In Search of Bloom's Missing Sigma:

Adding the conversational intelligence of human tutors to an intelligent tutoring system

A thesis submitted to the
Symbolic Systems Program at
Stanford University
in partial fulfillment of the
requirements for the degree of
Master of Science

Heather Pon-Barry

August 31, 2004

I certify that I have read this thesis, and that in my opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science.

_____

Stanley Peters
(Primary Adviser)

I certify that I have read this thesis, and that in my opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science.

_____

Martha Evens

Abstract

The project described in this thesis addresses an issue fundamental to human learning as well as an issue fundamental to the future of human learning: how do humans use language when teaching and learning from one another, and can we endow computers with enough language capability to teach us as effectively as humans do? Specifically, this thesis examines the language used in human-to-human tutorial dialogues with the goal of enhancing the SCoT[1] tutoring system to carry out more natural and more effective human-computer tutorial dialogues, and ultimately to understand what kind of linguistic and communicative devices are most important for successful tutorial dialogue.

This project was conducted in three phases. First, I examined data from human tutors to find principles that apply across domains about how human tutors take advantage of natural language in conversational interaction, and formed hypotheses about how tutors respond to signals of uncertainty in student utterances. Guided by my observations from this empirical data and with the goal of creating a system that can flexibly switch between various tutorial strategies, I redesigned the framework of the SCoT tutorial component and contributed to its reimplementation. Finally, using this new version of SCoT, I ran an evaluation comparing the effectiveness of two styles of tutoring that differed only in the language used by the tutor in order to test my hypotheses about responding to student uncertainty.

[1] See Chapter 1 for more information about the SCoT tutoring system.

Acknowledgements

I would like to thank my adviser, Stanley Peters, for his encouragement and his help in guiding my project, and for always asking questions that helped me refine my ideas and see things in new ways. I also wish to thank my second reader, Martha Evens, for her invaluable feedback on drafts of this thesis.

I could not have completed this project without the help of my colleagues at CSLI. Brady Clark, Karl Schultz, Elizabeth Owen Bratt, and Simon Berring all invested many hours in helping me prepare our tutoring system for the experiments described in this thesis. I am also grateful to Herb Clark, Teenie Matlock, Lee Martin, and Dan Schwartz, for their advice about experimental design and analysis. Finally, I'd like to thank my family and friends for their constant encouragement and support.

This work was supported by the Cognitive Science Program, Office of Naval Research, under Grant No. N00014-00-1-0660 to Stanford University. The content does not reflect the position or policy of the government and no official endorsement should be inferred.

Table of Contents

1 Introduction

Two decades ago, Benjamin Bloom reported on two studies showing that students who interacted with expert human tutors yielded test scores two standard deviations above those who received ordinary classroom instruction (Bloom, 1984). In other words, the average student in the tutoring condition performed better than 98% of the students in the classroom condition. Two decades later, this "2-sigma" effect is still commonly used as the gold-standard for measuring instructional effectiveness. Researchers in various fields have been building intelligent tutoring systems for over three decades—with a recent trend towards dialogue-based tutoring systems—yet the highest learning gains reported by current systems are only one standard deviation above classroom instruction or other control groups (e.g., Koedinger et al., 1997; Person et al., 2001). Naturally this leads one to wonder, what happened to the second sigma? What is it that expert human tutors do that makes their tutoring so effective?

At the Center for the Study of Language and Information (CSLI), we have been exploring the idea that natural language interaction may account for part of Bloom's missing sigma. Specifically, we are interested in whether the spoken nature of human-to-human tutorial dialogue is an essential part of its effectiveness. In order to test this hypothesis, we built SCoT (a Spoken Conversational Tutor). The first version of SCoT was completed in 2001, and led a conversation based on a simple dialogue move graph. The second version of SCoT was completed in 2002, and incorporated more sophisticated techniques for dialogue management and tutoring. The third version of SCoT (described in Chapter 4) was completed in 2004 in conjunction with the work presented in this thesis. This version of SCoT was designed with an emphasis on portability in application to new subject matter, and flexibility in planning and in switching between multiple tutoring styles.

The goal of my project is to get a better understanding of the linguistic and meta-communicative devices that human tutors utilize in tutorial interaction. I have examined certain features of the student language that can be used by tutors to guide their choice of response and to present information at the appropriate level. Adapting to their behavior in

this way should make it easier for students to build a clear mental representation of what is being discussed, facilitating self-reflection and a better understanding of the material.

This thesis is organized as follows. Chapter 2 gives an overview of relevant literature in the areas of human tutoring and learning as well as in intelligent tutoring systems. Chapter 3 describes the hypotheses I developed about how human tutors make use of linguistic devices and meta-communicative information in facilitating learning. Chapter 4 describes the framework of the SCoT tutorial engine, which I redesigned so that it could flexibly switch between multiple tutoring styles. Chapter 5 describes the evaluation I conducted to test out my hypotheses using the new version of SCoT. And finally, Chapter 6 discusses the results found and the conclusions drawn.

2 Previous and Related Work

The motivation for carrying out a project of this nature stems from related work in the areas of human tutoring, computer tutoring, and dialogue systems. In this chapter, I will discuss work that has been done in each of these areas, and explain how it relates to the research project I have undertaken.

2.1 Human Tutoring

The effectiveness of one-on-one human tutoring is well-documented and has motivated much of the development of computer tutoring systems. Understanding why human tutoring is particularly effective is an area which has also received much attention—although open questions remain regarding how to capture this effectiveness in a computer tutor. Findings from these two areas are discussed in Sections 2.1.1 and 2.1.2.

2.1.1 Effectiveness

One-on-one tutoring has been shown to be a highly effective mode of instruction, for problem-solving domains (e.g., physics, electronics, algebra) in particular. A meta-analysis of tutoring studies, consisting largely of studies of peer-tutoring, found larger gains in the tutoring condition than in the non-tutoring condition (an effect size of 0.4 standard deviations) (Cohen, Kulik & Kulik, 1982). With expert tutors, studies comparing one-on-one tutoring to ordinary classroom instruction found even larger differences in performance gains (an effect size of 2.0 standard deviations) (Bloom, 1984). Although Bloom's result is often cited as a gold-standard, it may be less ecologically valid than the Cohen et al. result because the studies he reports on compare students learning completely new material exclusively from expert tutors over a period of 3 to 4 weeks, whereas most tutoring occurs with non-expert tutors in conjunction with classroom instruction. While Bloom's result demonstrates the potential of what one-on-one tutoring can achieve, it may be hard to attain in practice.

Along these lines, it is important to note that while many studies report learning gains in terms of "sigmas" (standard deviations) most differ along some or all of the

following parameters: the expertise of the tutors (expert or novice), the age of the students, whether the students were learning a new subject or reviewing previously learned material, whether the tutoring was in place of or in conjunction with classroom instruction, and if in conjunction with classroom instruction—what the control condition was. Evens and Michael (unpublished manuscript) lay out a concise description of factors such as these worth considering when evaluating the effectiveness of human tutoring.

2.1.2 Theories of Learning

Studies of human-to-human tutorial interaction have identified particular features of tutorial dialogue that may account for its effectiveness. One such feature is the collaborative nature of tutorial dialogue—students learn more when they are actively solving problems rather than passively listening (Fox, 1993; Graesser, Person & Magliano, 1995; Rosé, Moore, VanLehn & Allbritton, 2001). Collaborative construction of knowledge is not unique to tutorial dialogue; it is a feature that characterizes most human conversational interactions. Clark (1996) characterizes conversational discourse as a joint activity—an activity where participants coordinate with each other to achieve public and private goals—because speakers help each other in the process of constructing messages. Another frequent feature in tutorial dialogue is self-explanation—the process of explaining one's steps during problem solving. It has been shown that students learn with deeper understanding when tutors elicit self-explanations during the dialogue (Chi et al., 1994; Aleven & Koedinger, 2002).

In a survey of human tutoring studies, Merrill et al. (1992) note that most studies agree that the effectiveness of experienced tutors is due to the ability to "maintain a delicate balance" between letting the students do as much as possible on their own, and providing appropriate guidance when necessary. Merrill et al. also note that the studies diverge in their claims about how and when tutors give feedback—while some researchers assert that effective tutors give subtle and implicit feedback (Lepper et al., 1990; Fox, 1991), others claim that effective tutors give feedback that is overt and often directive (McArthur et al., 1990).

Since Merrill et al.'s study, further analyses of human tutoring have identified specific dialogue tactics that human tutors use to give feedback when responding to student answers (Graesser et al., 1995; Heffernan, 2001; Evens & Michael, unpublished manuscript). These include tactics such as pumping the student for more information, giving a concrete example, and making reference to the dialogue history. Furthermore, transcripts have been analyzed in order to understand patterns relating the category of a student utterance (e.g., partial answer, error-ridden answer, request for clarification) with the category of a tutor response (e.g., positive feedback, leading question) (Person & Graesser, 2003).

The majority of dialogue-based tutoring systems currently rely on typed student input, so the information available from a student utterance is limited to the content of what the student typed. In contrast, human tutors have access not only to the words uttered by the student, but also to meta-communicative information such as timing, or the way a response is delivered; they use this information to diagnose the student and to choose appropriate tactics (Fox, 1993). My project aims to get a better understanding of how human tutors make use of this information, and to see if incorporating it into the behavior of an intelligent tutoring system can increase its effectiveness.

2.2 Intelligent Tutoring Systems

Intelligent tutoring systems (ITSs) are educational tools that draw upon ideas from cognitive psychology as well as artificial intelligence. Replicating the effectiveness of human tutors has been a motivational force driving the development of many ITSs over the past few decades, and many systems have proven useful in helping students understand complex problem-solving domains.

2.2.1 Past and Current ITSs

Some of the original intelligent tutoring systems were organized around ideas pioneered by Newell and Simon (1972) about human problem solving and how to understand cognitive tasks (Anderson, Boyle & Reiser, 1985). These "cognitive tutors" used a technique called model-tracing (Anderson et al., 1990) which compares student problem-solving steps to

expert problem steps and gives feedback when these steps differ in order to keep the student on the right solution path. The model-tracing methodology is still popular today, and although concerns have been raised (e.g., about the rigid path, or the lack of self-exploration), studies have found that the basic model-tracing paradigm is in fact very close to techniques employed by human tutors (Merrill et al., 1992).

Model-tracing tutors have been deployed in public schools and have been shown to be more effective than classroom instruction alone (Koedinger et al., 1997; Shelby et al., 2001). However, the effectiveness of both expert and novice human tutors (Bloom, 1984; Cohen et al., 1982) suggests that there is room for more improvement. One of the main differences between these early model-tracing tutors and human tutors is the fact that these early tutors used menu- and mouse-based interaction whereas human tutors use natural language to communicate.

This difference has motivated the development of a new wave of ITSs—intelligent tutoring systems with natural language capabilities, often referred to as tutorial dialogue systems. Some tutorial dialogue systems employ short, directed natural language dialogues when the student goes down an incorrect path or completes a step requiring explanation (Rosé et al., 2001; Zinn, Moore, & Core, 2002; Aleven, Koedinger, & Popescu, 2003). Other tutorial dialogue systems lead longer natural language dialogues throughout the problem-solving process (Evens et al., 2001; Graesser et al., 2001; Heffernan & Koedinger, 2002). Current results from dialogue-based tutoring systems are promising; Person et al. (2001) found average learning gains of 0.7 standard deviations greater than the control, and Rosé et al. (2001) found that adding natural language capabilities to an existing model-tracing tutor increased learning gains by 0.9 standard deviations. These results suggest that dialogue-based tutoring systems may be more effective than tutoring systems with no dialogue. However, most of these systems use either keyboard-to-keyboard interaction or keyboard-to-speech interaction (where the student's input is typed, but the tutor's output is spoken).

This progression towards human-like use of natural language suggests that tutoring systems with speech-to-speech interaction might be even more effective. The current state of speech technology has allowed researchers to build successful spoken dialogue systems in a

large variety of domains (see Section 2.3). There is reason to believe that spoken tutorial dialogue systems can be just as successful.

2.2.2 Spoken Interaction and ITSs

Spoken tutorial dialogue systems have the potential to be even more effective than typed dialogue systems because the system has access not only to what students say, but to *how* they say it. One idea currently being explored is that prosodic information from the speech signal can be used to detect student emotion, allowing developers to build a more responsive tutoring system (Litman & Forbes, 2003). Another advantage of speech is that spoken input contains meta-communicative information such as hedges, pauses, and disfluencies which can be used to make better inferences about the student's understanding[2] (Pon-Barry et al., 2004). Also, speech allows students to use their hands for other gestures (e.g., pointing to objects in the workspace) while speaking.

Also, recent evidence suggests that spoken tutorial dialogues are more effective than typed tutorial dialogues. A study of self-explanation (the process of explaining solution steps in the student's own words) has shown that spontaneous self-explanation is more frequent in spoken rather than typed tutorial interactions (Hausmann & Chi, 2002). In addition, a comparison of spoken vs. typed human tutorial dialogues showed that the spoken dialogues contained higher values for features such as proportion of student words to tutor words, which has been shown to correlate with student learning (Rosé et al., 2003).

Currently, there are just a few spoken tutorial dialogue systems. One existing system, ITSPOKE, was built by adding a speech interface to an existing (typed) tutorial dialogue system (Litman & Silliman, 2004). The SCoT tutoring system (Clark et al., 2001) was not built around an existing ITS, but rather around an existing architecture for dialogue management and conversational intelligence (see Section 2.3).

2.3 Spoken Dialogue Systems

[2] Typed dialogue contains many of these meta-communicative features as well, but there is reason to believe that some features are more frequent in spoken dialogue. See Section 3.2.1 for further discussion.

A spoken dialogue system is a system that interacts with users through natural language across a speech interface. Successful spoken dialogue systems have been built in domains ranging from travel planning (Walker et al., 2002), to in-car route navigation (Belvin, Burns & Hein, 2001), to command-and-control devices (e.g., human-controlled robots) (Lemon et al., 2002a). These systems vary greatly in dialogue complexity: most over-the-phone travel planning systems use simple slot-filling mechanisms, whereas command-and-control devices need more robust dialogue management in order to interpret commands in context.

Conversational dialogue is highly complex; not only do user utterances need to be interpreted in context, but system utterances need to be generated in light of both user and system goals (which vary depending on the activity at hand). In addition, the system must have some representation of the collaborative activities of the dialogue. At CSLI, a general purpose architecture supporting multi-modal, mixed-initiative dialogue has been developed (Lemon et al., 2002). This *Conversational Intelligence Architecture* is used for managing dialogue in the SCoT tutoring system as well as for other dialogue systems (e.g., controlling a semi-autonomous helicopter, and an in-car MP3 player). The *Conversational Intelligence Architecture* manages issues of conversational intelligence such as turn taking, construction of a structured dialogue history, anaphora resolution, and use of discourse markers.

For a conversational tutoring system, this structured representation of the dialogue and of the activity is extremely helpful because it organizes the dialogue into topics and interprets incoming student utterances—taking this burden off the artificially intelligent tutor, and making it easy for the tutor to do things like refer back to past dialogue and support the discussion of multiple topic threads.

3 Empirical Work

The question I set out to investigate is, how do human tutors facilitate learning through natural language interaction? How do they help students actively construct knowledge and integrate new information with existing knowledge? If students' language really does provide a window into their thinking process, what specifically do tutors attend to, and how does it affect their choice of response?

When I began this project, I had a notion that human tutors vary the way they respond to and present information to students depending on the students' language as well as the manner in which the answer was spoken. However, I did not have concrete ideas about how tutors did this, and as mentioned in Chapter 2, this issue is relatively unexplored in previous work. In order to refine my ideas into specific, operationalizable hypotheses, I analyzed transcripts of one-on-one human tutoring in multiple domains to discover exactly what human tutors do in practice.

3.1 Data

I examined transcripts of human tutorial dialogues in the domains of physiology, algebra, and shipboard damage control.

The dialogues in the domain of physiology came from the CIRCSIM-Tutor corpus of human tutorial dialogues collected by M. Evens, J. Michael, and A. Rovick at Rush Medical College. The corpus includes transcripts of 6 face-to-face tutoring sessions (approximately 2000 dialogue turns) and 75 keyboard-to-keyboard tutoring sessions. Of the 75 keyboard sessions, 5000 lines (excerpts from 13 sessions) have been annotated with tutorial goal structure, student answer classifications, and other relevant information. These dialogues were collected in order to guide the development of the CIRCSIM-Tutor system (Evens et al., 2001; Michael et al., 2003).

The dialogues in the domain of algebra came from the Ms. Lindquist (Algebra Tutor) corpus of human tutoring dialogues collected by Neil Heffernan (Heffernan, 2001). The corpus includes a transcript of a one-on-one hour-long tutoring session between an

experienced mathematics tutor and an eighth grade student working through 17 algebra problems. The transcript contains approximately 400 dialogue turns.

The dialogues in the domain of shipboard damage control were collected at the US Navy's Surface Warfare Officer's School (SWOS) in Newport, RI. Fifteen instructor-student debriefs (following student sessions with an early version of the DC-Train damage control simulator) were videotaped by J. Sniezek of the University of Illinois at Urbana-Champaign. The videotapes were subsequently transcribed at CSLI. The 15 debriefs contain approximately 240 dialogue turns in total. It should be noted that these dialogues are more like critiques than tutorial dialogues, so some instructor turns are fairly long (the longest was over 900 words, but most instructor turns ranged between 100 and 400 words).

3.2 Analysis

3.2.1 Motivation

It has been observed that "tutors use the timing of a student's response, and the way the response is delivered, in addition to what might be called the 'literal content' of the response, as a source of diagnostic information" (Fox, 1993). In my investigations, I was interested in both the timing and the delivery of responses—in particular, in cues that can signal student uncertainty. I focused my attention towards the cues shown below in Table 1.

Type of Cue	**Example**
Lexical	hedges (e.g., "I think…", "Maybe…")
Temporal	response latencies mid-sentence pauses filled-pauses (e.g., "uh", "um")
Other	trailing off at the end of a sentence fragmented or incomplete sentences

Table 1. Signals of Uncertainty

Studies in psycholinguistics have shown that when answering questions, speakers produce hedges, filled-pauses, and rising intonation when they have a lower "feeling-of-knowing" (Smith & Clark, 1993) and that listeners are sensitive to these phenomena (Brennan & Williams, 1995). However, it is not entirely clear if these same features are present in tutorial dialogue, and if they are present, how human tutors respond to them. My investigation was aimed at answering these questions.

One difference between the CIRCSIM 5000 lines of annotated dialogue and the other transcripts is that the annotated lines came from keyboard-to-keyboard interaction whereas the rest of the transcripts all came from spoken interaction. There is reason to believe that the signals of uncertainty I am interested in occur with different frequency in typed versus spoken interaction. In a Wizard-of-Oz style comparison of typed vs. spoken communication (to access an electronic mail system), the number of filled-pauses was found to be significantly higher in speech than in typing (Hauptmann & Rudnicky, 1988). There are no formal analyses comparing the relative frequencies of hedges in speech vs. typing. However, Bhatt (2004) classifies hedges in the CIRCSIM keyboard-to-keyboard corpus and lists the frequencies of various hedge categories. Looking at the same hedge categories as Bhatt, I counted the occurrences of the 5 most frequent hedges in an excerpt from the CIRCSIM face-to-face corpus of the same length (approximately 2000 dialogue turns), and found that some hedges (e.g., "I guess") are significantly more frequent in speech, while other hedges (e.g., "I think") are equally frequent in both speech and typing.

3.2.2 Methods

Due to the modality differences mentioned above, I divided the data into two groups—one for typed tutorial dialogue and one for spoken tutorial dialogue. The typed dialogue group contains just the 5000 lines of annotated transcripts from the CIRCSIM corpus. The spoken dialogue group contains the CIRCSIM face-to-face transcripts, the Ms. Lindquist transcript, and the damage control transcripts. In this chapter, I will refer to these two groups as the 'typed dialogue transcripts' and the 'spoken dialogue transcripts'.

Because the typed dialogue transcripts were annotated for tutorial goals, I used them as a starting point from which to get a preliminary understanding of the tactics tutors use in responding to uncertain student answers. The first signal of uncertainty I looked at was hedging.[3] Bhatt, Argamon, & Evens (2004) outline a list of hedge categories, a subset of which I adopted for this investigation. They are shown below in Table 2.

Hedge Keywords
"I think"
"I thought"
"probably"
"I guess"
"I'm not sure"
"kind of"
"I believe"
"maybe"
"it sounds as though"
"X should…"
"it shouldn't X, should it?"
answers phrased as questions

Table 2. Hedge keywords

In order to understand the distribution of tutor responses to hedged student answers, I analyzed answer-response pairs from the typed dialogue transcripts along the dimensions of incorrect vs. correct and hedged vs. non-hedged. Results are described below in Section 3.2.3.

Two of the response tactics identified in the typed dialogue transcripts, reminding the student of past dialogue and paraphrasing the student's answer, involved linguistic manipulation of the sort I was interested in, so as the next step I looked for further patterns involving these tactics in the spoken dialogue transcripts.

[3] I do not assume that hedging always indicates uncertainty, but rather that hedging *can* indicate uncertainty. Furthermore, I do not intend to suggest that hedged or uncertain answers are more likely to be incorrect. In fact, Bhatt (2004) found that students' hedges are not a reliable cue to errors or misconceptions.

3.2.3 Observations and Hypotheses

In the typed dialogue transcripts, tutor responses to hedged answers occurred with the following general distribution:

	Incorrect answers (n = 17)	**Correct answers (n = 25)**
Hedge	Refer back to past dialogue Point out misconception Follow incorrect line of reasoning	Paraphrase student answer
No Hedge	Inform of mechanism Try different line of reasoning	Acknowledge & move on

Table 3. Categories of Tutor Responses to Student Answers

At first it may appear that the various tactics for responding to student answers in Table 3 have no pattern to their distribution. However, a closer glance reveals that the tactics used in responding to hedged answers all involve elaboration on the current topic while the tactics used in responding to non-hedged answers do not. It makes sense that a tutor might elaborate on the current topic—either to fill in possible gaps in knowledge or to give positive reinforcement for known material.

The first tutorial tactic I examined was referring back to past dialogue. By referring back, I mean a construction where a tutor reminds the student of something previously discussed. Possible instances were identified in the transcripts by looking for key words such as "earlier", "we said", and "you told me" in utterances spoken by the tutor and then judging whether the utterance was indeed referring back to past dialogue. Of the 1600 turns (from the spoken dialogue transcripts) examined, 31 instances of a tutor reminding a student of something previously discussed were identified. Of these 31 instances, 21 followed an incorrect student answer containing a lexical hedge, filled-pause, mid-sentence pause, or trailing off at the end. Five of the 31 instances followed an incorrect answer without any

signs of uncertainty, and 5 instances were not in response to a student answer. Two examples of referring back after uncertain answers are shown below in Figures 1 and 2. The example in Figure 1 is from the CIRCSIM-Tutor face-to-face corpus, it shows a student answer containing the hedge "I guess" and a sentence that trails off at the end. The example in Figure 2 is from the Ms. Lindquist corpus, it shows a student answer containing many mid-sentence pauses.

Tutor: Which of these parameters, if any, reflects filling of the ventricle?
Student: Well, I guess the right atrial pressure certainly does, so it depends on how..
Tutor: OK.
Tutor: So you've told me that stroke volume was determined by contractility and right atrial pressure, but you haven't predicted how either of those change.

Figure 1. Example of Reference to Previous Dialogue

Student: 600-30+20 divided by :::::::::::::: two :::::::: no this parts wrong ::: [writes 600-(30+20)/2 and then scratches out the 600-]
Tutor: Right.
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?

Figure 2. Example of Reference to Previous Dialogue (colon = 0.5 sec pause)

These examples support the generalization that tutors typically refer back to past dialogue following incorrect student answers that contain hedges, mid-sentence pauses, or that trail off at the end. It is plausible for a tutor to purposefully remind a student of previous dialogue when the student shows signals of uncertainty because it encourages reflection. Chi (2000) argues that self-reflection often leads to self-repair, and that compared to hearing corrective feedback, students learn more when encouraged to reflect. Also, analyses of human tutorial dialogue have shown that tutors generally exploit prior

explanations rather than repeating the same information twice (Moore, Lemaire & Rosenblum, 1993).

The next tutorial tactic I examined was paraphrasing. Possible paraphrases in the transcripts were identified by looking for key words such as "so" or "said…?", as well as consecutive student-tutor turns where the same word occurred in both turns, and then judging whether the utterance was indeed a paraphrase of the student's answer. Of the 1600 turns (from the spoken dialogue transcripts) examined, 15 instances of a tutor paraphrasing a student's answer were identified.[4] Of these 15 cases, 11 followed a correct student answer containing a lexical hedge, a mid-sentence pause or a sentence that trailed off, 3 followed correct student answers with no signs of uncertainty, and 1 followed an incorrect student answer. Two examples of paraphrasing are shown below in Figures 3 and 4. The example in Figure 3 is from the CIRCSIM corpus, it shows a series of student utterances containing the hedge "I guess" as well as three sentences that trail off at the end. The example in Figure 4 is from the Ms. Lindquist corpus, it shows a series of student utterances containing the filled-pause "um" as well as mid-sentence pauses.

Tutor: And [initial fiber resting length] relates to which of these parameters?
Student: Let's see, initial fiber resting length would be...
Student: I'd say it's the preload which is...
Student: Well, it relates to the stroke volume, but that's ...
Tutor: Now the question is what determines stroke volume, and you told me contractility, and what else?
Student: Well, I guess if the right atrial pressure were a lot higher, then there would be more of an impetus for the blood to go into the right ventricle, and that would increase that somewhat.
Tutor: So right atrial pressure represents one of the determinants.
Student: Yes.
Tutor: OK.

Figure 3. Example of paraphrasing a student's answer

[4] Although only 15 instances of paraphrasing a student's answer were identified, many more instances of paraphrasing non-answers were found.

Student: [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?

Student: Um :::::::::::::::::::: writes "s/m+b" ::::::::::::::::::::::::::::::::

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?

Student: Um 5

Tutor: 5 and how did you get that 5? How did you use the numbers 100 and

Student: 100 miles divided by miles per hour

Tutor: So you took the miles and divided it by the speed.

Figure 4. Example of paraphrasing a student's answer

These examples support the generalization that tutors paraphrase correct student answers that contain hedges, mid-sentence pauses, or that trail off. It is plausible for a tutor to paraphrase a student's answer when there are signals of uncertainty because the paraphrasing reinforces knowledge that the student may be uncertain of and helps them think about the answer more concisely (as in Figure 3) or verbalize it with the appropriate language (as in Figure 4). Furthermore, paraphrasing can be seen as an attempt to *ground* the conversation, to establish joint actions as part of a common ground (Clark, 1996) and let the student know that s/he has succeeded in communicating the information s/he was attempting to convey.

The final construction I looked into was reasking versus rephrasing a question after giving an acknowledgement and/or hint. Reasking means repeating the original question whereas rephrasing means asking a different question that gets at the same concepts as the original question. The generalizations that I found in the typed dialogue transcripts are summarized below in Table 4.

Reasking is more frequent after:	**Rephrasing is more frequent after:**
Tutor gives "inform of rule" hint Tutor reminds of past dialogue	Tutor states possible misconception Tutor follows incorrect line of reasoning Student answer that was a near-miss

Table 4. Context before reasked versus rephrased questions

Based on all the generalizations found, the two hypotheses that I decided to evaluate using the SCoT tutoring system are:

1. Tutors that respond to correct student answers containing signals of uncertainty by paraphrasing the student's answer will be more effective than those who do not.
2. Tutors that respond to incorrect student answers containing signals of uncertainty by referring back to previous dialogue will be more effective than those who do not.

This evaluation is described further in Chapter 5.

4 SCoT

This chapter gives a brief overview of the SCoT tutoring system and then describes the work I did towards designing and implementing a new tutor component (one of multiple components that comprise SCoT). My work on redesigning the tutor component was done in conjunction with work conducted by my colleagues towards reimplementing other components of SCoT. The result was a new version (version 3) of SCoT that was more flexible and more modular than the previous version. Crucially, this new version made it possible to run an experiment comparing various tutoring styles in order to test out the hypotheses described in Chapter 3.

4.1 Overview of SCoT

SCoT (**S**poken **Co**nversational **T**utor) is an intelligent tutoring system developed at the Center for the Study of Language and Information at Stanford University in conjunction with research on natural language interaction in intelligent tutoring systems.

The design of SCoT is based on the assumption that the tutoring is a joint activity[5] where the content of the dialogue (language and other communicative signals) follows basic properties of conversation but is also driven by the activity at hand (Clark, 1996). Following this hypothesis, SCoT's architecture separates conversational intelligence (e.g., turn management, construction of a structured dialogue history, use of discourse markers) from the activity that the dialogue accomplishes (in this case, reflective tutoring).

SCoT-DC, the current instantiation of the SCoT tutoring system, is applied to the domain of shipboard damage control. Shipboard damage control refers to the task of containing the effects of fires, floods, explosions, and other critical events that can occur aboard Navy vessels. Students carry out a reflective discussion with SCoT-DC after completing a problem-solving session with DC-Train (Bulitko & Wilkins, 1999), a fast paced, real time, speech-enabled training environment for damage control. Figure 5 shows a screenshot of DC-Train.

[5] See explanation of joint activities in Section 2.1.2.

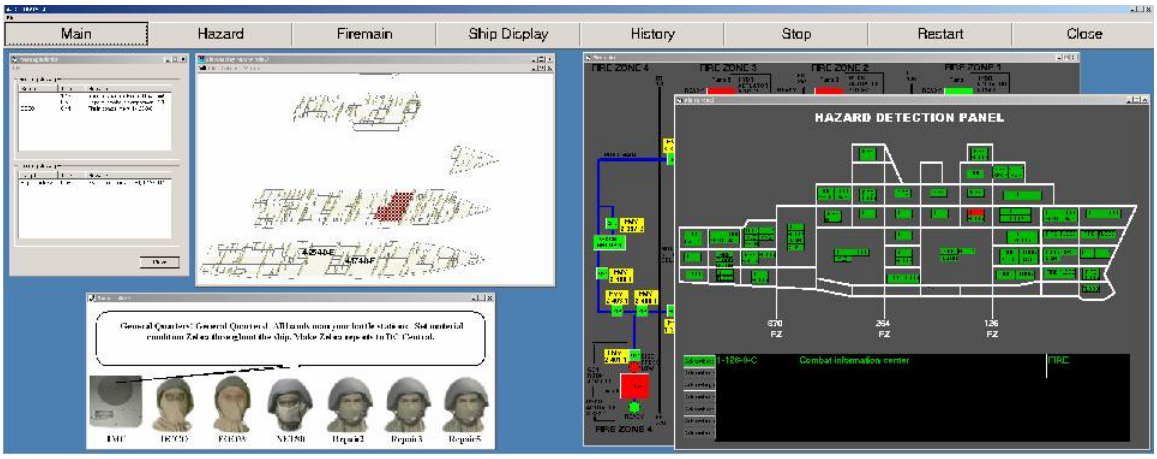

Figure 5. DC-Train Simulator Interface

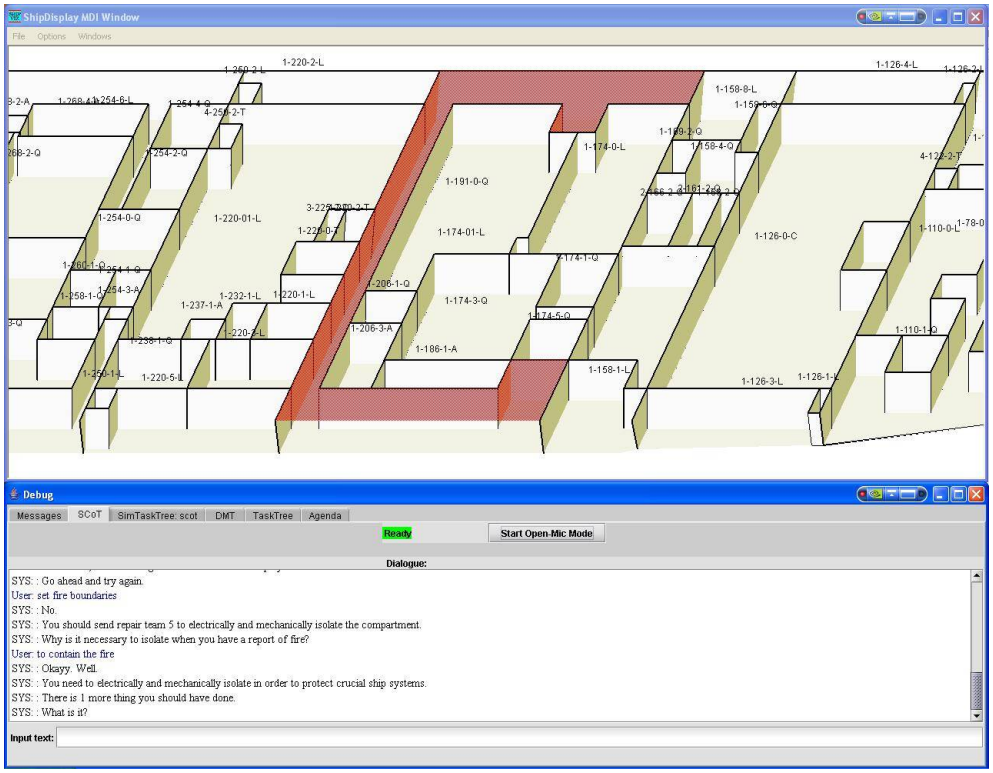

Figure 6. SCoT-DC Tutor Interface

Figure 6 above shows a screenshot of SCoT-DC. The bottom window contains a history of the tutorial dialogue; the top window is the common workspace—a space where both student and tutor can zoom in or out and select (i.e., point to) compartments, regions, or bulkheads (lateral walls) in the ship. In Figure 6, the tutor is "pointing" to the location of a crisis by zooming in to the deck it is on and highlighting the compartment

SCoT is composed of many separate components. In addition to the tutor component, which will be described further in the remainder of this chapter, the other primary components are the dialogue manager, the knowledge representation, and a set of natural language processing components.

The *dialogue manager* mediates communication between the system and the user by handling aspects of conversational intelligence such as turn management and coordination of multi-modal input and output. It contains multiple dynamically updated components—the two main ones are (1) the dialogue move tree, a structured history of dialogue moves, and (2) the activity tree, a hierarchical representation of the past, current, and planned activities initiated by either the tutor or the student.

The *knowledge representation* provides SCoT with a domain-general interface to domain-specific information. In accordance with production-system theories of cognition (Anderson, 1993), knowledge specifying causal relationships between events on the ship and between actions and their preconditions is encoded in a set of production rules. A knowledge reasoner operates over this production system to provide the tutor with procedural explanations of domain-specific actions, as well as information about the problem-solving session.

The *natural language processing* components that make the spoken dialogue possible include a bi-directional unification grammar and state-of-the art software for automatic speech recognition, parsing, sentence generation, and text-to-speech synthesis. Incoming student utterances are handled by SCoT in the following way. First, the utterance is recognized using Nuance[6] speech recognition, which uses a language model generated from a Gemini natural language understanding grammar. Gemini (Dowding, Gawron, Appelt,

[6] http://www.nuance.com

Cherny, Moore & Moran, 1993) translates word strings from Nuance into logical forms, which the dialogue manager interprets in context and routes to the tutor. The system responds to the student via a FestVox[7] limited domain synthesized voice.

Further information about the architecture of SCoT can be found in (Clark, Lemon, Gruenstein, Bratt, Fry, Peters, et al., in press) and (Schultz, Bratt, Clark, Peters, Pon-Barry, & Treeratpituk, 2003).

4.2 Motivations for a New Design

The second version of SCoT's tutor component improved upon the first in many ways, and in the Spring 2004 evaluation of SCoT (Pon-Barry, Clark, Bratt, Schultz & Peters, 2004b) it successfully helped students learn damage control. However, it also had a number of shortcomings. It was only able to lead one kind of dialogue (i.e., support only one overall tutoring strategy), and domain knowledge was built directly into the tutor component, so altering low-level tactics (e.g., how to respond to a student answer) was very difficult.

In order to evaluate the hypotheses discussed in Chapter 3, it was necessary to control SCoT to switch between multiple tutoring styles. The previous version of SCoT's tutoring component did not easily support this—it was clear that the previous version of SCoT's tutoring component had to be improved further.

4.3 Design of SCoT Tutor Component

The tutor component in the third version of SCoT differs from the one in the previous version in the following ways:

1. There is a clean separation between domain knowledge and tutorial knowledge (domain knowledge is extracted out into its own module/component)
2. Tutorial knowledge is divided between a *planning and execution system* and a *recipe library* (see Figure 7 below).

[7] http://festvox.org

These changes not only make it possible to easily switch between multiple tutoring styles, but by modularizing the components, they increase SCoT's portability, i.e., make it easier to move SCoT to new domains.

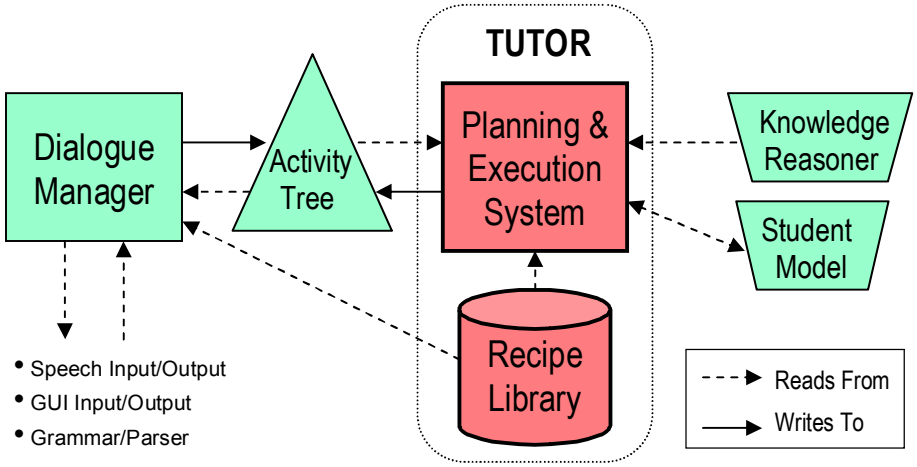

Figure 7. Subset of SCoT Architecture

The *planning and execution system* is responsible for selecting initial dialogue plans, revising plans during the dialogue, classifying student utterances, and deciding how to respond to the student. All of these tasks rely on external knowledge sources such as the knowledge reasoner, the activity tree, and the dialogue move tree (collectively referred to as the Information State). The planning and execution system "executes" tutorial activities by placing them on the activity tree, where they get interpreted and executed by the dialogue manager. By separating tutorial knowledge from external knowledge sources, this architecture allows SCoT to lead a flexible dialogue and to continually re-assess information from the Information State in order to select the most appropriate tutorial tactic.

The *recipe library* contains activity recipes that specify how to decompose a tutorial activity into other activities and low-level actions. An activity recipe can be thought of as a tutorial goal and a plan for how the tutor will achieve the goal. The recipe library contains a large number of activity recipes for both low-level tactics (e.g., responding to an incorrect answer) and high-level strategies (e.g., specifications for initial dialogue plans). The recipes

are written in a scripted language (Gruenstein, 2002) allowing for automatic translation of the recipes into system activities.

By explicitly encoding activities in a recipe library separate from the tutorial planning and execution system, we not only make it easier to integrate new tutorial activities, but we also move closer to our ideal activity model. An activity model is a formalized description of how a joint activity (in this case, tutoring) breaks down into sequences of actions and sub-actions of the participants in the activity. Furthermore, it models the activities at the level of granularity necessary for the system to relate what it is doing to natural language descriptions of what it is doing (Gruenstein, 2002).

In the previous version of SCoT, the activity model and the planning and execution system were conflated. Rather than operating over a set of activity recipes, the planning and execution system *implemented* the activity recipes. With the abstraction of the activity model in the new tutor component, the planning and execution system encapsulates domain-independent tutorial knowledge such as how to use information from the information state in generating initial plans, revising current plans, and responding to student answers.

The activities are domain-independent as well (e.g., acknowledge correct answer, elicit action), although people interested in applying SCoT to a new domain would most likely want to augment the library with new activity recipes depending on the kind of dialogue they wish to lead and the kind of multi-modal interaction they wish to support.

An activity recipe corresponding to the tutorial goal ***discuss_problem_solving_sequence*** is shown below in Figure 8. A recipe contains three primary sections: *DefinableSlots*, *MonitorSlots*, and *Body*. The *DefinableSlots* specify what information is passed in to the recipe, the *MonitorSlots* specify which parts of the Information State are used in determining how to execute the recipe, and *Body* specifies how to decompose the activity into other activities or low-level actions. The recipe below decomposes the activity of discussing a problem solving sequence into either three or four other activities (depending on whether the problem has already been discussed). The planning and execution system places these activities on the activity tree, and the dialogue manager begins to execute their recipes.

```
Activity <discuss_problem_solving_sequence> {

    DefinableSlots {
        currentProblem;
    }
    MonitorSlots {
        currentProblem.alreadyDiscussed;
    }
    Body {
        if (!currentProblem.alreadyDiscussed) {
            situate_problem_context;
        }
        state_review_purpose;
        state_correct_steps;
        elicit_missing_steps;
    }
}
```

Figure 8. Example of an Activity Recipe

All activity recipes have this same structure. The modular nature of the recipes helps us test our hypotheses by making it easy to alter the behavior of the tutor. For example, two tutoring styles that differ only in whether the tutor gives generic hints versus hints that refer back to past dialogue can be realized by the same set of activity recipes except that when responding to incorrect answers, one places the ***give_generic_hint*** activity on the activity tree, whereas the other places the ***give_referring_back_hint*** activity on the activity tree. An activity tree from an actual SCoT dialogue can be found in Appendix A.

5 Evaluation

An experiment was conducted using the new version of SCoT in order to evaluate the following hypothesis (discussed in Chapter 3):

- Tutors who respond to student uncertainty with particular linguistic devices will be more effective than those who do not. The specific linguistic devices tested were:
    i. paraphrasing the student's answer, if answer was correct.
    ii. referring back to previous dialogue, if answer was incorrect.

These two patterns of responding (paraphrasing and referring back) were collapsed into one tutorial "treatment" in SCoT (which I call "T1") and compared to a control treatment where the tutor responds to correct answers with simple acknowledgements and to incorrect answers with generic hints. The T1 treatment differed from the control only in the language used. Two pairs of SCoT dialogues are shown below. Paraphrases are tagged with "<P>" and referring-back statements are tagged with "<RB>". In both examples, the (a) and (b) dialogues are minimal pairs differing only in the lines italicized.

(1a) Paraphrase (T1)

Tutor: Assuming you have a report of fire, there are three other things you should have done before ordering firefighting. What is one of them?

Student: isolate compartment

Tutor: Yes, that's right.

Tutor: <P> You should send repair five to electrically and mechanically isolate the compartment.

(1b) No paraphrase (control)

Tutor: Assuming you have a report of fire, there are three other things you should have done before ordering firefighting. What is one of them?

Student: isolate compartment

Tutor: Yes, that's right.

(2a) "Referring back" hint (T1)

Tutor: Ok, which repair team has jurisdiction over this compartment?

Student: repair three

Tutor: No, That is incorrect.

Tutor: <RB> When we went over the regions of the ship earlier, which region did we say this was? [highlight]

Student: fore

Tutor: Yes, and which repair team has jurisdiction over the fore region?

(2b) Default hint (control)

Tutor: Ok, which repair team has jurisdiction over this compartment?

Student: repair three

Tutor: No, That is incorrect.

Tutor: Which region is this? [highlight]

Student: fore

Tutor: Yes, and which repair team has jurisdiction over the fore region?

The signals of uncertainty that SCoT detected and made use of are listed below.

- Lexical hedges ("I think", "I thought", "I guess", "maybe")
- Filled-pauses ("uh", "um")
- Response latencies (time between tutor's question and student's response)

Other signals of interest (e.g., mid-sentence pauses, rising intonation) were not included in this study due to time constraints on system development.

5.1 Method

5.1.1 Participants

Forty native English speakers from the Stanford community were recruited to participate in this experiment (17 female, 23 male). All subjects were novices in the domain of damage control, thirty-six had no prior experience using spoken dialogue systems.

5.1.2 Experiment Design

Subjects were randomly assigned to one of four groups (10 per group) and each group received a different style of tutoring, summarized below in Table 5. "T1" refers to the tutorial treatment of paraphrasing correct answers and referring back to past dialogue after incorrect answers (regardless of uncertainty). "Control" refers to the control treatment responding to correct answers with simple acknowledgements and to incorrect answers with generic hints.

Group	Treatment for Knowledge Area A (Sequencing)	Treatment for Knowledge Area B (Drilling)
I	T1	control
II	control	T1
III	T1 if uncertain; otherwise control	control
IV	control	T1 if uncertain; otherwise control

Table 5. Four Experimental Conditions

In order to counter-balance for subject differences, the damage control knowledge that SCoT tutors on was divided into two areas (sequencing and drilling) and all subjects received the T1-style tutoring in one knowledge area and the control tutoring in the other.

Sequencing refers to issuing orders for actions in response to crises (e.g., fires, floods) at the correct times. Drilling consists of two sub-areas: boundaries and jurisdiction. Setting boundaries refers to the task of correctly specifying six parameters that determine the location of the bulkheads (upright partitions separating ship compartments) that need to be cooled or sealed to prevent a crisis from spreading. Jurisdiction refers to the task of giving orders to the appropriate personnel on the ship—personnel are assigned to different regions such as forward, aft, and midship.

The experiment was run in two rounds. Round 1 consisted of subject groups I and II, and Round 2 consisted of subject groups III and IV. The contingency "T1 if uncertain, otherwise control" employed in groups III and IV corresponds directly to the hypotheses discussed in Chapter 3. However, Round 1 (groups I and II) was run beforehand in order to determine whether the T1 responses (paraphrasing and referring back), when employed regardless of the student's indications of uncertainty, had any effect on learning. Also, the median latencies for each of three question-types from Round 1 (20 subjects) were used as the thresholds for classifying latencies in Round 2. In this chapter, I will refer to the treatments in Round 1 as "non-contingent T1" and the treatments in Round 2 as "contingent T1".

Each subject ran through the same three DC-Train simulator scenarios interspersed with two SCoT dialogues (where they were debriefed on the preceding DC-Train session). In each SCoT dialogue, both knowledge areas were discussed—so each dialogue contained both T1-style responses and control responses, but associated with different knowledge areas. See Appendix B for transcripts. This between-subjects design allows us to see how the four conditions affect learning gains in each knowledge area.

5.1.3 Procedure

The experimental procedure is illustrated below in Table 6. Steps 4 through 8 (shown in boldface) constitute the main body of the experiment. In addition to these main steps, all subjects went through an interactive multimedia introduction to (1) familiarize them with DC-Train and basic damage control knowledge, and (2) give them practice using the speech recognition interface. After the multimedia introduction, subjects took a 22 question multiple-choice pre-test, and had one practice DC-Train session. Following the main body of the experiment, subjects took a 22 question post-test and filled out a questionnaire.

Step 1	Multimedia Introduction	30 min
Step 2	Pre-test	5-10 min
Step 3	Practice DC-Train session	10 min
Step 4	**DC-Train session 1**	15 min
Step 5	**SCoT Tutoring**	20-25 min
Step 6	**DC-Train session 2**	15 min
Step 7	**SCoT Tutoring**	20-25 min
Step 8	**DC-Train session 3**	15 min
Step 9	Post-test	5-10 min
Step 10	Questionnaire	5 min

Table 6. Experiment Procedure

The total duration of the experiment ranged between two and a half and three hours. All subjects ran through the experiment in one sitting[8] with a 5 minute break in the middle.

5.1.4 Measuring Learning Gains

Learning was measured in two ways. General knowledge was tested in the 22 question multiple-choice pre-test and a post-test of the same format (11 questions in each knowledge area). In addition, quantitative performance measures were drawn from each of the three DC-Train scenarios. For each of the knowledge areas, both raw scores and percentages were calculated.

Problem solving in the damage control domain is different from traditional tutoring domains (e.g., algebra) because the problem state is dynamically changing and there is not one unique solution path per scenario. The DC-Train sessions consist primarily of the user issuing commands and receiving reports. While we do control for time on task (scenarios end after 15 minutes), there is no way to control how many commands a user issues or how many "expert" actions will be suggested. For this reason, a variety of performance measures (both raw scores and percentages) were calculated.

[8] Two subject's sessions were interrupted roughly 1 hour after starting when the building had to be evacuated. They both picked up where they had left off 25 minutes later. Their performance was not significantly different than the other subjects in their group.

5.2 Results

The results did not show overwhelming evidence in support of my hypotheses, but there were significant differences between the non-contingent T1 and control tutoring styles. Interestingly, results from the written test differed from the simulator performance results, and for a few performance measures, the data did in fact match our predictions based on the hypotheses from Chapter 3.

Learning gains between the pretests and posttests are summarized below in Table 7. Raw gains are simply the posttest scores minus the pretest scores (as percentages), and normalized gains are: [(post-test – pre-test) / (1 – pre-test)]. The normalized means are also shown graphically in Figure 8.

Group	**Raw Gain Sequencing (Stdev)**	**Raw Gain Drilling (Stdev)**	**Normalized Gain Sequencing (Stdev)**	**Normalized Gain Drilling (Stdev)**
I	0.191 (0.100)	0.246 (0.201)	0.493 (0.233)	0.796 (0.351)
II	0.082 (0.109)	0.255 (0.147)	0.227 (0.309)	0.872 (0.182)
III	0.082 (0.125)	0.209 (0.187)	0.153 (0.412)	0.819 (0.349)
IV	0.118 (0.149)	0.255 (0.159)	0.330 (0.365)	0.827 (0.189)

Table 7. Learning gains between pre and post tests

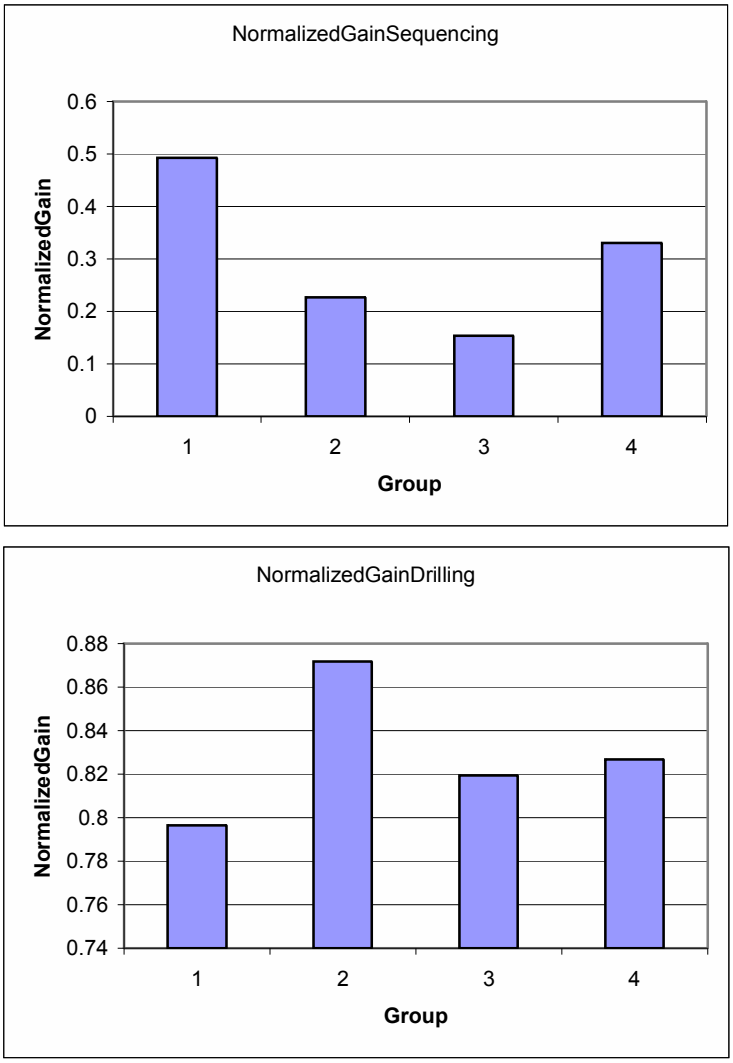

Figure 8. Normalized learning gains for sequencing (top) and drilling (bottom)

A one-way ANOVA on the normalized test score gains between the four groups suggested the differences were not statistically significant (sequencing: $p = .143$, drilling: $p = .945$). However, there was a significant difference between groups I and II. An independent samples T-test between groups I and II on the normalized sequencing gains yielded a significance of 0.042, however a T-test between the same two groups on the normalized drilling gains yielded a significance of only 0.559.

Performance measures from the simulator were examined in the areas of sequencing and drilling. The scores reported for drilling are actually a combination of scores for

boundaries and for jurisdiction—the two sub-areas that compose the drilling portion of the tutoring. The results for sequencing showed little difference between all four groups, while the results for drilling showed significant differences between groups I and II, and non-significant differences between groups III and IV.

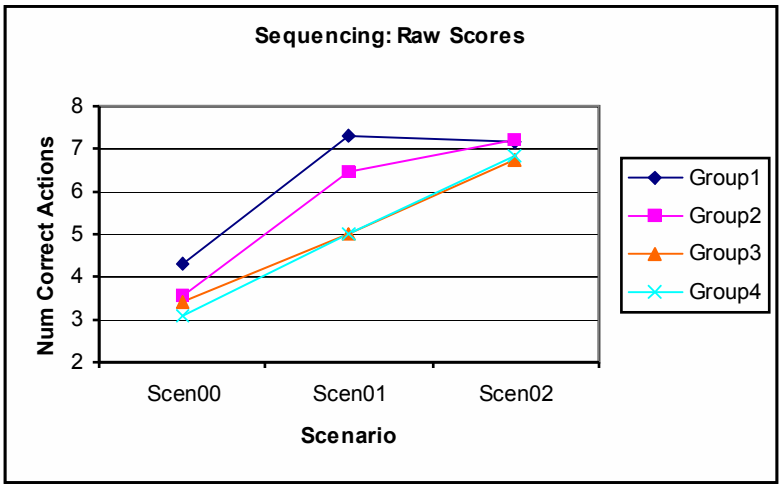

Figure 9. Raw Scores for Sequencing

Figure 9 above show the raw sequencing scores for each group across the three DC-Train scenarios. Every action that a student performs (i.e., every command that s/he issues) is graded as either on-time, early, late, or extra. The y-axis in Figure 9 represents the raw number of on-time (i.e., correct) actions. The fact that all four groups ended up with approximately the same score in the final scenario suggests that there might be a ceiling effect for sequencing, and that the differences between groups at Scenario 01 would be more telling than those at Scenario 02. A one-way ANOVA on the Scenario 01 raw scores yielded a significance of 0.157. Figure 10 below shows the gains in raw scores for each group between Scenarios 00 and 01.

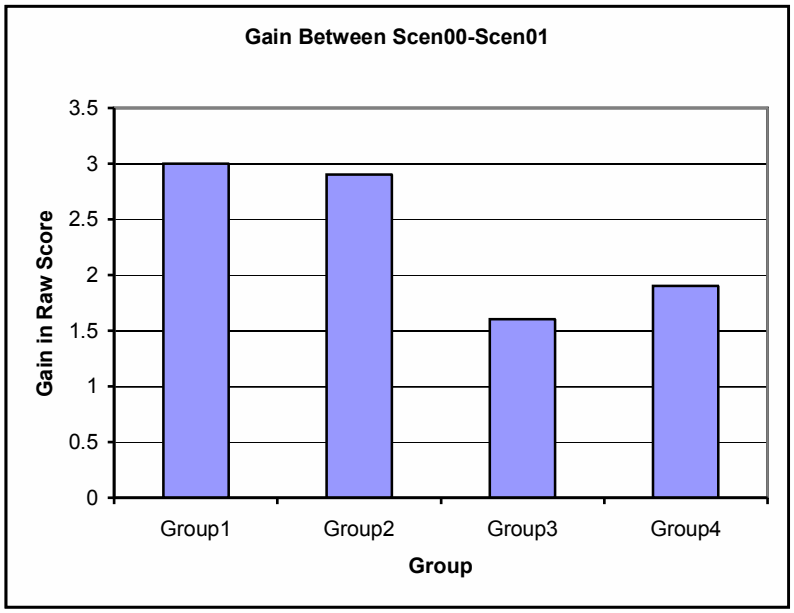

Figure 10. Sequencing performance gains between Scenario 00 and Scenario 01

In the area of sequencing, group I received non-contingent T1 and group III received contingent T1, whereas groups II and IV received the control. So if T1 really is more effective than the control, we would expect group I to have larger gains than group II, and group III to have larger gains than group IV. If the hypotheses from Chapter 3 are correct, we would expect group III to have larger gains than group I. Unfortunately, the sequencing raw scores did support these expectations. It is interesting to note though that groups I and II showed greater initial learning gains than groups III and IV.

Because there is no way to control how many actions a student performs in one scenario, another measure of interest is the correct actions as a percent of the total recommended actions (i.e., expert actions). Figure 11 below shows the percent recommended actions achieved for each group across the three DC-Train scenarios. In this case, all four groups behaved similarly. A one-way ANOVA at each scenario showed none of the differences to be significant. The plateau between Scenario 01 and Scenario 02 supports the hypothesis that there is a ceiling effect for sequencing performance.

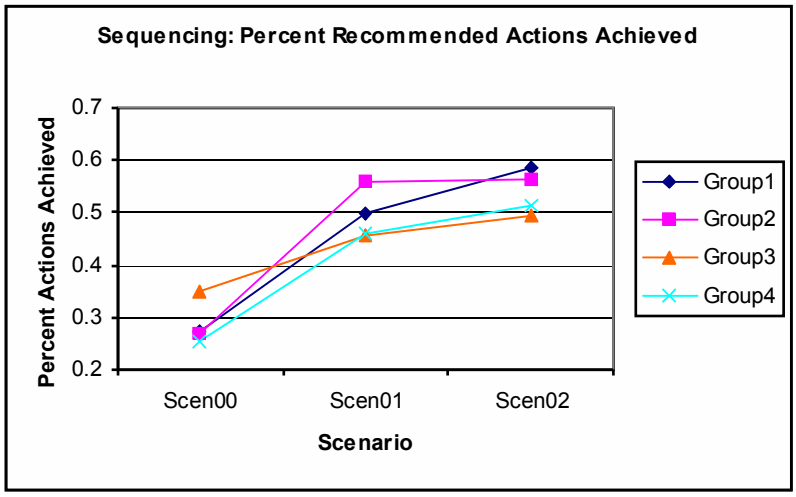

Figure 11. Percent correct for Sequencing

The performance results for drilling showed more variation. Figure 12 below shows the raw scores for drilling. The drilling raw score is a composite of the raw scores for boundaries and for jurisdiction. The raw score for boundaries represents the total number of boundary commands issued with all 6 parameters (4 bulkheads + 2 decks) correct. The raw score for jurisdiction represents the total number of commands issued with the correct repair team specified.

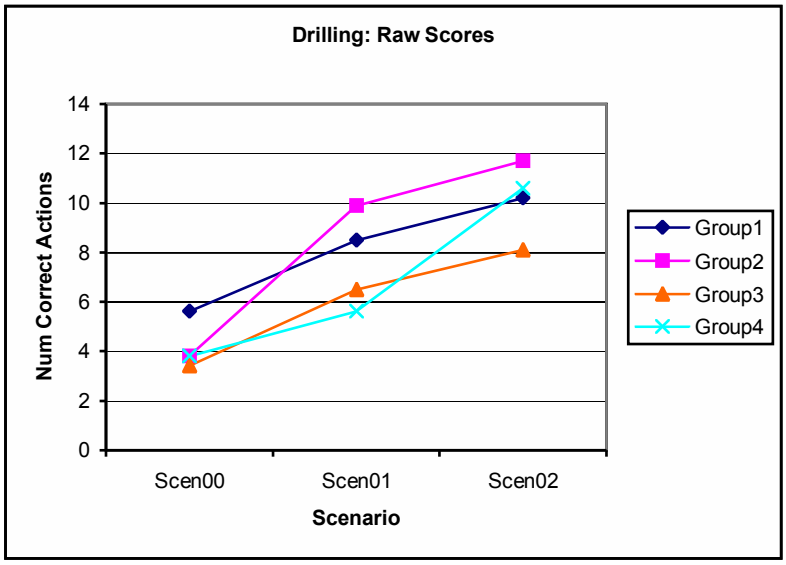

Figure 12. Raw scores for drilling

Figure 12 shows an interesting pattern: groups 1 and 3 have very similar learning gains (slopes), whereas group 2 has a steep learning gain between Scenarios 00 and 01 and group 4 has a similarly steep gain between Scenarios 01 and 02. The total gains in raw score (Scenario02 score – Scenario00 score) are shown below in Figure 13.

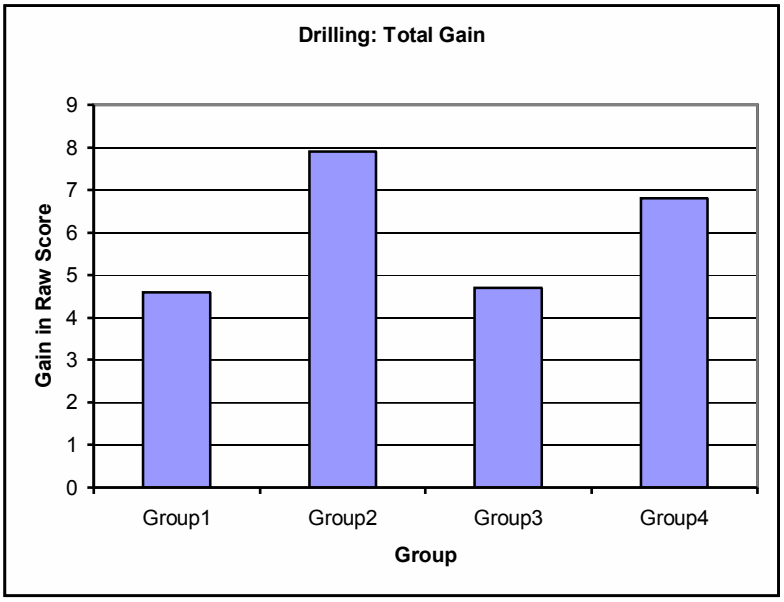

Figure 13. Total gains in raw score

In the area of drilling, group II received non-contingent T1 and group IV received contingent T1, whereas groups I and III received the control. So if T1 really is more effective than the control, we would expect group II to have larger gains than group I, and group IV to have larger gains than group III. If the hypotheses from Chapter 3 are correct, we would expect group IV to have larger gains than group II. Figure 13 shows that group II did indeed have larger gains than group I and that group IV did indeed have larger gains than group III. Independent samples T-tests showed the difference between groups I and II to be significant ($p = 0.022$), but the difference between the groups III and IV to be non-significant ($p = 0.405$).

As with sequencing, it is important to look at not only the raw values, but also the percentages correct. Figure 14 below shows percentages representing the number of correct commands (the raw score) out of the number of commands attempted. This helps

differentiate subjects who got 4 out of 4 attempts correct from subjects who got 4 out of 6 attempts correct, for example.

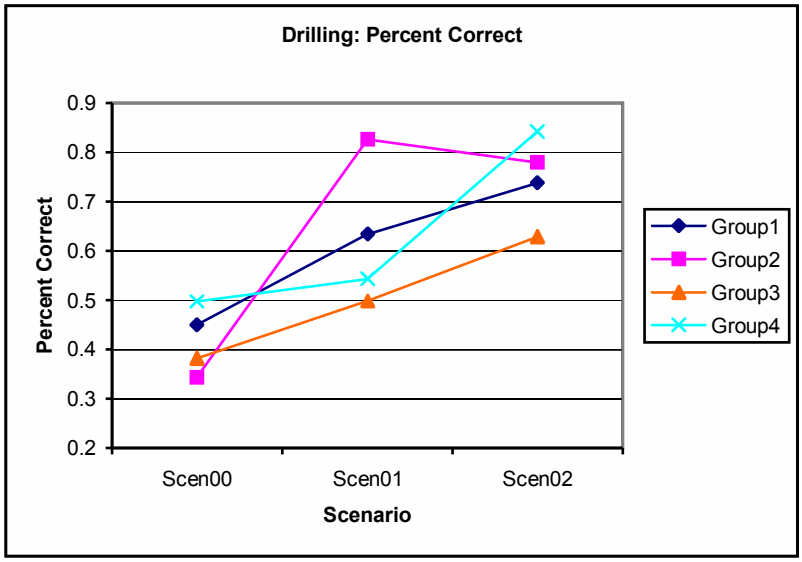

Figure 14. Percent correct for drilling

The learning gains in Figure 14 follow the same pattern as the ones in Figure 12. Groups I and III show improvement at approximately the same rate, whereas groups II and IV (the groups that received T1 in drilling) both show sharp gains, but in between Scenarios 00 and 01 for group II and in between Scenarios 01 and 02 for group IV. A one-way ANOVA was run between groups at each scenario, and the only one to show significant differences was Scenario 01 ($p = 0.019$).

The total gains in percent correct are shown below in Figure 15. As with the raw scores, group II showed greater gains than group I and group IV showed greater gains than group III. However, neither of these differences was significant.

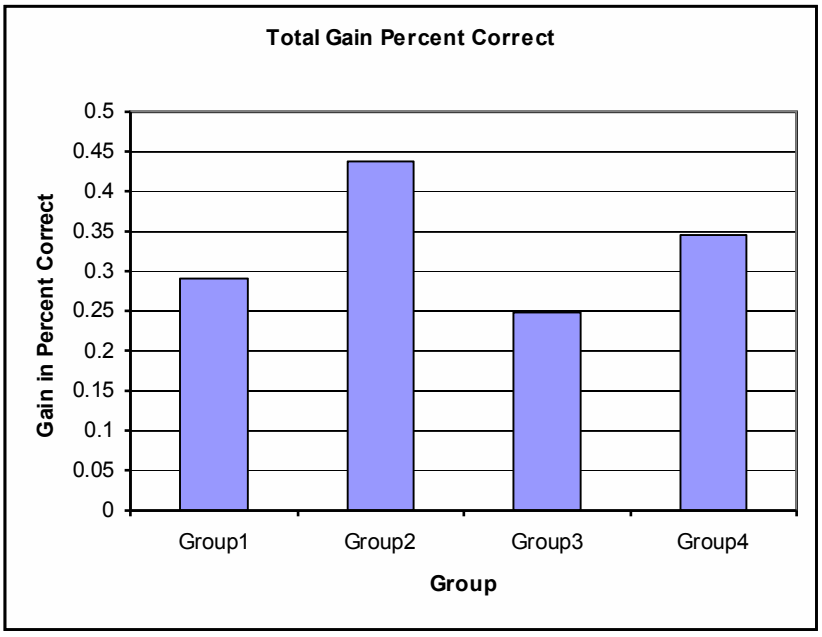

Figure 15. Total gains in percent correct

One area in particular showed learning gains that matched our expectations. In addition to raw scores and percentage scores, *weighted* scores were also calculated. A weighted score is simply the raw score multiplied by the percentage correct—it gives the most credit to students who had many correct actions *and* few mistakes. For the most part, the weighted scores looked very similar to the raw scores, so they were omitted from the previous discussion. Likewise, the individual scores for boundaries and jurisdiction were very similar, so only the composite was reported. However, the weighted boundary performance scores are worth discussion. Figure 16 below shows the weighted scores for the four subject groups in each scenario.

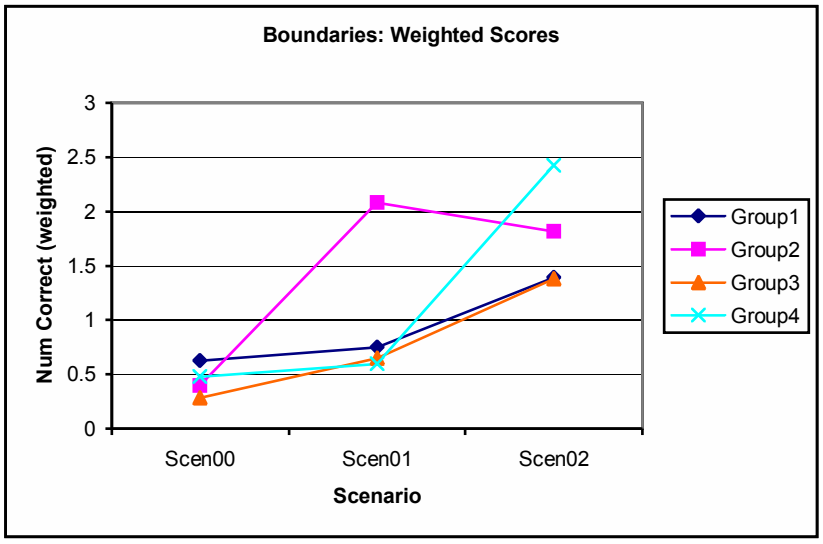

Figure 16. Weighted boundary performance scores

Figure 16 is similar to Figures 12 and 14 in that there is a sharp learning gain for group II between Scenarios 00 and 01 and one for group IV between Scenarios 01 and 02, but notably, the final score for group IV is well above the other three, and the final score for group II is also higher than groups I and III. An independent samples T-test between groups II and IV at Scenario01 yielded a significance of 0.025, however a T-test between the same groups at Scenario02 yielded a significance of only 0.452.

Figure 17 below shows the total gains (Scenario02 score – Scenario01 score) in weighted boundary scores for each group. In Figure 17, the relative gains between the four groups match the relative gains predicted by the hypotheses—groups II and IV received T1 style tutoring in drilling, so they are predicted to show greater gains than groups I and III. Also, group IV receives T1 style tutoring only when showing signs of uncertainty whereas group II receives T1 across the board, so group IV is predicted to have greater gains than group II. This much is true about the data in Figure 17, however independent samples T-tests suggest that none of these differences are significant.

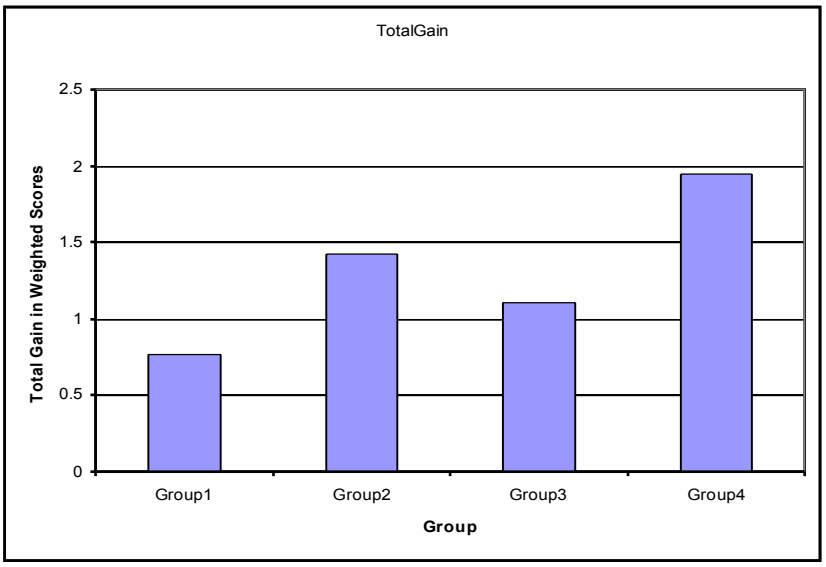

Figure 17. Gains in weighted boundary performance scores

One other result of interest is how often uncertainty was detected for groups III and IV—in other words, how often how often T1 actually "kicked-in". If uncertainty ware never detected, for example, then T1 would never kick in and we would expect groups III and IV to look the same. Or, if uncertainty ware always detected, then we would expect group III to look like group I and group IV like group II.

Group	**SCoT Dialogue 1: T1 used/T1 opportunities**	**SCoT Dialogue 2: T1 used/T1 opportunities**
III	0.623	0.475
IV	0.605	.0486

Table 8. Percent of times T1 "kicked-in" for groups III and IV

Both groups showed less uncertainty in their second tutoring session than in their first, which suggests that they were improving over time. The fact that for both sessions, the two groups showed roughly the same amount of uncertainty suggests that the knowledge area of

the question asked did not affect how often T1 "kicked-in" (for group III T1 opportunities are limited to sequencing questions; for group IV they are limited to drilling questions).

The number of times that each "uncertainty cue" was detected (out of 1600 total student responses from Round 2) is summarized below in Table 9. Unfortunately, hedges and filled-pauses were scarce to non-existent in the data. See Section 5.3 for a discussion of what this may mean.

Uncertainty Cue	**Number of times cue detected in Round 2 (Groups III and IV)**
Hedge	0
Filled-pause	20
Latency > threshold	893

Table 9. Number of occurrences of uncertainty cues

To summarize, the test results showed group I to have significantly larger learning gains than group II in sequencing, and group II to have (non-significantly) larger learning gains than group I in drilling. The test results for groups III and IV were not significantly different in either knowledge area. The performance results for sequencing showed an interesting ceiling effect, but no significant differences between groups. The performance results for drilling showed group II to have significantly larger learning gains than group I, and group IV to have (non-significantly) larger learning gains than group III. Overall, the differences between groups I and II were more pronounced than the differences between groups III and IV.

5.3 Discussion

Although the differences between groups III and IV were not significant (i.e., did not validate my primary hypothesis), the fact that there were significant differences between groups I and II supports the general hypothesis that even subtle variations in language (paraphrasing and referring back) can affect learning gain.

The fact that the test scores showed different patterns than the performance scores is not surprising. The written test tests a student's understanding of concepts without any time pressure, and because the questions were multiple choice a student at chance will get 25% correct. The simulator tests how well students can turn their knowledge into actions in a fast-paced time-pressured environment. Because the space of possible actions is so large and the grading of actions depends on a dynamically changing state, a student at chance would get far less than 25% of actions correct. Furthermore, in the area of sequencing, a student must be keeping track not only of the commands he or she issues, but also of incoming reports (about multiple crises) in order to order an action on-time. So, it seems that while the T1-style tutoring in sequencing gave group I a better understanding of the conceptual knowledge, it was not enough to affect their performance in the simulator. This finding is relevant for developers of ITSs in general—where learning gains are often measured with written tests alone. This may be fine for domains where the goal of the tutoring is to improve test scores (e.g., in the classroom), but if the goal is to give students a deeper understanding and the ability to apply their knowledge in practice, then it is important to look at other measures of learning as well.

Because there were significant differences in performance gains between groups for drilling but not for sequencing, it seems safe to assume that the extent to which subtle tutoring style differences affect learning depends on the knowledge area. In the area of drilling, group II had greater learning gains than group I and a steep gain in performance between Scenarios 00 and 01, whereas group IV had greater learning gains than group III and a steep gain in performance between Scenarios 01 and 02. In the drilling portion of the dialogue, group II received T1 100% of the time, and group IV received T1 61% of the time (on average) in the first tutoring dialogue, and 49% of the time (on average) in the second. This suggests that there might be a point where students suddenly understand the concepts tutored in the drilling area, and that the more T1 style responses they receive, the faster they reach this point of sudden clarity.

Another interesting result from the performance data is that the gains in weighted boundary scores matched our predictions better than any other measures. This may be due

to the nature of the task—issuing a correct boundary command involves specifying 6 parameters in the format "primary forward 300, primary aft 330, secondary forward 254, secondary aft 338, above 1, below 2". Although all subjects practiced issuing boundary commands in the multimedia introduction, it was by far the most difficult task early on. However, by the end of the experiment, most subjects had a fairly good grasp of it. Like issuing commands in the proper sequence, issuing boundary commands is difficult to get correct purely by chance. However, unlike sequencing, it does not depend on the dynamically changing state of the ship, so the tutoring SCoT gave was sufficient to affect performance, and to show differences between conditions. This suggests that the hypotheses from Chapter 3 may indeed be correct, but that the effectiveness of T1 versus control is affected by the knowledge area.

A possible explanation for the non-significant differences found between groups III and IV is that the times T1 "kicked in" may not have been representative of the student's actual uncertainty. Table 9 shows that the vast majority of contingent uses of T1 in Round 2 were because the student's latency in responding was greater than the threshold. Response latency is not a cue that I looked into (or had access to) in the empirical work described in Chapter 3. It was added as a substitute for mid-sentence pauses due to development delays that precluded detecting mid-sentence pauses in real-time. Also, grammar development constraints limited the types of phrases containing a hedge or a filled-pause that could be understood. Most subjects began their sessions with SCoT speaking verbosely, but after realizing that many long or complicated phrases could not be understood, switched to giving terse and less natural answers (see transcript in Appendix B). This is often the case in human-computer conversations, and the grammar coverage issue is one of the trade-offs that affects any system using deep semantic parsing. One interesting point, though, is that prior to this set of experiments, SCoT used a push-to-talk style of interaction. For these experiments, we switched to an open-mic style of interaction (the system is continuously listening) in hopes that it would lead to more hedges, filled-pauses, and other features that are common in human-human conversation. Subjects were in fact much more "chatty" with this version of SCoT than they were with the previous version, but as previously mentioned,

this talkativeness diminished as the dialogue progressed. This suggests that with better coverage of natural language phrasings and the ability to detect features such as mid-sentence pauses, a future study like this one might show different or more significant results.

Finally, one factor that may have affected test scores and performances scores is student fatigue. The entire experiment demands a lot of concentration from the student, and by the end of 2.5 to 3 hours, many students were tired or mentally worn out (see questionnaire results about "effort required" in Appendix C). It is likely that this mental fatigue may have affected their performance in the final simulator session and/or on the post-test. On average, subjects completed the post-test in half the time it took them to complete the pre-test. Obviously, they understood the material better, but it is possible that they were not putting as much into the questions as they had in the pre-test.

Another factor which may account for differences between knowledge areas is the number of opportunities for T1 to apply in the sequencing portion of the dialogue versus the drilling portion of the dialogue. The sequencing portion of the dialogue steps through the scenario, and elicits from the student actions that they should have done but failed to do. Thus, the number of actions elicited depends on the student's performance, whereas in the drilling portion of the dialogue, there is a fixed number of main (as opposed to follow-up) questions asked. This difference does not affect the validity of the results (sequencing measures were compared to sequencing measures between groups, and same for drilling), but it may help explain why T1 caused greater differences in the drilling performance measures than in the sequencing performance measures.

To sum up, the results showed that subtle differences in tutoring style can affect how well material is learned and how well learned material can be applied in practice. The variation in the results may be explained by differences between the content of the two knowledge areas and by the distribution of cues of uncertainty actually detected by the system.

6 Conclusions

This research project was undertaken in an attempt to understand which parts of natural language interaction are most responsible for the effectiveness of tutorial dialogue. The first step taken was to look for patterns in transcripts of one-on-one human tutoring. My observations led me to the following hypothesis:

- Tutors who respond to student uncertainty with particular linguistic devices (paraphrasing and referring to past dialogue) will be more effective than those who do not.

With the goal of allowing SCoT to flexibly switch between strategies and test out these hypotheses, I redesigned and reimplemented the framework of SCoT's tutor component. Finally, I conducted an evaluation using SCoT to compare two subtly different styles of tutoring: in one style, the tutor paraphrased and referred back at every possible opportunity (non-contingent treatment), in the other, the tutor paraphrased and referred back only when a signal of uncertainty was detected (contingent treatment). Both of these styles of tutoring were compared to a control style where the tutor gave simple acknowledgements in place of paraphrases and generic hints in place of referring back hints.

The results showed statistically significant differences in learning gain between the non-contingent tutoring and the control, and non-significant differences in learning gain between the contingent tutoring and the control. This affirms the general hypothesis that subtle differences in language can affect the outcome of the tutoring. Furthermore, the fact that paraphrasing and referring back are general linguistic devices and not specific to tutorial dialogue suggests that the effectiveness of human tutoring may be due to general characteristics of conversation in addition to the specific tutoring techniques that have been identified in the literature (see Section 2.1.2).

One other important lesson learned is that the signals of uncertainty present in human-to-human spoken dialogue may not occur with the same frequency in human-to-computer spoken dialogue. Because most people talk to computers differently than they talk to other humans, the best way to choose appropriate signals of uncertainty would be based on an analysis of comparable human-to-computer dialogues. Our experience using open-

mic interaction in SCoT (compared to a push-to-talk interface in the previous evaluation) suggests that the interface itself can encourage (or discourage) more natural speech.

Although the experiment results did not validate my specific hypotheses about paraphrasing and referring back in response to student uncertainty, they did validate the general hypothesis that linguistic devices in tutorial dialogue such as paraphrasing and referring back *do* affect learning. Given the patterns of student language observed in this study, we are now in a better position to generate a list of signals of uncertainty applicable to human-computer tutorial interaction (e.g., by hand-annotation). With such a list, a future study could be conducted in order to determine whether tutoring is more effective when these sorts of linguistics devices are employed all of the time versus when they are employed only after signs of uncertainty.

References

Aleven V., Koedinger, K. R., & Popescu, O. (2003). A Tutorial Dialog System to Support Self-Explanation: Evaluation and Open Questions. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), *Proceedings of the 11th International Conference on Artificial Intelligence in Education, AI-ED 2003*, 39-46.

Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring systems. *Science, 228*, 456-467.

Anderson, J. R., Boyle, C. F., Corbett, A., and Lewis, M. W. (1990). Cognitive modelling and intelligent tutoring. *Artificial Intelligence*, *42*, 7-49.

Belvin, R., Burns, R., & Hein, C. (2001). Development of the HRL Route Navigation Dialogue System. In *Proceedings of the First International Conference on Human Language Technology Research,* Paper H01-1016.

Bhatt, K. (2004). Classifying student hedges and affect in human tutoring sessions for the CIRCSIM-Tutor intelligent tutoring system. Unpublished M.S. Thesis, Illinois Institute of Technology.

Bhatt, K., Argamon S., and Evens, M. (2004). Hedged responses and expressions of affect in human/human and human/computer tutorial interactions. In *Proceedings of COGSCI 2004*, Chicago, IL, pp. 114-119.

Bloom, B. (1984). The 2 sigma problem: The search for methods of group instruction as effective one-on-one tutoring. *Educational Researcher, 13*, 4-16.

Brennan, S. E., & Williams, M. (1995). The feeling of another's knowing: Prosody and filled pauses as cues to listeners about the metacognitive states of speakers. *Journal of Memory and Language, 34*, 383-398.

Bulitko, V., & Wilkins., D. C. (1999). Automated instructor assistant for ship damage control. In *Proceedings of AAAI-99*.

Chi, M. T. H. (2000). Self-explaining expository texts: The dual processes of generating inferences and repairing mental models. In R. Glaser (Ed.), *Advances in Instructional Psychology*. Hillsdale, NJ: Lawrence Erlbaum Associates. 161-238.

Chi, M.T.H., de Leeuw, N., Chiu, M., LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science, 18*, 439-477.

Chi, M. T. H., Siler, S. A., Jeong, H., Yamauchi, T., & Hausmann, R. G. (2001). Learning from human tutoring. *Cognitive Science, 25*, 471-533.

Clark, B., Fry, J., Ginzton, M., Peters, S., Pon-Barry, H., & Thomsen-Gray, Z. (2001). A Multimodal Intelligent Tutoring System for Shipboard Damage Control. In *Proceedings of 2001 International Workshop on Information Presentation and Multimodal Dialogue (IPNMD-2001).* Verona, Italy. 121-125.

Clark, B., Lemon, O., Gruenstein, A., Bratt, E., Fry, J., Peters, S., Pon-Barry, H., Schultz, K., Thomsen-Gray, Z., & Treeratpituk, P. (In press). A General Purpose Architecture for Intelligent Tutoring Systems. In *Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems.* Edited by Niels Ole Bernsen, Laila Dybkjaer, and Jan van Kuppevelt. Dordrecht: Kluwer.

Clark, H. H. (1996). *Using Language*. Cambridge: University Press.

Cohen, P. A., Kulik, J. A., & Kulik, C. L. C. (1982). Educational outcomes of tutoring: a meta-analysis of findings. *American Educational Research Journal, 19,* 237–248.

Dowding, J., Gawron, M., Appelt, D., Cherny, L., Moore, R., and Moran, D. (1993). Gemini: A natural language system for spoken language understanding. In *Proceedings of ACL 31.*

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

Evens, M., & Michael, J. (Unpublished manuscript). *One-on-One Tutoring by Humans and Machines.* Computer Science Department, Illinois Institute of Technology.

Fox, B. (1993). *The Human Tutorial Dialogue Project: Issues in the Design of Instructional Systems.* New Jersey: Lawrence Earlbaum.

Graesser, A. C., Person, N. K., Harter, D., & TRG. (2001). Teaching tactics and dialog in AutoTutor. *International Journal of Artificial Intelligence in Education, 12,* 257-279.

Graesser, A. C., Person, N. K., & Magliano J. P. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring sessions. *Applied Cognitive Psychology, 9,* 1-28.

Gruenstein, A. (2002). Conversational Interfaces: A Domain-Independent Architecture for Task-Oriented Dialogues. Unpublished M.S. Thesis, Stanford University.

Hauptmann, A. G. & Rudnicky, A. I. (1988). Talking to Computers: An Empirical Investigation. *International Journal of Man-Machine Studies 28(6),* 583-604.

Hausmann, R., & Chi, M. T. H. (2002). Can a computer interface support self-explaining? *Cognitive Technology, 7(1),* 4-15.

Heffernan, N. T. (2001). Intelligent Tutoring Systems have Forgotten the Tutor: Adding a Cognitive Model of Human Tutors. Dissertation. Computer Science Department, School of Computer Science, Carnegie Mellon University. Technical Report CMU-CS-01-127.

Heffernan, N. T., & Koedinger, K. R. (2002). An Intelligent Tutoring System Incorporating a Model of an Experienced Human Tutor. In *Proceedings of the 6th International Conference on Intelligent Tutoring Systems, ITS 2002.* Biarritz, France.

Koedinger, K. R., Anderson, J.R., Hadley, W.H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education, 8,* 30-43.

Lemon, O., Bracy, A., Gruenstein, A., & Peters, S. (2002a). Collaborative dialogue for controlling autonomous systems. In *Proceedings of the AAAI Fall Symposium*, 2002.

Lemon, O., Gruenstein, A., & Peters, S. (2002b). Collaborative activities and multitasking in dialogue systems. In C. Gardent (Ed.), *Traitement Automatique des Langues (TAL, special issue on dialogue), 43(2),* 131-154.

Litman, D., & Forbes, K. (2003). Recognizing Emotions from Student Speech in Tutoring Dialogues. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU).*

Litman, D., & Forbes-Riley, K. (2004). Annotating student emotional states in spoken tutoring dialogues. *Proceedings of the Fifth Workshop on Discourse and Dialogue (SIGDIAL),* Cambridge, MA, pp. 144-151.

Litman, D., & Silliman, S. (2004). ITSPOKE: An Intelligent Tutoring Spoken Dialogue System. In *Proceedings of the Human Language Technology Conference: 4th Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*.

Merrill, D., Reiser, B., Ranney, M., & Trafton, J. G. (1992). Effective Tutoring Techniques: A Comparison of Human Tutors and Intelligent Tutoring Systems. *The Journal of the Learning Sciences, 2(3),* 277-305.

Michael, J. A., Rovick, A. A., Glass, M. S., Zhou, Y., and Evens, M. (2003). Learning from a computer tutor with natural language capabilities. *Interactive Learning Environments, 11(3),* 233-262. November 2003.

Moore J. D., Lemaire B., Rosenblum J. A. (1996). Discourse Generation for Instructional Applications: Identifying and Exploiting Relevant Prior Explanations. *The Journal of the Learning Sciences, 5(1),* 49-94.

Newell, A., & Simon, H. A. (1972). *Human Problem Solving.* Englewood Cliffs, NJ: Prentice-Hall.

Person, N.K., Graesser, A.C., Bautista, L., Mathews, E., & the Tutoring Research Group. (2001). Evaluating student learning gains in two versions of AutoTutor. In J. D. Moore, C. L. Redfield, & W. L. Johnson (Eds.), Proceedings of *Artificial intelligence in education: AI-ED in the wired and wireless future*, 286-293.

Pon-Barry, H., Clark, B., Schultz, K., Bratt, E., & Peters, S. (2004a). Advantages of Spoken Language Interaction in Tutorial Dialogue Systems. In *Proceedings of ITS 2004, 7th International Conference on Intelligent Tutoring Systems*. Maceió, Brazil.

Pon-Barry, H., Clark, B., Bratt, E., Schultz, K., & Peters, S. (2004b). Evaluating the Effectiveness of SCoT: a Spoken Conversational Tutor. In *Proceedings of ITS 2004 Workshop on Dialog-based Intelligent Tutoring Systems*. Maceió, Brazil.

Rosé, C. P., Jordan, P., Ringenberg, M., Siler, S., VanLehn, K., & Weinstein, A. (2001). Interactive Conceptual Tutoring in Atlas-Andes. In J. D. Moore, C. L. Redfield & W. L. Johnson (Eds.), Proceedings of *Artificial intelligence in education: AI-ED in the wired and wireless future*, 256-266.

Rosé, C. P., Moore, J. D., VanLehn, K., & Allbritton, D. (2001). A Comparative Evaluation of Socratic versus Didactic Tutoring, In *Proceedings of the 23rd Annual Meeting of the Cognitive Science Society.* Edinburgh, Scotland, UK.

Rosé, C. P., Litman, D., Bhembe, D., Forbes, K., Silliman, S., Srivastava, R., & VanLehn, K. (2003). A Comparison of Tutor and Student Behavior in Speech Versus Text Based Tutoring. In *Proceedings of the HLT-NAACL 03 Workshop on Educational Applications of NLP.*

Schultz, K., Bratt, E., Clark, B., Peters, S., Pon-Barry, H., & Treeratpituk, P. (2003). A Scalable, Reusable Spoken Conversational Tutor: SCoT. In *Proceedings of the AIED 2003 Workshop on Tutorial Dialogue Systems: With a View Towards the Classroom*.

Shelby, R., Schulze, K., Treacy, D., Wintersgill, M., VanLehn, K., & Weinstein A. (2001). An assessment of the Andes tutor. In *Proceedings of the Physics Education Research Conference*, July 21-25, Rochester, NY.

Smith, V. L., & Clark, H. H. (1993). On the course of answering questions. *Journal of Memory and Language, 32,* 25-38.

Walker, M., Rudnicky, A., Prasad, R., Aberdeen, V., Bratt, E., Garofolo, J., Hastie, H., Le, A., Pellom, B., Potamianos, A., Passonneau, R., Roukos, S., Sanders, G., Seneff, S., & Stallard,

D. (2002). DARPA Communicator: Cross-System Results for the 2001 Evaluation. In *Proceedings of the 7th International Conference on Spoken Language Processing, ICSLP 2002*.

Zinn, C., Moore, J., & Core, M. (2002). A 3-tier planning architecture for managing tutorial dialogue. In *Proceedings of the 6th International Conference, ITS 2002*, 574-584.

Appendix A: Sample Activity Tree

- hello (done) [DMTASK0:scot]
  - introduction (done) [SIM0:scot]
    - say_hello (done) [SIM1:scot]
    - get_username (done) [SIM2:scot]
  - select_session (done) [SIM3:scot]
    - preface_get_session (done) [SIM4:scot]
    - get_session (done) [SIM5:scot]
    - acknowledge_session (done) [SIM6:scot]
  - confirm_user_ready (done) [SIM7:scot]
    - okay (done) [DMTASK1:scot]
  - tutoring_from_session (done) [SIM8:scot]
    - summarize_session (done) [SIM9:scot]
      - qualitative_session_summary (done) [SIM10:scot]
      - quantitative_session_summary (done) [SIM11:scot]
        - state_action_performance (done) [SIM12:scot]
        - state_boundary_performance (done) [SIM13:scot]
        - state_jurisdiction_performance (done) [SIM14:scot]
      - qualitative_assessment (done) [SIM15:scot]
        - performance_summary (done) [SIM16:scot]
        - tutoring_session_state_purpose (done) [SIM17:scot]
    - tutoring (done) [SIM18:scot]
      - plan_session (done) [SIM19:scot]
        - discuss_one_problem (done) [SIM20:scot]
          - plan_problem_discussion (done) [SIM21:scot]
          - introduce_problem (done) [SIM22:scot]
            - highlight_location (done) [SIM23:scot]
            - state_problem_type_and_description (done) [SIM24:scot]
          - discuss_actions (done) [SIM25:scot]
            - discuss_action (done) [SIM26:scot]
              - plan_discuss_action (done) [SIM27:scot]
              - state_current_action (done) [SIM28:scot]
                - state_precontext (done) [SIM29:scot]
              - state_ontime_action (done) [SIM30:scot]
            - discuss_action (done) [SIM31:scot]
              - plan_discuss_action (done) [SIM32:scot]
              - state_current_action (done) [SIM33:scot]
                - state_precontext (done) [SIM34:scot]
              - state_ontime_action (done) [SIM35:scot]
            - discuss_action (done) [SIM36:scot]
              - plan_discuss_action (done) [SIM37:scot]
              - state_current_action (done) [SIM38:scot]
                - state_precontext (done) [SIM39:scot]
              - state_ontime_action (done) [SIM40:scot]

- discuss_action (done) [SIM41:scot]
  - plan_discuss_action (done) [SIM42:scot]
  - state_current_action (done) [SIM43:scot]
    - state_precontext (done) [SIM44:scot]
  - state_ontime_action (done) [SIM45:scot]
- discuss_action (done) [SIM46:scot]
  - plan_discuss_action (done) [SIM47:scot]
  - state_current_action (done) [SIM48:scot]
    - state_precontext (done) [SIM49:scot]
  - set_window_of_problem_steps (done) [SIM50:scot]
  - iterate_over_steps (done) [SIM51:scot]
- discuss_action (done) [SIM52:scot]
  - plan_discuss_action (done) [SIM53:scot]
  - state_current_action (done) [SIM54:scot]
    - state_precontext (done) [SIM55:scot]
  - set_window_of_problem_steps (done) [SIM56:scot]
  - iterate_over_steps (done) [SIM57:scot]
- plan_discuss_last_action (done) [SIM58:scot]
- discuss_last_action (done) [SIM59:scot]
  - state_last_action (done) [SIM60:scot]
  - iterate_over_steps (done) [SIM61:scot]
    - state_unperformed_necessary_actions_for_window (done) [SIM62:scot]
    - plan_discuss_unperformed_necessary_actions_in_step (done) [SIM63:scot]

- get_errors_in_step (done) [SIM64:scot]
    - discuss_unperformed_necessary_actions_in_step (done) [SIM65:scot]
- introduce_step (done) [SIM66:scot]
- elicit_unperformed_necessary_actions (done) [SIM67:scot]
  - plan_elicit_action (done) [SIM68:scot]
    - elicit_action (done) [SIM69:scot]
      - acknowledge_neutral (done) [DMTASK2:scot]
      - give_CI_hint (done) [DMTASK3:scot]
      - reask_question (done) [DMTASK4:scot]
        - acknowledge_correct_answer (done) [DMTASK5:scot]
  - decrement_num_actions_remaining (done) [SIM70:scot]
      - discuss_one_problem (done) [SIM71:scot]
        - plan_problem_discussion (done) [SIM72:scot]
        - introduce_problem (done) [SIM73:scot]
          - highlight_location (done) [SIM74:scot]
          - state_problem_type_and_description (done) [SIM75:scot]
        - discuss_actions (done) [SIM76:scot]
          - discuss_action (done) [SIM77:scot]
            - plan_discuss_action (done) [SIM78:scot]

- state_current_action (done) [SIM79:scot]
  - state_precontext (done) [SIM80:scot]
- state_ontime_action (done) [SIM81:scot]

- discuss_action (done) [SIM82:scot]
  - plan_discuss_action (done) [SIM83:scot]
  - state_current_action (done) [SIM84:scot]
    - state_precontext (done) [SIM85:scot]
  - set_window_of_problem_steps (done) [SIM86:scot]
  - iterate_over_steps (done) [SIM87:scot]
- discuss_action (done) [SIM88:scot]
  - plan_discuss_action (done) [SIM89:scot]
  - state_current_action (done) [SIM90:scot]
    - state_precontext (done) [SIM91:scot]
  - set_window_of_problem_steps (done) [SIM92:scot]
  - iterate_over_steps (done) [SIM93:scot]
    - state_unperformed_necessary_actions_for_window (done) [SIM94:scot]
    - plan_discuss_unperformed_necessary_actions_in_step (done) [SIM95:scot]

- get_errors_in_step (done) [SIM96:scot]
  - discuss_unperformed_necessary_actions_in_step (done) [SIM97:scot]
- introduce_step (done) [SIM98:scot]
- elicit_unperformed_necessary_actions (done) [SIM99:scot]
  - plan_elicit_action (done) [SIM100:scot]
    - elicit_action (done) [SIM101:scot]
      - acknowledge_correct_answer (done) [DMTASK6:scot]
  - decrement_num_actions_remaining (done) [SIM102:scot]
    - plan_discuss_last_action (done) [SIM103:scot]
    - discuss_last_action (done) [SIM104:scot]
      - state_last_action (done) [SIM105:scot]
      - iterate_over_steps (done) [SIM106:scot]
  - start_drilling (done) [SIM107:scot]
    - break_flow (done) [SIM108:scot]
    - pick_topic (done) [SIM109:scot]
    - drill_on_topic (done) [SIM110:scot]
      - pick_question (done) [SIM111:scot]
      - initialize_hints (done) [SIM112:scot]
      - set_stage (done) [SIM113:scot]
      - ask_question (done) [SIM114:scot]
      - keep_drilling_topic (done) [SIM115:scot]
      - pick_question (done) [SIM116:scot]
      - initialize_hints (done) [SIM117:scot]
      - set_stage (done) [SIM118:scot]
      - ask_question (done) [SIM119:scot]

- keep_drilling_topic (done) [SIM120:scot]
- keep_drilling_session (done) [SIM121:scot]
- pick_topic (done) [SIM122:scot]
- intro_utter6 (done) [SIM123:scot]
- drill_on_topic (done) [SIM124:scot]
  - pick_question (done) [SIM125:scot]
  - initialize_hints (done) [SIM126:scot]
  - set_stage (done) [SIM127:scot]
  - ask_question (done) [SIM128:scot]
    - acknowledge_positive (done) [DMTASK7:scot]
    - paraphrase_correct_answer (done) [DMTASK8:scot]
    - restate_question (done) [DMTASK9:scot]
      - acknowledge_positive (done) [DMTASK10:scot]
      - state_complete_answer (done) [DMTASK11:scot]
  - keep_drilling_topic (done) [SIM129:scot]
  - pick_question (done) [SIM130:scot]
  - initialize_hints (done) [SIM131:scot]
  - set_stage (done) [SIM132:scot]
  - ask_question (done) [SIM133:scot]
    - acknowledge_positive (done) [DMTASK12:scot]
    - restate_question (done) [DMTASK13:scot]
      - acknowledge_positive (done) [DMTASK14:scot]
      - state_complete_answer (done) [DMTASK15:scot]
  - keep_drilling_topic (done) [SIM134:scot]
- keep_drilling_session (done) [SIM135:scot]
- pick_topic (done) [SIM136:scot]
- drill_on_topic (done) [SIM137:scot]
  - pick_question (done) [SIM138:scot]
  - initialize_hints (done) [SIM139:scot]
  - set_stage (done) [SIM140:scot]
  - ask_question (done) [SIM141:scot]
    - acknowledge_positive (done) [DMTASK16:scot]
    - restate_question (done) [DMTASK17:scot]
      - acknowledge_positive (done) [DMTASK18:scot]
      - state_complete_answer (done) [DMTASK19:scot]
  - keep_drilling_topic (done) [SIM142:scot]
  - pick_question (done) [SIM143:scot]
  - initialize_hints (done) [SIM144:scot]
  - set_stage (done) [SIM145:scot]
  - ask_question (done) [SIM146:scot]
    - acknowledge_positive (done) [DMTASK20:scot]
    - restate_question (done) [DMTASK21:scot]
      - acknowledge_positive (done) [DMTASK22:scot]
      - state_complete_answer (done) [DMTASK23:scot]
  - keep_drilling_topic (done) [SIM147:scot]

- keep_drilling_session (done) [SIM148:scot]
- pick_topic (done) [SIM149:scot]
- intro_utter1 (done) [SIM150:scot]
- intro_utter2 (done) [SIM151:scot]
- intro_utter3 (done) [SIM152:scot]
- intro_utter4 (done) [SIM153:scot]
- drill_on_topic (done) [SIM154:scot]
  - pick_question (done) [SIM155:scot]
  - initialize_hints (done) [SIM156:scot]
  - set_stage (done) [SIM157:scot]
  - ask_question (done) [SIM158:scot]
    - acknowledge_positive (done) [DMTASK24:scot]
    - restate_question (done) [DMTASK25:scot]
      - acknowledge_positive (done) [DMTASK26:scot]
      - state_complete_answer (done) [DMTASK27:scot]
  - keep_drilling_topic (done) [SIM159:scot]
- keep_drilling_session (done) [SIM160:scot]
- pick_topic (done) [SIM161:scot]
- drill_on_topic (done) [SIM162:scot]
  - pick_question (done) [SIM163:scot]
  - initialize_hints (done) [SIM164:scot]
  - set_stage (done) [SIM165:scot]
  - ask_question (done) [SIM166:scot]
    - acknowledge_positive (done) [DMTASK28:scot]
    - restate_question (done) [DMTASK29:scot]
      - acknowledge_positive (done) [DMTASK30:scot]
      - state_complete_answer (done) [DMTASK31:scot]
  - keep_drilling_topic (done) [SIM167:scot]
- keep_drilling_session (done) [SIM168:scot]
- pick_topic (done) [SIM169:scot]
- drill_on_topic (done) [SIM170:scot]
  - pick_question (done) [SIM171:scot]
  - initialize_hints (done) [SIM172:scot]
  - set_stage (done) [SIM173:scot]
  - ask_question (done) [SIM174:scot]
    - acknowledge_positive (done) [DMTASK32:scot]
    - give_CI_drill_hint (done) [DMTASK33:scot]
    - restate_question (done) [DMTASK34:scot]
      - acknowledge_positive (done) [DMTASK35:scot]
      - state_complete_answer (done) [DMTASK36:scot]
  - keep_drilling_topic (done) [SIM175:scot]
  - pick_question (done) [SIM176:scot]
  - initialize_hints (done) [SIM177:scot]
  - set_stage (done) [SIM178:scot]
  - ask_question (done) [SIM179:scot]

        - acknowledge_positive (done) [DMTASK37:scot]
    - keep_drilling_topic (done) [SIM180:scot]
- keep_drilling_session (done) [SIM181:scot]
- pick_topic (done) [SIM182:scot]
- intro_utter5 (done) [SIM183:scot]
- drill_on_topic (done) [SIM184:scot]
    - pick_question (done) [SIM185:scot]
    - initialize_hints (done) [SIM186:scot]
    - set_stage (done) [SIM187:scot]
    - ask_question (done) [SIM188:scot]
        - acknowledge_positive (done) [DMTASK38:scot]
    - keep_drilling_topic (done) [SIM189:scot]
    - pick_question (done) [SIM190:scot]
    - initialize_hints (done) [SIM191:scot]
    - set_stage (done) [SIM192:scot]
    - ask_question (done) [SIM193:scot]
        - acknowledge_positive (done) [DMTASK39:scot]
    - keep_drilling_topic (done) [SIM194:scot]
    - pick_question (done) [SIM195:scot]
    - initialize_hints (done) [SIM196:scot]
    - set_stage (done) [SIM197:scot]
    - ask_question (done) [SIM198:scot]
        - acknowledge_positive (done) [DMTASK40:scot]
    - keep_drilling_topic (done) [SIM199:scot]
- keep_drilling_session (done) [SIM200:scot]
- pick_topic (done) [SIM201:scot]
- drill_on_topic (done) [SIM202:scot]
    - pick_question (done) [SIM203:scot]
    - initialize_hints (done) [SIM204:scot]
    - set_stage (done) [SIM205:scot]
    - ask_question (done) [SIM206:scot]
        - acknowledge_positive (done) [DMTASK41:scot]
    - keep_drilling_topic (done) [SIM207:scot]
    - pick_question (done) [SIM208:scot]
    - initialize_hints (done) [SIM209:scot]
    - set_stage (done) [SIM210:scot]
    - ask_question (done) [SIM211:scot]
        - acknowledge_positive (done) [DMTASK42:scot]
    - keep_drilling_topic (done) [SIM212:scot]
    - pick_question (done) [SIM213:scot]
    - initialize_hints (done) [SIM214:scot]
    - set_stage (done) [SIM215:scot]
    - ask_question (done) [SIM216:scot]
        - acknowledge_positive (done) [DMTASK43:scot]
    - keep_drilling_topic (done) [SIM217:scot]

- keep_drilling_session (done) [SIM218:scot]
- pick_topic (done) [SIM219:scot]
- drill_on_topic (done) [SIM220:scot]
  - pick_question (done) [SIM221:scot]
  - initialize_hints (done) [SIM222:scot]
  - set_stage (done) [SIM223:scot]
  - ask_question (done) [SIM224:scot]
    - acknowledge_positive (done) [DMTASK44:scot]
    - paraphrase_correct_answer (done) [DMTASK45:scot]
  - keep_drilling_topic (done) [SIM225:scot]
- keep_drilling_session (done) [SIM226:scot]
- pick_topic (done) [SIM227:scot]
- drill_on_topic (done) [SIM228:scot]
  - pick_question (done) [SIM229:scot]
  - initialize_hints (done) [SIM230:scot]
  - set_stage (done) [SIM231:scot]
  - ask_question (done) [SIM232:scot]
    - acknowledge_positive (done) [DMTASK46:scot]
  - keep_drilling_topic (done) [SIM233:scot]
  - pick_question (done) [SIM234:scot]
  - initialize_hints (done) [SIM235:scot]
  - set_stage (done) [SIM236:scot]
  - ask_question (done) [SIM237:scot]
    - acknowledge_positive (done) [DMTASK47:scot]
  - keep_drilling_topic (done) [SIM238:scot]
  - pick_question (done) [SIM239:scot]
  - initialize_hints (done) [SIM240:scot]
  - set_stage (done) [SIM241:scot]
  - ask_question (done) [SIM242:scot]
    - acknowledge_positive (done) [DMTASK48:scot]
  - keep_drilling_topic (done) [SIM243:scot]
  - pick_question (done) [SIM244:scot]
  - initialize_hints (done) [SIM245:scot]
  - set_stage (done) [SIM246:scot]
  - ask_question (done) [SIM247:scot]
    - acknowledge_positive (done) [DMTASK49:scot]
  - keep_drilling_topic (done) [SIM248:scot]
  - pick_question (done) [SIM249:scot]
  - initialize_hints (done) [SIM250:scot]
  - set_stage (done) [SIM251:scot]
  - ask_question (done) [SIM252:scot]
    - acknowledge_negative (done) [DMTASK50:scot]
    - give_RB_hint_question (done) [DMTASK51:scot]
      - acknowledge_positive (done) [DMTASK52:scot]
      - ask_followup_RB_question (done) [DMTASK53:scot]

                - acknowledge_positive (done) [DMTASK54:scot]
            - keep_drilling_topic (done) [SIM253:scot]
        - keep_drilling_session (done) [SIM254:scot]
    - summarize_tutoring (done) [SIM255:scot]
        - initialize_summary_problems (done) [SIM256:scot]
        - introduce_end_summary (done) [SIM257:scot]
        - summarize_positive (done) [SIM258:scot]
        - summarize_negative (done) [SIM259:scot]
            - introduce_neg_summary (done) [SIM260:scot]
            - wrap_up_problems (done) [SIM261:scot]
- goodbye (done) [SIM262:scot]

Appendix B: Example Transcript from Experiment

Note: In the transcript below, student turns contain two parts. The first part is the actual utterance (as transcribed by a human). The second part, shown in parentheses, is the utterance recognized by SCoT. The dialogue below is from a subject in Group I.

Student: hello (hello)
Tutor: Hello, please select your name from the box.
Tutor: Please select the session you would like to discuss from the box.
Tutor: OK, I am ready to discuss your session.
Tutor: Are you ready to begin?
Student: yes (yes)
Tutor: OK, In this session there were 2 fire crises and 1 flood crisis
Tutor: There were 10 necessary actions, you performed 1 at the appropriate time.
Tutor: There were 3 crises requiring boundaries, and you set boundaries in the correct place 0 out of 4 times.
Tutor: There were 8 actions you ordered requiring a repair team, and you sent the correct repair team 5 times.
Tutor: You had some problems in this session
Tutor: Lets discuss the areas that were problematic for you
Tutor: Ok, the first crisis was a fire here in Crew living space No. 2 (compartment 3-97-02-L)
Tutor: After the fire alarm sounded, the first thing you did was order repair team 2 to fight the fire.
Tutor: After ordering fire fighting, the next thing you did was order repair team 2 to electrically and mechanically isolate the compartment.
Tutor: However, there is 1 other thing you should have done before ordering isolation.
Tutor: What is it?
Student: setting boundaries [question-rise] (flooding boundaries)
Tutor: No.
Tutor: What if its a false alarm?
Tutor: Go ahead and try again.
Student: investigate compartment (investigate compartment)
Tutor: Yes, that's right.
Tutor: You should send repair team 2 to investigate.
Tutor: Why is it necessary to investigate after the alarm sounds?
Student: to see if it's a false fire (receive all fires)
Tutor: Okay. Well.
Student: investigate false alarm (investigate all alarms)
Tutor: You need to investigate in order to verify the alarm.
Tutor: After ordering isolation, the next thing you did was order repair team 2 to dewater the compartment.
Tutor: After ordering dewatering, the next thing you did was order repair team 2 to electrically and mechanically isolate the compartment.

Tutor: This was correct.
Tutor: Ordering repair team 2 to electrically and mechanically isolate the compartment was the last thing you did towards addressing this crisis.
Tutor: Assuming the compartment is isolated, the fire is out, and you have a smoke report, there is 1 other thing you should have done after ordering isolation.
Tutor: What is it?
Student: investigate compartment (investigate compartment)
Tutor: No.
Tutor: What if the smoke starts spreading to other compartments?
Tutor: Go ahead and try again.
Student: establish boundaries ([NOT RECOGNIZED])
Student: set boundaries (set boundaries)
Tutor: Yes, that's right.
Tutor: You should send repair team 2 to set smoke boundaries.
Tutor: Why is it necessary to set smoke boundaries when you have a report of smoke?
Student: prevent smoke from spreading to other compartments ([NOT RECOGNIZED])
Student: stop smoke from spreading to other compartments (stop some first other pumps)
Tutor: Okay. Well.
Tutor: You need to set smoke boundaries in order to contain the smoke.
Tutor: Ok, the next crisis was a flood here in Engine room No. 2 (compartment 4-254-0-E)
Tutor: After the flood alarm sounded, the first thing you did was order repair team 5 to dewater the compartment.
Tutor: However, there are 3 other things you should have done before ordering dewatering.
Tutor: Lets begin with the first 1.
Tutor: What is it?
Student: set boundaries (set boundaries)
Tutor: No.
Tutor: When we discussed the previous casualty in 3-97-02-L what did we say needed to be done in order to verify the fire alarm?
Student: investigate compartment (investigate compartments)
Tutor: Yes, that's right.
Tutor: Flood alarms need to be verified just like fire alarms.
Tutor: So what should you do in order to verify a flood alarm?
Student: investigate compartment (investigate compartment)
Tutor: Yes, that's right.
Tutor: You should send repair team 5 to investigate.
Tutor: Assuming you have a flood report, lets move on to the next 2 things you should do.
Tutor: What is one of them?
Student: dewater compartment (dewater compartment)
Tutor: No.
Tutor: When we discussed the previous casualty in 3-97-02-L what did we say needed to be done in order to contain the fire crisis?
Student: isolate compartment (isolate compartment)

Tutor: No, thats not right.
Tutor: We said a repair team needed to set fire boundaries.
Tutor: Floods need to be contained just like fires.
Tutor: So what should you do in order to contain a flood crisis?
Student: set boundaries (set boundaries)
Tutor: Yes, that's right.
Tutor: You should send repair team 5 to set flood boundaries.
Tutor: Why is it necessary to set flood boundaries when you have a report of flood?
Student: contain th(e)- flood (contain flood)
Tutor: Yes, that's right.
Tutor: You need to set flood boundaries in order to contain the flood.
Tutor: There is 1 more thing you should have done.
Tutor: What is it?
Student: isolate compartment (isolate compartment)
Tutor: Yes, that's right.
Tutor: You should send repair team 5 to electrically and mechanically isolate the compartment.
Tutor: Ordering repair team 5 to dewater the compartment was the last thing you did towards addressing this crisis.
Tutor: OK, moving on.
Tutor: Let's talk about ordering boundaries around compartments, starting with some terminology.
Tutor: Primary boundaries are the first bulkheads forward and aft of the crisis.
Tutor: Secondary boundaries are the second bulkheads forward and aft of the crisis.
Tutor: OK, now let's work through some drills. I will pick a couple compartments, and you answer the questions about boundaries for each one.
Tutor: Let's start some drills with primary boundaries, and then move on to the secondary boundaries.
Tutor: Let's assume there is a fire here, in 2-256-2-F.
Tutor: What are the primary bulkheads you should use for setting boundaries on compartment 2-256-2-F?
Student: primary compartment two two zero [hesitant-start] forward [hesitant-end] ([NOT RECOGNIZED])
Student: primary compartment two five four forward ([NOT RECOGNIZED])
Student: primary forward two five four (primary forward two five four)
Tutor: Good job!
Tutor: Please try again for the primary aft bulkhead.
Student: primary aft three zero zero (primary aft three zero zero)
Tutor: Good job!
Tutor: The complete primary boundaries are primary aft 300 and primary forward 254.
Tutor: Now, let's assume there is a fire here, in 1-330-0-Q.
Tutor: What are the primary bulkheads you should use for setting boundaries on compartment 1-330-0-Q?

Student: primary forward three hundred (primary forward three hundred)
Tutor: Good job!
Tutor: Please try again for the primary aft bulkhead.
Student: primary aft three three eight (primary aft three three eight)
Tutor: Good job!
Tutor: The complete primary boundaries are primary aft 338 and primary forward 300.
Tutor: Now let's try identifying the secondary boundaries for a couple compartments.
Tutor: Let's assume there is a flood here, in 1-78-1-Q.
Tutor: What are the secondary bulkheads you should set boundaries on?
Student: secondary forward eighteen (secondary forward eighteen)
Tutor: No, thats not right.
Tutor: The 18 bulkhead is forward the desired secondary bulkhead.
Student: secondary forward forty two (secondary forward forty two)
Tutor: Remember, secondary bulkheads are the second pair of bulkheads forward and aft of the crisis.
Tutor: Please try again for the secondary boundaries.
Student: secondary forward seven eight (secondary forward seventy eight)
Tutor: No, thats not right.
Student: secondary forward forty two (secondary forward forty two)
Tutor: In this case the secondary boundaries are secondary aft 174 and secondary forward 42.
Tutor: Now, let's assume there is a flood here, in 1-330-2-Q.
Tutor: What are the secondary bulkheads you should set boundaries on?
Student: secondary forward two five four (secondary forward two five four)
Tutor: Good job!
Student: secondary aft three seven zero (secondary aft three seven zero)
Tutor: Please try again for the secondary aft bulkhead.
Student: secondary aft three seven zero (secondary aft three seven zero)
Tutor: Good job!
Tutor: The complete secondary boundaries are secondary aft 370 and secondary forward 254.
Tutor: Okay, one additional wrinkle to this is if a crisis is so close to the edge of the ship that you can not have distinct primary and secondary boundaries on that side.
Tutor: In that case, you use the same number as both primary and secondary boundary.
Tutor: For example, if you consider compartment 3-38-0-Q, both of the forward boundaries use bulkhead 18.
Tutor: The boundaries for this compartment are primary forward 18, primary aft 42, secondary forward 18, secondary aft 78.
Tutor: Now lets try identifying all the bulkheads for compartments near an edge of the ship.
Tutor: Let's assume there is a fire here, in 4-22-0-L
Tutor: What are the primary bulkheads you should use for setting boundaries on compartment 4-22-0-L?
Student: primary forward eighteen (primary forward eighteen)
Tutor: Good job!
Tutor: Please try again for the primary aft bulkhead.

Student: primary aft forty two (primary aft forty two)
Tutor: Good job!
Tutor: The complete primary boundaries are primary aft 42 and primary forward 18.
Tutor: What are the secondary bulkheads you should set boundaries on compartment 4-22-0-L?
Student: secondary forward eighteen (secondary forward eighteen)
Tutor: Good job!
Tutor: Please try again for the secondary aft bulkhead.
Student: secondary aft seventy eight (secondary aft seventy eight)
Tutor: Good job!
Tutor: The complete secondary boundaries are secondary aft 78 and secondary forward 18.
Tutor: Now, just to make sure you have a handle on all the parameters for a boundary order, we should practice the final two parameters, the above and below decks.
Tutor: Suppose that compartment 2-338-1-L, here, floods.
Tutor: What decks would you use to set boundaries?
Student: above deck two (above deck two)
Student: below deck three (below deck tree)
Tutor: No, thats not right.
Tutor: The decks involved in boundaries are immediately above and below the crisis.
Tutor: Please try again for the above and below decks.
Student: above deck two below deck three (above deck two below deck three)
Tutor: No, thats not right.
Tutor: In this case the above and below decks are above deck 1 and below deck 2.
Tutor: Now, suppose that compartment 4-174-0-E, here, floods.
Tutor: What decks would you use to set boundaries?
Student: i don't know (i_dunno)
Tutor: Well.
Tutor: The decks involved in boundaries are immediately above and below the crisis.
Tutor: Please try again for the above and below decks.
Student: above deck three below deck four (above deck three below deck four)
Tutor: Good job!
Tutor: Now, let's practice identifying repair team jurisdictions.
Tutor: First of all, let's go over some terminology.
Tutor: Which region of the ship is this?
Tutor: Fore, midship, or aft?
Student: midship (midship)
Tutor: Good job!
Tutor: Now, which region of the ship is this?
Student: forward (forward)
Tutor: Fore, midship, or aft?
Student: fore (fore)
Tutor: Good job!
Tutor: Now, which region of the ship is this?

Student: aft (aft)
Tutor: Fore, midship, or aft?
Student: aft (aft)
Tutor: Good job!
Tutor: Lets move on to the areas covered by each of these repair teams.
Tutor: Which repair team has jurisdiction over the fore region?
Student: repair two (repair two)
Tutor: Good job!
Tutor: Which repair team has jurisdiction over the aft region?
Student: repair three (repair three)
Tutor: Good job!
Tutor: Which repair team has jurisdiction over the midship region?
Student: repair five (repair five)
Tutor: Good job!
Tutor: Now let's review the locations where the three repair teams are based.
Tutor: Here is where one of the repair teams is based.
Tutor: Which repair team is this?
Student: repair three (repair three)
Tutor: Good job!
Tutor: Now, here is where one of the repair teams is based.
Tutor: Which repair team is this?
Student: repair five (repair five)
Tutor: Good job!
Tutor: Now, here is where one of the repair teams is based.
Tutor: Which repair team is this?
Student: repair two (repair two)
Tutor: Good job!
Tutor: Now we are ready to move on to relating repair teams to individual compartments that need damage control.
Tutor: Can you tell me which repair team you should send to compartment 4-300-0-E?
Student: repair three (repair three)
Tutor: Good job!
Tutor: Can you tell me which repair team you should send to compartment 1-100-2-Q?
Student: repair two (repair two)
Tutor: Good job!
Tutor: Can you tell me which repair team you should send to compartment 2-174-6-Q?
Student: repair five (repair five)
Tutor: Good job!
Tutor: Can you tell me which repair team you should send to compartment 4-402-0-J?
Student: repair three (repair three)
Tutor: Good job!
Tutor: Can you tell me which repair team you should send to compartment 3-97-1-T?
Student: repair two (repair two)

Tutor: Good job!
Tutor: Alright, let's go over the most important points from this session.
Tutor: You performed well in the areas of investigation, and isolation.
Tutor: We discussed your errors relating to dewatering.
Tutor: Remember, It's important to order dewatering in order to eliminate the flood.
Tutor: Ok, this ends our review. Goodbye.

Appendix C: Post-Experiment Questionnaire

Questions

For the statements below, please click a number 1-7, where 1 means strongly disagree and 7 means strongly agree.

The tutor was clear about what I needed to learn.
The tutor's explanations were easy to understand.
The tutor's voice was easy to understand.
I feel like the tutor understood what I said.
I feel that the information the tutor told me was accurate.
The effort that was needed to interact with the tutor was manageable.
I found interacting with the tutor to be interesting and engaging.
I found the speech interface of the tutor interesting and engaging.
The spoken messages in the simulator were easy to understand.
The effort that was needed to interact with the simulator was manageable.
I found interacting with the simulator to be interesting and engaging.
I enjoyed interacting with this system.
I am confident that I could explain to a friend how to use this system.
I think I could study efficiently with this sort of automated tutoring system.

For the questions below, please type in your answers.

1. What did you like the most about interacting with this system?
2. What did you like the least about interacting with this system?
3. What areas do you wish the tutor covered but didn't?
4. Do you have any general comments about the system or your experience with it?

Student Answers

Average user ratings (1 = strongly disagree, 2 = strongly agree)

The tutor was clear about what I needed to learn. (4.60)
The tutor's explanations were easy to understand. (4.31)
The tutor's voice was easy to understand. (4.21)
I feel like the tutor understood what I said. (2.55)
I feel that the information the tutor told me was accurate. (4.86)
The effort that was needed to interact with the tutor was manageable. (3.57)
I found interacting with the tutor to be interesting and engaging. (4.29)

I found the speech interface of the tutor interesting and engaging. (4.07)
The spoken messages in the simulator were easy to understand. (5.02)
The effort that was needed to interact with the simulator was manageable. (4.81)
I found interacting with the simulator to be interesting and engaging. (4.90)
I enjoyed interacting with this system. (4.48)
I am confident that I could explain to a friend how to use this system. (5.29)
I think I could study efficiently with this sort of automated tutoring system. (3.67)

User responses to "What did you like the most about interacting with this system?"

- it was easy to understand
- I was amazed how accurate the system was in recognizing my messages and how it acted upon them.
- the simulations were for the most part realistic and you got to practice your skills as if you were in a real situation.
- the re-explanation of concepts that were glazed over in the introduction
- The simulator was really enjoyable, and the tutor really helped clarify some things (even though it was annoying).
- The action of damage control.
- It was helpful, and interesting.
- I liked that I did not have to type in what I said. I liked that it was forward yet polite.
- Tutor on problems I had
- it was fun like a computer game that did what i told it to. it was like playing god with a battleship. and they did what i told them to do. plus, the tutor gave alot of positive reinforcement.
- When it recalled what I said
- It was interactive and had a nice visual interface.
- simulator was interesting, fun
- The tutor's reading was pleasant.
- I learned a lot from the second and third tutor sessions.
- there was a lot of simulations
- It allowed me to learn a lot
- Being recognized accurately and having orders followed by a computer.
- Issuing commands.
- tutor was very helpful and it was easy to hear instructions
- I liked seeing how far speech recognition has come
- its comments were clear and direct.
- Organizing the script under time constrain and pressure or occurring event
- The second two tutoring sessions with the open mic were nice: I liked that he sounded as if he were responding directly to me, and how he had a good sense of my performance on the preceeding simulations.
- I've never done anything like this before so it was very interesting. I did feel like I was learning, and the program itself was fun.

- The clear highlighting and numbering of compartments
- I liked how the system was able to show me which areas I struggled in.
- The problem solving aspect, and the tutor's help.
- It was fun to do the simulations after I understood them better.
- story like learning environment
- The tutor -- it seems to take context into account well, and can handle various response styles.
- It was like a little computer game
- I liked that the system was able to recognize my natural voice right away.
- The relative realism of the simulation
- The speach recognition was pretty neat.
- The tutor could recognize common phrasing rather then needing one particular phrase
- It was an interesting way to learn something new
- I think it's amazing how easily the tutor recognized what I was saying.

User responses to "What did you like the least about interacting with this system?"

- The tutor could not comprehend most of what I was saying, and in the beginning of the simulations, I felt nervous and intimidated. I was completely unsure of what I was doing at first before the tutor gave me feedback.
- sometimes it was frustrating when it couldn't accurately understand what I was saying
- not being able to ask questions, and that it often didn't understand what i said
- Sometimes in the simulator, my voice would be interpreted incorrectly and there would be no way to reverse things. The tutor program was just really annoying when it would not pick up one of my responses and I'd have to listen (and wait for it to) repeat the same thing for 30 more seconds.
- The tutor's program was too remedial.
- It couldn't understand what I was saying.
- It frequently did not receive the commands I said correctly. It also needed to give me more time to speak, at times.
- Not understanding what I said
- he was mean and didnt want to listen to anything i wanted to say. and sometimes, he was patronizing. i didnt like that. i think he's just kinda angry.
- When it inaccurately recalled what I said or ignored my attempts to correct my answer
- The tutoring program was slow and spent an equal amount of time on my strong and weak areas rather than more on my weak areas.
- speech recognition in the tutor system is very weak
- The computerized responses were long, redundant, and interrupted my attempts to string commands together. Commands that were strung together in order to avoid verbose feedback failed.. I imagine the speech recognition couldn't handle a very long audio clip. This required me to repeat several long commands or listen to verbose feedback, both of which was frustrating.
- speech recognition was horrible and slow

- I had to repeat a lot of things. If I was unclear about a tutor explanation, I couldn't ask questions.
- sometimes the voice recognition was incorrect
- It could not understand what I was saying sometimes.
- I didn't like that the tutor made me do all the drills both times even when it said I correctly called the right repair team every time. The drills got redundant. I didn't like that the repair teams would notify me several times that there was a fire or flood, even after I had already ordered actions to be done to control it.
- Conversing with the tutor. He is mean.
- frustrating when my speech was not recognized by the system
- It is hard for me to learn without having an "instruction list" written down on paper- I am terrible at following auditory directions
- it often misunderstood what i said. the simulator didnt give me enough information. the tutor often would repeat things that i already clearly knew so it was very redundant.
- if one makes a mistake in the command one should be able to continue when mistake was made instead of giving entire command again.
- When it wouldn't understand what I said.
- It was very repetitive at times, and the tutor didn't always seem to know precisely where my problems were. I also didn't think the introduction part was helpful because many things were left unexplained or weren't explained well enough. It was also a little frustrating when the program couldn't recognize what I had said, or when it took a while to register what I had said.
- Speech not correctly recognized, short amount of time messages from repair teams, etc. remained on screen
- I didn't like how I had to review info that I already knew in the tutorial with the tutor. I also didn't like how I had to repeat myself many times before the tutor understood what I was saying.
- The poor recognition (at times) in the tutor.
- The first simulation
- didn't seem possible to ask questions
- The simulator display is confusing, could use a redesign. Push-to-talk was strenuous in the times of crises.
- It didn't always understand what I was saying, and I couldn't remember to rephrase thing
- I was slow and very confusing at times.
- Not having the speech recognition system understand me, not having some things explained, the tutor is kind of irritating when he says, "Well…" all drawn out
- The tutor was not very helpful at correcting mistakes in identifying the deck system.
- The processing took way too long- it was especially frustrating when I said something wrong and had to wait two or three minutes to get the right thing out (if there was an abort it would help)
- saying "affirmative" was hard for me.

- It tends to be repetitive
- The simulater had a great deal of difficulty understanding what I was saying. I would often repeat orders at least once or twice.

User responses to "Do you have any general comments about the system or your experience with it??"

- It was fun and felt like playing a game. I feel like the tutor would have been more helpful before I started any simulations, such as immediately following the introduction.
- at first I was overwhelmed by the wealth of information I recieved in the introduction. But after trying it out and having the tutor point out my mistakes the first time I started to really enjoy the excercises.
- i did not like this system. it needs to understand me better
- It was fun. The simulator was really enjoyable; the tutoring was helpful, but monotonous.
- I think the voices could be better - some are easy to understand but some are difficult. I think the command log could be larger, which would make it easier to use. Also, the ship display would sometimes display too many compartment names making it difficult to read the compartment with a crisis.
- I felt that an oral/typing combination interface may be more versatile.
- Tutor is too slow
- it's an interesting, different experience. i was pretty happy with it.
- The simulator spoke naturally but the tutor was much more dificult to understand and I relied more on the text. It was a fun experience overall and I wish I had more time to save the ship!
- One thing that wasn't clear to me was how to know when the below deck was the same number as the first number and when it was one more. The tutoring system was sometimes frustratingly slow. Overall though it was fun to use and pretty good at voice recognition.
- n/a
- Way too verbose.. I couldve learned the proccess quickly by reading about it. Why should people go throught the hassles of voice recognition when typing is much more accurate and just as fast?
- I felt I didn't accurately understand primary and secondary boundaries. I think I would have done better if there were visual explanations.
- i learned a lot in a short amount of time,. it can be frustrating when the computer doesn't know what i'm saying. I feel that it looks for a few key phrases, so after I just started saying "desmoke" or "dewater" to answer the debriefing quesitons, it became a lot easier. So i guess answering with the bold letters on the guide helped.
- If this system could improve understanding my speech, it would become a very effective learning tool.
- we didn't need to know about firepumps

- It's fun.
- overall system worked well, but was difficult given the complexity of the task so had some disadvantages
- No.
- i think i had a particularly unlucky experience because i encountered problems with the program freezing and not recognizing my voice. this made the test very frustrating and annoying. i also think the simulater should give me more information and it should do more to step you through a crisis one at a time. gernerally the voice recognition was good, but sometimes slow and very tedious.
- Why do the words sound inhuman? in real time the wrong answer script interuppts the attempt to give command again. the scripted command ad some flexibility. the commands did not require verbatim commands, so that was easier.
- This was an enjoyable, if stressful test. I think the tutor was pretty effective.
- Finding out which compartments are in fore, aft, midship. Without a color diagram, some of the ambiguous ones are hard to discern which area they belong to, especially in the upper parts of the ship.
- Felt uncomfortable throughout because I never got a lock on boundaries or decks, which slowed me down. Speech recognition problems also troublesome, beyond my choosing words correctly.
- I think the system is interesting. I found giving commands rather frustrating.
- The order of procedures was not clear until I used the tutor and it quizzed me on what to do next, before, etc. Chain of command was difficult to keep straight.
- Excellent, very elaborate and responsive interaction. Has several annoyances (GUI and TTS, mostly), but those are minor.
- The tutor only ever explained things one way (pedagogically not effective, because if I didn't understand him the first time, I probably wouldn't understand him the fifth time)
- It was fun, and if the learning curve is worked out, this could be its own video game.
- It was kind of odd that the intro covered things that I never used at all during the simulation, and also that a couple of the windows (the one with pumps and the one with checkboxes in the lower left) were not useful or needed at all.
- Communicating with the computer was a little frustrating at first, but I got used to it and learned to tell it what it wants to hear.
- Overall I was rather frustrated by the general time it took to correct mistakes and the repetitiveness of the two sessions. However I was impressed with how common the speech was within the tutor session and the hinting system. It would be kind of nice (as always) to have a non-robotic voice (something pre-recorded) as well as an abort option or perchaps a menu that pops up with some suggestions about wording
- It seems to have done ok with my slight accent