor gave positive feedback for just a small kernel of a correct response.

> Tutor:Good. You now have told me that the amount she made at bagging groceries was 5*h and the amount she made delivering newspapers was (30-h)*7. What is the total amount she earned at both jobs?

Op15: Simple Feedback on an identifiable bug category

> Tutor:Your answer is missing parenthesis

or

> Tutor:You reversed the order of the subtraction sign.

or

> tutor:You calculated the amount she rowed, not the distance she has left to the dock

Op16: Ask the student to figure out what sub-goal to set

An example where the tutor asks the student what information is needed for a sub-goal is at line 286 See Catrombone, (1996) for an argument that helping the students see the sub-goal structure is of crucial importance. Line 37 contains an instance of the student having set too large a goal (trying to find the cost of all 4 jackets), and the tutor points this out by asking the student what the goal for the problem is (cost of a single jacket).

> Tutor:That is not right. Let me get you to simplify the problem for yourself. In order to figure out the distance she has left to the dock, what quantity do you have to first figure out? (Answer: "the distance Ann has rowed after "m" minutes.")

Op17: Socratic Technique showing a contradiction from a student's error

Line 32 is an example showing a successful use of a Socratic technique that makes the student aware of an error in his answer by asking him to evaluate an expression. Line 260 is another example of the tutor showing the student that his symbolization does not agree with his arithmetic computation. Line 359 is an example where the tutor points out a conflict to *the student in that their expression does not give an answer in the units the student was expecting.*

Tutor:Let's take a look at your answer of 800-m. What does your answer predict the distance Ann is from the dock will be after 3 minutes?
Student:800-3=797.
Tutor:OK good. I agree that that is what your answer predicts. But let me ask you to now consider how far Ann rows in 3 minutes when she is going 40 yards per minute?
Student:120 yards
Tutor:So what does that predict should be Ann's distance from the dock after 3 minutes?
Student:680.
Tutor:Good. That is correct. You now see that your answer of "800-m" could not be correct, since it predicts she will be 797 yards form the dock after 3 minutes, which is not correct. Please take another guess as to what you think her distance from the dock is after "m" minutes.

Op18: Order of Operations Sub-Dialogue

Line 161 is an example of the tutor engaging the missing parenthesis errors possibly on problems that require parenthesis.

Tutor:Your answer of "70-m/4 "is wrong because it is missing parenthesis. Try to fix it.*[correct answer is (70-m)/4]*
Student:70-(m/4)
Tutor:Suppose you knew what value "m" had. What is the first math operation you would perform in your head? (subtract or divide)
Student:subtract
Tutor:Correct. But the answer you wrote above "70-(m/4)" would have which math operation done first?
Student:subtract
Tutor:No. Remember that what is inside parenthesis is done first. Try again. What math operation would be done first in your answer 70-(m/4)?
Student:divide
Tutor:Correct. Now see if you can put parenthesis around 70-m/4 to make the subtraction happen first.
Student:(70-m)/4

Op19: Teach students how to do unit analysis

On line 359-361 the tutor attempts to use unit analysis to convince the student that his answer must not be correct.

Tutor:Your answer of 800-40/m has a problem. Let's look at the 40/m that you probably mean to be the distance she has rowed so far. What are the units on the 40?
Student:yards per minute
Tutor:Good. What is the unit on the "m"?
Student:minutes
Tutor:If you divide yards per minute by minute what do you get?
Student:yards per minute squared
Tutor:Correct. So you see your answer of "40/m" couldn't possible be correct because it gives a number in the wrong units. You want an answer in yards. Try again at what you think is the correct expression for Ann's distance form the dock.

Op20: Coach the student to realize distractor numbers are not needed

The student only made one error involving a distractor number, and the tutor early on coached the student to realize this error in lines 37-41.

Tutor:In your answer 2400-40*m you used a number that is not needed. Which number do you think is not needed?
Student:2400
Tutor:Good- Now that you have identified that 2400 is not needed, take another guess at the distance she is from the dock.

Op21: Slips and other mistakes that tutors don't dwell on

Line 278 is an example where the student wrote the expression with the units in the expression and the tutor did not dwell on the error and simply told the student to leave off the units inside expressions. Possibly, some errors would be explored with tutoring if a novice made them but those same errors might be interpreted as slips if a more advanced student made them.

Tutor:Your answer of 80-40*m looks like it might be right but you used 80 instead of 800. Please fix your answer.

Op22: Tutor focuses attention on a previous answer that was more correct

This appears to happen on the many different answers the student gave between 17 and 145, but is hard to identify.

Tutor:Your answer of 800+40m is close. Try again.
Student:2400+40*m.
Tutor:No- you are getting colder. Let us go back to your first answer of "800+40*m" and work from there. Now what is...

Op23: Tutor comments on the repair the student attempted

The lines 174-182 show a student who was able to get the two separate parts, and then got tutored simply on the repair to combine the two expressions together. In general, this requires additions to student model to model the process of creating new answers by modification of existing answers.

Student:-800-40*m
Tutor:No, that is not correct. See if you can see for yourself what is wrong and try again
Student:-800+40*m
Tutor:No. You changed the expression to be an expression that gets larger as "m" gets larger, but in fact the distance to the dock should decrease as the number of minutes she rows increases.

Op24: Levels of Specificity

This operator is orthogonal to the above operators, since all the tutors remarks can be more or less specific.

Tutor:1) Your answer has 2 errors
2) You are missing parentheses
3) You need parentheses around the 30-b.
4) Your answer should be 5*b + 7*(30-b)

Op25: Be able to differentiate a close answer from a very wrong answer, from an unintelligible one

This operator is also orthogonal to the above operators. Presumably, most of the above operators can be made more or less helpful in the degree that they get more specific in hinting as to what the student should do.

Tutor: No, but that is close. I can see you only made one mistake.
versus Tutor:No that is not correct. Let me help you...
versus Tutor:I am sorry, but I can't understand what you wrote. ...

Op26: Engage the student to try to diagnosis what the student was thinking

Line 20 and 24 are both examples where the tutor asks the student first for simple clarification, and then for an explanation as to how the student arrived at a part of the answer.

Tutor:Why do you say 40/m?

Appendix E: An Transcript Showing a Student with a Composition Difficulty

This is a verbatim transcript of a tutorial dialog I had with an advanced seventh grade student over a teletype. It is included to show that students can sometimes be able to do decomposed problems separately, but experience great difficulty when they try to put it all together. The time is indicated in square brackets. Each line was sent once the participant hit the return key. Typos have been left as they appeared to the student.

Tutor:: [12:08:00]
Good
Debbie has two jobs over the summer. At one job she bags groceries at
Giant Eagle and gets paid 5 dollars an hour. At the other job she
delivers newspapers and gets paid 7 dollars an hour. She works a
total of 30 hours a week. She works "b" hours bagging groceries.
Write an expression for the total amount she earns a week.
Student::
30 - b = n   n * 7 = m   b * 5 = o   m + o = y
Tutor:: [12:11:55]
Ok Good- That is a good first step.
Now let me ask you what is "y" interms of the numbers given in the problem?
Studnet::
y is the amount of money she earns in a week
Tutor:: [12:13:08]
That is true but was not what I meant to ask. I meant to ask can you tell me what y is
me what Y"" is only mentioning the numbers in the problem? (Like is it "5*7/30+b"?
Student::
it is the money she makes from both jobs (m and o) added together
Tutor:: [12:14:57]
again that is correct but not what I was looking for. Let me ask you something different. Can you tell me
how much money she makes baggin groceireis?
Student:: she makes 5 * b bagging groceries
Tutor:: [12:16:22] good.
Now can you give me the expression for how much she made deliveirng newspapers?
Student:: 7 * (30 - 5 * b)
Tutor:: [12:18:07] No. It is 7 dollars times the number of hours she worked, but that is not
(30 -5 *b). Can you give me the expression for the number of hours she worked?
Student:: i'm stuck
Tutor:: [12:20:37] Ok- I will help
The expressions you wrote before wehere correct when you said
the stuff at the begining
Ok. Let me ask you hour many hours she worked deliveing newspapers if instead of "b" it said 20?
Student:: 30 - 20 = 10
Tutor:: [12:21:32] Good. So now how many hours delivierng nespapers if it really was "b"?
Student:: 30 - b = c
Tutor:: [12:22:02]
Good. Now haw much does she make from deliveirng newspapers?
Student:: 7 * c
Tutor:: [12:22:58]
Ok-
Now combine that
7*c
30-b=c
into one expression which tells us how much she made baggin groceriies
Student::
(30 - c) * 5 = y
Tutor:: [12:24:55]
no-
It shoudl be (30-b)*7
Do you see how you can get this from the two things you wrote?
You had said
7*c and 30-b=c
do you see how these two can be put together to get
(30-b)*7
?
Student:: you said bagging groceries, not delivering newspapers. She makes $5

an hour bagging groceries.
Tutor:: [12:27:26] Opps you are correct that I made a mistake. But you also made a mistake.
so lets talk about magging groceires for a minutes and rmemebr that she makes
(30-b)*7 at the newspapers.
So now bagging grocieres. Can you state agin how much she makes at that job?
Student:: (30 - c) * 5
Tutor:: [12:28:42] Ok the 5 is correct. But she only works b bours bagging rocieris, not the (30-c) you have said.
So let me ask you to give a shot at the fianl answer for the total she makes
Student:: b * 5 + 7* c
Tutor:: [12:30:01] That is close to correct. Now rewrite that showing what c is. (You see that c is not givne in the problem so you final answer should only have numbers form the problem (5,7,30 and b))
Student:: 5 * b + (30 - b) * 7
Tutor:: [12:30:51] Yes!

Appendix F: Production Model-Traces for the student model

These are the 33 productions that fire when the student correctly symbolizes "(x-20)/4".

```
! Firing: (READ_SYMB_QUESTION SET_BEGIN_MODEL_WITH_QUES)
!! Firing: (SET_CURRENT_INPUTS SET_SET_CURRENT_INPUTS)
!!! Firing: (PICK_RELATION_THAT_HAS_GOAL SET_TRAVESE_QUAN_NET)
!!!! Firing: (PROD_HELP_FINISH_PICK_REALTION SET_FINISH_PICK_REALTION)
!!!!! Firing: (MARK_MENTAL_EXPR_WITH_REALTION_USED SET_MARK_EXPR_WITH_REALTION_USED)
!!!!!! Firing: (MARK_MENTAL_EXPR_WITH_REALTION_USED SET_MARK_EXPR_WITH_REALTION_USED)
!!!!!!! Firing: (PICK_RELATION_THAT_HAS_GOAL SET_TRAVESE_QUAN_NET)
!!!!!!!! Firing: (PROD_HELP_FINISH_PICK_REALTION SET_FINISH_PICK_REALTION)
!!!!!!!!! Firing: (MARK_MENTAL_EXPR_WITH_REALTION_USED SET_MARK_EXPR_WITH_REALTION_USED)
!!!!!!!!!! Firing: (MARK_MENTAL_EXPR_WITH_REALTION_USED SET_MARK_EXPR_WITH_REALTION_USED)
!!!!!!!!!!! Firing: (FILL_IN_LITERAL SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!! Firing: (FILL_IN_LITERAL SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!! Firing: (GET_OPERATOR_AND_FINISH SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!! Firing: (SETUP_OP_AND_ARG_FOR_OUPUT SET_RETRIVE_OP_AND_ORDER)
!!!!!!!!!!!!!!! Firing: (SELECT_OPERATOR_CORRECT SET_SELECT_OPERATOR)
!!!!!!!!!!!!!!!! Firing: (IF_OPERATOR_DONE_CORRECT_SELECT_ORDER SET_CHECK_TO_SEE_IF_BUGGY_OPERATOR)
!!!!!!!!!!!!!!!!! Firing: (SELECT_ORDER_NON_COM_CORRECT SET_SELECT_ORDER_IF_OP_CORRECT)
!!!!!!!!!!!!!!!!!! Firing: (PARENS_NOT_NEEDED_AND_NOT_USED_FOR_LITERALS SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!! Firing: (PARENS_NOT_NEEDED_AND_NOT_USED_FOR_LITERALS SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!!! Firing: (ALGEBRA_SYMBOLIZE SET_COMPUTE_OR_SYMBOLIZATION)
!!!!!!!!!!!!!!!!!!!!! Firing: (FILL_IN_LITERAL SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!!!!!!!!!! Firing: (SYMB_FILL_IN_VALUE_OF_SUB_EXPRESSION SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!!!!!!!!!!! Firing: (GET_OPERATOR_AND_FINISH SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!!!!!!!!!!!! Firing: (SETUP_OP_AND_ARG_FOR_OUPUT SET_RETRIVE_OP_AND_ORDER)
!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (SELECT_OPERATOR_CORRECT SET_SELECT_OPERATOR)
!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (IF_OPERATOR_DONE_CORRECT_SELECT_ORDER
SET_CHECK_TO_SEE_IF_BUGGY_OPERATOR)
!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (SELECT_ORDER_NON_COM_CORRECT SET_SELECT_ORDER_IF_OP_CORRECT)
!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (LEFT_QUAN SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (LEFT_GOODUSE SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (PUT_IN_PARENLEFT SET_PUT_IN_PAREN)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (PARENS_NOT_NEEDED_AND_NOT_USED_FOR_LITERALS
SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (ALGEBRA_SYMBOLIZE SET_COMPUTE_OR_SYMBOLIZATION)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (RECOGINZE_SYMB_DONE SET_TRAVESE_QUAN_NET)the number of bugs =0
```

This is the list of 65 productions to trace 5g+7(30-g)

```
! Firing: (READ_SYMB_QUESTION SET_BEGIN_MODEL_WITH_QUES)
!! Firing: (SET_CURRENT_INPUTS SET_SET_CURRENT_INPUTS)
!!! Firing: (PICK_RELATION_THAT_HAS_GOAL SET_TRAVESE_QUAN_NET)
!!!! Firing: (PROD_HELP_FINISH_PICK_REALTION SET_FINISH_PICK_REALTION)
!!!!! Firing: (MARK_MENTAL_EXPR_WITH_REALTION_USED SET_MARK_EXPR_WITH_REALTION_USED)
!!!!!! Firing: (MARK_MENTAL_EXPR_WITH_REALTION_USED SET_MARK_EXPR_WITH_REALTION_USED)
!!!!!!! Firing: (PICK_RELATION_THAT_HAS_GOAL SET_TRAVESE_QUAN_NET)
!!!!!!!! Firing: (PROD_HELP_FINISH_PICK_REALTION SET_FINISH_PICK_REALTION)
!!!!!!!!! Firing: (MARK_MENTAL_EXPR_WITH_REALTION_USED SET_MARK_EXPR_WITH_REALTION_USED)
!!!!!!!!!! Firing: (MARK_MENTAL_EXPR_WITH_REALTION_USED SET_MARK_EXPR_WITH_REALTION_USED)
!!!!!!!!!!! Firing: (FILL_IN_LITERAL SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!! Firing: (FILL_IN_LITERAL SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!! Firing: (GET_OPERATOR_AND_FINISH SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!! Firing: (SETUP_OP_AND_ARG_FOR_OUPUT SET_RETRIVE_OP_AND_ORDER)
!!!!!!!!!!!!!!! Firing: (SELECT_OPERATOR_CORRECT SET_SELECT_OPERATOR)
!!!!!!!!!!!!!!!! Firing: (IF_OPERATOR_DONE_CORRECT_SELECT_ORDER SET_CHECK_TO_SEE_IF_BUGGY_OPERATOR)
!!!!!!!!!!!!!!!!! Firing: (SELECT_THIS_ORDER_COMMUNTIVE SET_SELECT_ORDER_IF_OP_CORRECT)
!!!!!!!!!!!!!!!!!! Firing: (PARENS_NOT_NEEDED_AND_NOT_USED_FOR_LITERALS SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!! Firing: (PARENS_NOT_NEEDED_AND_NOT_USED_FOR_LITERALS SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!!! Firing: (ALGEBRA_SYMBOLIZE SET_COMPUTE_OR_SYMBOLIZATION)
!!!!!!!!!!!!!!!!!!!!! Firing: (PICK_RELATION_THAT_HAS_GOAL SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!!!!!!!!!! Firing: (PROD_HELP_FINISH_PICK_REALTION SET_FINISH_PICK_REALTION)
!!!!!!!!!!!!!!!!!!!!!!! Firing: (MARK_MENTAL_EXPR_WITH_REALTION_USED
SET_MARK_EXPR_WITH_REALTION_USED)
```

```
!!!!!!!!!!!!!!!!!!!!!!!! Firing: (MARK_MENTAL_EXPR_WITH_REALTION_USED
SET_MARK_EXPR_WITH_REALTION_USED)
!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (FILL_IN_LITERAL SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (PICK_RELATION_THAT_HAS_GOAL SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (PROD_HELP_FINISH_PICK_REALTION SET_FINISH_PICK_REALTION)
!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (MARK_MENTAL_EXPR_WITH_REALTION_USED
SET_MARK_EXPR_WITH_REALTION_USED)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (MARK_MENTAL_EXPR_WITH_REALTION_USED
SET_MARK_EXPR_WITH_REALTION_USED)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (FILL_IN_LITERAL SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (FILL_IN_LITERAL_SECOND_TIME_IN_PROBLEM SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (GET_OPERATOR_AND_FINISH SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (SETUP_OP_AND_ARG_FOR_OUPUT SET_RETRIVE_OP_AND_ORDER)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (SELECT_OPERATOR_CORRECT SET_SELECT_OPERATOR)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (IF_OPERATOR_DONE_CORRECT_SELECT_ORDER
SET_CHECK_TO_SEE_IF_BUGGY_OPERATOR)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (SELECT_ORDER_NON_COM_CORRECT SET_SELECT_ORDER_IF_OP_CORRECT)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (PARENS_NOT_NEEDED_AND_NOT_USED_FOR_LITERALS
SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (PARENS_NOT_NEEDED_AND_NOT_USED_FOR_LITERALS
SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (ALGEBRA_SYMBOLIZE SET_COMPUTE_OR_SYMBOLIZATION)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (SYMB_FILL_IN_VALUE_OF_SUB_EXPRESSION SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (GET_OPERATOR_AND_FINISH SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (SETUP_OP_AND_ARG_FOR_OUPUT SET_RETRIVE_OP_AND_ORDER)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (SELECT_OPERATOR_CORRECT SET_SELECT_OPERATOR)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (IF_OPERATOR_DONE_CORRECT_SELECT_ORDER
SET_CHECK_TO_SEE_IF_BUGGY_OPERATOR)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (SELECT_THIS_ORDER_COMMUNTIVE
SET_SELECT_ORDER_IF_OP_CORRECT)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (PARENS_NOT_NEEDED_AND_NOT_USED_FOR_LITERALS
SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (RIGHT_QUAN SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (RIGHT_GOODUSE SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (PUT_IN_PARENRIGHT SET_PUT_IN_PAREN)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (ALGEBRA_SYMBOLIZE SET_COMPUTE_OR_SYMBOLIZATION)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (SYMB_FILL_IN_VALUE_OF_SUB_EXPRESSION
SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (SYMB_FILL_IN_VALUE_OF_SUB_EXPRESSION
SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (GET_OPERATOR_AND_FINISH SET_TRAVESE_QUAN_NET)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (SETUP_OP_AND_ARG_FOR_OUPUT SET_RETRIVE_OP_AND_ORDER)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (SELECT_OPERATOR_CORRECT SET_SELECT_OPERATOR)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (IF_OPERATOR_DONE_CORRECT_SELECT_ORDER
SET_CHECK_TO_SEE_IF_BUGGY_OPERATOR)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (SELECT_THIS_ORDER_COMMUNTIVE
SET_SELECT_ORDER_IF_OP_CORRECT)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (LEFT_QUAN SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (LEFT_DONT_NEED_AND_DIDNT_USE
SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (DONT_PUT_IN_PARENLEFT SET_DONT_PUT_IN_PARENS)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (RIGHT_QUAN SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (RIGHT_DONT_NEED_AND_DIDNT_USE
SET_PARENTHESES_POTENTIAL)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (DONT_PUT_IN_PARENRIGHT SET_DONT_PUT_IN_PARENS)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (ALGEBRA_SYMBOLIZE
SET_COMPUTE_OR_SYMBOLIZATION)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Firing: (RECOGINZE_SYMB_DONE SET_TRAVESE_QUAN_NET)the
number of bugs =0
```

Appendix G: The 11-item posttest given in the Ms. Lindquist Controlled Experiment

Name: ___________________________ Date: _________________

Answer the Following Problems. Some questions ask for an equation while other only ask for an expression. Remember that an equation always has an equal sign, while an expression does not.

For each problem you will also be asked to rate your confidence in that answer. Use 10 to indicate that you are sure the answer is correct, and use 1 to indicate that you think the answer is probably wrong and just a guess.

Remember that an ***equation*** has an "=" sign but an ***expression*** does not. Examples of an *expression* are "3x+7", and "4*5-18*2". Examples of *equations* are "3x-5=2y" and "4*5=4*9-10."

1) Joelyn has decided to save $12 a week to buy a stereo system costing $125. Write an expression which shows how much money she will still have to save after "n" weeks.

2) For lunchtime exercise Rachel runs laps around a track. It takes her an average of 7 minutes to run around the track once, and it takes her an 4 minutes to change clothes afterward. Which equation could be used to find "m", the number of minutes it takes for Rachel to job 5 laps and change clothes afterwards.

A) m=(7 * 5) + 4
B) m=(7 * 4) + 5
C) m=(7 * 4) * 5
D) 5 = (m +4) * 7
E) m = 7 * (5 + 4)

3) Juan has 5 fewer hats than Maria, and Clarissa has 3 times as many hats as Juan. If Maria has "n" hats, which of these represents the number of hats that Clarissa has?

a) 5-3n
b) 3n
c) n-5
d) 3n-5
e) 3(n-5)

4) You go on a bike ride. You ride at a speed of "s" miles per hour. You bike "m" miles. You take a "h" hour break. Write an expression for how long the trip took.

5) Tom is exactly one year younger than Bill. Write an equation that precisely shows the relationship between Tom's age and Bill's age. Use "T" for Tom's age and "B" for Bill's age.

Answer: ___________________ Confidence: _____________

6) Penny is eager to drive a car but must wait until she is 16 to apply for a driver's license. Write an equation that relates "n" and "p", where "n" is the number of years Penny must wait to apply for her license and "p" is her present age.

Answer: ___________________ Confidence: _____________

7) Michael is doing his federal income taxes. He comes across a receipt for the cost of the moving van he rented for $900 which included sales tax. The federal government allows to declare your moving expenses and thus to reduce the amount of tax you have to pay, but you can not count the amount you pay in sales tax. Therefore Michael needs to figure out what portion of the $900 dollars was sales tax and what portion was the expense for the moving van. The sales tax rate where Michael is from is 5%. Show you work and put your final answers in the blanks given below.

Cost paid in sales tax:_________ Cost paid for the van:__________ Confidence: _____

8)A grocer noticed that a farmer delivered twice as many brown eggs as white eggs. If there are 144 eggs in the delivery, how many of them are brown? Write an equation you could use to solve this problem.

Answer: ___________________ Confidence: _____________

9) The total cost of a stereo is $230. Jaime put "z" dollars down on the stereo and will pay the rest in 3 equal payments. If "x" is the amount of each payment, write an equation for this problem that you could then use to find the amount of each payment.

Answer: ___________________ Confidence: _____________

10)A factory makes two types of TVs, called Vegas and Romys. The assemble line always produces 200 TV each day but they can be of any type. A Vega TV sells for 100 dollars and a Romy TV sells for 300 dollars. If they make "v" Vegas TVs one day what is the total retail value of all the TV's make that day?

Answer: ___________________ Confidence: _____________

Name:____________________________

11) Mary starts a car washing business. She spends 15 dollars on supplies. She charges 5 dollar per car. On her first day of business she washes "w" cars. Please write an expression for the profit she makes.

12) John and his wife Peg have been saving for 7 months to give the 5 children presents for the holidays. John has saved 972 dollars for presents and Peg has saved "p" dollars. They give each child the same amount. Write an expression for how much each child gets.

13) Bob left at 3 P.M. and drove 550 miles from Boston to Pittsburgh to visit his grandmother. Normally this trip takes him "h" hours, but on Tuesday there was little traffic and he saved 2 hours. What was his average driving speed?

14) A ice cream truck sells ice cream cones in two sizes, small and medium. The price of a medium cone is 3 dollars and a small costs 2 dollars. Today the truck sold "s" small and "m" medium ice cream cones. Write an expression for the amount for the total amount of money the truck took in today.

15) Rebecca makes a "h" hour car trip. For 3 of those hours it was raining and Rebecca drives at 40 miles per hour. The rest of the time it was sunny and she drove 55 miles per hour. Write an expression for total distance Rebecca drives.

16) Jaime drove to the library which is "x" miles away at a speed of 30 miles an hour. Then he drove to his friend's house which is "y" miles away at a speed of 40 miles per hour. Write an expression for how long a period of time Jaime spent driving.

11)Ms. Hart puts a math problem in each square of a 3x3 grid. Each problem solved correctly is one point, with an additional point for every three correct in a row (including up and down, left to right, and diagonal.) The highest score for a 3x3 grid would be 17 points.

Sometimes she uses a 4 x 4 grid, where the students need to now get 4 in a row to get the additional point. Sometimes the grid that is a 5 x 5, 6 x 6, etc. Find a pattern to determine the highest score possible no matter what size the square grid Ms. Hart uses. Write this rule as an expression using "n" to stand for the width of the grid.

Answer: ____________________ Confidence: ______________

Appendix H: Worksheet used during the in classroom teaching portion of the transfer study.

Name: _____________
Teacher: _______________
Period : _______________

To The Student: The objective of this lesson is to help you learn how to compose multiple steps into a single expression.

Here is a problem

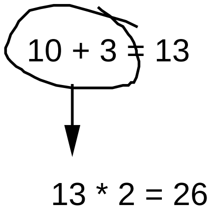
Start with 10 and add 3. Multiply the result by two.

Write the two steps out. Then write one single expression.

Here are the two steps

10+3=13
13*2=26

You will be tested to combine the arithmetic steps into a single expression. This is often a stumbling block for many students. But the idea is that you just substitute one expression into the correct place in the following step. And remember to put parenthesis around the expression that you are substituting in since parenthesis tell you what to do first.



To get (10+3) * 2 = 26

Student Practice:

1) Step 1: 10 + 5 = 15
Step 2: 30 / 15 = 2
Write the composed expression ____________

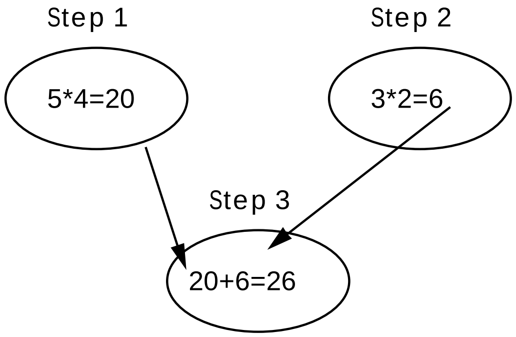
2) Step 1: 20 - 7 = 13
Step 2: 2 * 13 = 26
Write the composed expression ____________

3) Step 1: 5 * 4 = 20
Step 2: 20 / 2 = 10
Write the composed expression ____________

4) Step 1: 4 + 3 = 7
Step 2: 7 * 2 = 14
Write the composed expression ____________

5) Step 1: 10 - 2 = 8
Step 2: 32 / 8 = 4
Write the composed expression ____________

Three Steps



All three steps combined = ________________

6) Step 1: 5 * 4 = 20
Step 2: 3 * 2 = 6
Step 3: 20 + 6 = 26
Write the composed expression ___________

7) Step 1: 6 * 3 = 18
Step 2: 10 - 3 = 7
Step 3: 18 - 7 = 11
Write the composed expression ___________

8) Step 1: 5 - 3 = 2
Step 2: 7 + 1 = 8
Step 3: 8 / 2 = 4
Write the composed expression ___________

Variables for Intermediate Quantities

9) Step 1: 173 * 241 = ? _____________
Step 2: ? / 5 = something new
Write the composed expression ___________

10) Step 1: 151 + 24 = something
Step 2: 10 - something = answer
Write the composed expression for the answer ___________

11) Step 1: 3 - 2 = X
Step 2: X / 4 = 10
Write the composed expression ___________

12) Step 1: 4 * 3 = A
Step 2: 17 - A = 5
Write the composed expression ___________

13) Step 1: 14 - 11 = W
Step 2: 45 / W = 15
Write the composed expression ___________

Variables for Intermediate Quantities and Three Steps

14) Step 1: 5 - 3 = A
Step 2: 7 + 1 = B
Step 3: A / B = 2
Write the composed expression ____________

15) Step 1: 3 + 4 = A
Step 2: 10 - 2 = B
Step 3: B * A = 56
Write the composed expression ____________

With Variables as Givens

16) Step 1: 10 - x = 7
Step 2: 7 * 2 = 14
Write the composed expression ____________

17) Step 1: 8 - 2 = 6
Step 2: 6 * a
Write the composed expression ____________

18) Step 1: a + b = 10
Step 2: 20 / 10 = 2
Write the composed expression ____________

With Variables as Givens and Intermediate Quantities

19) Step 1: x + 3 = A
Step 2: 5 / A
Write the composed expression ____________

20) Step 1: a - b = X
Step 2: 5 / X
Write the composed expression ____________

21) Step 1: a - b = X
Step 2: f * X
Write the composed expression ____________

More Complicated Patterns : Intermediate Quantities being used twice

You can do more complicated substitutions
22) Step 1: a + b = X
Step 2: X * X
Write the composed expression ____________

23) Step 1: a + b + c = X
Step 2: X * 2
Write the composed expression ____________

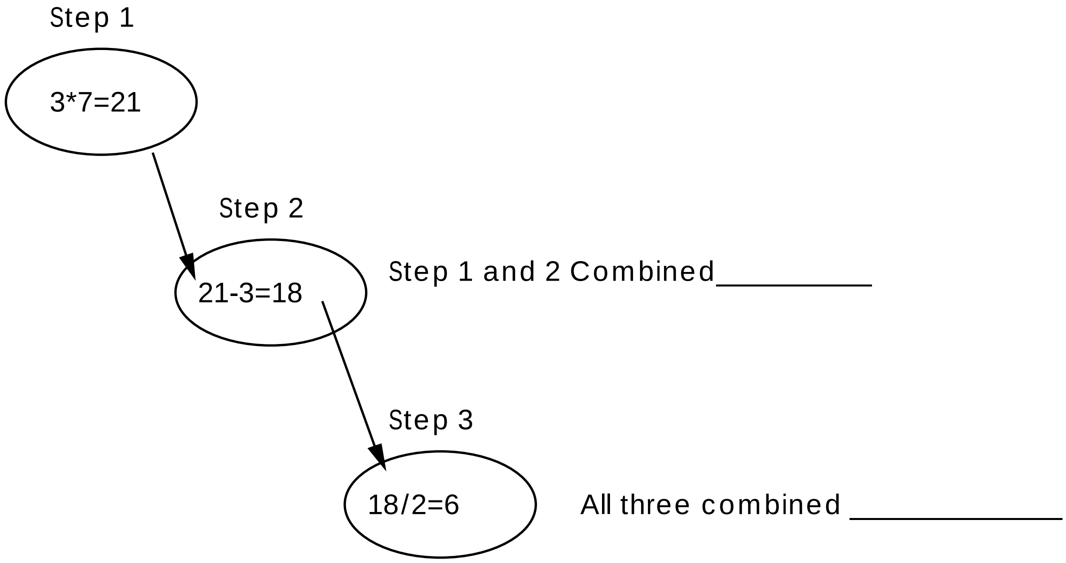
24) Step 1: a + b + c = X
Step 2: X * X
Write the composed expression ____________

25) Step 1: (7+a)*3 = W
Step 2: 4 / W + 57 * W
Write the composed expression ____________

26) Step 1: (7+a)*3 = W

Step 2:   4 / W + 57 * W
Write the composed expression   ___________

Three Steps **Chained** together


Step 1
3*7=21
Step 2
21-3=18
Step 1 and 2 Combined__________
Step 3
18/2=6
All three combined ______________

27) Step 1:   9 - 3 = 6
Step 2:   6 * 3 = 18
Step 3:   20 - 18 = 2
Combine steps 1 & 2   _______________
Now combine your answer with step 3   ________________

28) Step 1:   8 - 3 = 5
Step 2:   10 / 5 = 2
Step 3:   2 + 7 = 9
Combine steps 1 & 2   _______________
Now combine your answer with step 3   ________________

29) Step 1:   17 * 33 = A
Step 2:   567 / A = B
Step 3:   1254 - B
Combine steps 1 & 2   _______________
Now combine your answer with step 3   ________________

30) Step 1:   5 - f   = A
Step 2:   10 / A   = B
Step 3:   13 - B
Combine steps 1 & 2   _______________
Now combine your answer with step 3   ________________

31) Step 1:   9 + 6   = X
Step 2:   X * X   = B
Step 3:   17 - B
Combine steps 1 & 2   _______________
Now combine your answer with step 3   ________________

32) Step 1: a + b = X
Step 2: a / X = Y
Step 3: X + Y
Combine steps 1 & 2 _______________
Now combine your answer with step 3 ________________

Mixed Review

33) Step 1: 10 - a = X
Step 2: X * 2
Write the composed expression ___________

34) Step 1: 5 - 3 = A
Step 2: 7 + 1 = B
Step 3: A / B = 2
Write the composed expression ___________

35) Step 1: 5 - 3 = A
Step 2: 7 + 1 = B
Step 3: A / B = 2
Write the composed expression ___________

36) Step 1: 5 - f = A
Step 2: 10 / A = B
Step 3: 13 - B
Write the composed expression ___________

Transfer
1) Take 50 and subtract “x”. Then subtract the result from 100.
Step 1 Let A = ____________
Step 2 ______________
Composed expression __________________

2) First multiply “x” and “y”. Then add that to 10
Step 1 Let A = ____________
Step 2 ______________
3) Multiply 5 by “g”. Then divide 20 by “f”. Add these two results together.
Step 1 Let A = ____________
Step 2 Let B = ____________
Step 3 ______________
Composed expression __________________

Appendix I: Effect Size and P-values

In chapter 5 we report tests of significance (p-values) as well as effective size (sigma). The test of significance determines what is the probability that the two groups were drawn from the same mean given the differences between the two groups averages and the variance between the groups. Unlike significance tests, effect size indices are independent of sample size. The effect size is a way to compare the size of effects across different experiments. The effect size measurement compares the difference in the means between two conditions and compares that to the size of the variation found in the control group. Specifically, the effect size is calculated by taking the difference in the means between the two groups and dividing by the standard deviation found in the control group. Effect size has units in terms of the standard deviation. Effect size of .5 sigma is respectable. For example, an effect size of 0.8 means that the score of the average person in the experimental group exceeds the scores of 79% of the control group. Here are a few more pairs to give readers an idea of how to interpret an effect size number.

Effect Size	0	.1	.2	.3	.4.	.5	.6	.7	.8	.9
Percentile	50%	54%	58%	62%	66%	69%	73%	76%	79%	.82

For one final example, the often reported result by Bloom (1984) is that tutors are better then classroom instruction with an effect size of two sigma effect size. This means that 98% of the control group would perform below the average of the experimental group (i.e. human tutors).

Appendix J: How Can You Extend the Tutorial Model?

Generally speaking, adding a new KRD (or KCD) means that you need to do the following four steps:

5) Write the tutorial rule that adds the new question(s).
6) Write the student model rule(s) that allow you to trace this answer for each new question asked.
7) Write a tutorial rule that generates the response for each question type added.
8) If you are interested in tracing certain types of errors, you need to add buggy rules to the student model to generate those error types.

Note that, adding a KCD is even easier then a KRD since you do not have to define a specific condition in which it applies (e.g., parentheses error). We will consider each of the four steps in order. We will also illustrate each of these steps with the "error of omission" KCD. An example of an error of omission is when a student is supposed to symbolize a quantity, but instead the student symbolizes only a portion, and that portion represents an answer but for a different quantity. Consider the following problem:

> *Anne is rowing a boat in a lake and is 800 yards from the dock from which she started. She rows back towards the dock at 40 yards per minute for "m" minutes and stops to rest. How far is she from the dock now?*

If the student says "40*m" instead of "800-40*m" they have committed an error of omission. The tutor then says "That is part of the answer. Tell me this about your answer. What does 40*m represent in words?" The student is given a menu to choose from[51]. The tutor is hoping the student will answer correctly that the "40*m" represents " the distance rowed so far". If the student fails, we want to give the hint "Try again. You can do it! What does 40m represent in words?"[52]

Now we are ready to look at each of the 4 steps for adding the "error of omission" KCD. The first step is to add a tutorial rule that adds the new questions. In this case, there is only one new question to add.

Step 1: Write the tutorial rule that represents the strategy

The tutorial rule we have in the model is the following (Given in pseudocode)

```
If the student's answer is correct for some other quantity in the problem
Then
        Create the text of the question (i.e. "That is part of the answer.  What does '40m' represent in words?")
        Create the text for the hints and add those to the question
        Create a new question (of type QREPRESENT)
```

---

[51] The productions given ignore the fact that this particular question actually involves the displaying to the used a menu to choice choices from. As far as the tutorial model is concerned, the student's answer is treated as if it had been typed in like by the user. The actual generation of the choices of the pull down menu and the display of the actual, user interface elements are not done with production rules. Once the student selects a menu item its selection is turned into text and fed into the system as if the student typed it.

[52] In the actual Ms. Lindquist system, there is no hint, but for illustrative purposes we will added this.

Add that to the Agenda

Now we look at the same rule encoded in a TERT1 production. I provide some comments and refer the reader to the TERT1 documentation for the details of TERT1.

```
; ;If we are in a state such that our focus of attention is on the variable =tutor_context
(defproduction error_of_omission SETt_error_of_omission (=tutor_context)
!eval! (not (my_string= "cut" *strat_slctd*));Don't use with Cut to the chase strategy
!eval! (not (my_string= "worked" *strat_slctd*)); don't use with worked example strategy

 =tutor_context>                                      ;and that variable
    isa tutor_context                                 ;is of type tutor_context( The wme just happens to
                                                              ;have the same name as the the wme type )
    current_problem =first_prob                       ;This tutor_context has a current problem we will call =first_prob
    tutors_question_stack ($begin =ques $other)       ;And this tutor_context has The AGENDA(i.e stack)
                                                              ;and that stack has a variable (we label "=ques")
 =ques>                              ;And That question(i.e., "Please symbolize the distance left to row"
                                                      ;that has an answer of "800-40*m")
   isa Question_wme
   goal_slot =goal                   ;has a goal (which we will refer to as =goal) (i.e."The distance left to row")
   Q_type =Q_type                    ;and has a question type (which we will refer to as =Q_type) (i.e. Qsymb)
 =goal>
   isa genrl_goal                    ;And that =goal (i.e. the goal for "the distance left to row")
   mental_expr =mental_expr          ;has a mental-expression (where we store student thinking)
=mental_expr>
   isa mental_expression
    symbolic_string =symbolic_string ;This it the expression the student typed
                                                      ;(i.e. "40*m" but it should have been "800-40m")

;;The next two lines are to find a quantity that has an equivalent symbolic representation.  If this fails, this production won't
fire
  =first_prob>                                        ;And that =first_prob
   isa problem                                        ; is  of type "problem"
   LIST_OF_INST_QUANTITIES ($ =quan $)                ;And there is a quantity (=quan) in this problem

  =quan>
   isa quan_instance                                  ;This is the other quantity
   a_correct_symb =a_correct_symb                     ;that has a symbolization (i.e. "m*40") (we don't yet know it's the same)
   LABEL_OPTIONS (=the_answer)                        ;(i.e. "The distance rowed")
   MENTAL_LIST (=mental)
   - SUB_QUAN_AND_SYMB_LIST nil                       ;this is to make sure we don't ask this sort of question if the
                                                              ;student types just "m", "40" or "800"
   - correct_p ($ Qrepresent $)                       ;and we haven't already done this once

               ;The next line makes some calls to Lisp that runs some function
               ;  that see if the text string evaluates to same sort of answer (i.e.  "m*40 " is the same as "40*m")
               ;!eval! makes a call out to lisp
!eval! (question_correctp =symbolic_string =Q_type =a_correct_symb);is the student's answer correct for some quantity?

==>  ;everything above this line is the "if"  part; everything below this is the "then" part

!eval! =ques_text_stored(format nil "What does ~a represent in words?" =symbolic_string)
               ;this makes a call to Lisp that creates the actual text of the question and is then stored in
                                              ; the variable named "=ques_text_stored"

!eval! =hint1(format nil "Try again.  You can do it!  What does ~a represent in words?" =symbolic_string)
               ;This is the first hint
!eval! =hint2(format nil "The answer is that ~a is ~a.  Select ~a from the pull-down menu." =symbolic_string  =the_answer
=the_answer) ;second hint tells then what to do


  =new_goal>                         ;not important- book keeping but this is creating a new genrl_goal
    isa genrl_goal                   ;
    quantity_of_interest =quan
    MENTAL_EXPR =mental

 =new_ques_represent>                ;This is important;  it create a new Question_wme the the follwing slots:
   isa Question_wme
   goal_slot =new_goal
   Q_type Qrepresent                 ;we call this new question type "Qrepresent"
   ques_text =ques_text_stored       ;store the text of the question
   hint_list_available (=hint1 =hint2) ;store the hints in order
   back_pointer_q =ques              ;complicated so we ignore
   a_correct_answer =the_answer      ;Store the answer to this question so as to make it easy
```

```
                                                   ; for the student model to trace this question

 =tutor_context>
    tutors_question_stack ($begin =ques =new_ques_represent $other)          ;Add that question to the agenda!!!!

  !chain! SETt_utter (=tutor_context  "That is part of the answer. Tell me this about your answer. " )
              ;This line causes the text to be output to the user.
);done
```

We also have to add one control productions that says for what sort of question this KCD can be used. Since we only want this KCD to be applicable for the first symbolization question, the production applies only when the Q_TYPE is Q_SYMB.

```
(defproduction no_and_pos_feed_SYMB_with_only_SUPRABUGs SETt_no_and_pos_feed (=tutor_context
=last_question =q_type =wrong_or_uninter)
=last_question>   ;t
   isa Question_wme
   Q_TYPE Qsymb              ;This means that an error of omission KCD is possible for question of the type Qsymb
==>
!chain! SETt_error_of_omission (=tutor_context) ;should only succeeded if true
)
```

If one also wanted this KRD to be applicable on a symbolize question (i.e. Q_SYMB_AFTER_VERBAL) right after a student is asked an explain question (i.e. QEXPLAIN_VERBAL that is part of the "Explain in English" strategy) we could change the production to the following.

```
(defproduction no_and_pos_feed_SYMB_with_only_SUPRABUGs SETt_no_and_pos_feed (=tutor_context
=last_question =q_type =wrong_or_uninter)
=last_question>   ;t
   isa Question_wme
   Q_TYPE =x
!eval! (or (string= =x "Qsymb")
              (string= =x " Q_SYMB_AFTER_VERBAL")) ;this means we can have this KCD for either question type
==>
!chain! SETt_error_of_omission (=tutor_context)
)
```

Step 2: Write the student model rule(s) that allow you to trace this answer

Then the student model must be able to answer the question. For many simple questions, the answer can already be specified by the tutorial rule, so that the student model can use a single simple rule that tests to see if the answer is correct. Sometimes the questions are more complex, particularly if they are allowing the student to answer with a complete algebraic expression. We have already built a large number of rules that deal with interpreting algebraic expressions and sometimes they can be used. The general idea is that for each new question that the tutorial model asks often, only a small number of rules (usually two) need to be added to the student model.

```
(defproduction read_REPRESENT_question SET_begin_model_with_ques (=problem )

 =problem>                              ;if there is a problem
   isa problem
   the_stack ($ =Question_wme  )        ;and there is question at the top of the stack =Question_wme>
   isa Question_wme
   goal_slot  =goal
   Q_type Qrepresent                    ;and it’s a Qrepresent
   a_correct_answer =the_answer         ;and this is its answer

!eval! (my_string= (string *current-input*) =the_answer);here is where we compare the student's input
                                                          ; with the correct answer stored in the
                                                  variable"=the_answer")
                                                  ;if it is not true this production will not fire and the
                                                          ;student's answer will be considered wrong and un-
                                                  interpretable
==>
   !chain! SET_set_current_inputs (=problem)
  )
```

Step 3: Write a tutorial rule that generates the responses to each question type added

So far we have described how the system generates the new question(step 1) as well as how the student model is used to determine if the student’s response is correct (step2). No additional rules need to be written for the case that the student’s response is correct. But, if the student’s response to the new questions is wrong, we need to specify in the tutorial model how the system should respond. Usually the system simply gives the next hint.

> If the student answered a question of type Qrepresent incorrectly
> Then Give them the next hint.

This is shown by the following rule

```
 (defproduction no_and_pos_feed_Qrepresent SETt_no_and_pos_feed (=tutor_context =last_question Qrepresent
=wrong_or_uninter)
==>
!chain! SETt_stay_on_question (=tutor_context =last_question);this means give the next hint if there is one
)
```

The important point is that for each new question type, we need to specify how the system responses to a wrong answer. This example shows that you can do this very simply with one short rule, if all you want to happen is giving the next hint. If we had wanted the system to give a response like "No, the speed rowing is 40 " we could have made that happen in this rule.

Step 4: If you are interested in tracing certain types of errors, you need to add buggy rules to the student model to generate those error types.

Since this KCD doesn't have any special responses like asking new question if we get some particular response, we don't need to add anything .

We have now shown all the code associated with the error of omission KCD. However, this is a very simple KCD. It have one new question, plus the responses of the tutor to unexpected errors is very the same ("It says 'no""). And it is also true that this KCD does not have an embedded sub-dialogs possible as says does the concrete articulation strategy. Nevertheless, we have shown that it is easy to add a KRD, and not that much harder to add a more complicated KRD. The "Convert The Problem into an Example to Explain" Strategy has just a few additional rules to deal with the three questions it is concerned with. I added that strategy in about one hour and had it fully debugged within another hour. One of the reasons it was easy to do this is that the dynamic scaffolding has been abstracted out so it was easy to change the tutorial questions without having to redo the dynamic scaffolding.

However, this architecture does have its drawback. Coding production rules is hard because it is easy to forget how changes in one area will affect another area. Unfortunately, I have had no time to look at making it easier for non-programmers. Nevertheless, I think the architecture and the system I have built it in are good enough that I plan to continue to build onto this system so that we can experiment and learn more about what makes for good tutorial dialog.