Empirical Foundations for Intelligent Coaching Systems

Elizabeth Owen Bratt, Karl Schultz, Stanley Peters, Tina Chen
Stanford University
Stanford, CA

ebratt@csli.stanford.edu, schultzk@stanford.edu, peters@csli.stanford.edu, tinachen@stanford.edu

Heather Pon-Barry

Robert Bosch Corp.
Palo Alto, CA

heather.pon-barry@rtc.bosch.com

ABSTRACT

The Navy is shifting its training and education from traditional methods, such as on-site instruction, texts, and observing students during drills, to computer-supported learning such as web-based instruction and computer simulations in lieu of live drills. This transition presents the challenge of keeping the best parts of traditional methods of instruction while obtaining the advantages that computers afford. The challenge is more difficult because to maximize savings in manpower, money and time, computer-based learning must be able to teach, evaluate and give feedback to students without any instructor in the loop.

A valuable aspect of traditional training methods, in which computers currently fall short, is the mentor/student relationship: an experienced person monitoring and guiding a novice's performance. The mentor gives the student direct, personalized feedback in a setting where the student can ask questions and discuss issues. Most computer simulations are lacking in this type of interaction.

We propose that giving computers the ability to hint, question, prompt and guide a student's actions using natural language will more closely simulate this relationship and greatly improve the effectiveness of computer-based learning. To assess this hypothesis, we are utilizing natural language technology to (1) allow students to use a damage control trainer for surface ships by speaking with the simulation system, and to (2) support a concurrent spoken discussion with an intelligent coaching system that aims to improve the student's immediate and future performance. The combined system performs a mentoring function, helping students learn correct actions and avoid practicing mistakes. We present data from United States Naval Academy cadets using the spoken damage control trainer and a spoken tutoring system, categorizing the opportunities their sessions present for coaching and organizing these opportunities within a mentoring framework. Additionally, natural language interaction has the advantage that students train as they will perform on duty.

ABOUT THE AUTHORS

Elizabeth Owen Bratt is a Senior Research Engineer at the Center for Study of Language and Information at Stanford University. Her research is in the areas of spoken language understanding and concept-to-speech generation in dialogue systems. She holds a Ph.D. in Linguistics from Stanford University, and she was previously a Research Linguist at SRI International.

Stanley Peters is Professor of Linguistics at Stanford University. His research is in the areas of meaning and of computation on language. His current research on spoken dialogue systems includes artificially intelligent tutoring systems. His education was in mathematics and linguistics at MIT, and he previously taught at the University of Texas at Austin.

Karl Schultz is a Research Engineer at the Center for Study of Language and Information at Stanford University. He holds a B.S. in Computer Science from the University of Illinois, where he worked on the development of the DC-Train simulator.

Tina Chen is an undergraduate Symbolic Systems major at Stanford University, working on analysis of coaching opportunities.

Heather Pon-Barry is a Research Engineer at the Research and Technology Center of Robert Bosch Corp, working on in-car dialogue systems She holds an A.M. in Symbolic Systems from Stanford University, where her thesis focused on tutoring strategies using natural language to improve the effectiveness of automated tutoring.

Empirical Foundations for Intelligent Coaching Systems

Elizabeth Owen Bratt, Karl Schultz, Stanley Peters, Tina Chen
Stanford University
Stanford, CA
ebratt@csli.stanford.edu, schultzk@stanford.edu, peters@csli.stanford.edu, tinachen@stanford.edu

Heather Pon-Barry
Robert Bosch Corp.
Palo Alto, CA
heather.pon-barry@rtc.bosch.com

INTRODUCTION

Automated tutoring and coaching systems can be effective in supporting learning, and in providing new opportunities for instruction (Collins, Brown & Newman, 1987, Lesgold et al., 1988). Various automated tutors (Koedinger et al., 1997, Gertner & VanLehn, 2000, Graesser et al., 2001, Evens et al. 2001) and coaches (Constantino-Gonzalez & Suthers, 2003) have been developed.

A spoken natural language interface to an automated tutor may make it even more powerful and effective. (e.g. Aleven, 2001). Using natural language for coached practice during a problem-solving simulation has similar promise, particularly when paired with natural language post-practice review (Katz, O'Donnell & Kay, 2000).

The advantages from using natural language come in several areas. A natural language interface encourages free responses, rather than choice from a menu, and thus gives greater opportunity for students to articulate their current state of knowledge in their own words (cf. Chi et al., 1989 on the value of self-explanation). For any domain with specialized terminology, a natural language interface allows the system to judge the student's usage of correct terms and phrasing, in the context of their overall understanding (Evens & Michael, 2005). Spoken natural language gives additional advantages to the system of detecting metacommunicative detail, such as the speaker's confidence in their answer, indicated by intonation, pauses, speech rate, etc., (Pon-Barry et al., 2004a) and their likelihood of recognizing the correct answer (Smith & Clark, 1993) as well as the speaker's emotion (Litman & Forbes-Riley, 2004). A natural language *dialogue* interface involves context, which permits references to past items of discussion. Dialogue can support clarification questions by either system or student. Furthermore, dialogue allows a tutor or coach to construct a point of discussion over several distinct utterances, as in a directed line of reasoning (Sanders, 1995).

The SCoT-DC (Spoken Conversational Tutor – Damage Control) tutoring system, with demonstrated effectiveness in the domain of after-action review for shipboard damage control, is the foundation for development of a spoken language coach in the same domain, seeking answers to the following questions: Does spoken language have particular benefits for coaching systems? Can a spoken language coach work effectively in a fast-paced, dynamic, heavily verbal domain?

These questions lead to a more basic concern: Does human spoken coaching work in this environment? Observations of human coaching answer this question in the affirmative, leading to an examination of how the successful features of human coaching could be replicated in an automated system. Assuming success in developing an automated system with the characteristics of human coaching raises a number of additional issues: Would students find a speech-enabled automated tutor or coach enjoyable to use? Do students want automated help? Would it be capable and effective? Can a system detect when coaching should occur in this complex environment? Can the system fit the coaching smoothly into the rich spoken environment? Experiments at Stanford and at the United States Naval Academy (USNA) provide some answers.

THE DAMAGE CONTROL DOMAIN

Shipboard damage control refers to the task of containing fires, floods, and other critical events that occur aboard Navy vessels. The SCoT-DC spoken tutor conducts a reflective discussion with students after they complete a simulator session with DC-Train (Bulitko & Wilkins, 1999), a fast-paced, real-time, multimedia training environment for damage control. Several crises may occur within the same short time interval and demand immediate attention (e.g., in an average scenario, there may be 3 fires and 1 flood occurring simultaneously). The experiments use Voice-Enabled

DC-Train, in which the student plays the role of the DCA (Damage Control Assistant) giving spoken orders to three simulated repair teams, while receiving spoken reports and coordinating with other simulated officers on the ship. While the student may consult written versions of the communications as *Message Blanks*, the overall interaction is highly verbal and intense. During the USNA experiment, the average DC-Train session was 10 minutes long, during which there was a mean of 37 user utterances, 26 dialogue-related system responses, and many additional system reports and communications.

DC-Train incorporates several of the complexities of real-life shipboard damage control. The solution to the events in a DC-Train scenario may not be obvious even to experts, and there may be more than one valid solution. The student is under time-pressure, since failure to deal with fires, flood, and smoke soon enough leads to them spreading and requiring additional actions to eliminate.

Several kinds of complexity in DC-Train will make it challenging for an automated coach. Higher level goals are often achieved through multiple actions. If a student forgets some necessary actions, the coach will need to evaluate those errors of omission in the context of the higher level goal and whether the student appears to recognize and work toward that goal with other actions. This will influence the timing of guidance by the coach, since the coach may not be able to assess the student's understanding adequately till some time after an error. Also, the discrete nature of the commands issued in DC-Train means that if the coach provides guidance after an error, the student may not face a similar situation for several minutes, so the coach cannot immediately assess the effect of the advice.

THE SCOT-DC TUTORING SYSTEM

The architecture of the SCOT-DC tutoring system will form the basis of the new coaching system, expected to reach prototype status by November 2005.

Figure 1 shows the graphical user interface of SCoT-DC. The lower window contains a history of the tutorial dialogue. The upper window depicts the physical layout of the ship and serves as the *common workspace*—a space where both the student and the tutor can focus on items by zooming in or out, and also select, circle, or click-and-drag various entities in the display (e.g., compartments, regions, labels, and bulkheads, which are lateral walls in the ship).

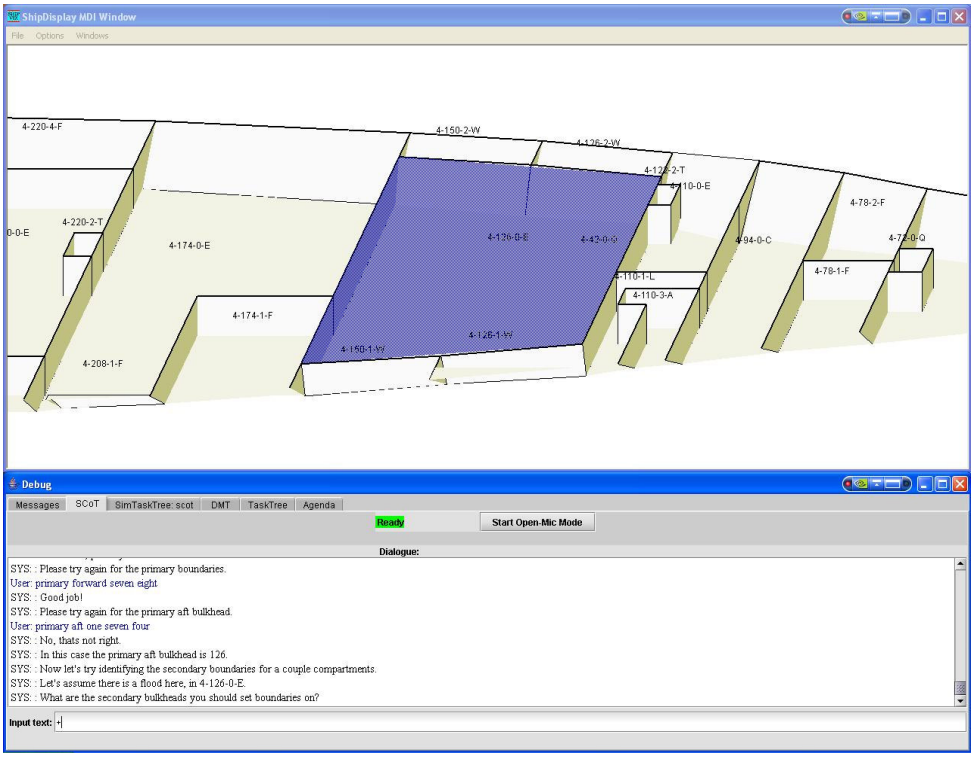

Figure 1. SCoT-DC Graphical User Interface

Overview of the SCoT Tutoring approach

While the SCoT-DC system is in the damage control domain, the SCoT framework is domain-neutral. SCoT is based on the assumption that tutorial dialogue is a *joint activity*—an activity in which participants have to coordinate their actions in order to succeed (Clark, 1996). The content of the dialogue (language and other communicative signals) is driven by both the problem being discussed as well as the nature of tutorial help. Following this hypothesis, SCoT's architecture separates conversational intelligence (e.g. turn management, use of discourse markers such as 'so' and 'OK') from the activity that the dialogue accomplishes (in this case, reflective tutoring). This separation provides for a clearer representation of how and why the nature of a task affects the dialogue.

SCoT is developed within the Architecture for Conversational Intelligence (Lemon, Gruenstein & Peters, 2002), a general purpose architecture which supports multimodal, mixed-initiative dialogue. SCoT is composed of three primary components: a dialogue manager, a knowledge representation, and a tutor.

Dialogue Manager

The dialogue manager handles aspects of conversational intelligence, helping the tutor interpret student utterances in context. It contains the following dynamically updated components:

1. The *Dialogue Move Tree*: a structured history of dialogue moves and dialogue threads
2. The A*ctivity Tree*: a temporal and hierarchical representation of the past, current, and planned activities initiated by either the tutor or the student
3. The *System Agenda*: issues to be raised by the system
4. The *Salience List*: objects referenced in the dialogue thus far, used for anaphora resolution
5. The *Pending List*: questions asked by the system but not yet answered
6. The *Modality Buffer*: a place to store (mouse) gestures for later resolution

In SCoT, the activity tree serves as the communication interface between the tutor component and the rest of the dialogue manager. Each activity initiated by the tutor corresponds to a tutorial goal such as discussing actions the student forgot to perform, or drilling the student on a particular knowledge area. The decompositions of these goals are specified by activity recipes contained in the *recipe library*.

Knowledge Representation in a Production System

The knowledge representation provides a domain-general interface to domain-specific information. In accordance with production-system theories of cognition (Anderson, 1993), knowledge specifying causal relationships between problem states (events and crises on the ship) and student actions 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 student's session.

Tutor

SCoT's tutor component contains the tutorial knowledge necessary to plan and carry out a flexible and coherent tutorial dialogue. Students will likely provide evidence during the dialogue that alters the tutor's original assessment. This emphasizes the need for a planning architecture that allows for revisions to the original dialogue plan. SCoT separates tutorial knowledge (i.e. how to lead a tutorial dialogue) from all other sources of information (e.g. domain knowledge, knowledge of the student). The tutorial knowledge is divided between a *planning and execution system* and a *recipe library*. Figure 2 depicts how the planning and execution system and the recipe library fit into the overall architecture. ASR (Automatic Speech Recognition) refers to the speech recognizer, and TTS (Text-to-Speech) refers to the speech synthesizer.

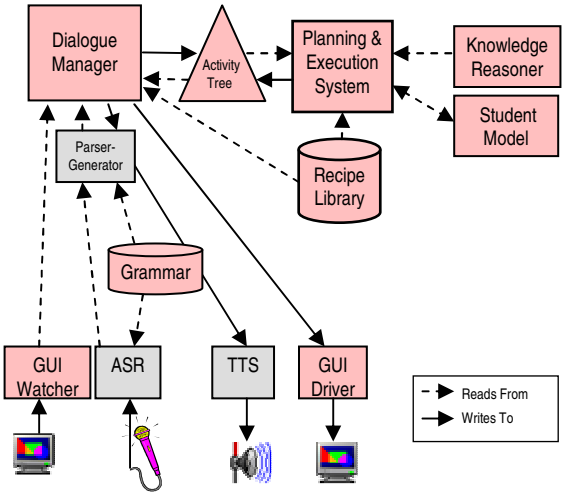

Figure 2. SCoT Architecture

Planning and Execution System

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 *student model*, and the *activity tree* (collectively referred to as the *Information State*). The Information State provides the tutor with information such as the current context of the dialogue, or a history of the student's interactions with SCoT. The separation of tutorial knowledge from other knowledge sources makes the tutor reusable in other domains (Schultz et al., 2003), and also provides a clean and consistent way for each component of the tutor to access information pertaining to the current context. This allows SCoT to lead a flexible dialogue and to reassess information from the Information State continually in order to select the most appropriate tutorial tactic. 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.

Recipe Library

The *recipe library* contains activity recipes that specify how to decompose a tutorial activity into other activities and low-level actions. An *activity recipe* contains a tutorial goal and a plan for how the tutor will achieve the goal. The recipes are written in a scripted language (Gruenstein, 2002) which allows for automatic translation into system activities—the same activities that are stored in the activity tree (each node on the activity tree corresponds to one tutorial goal).

An activity recipe contains three sections: *DefinableSlots*, *MonitorSlots*, and *Body*. The *DefinableSlots* specify what information is passed into the recipe, the *MonitorSlots* specify which parts of the information state are used in determining how to execute the recipe, and the *Body* specifies how to decompose the activity into other activities and low-level actions. The modular nature of the recipes makes it easy to test the effect of different pedagogical and conversational approaches.

Natural Language Components

Incoming student utterances are recognized using Nuance speech recognition (http://www.nuance.com). For the USNA experiment, a trigram language model was trained on the corpus of user utterances from the preceding Stanford experiments, in which a grammar-based language model was used. The Gemini natural language system (Dowding et al., 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 (http://festvox.org) limited domain synthesized voice using the Festival speech synthesizer (Black & Taylor, 1997).

Multimodal Interaction

By coordinating spoken and visual output in the *common workspace*, the tutor has increased flexibility in how it chooses to present information. Because the dialogue in SCoT is spoken rather than typed, the student is free to use the mouse to make gestures by clicks, circling, or dragging in the common workspace while speaking. This allows *pointing to* compartments, regions, and bulkheads (for setting boundaries) in the ship display while explaining an action from the session, or asking a question. Figure 3 illustrates the common workspace with four rectangles highlighting particular bulkhead walls selected by the student.

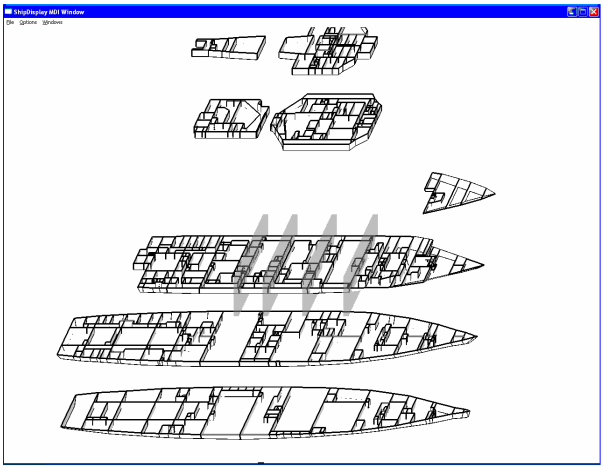

Figure 3. Boundary Selection Mode

The common workspace can also incorporate a symbolic representation of the changing states of various crises on the ship, as in Figure 4. Standard damage control symbology starts with a line drawn from the affected compartment in a ship diagram out to the side, where lines incrementally build triangles with each action (e.g., the report of smoke, the start of clearing smoke, and the completion of clearing smoke). This traditional DCA symbology is implemented abstractly, to allow future substitution of the newer computer symbology, based on graphic images and circles, as in the DCAMS (Damage Control Action Management Software) system.

While current tutoring recipes do not use symbology, the graphics capability is available for future tutor and coach development using a concise presentation of a

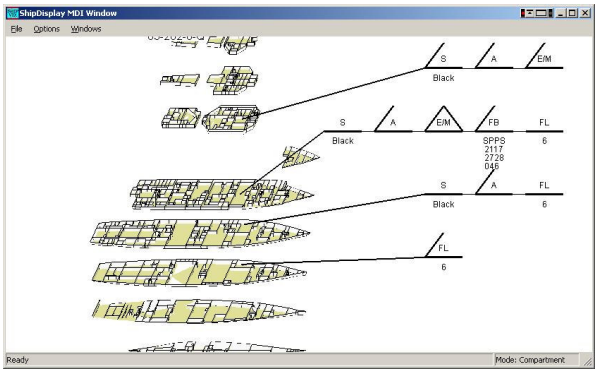

Figure 4. DCA Symbology

sequence of events. The tutor can also show patterns of mistakes by highlighting portions of symbology. For example, repeatedly forgotten *desmoke* commands can be colored red within the symbology representing the actions of an entire session.

EXPERIMENTS WITH SCOT-DC

Data from three experiments with the SCoT-DC tutor illuminates the issues raised in developing a spoken coach for a domain like damage control. The first two experiments were conducted at Stanford University, using students unfamiliar with damage control as subjects. One experiment timed tutoring of three knowledge areas differently in three subject pools, and demonstrated that tutoring in a particular area raised student performance in that area more than just simulator practice alone (Peters et al., 2004, Pon-Barry et al., 2004b). A second experiment demonstrated that tutoring strategies specifically enabled by natural language (referring back to past explanations and paraphrasing correct answers) result in greater learning gains (Pon-Barry, 2004) than in a control condition. These experiments had 30 and 40 subjects respectively.

The largest experiment was took place in February and March 2005 at the United States Naval Academy (USNA), using approximately 210 USNA midshipmen, in 12 sections of about 18 cadets, as subjects. The experiment was conducted during a section meeting for a 1 hour 50 minute lab. To control the effect of many subjects speaking close together in the same room, the cadets used Protravelgear PlaneQuiet noise-canceling headphones and Andrea ANC-700 noise canceling microphones. Each student engaged in one practice DC-Train session, followed by an actual DC-Train session, a tutoring session if applicable to the subject condition (see Table 1), and then a final DC-Train session.

Table 1. USNA Experiment Conditions

Condition	1	2	3	4	5	6	7
# of subjects	27	32	24	35	20	21	13
System high-lights, circles	*	*			*		
User can click, drag and circle	*		*				
Recognition of user speech Tutoring on user's session	*	*	*	*			
User reads a sample tutoring session. Clicks to advance page					*	*	
User is not tutored. Plays solitaire instead							*

Transcription of the speech and analysis of the tutoring data from the USNA experiment is underway.

HUMAN COACHING FOR DC-TRAIN

Given the fast-paced, intense, highly verbal domain of damage control, a first question is whether human spoken coaching is an effective way to improve student performance in DC-Train. The primary model coach observed in studying this issue is a former Navy DCA, familiar with DC-Train and experienced with simulation-based training (Perry McDowell of Naval Postgraduate School). He coached two novices to DC-Train through several videotaped sessions, which have been transcribed and analyzed with keywords marking types of events. These sessions will be referred to as the "Navy DCA sessions." A system developer also coached three other novices to DC-Train, and these videotaped sessions are currently undergoing transcription and analysis. All novices were Stanford students joining the project as new employees.

Impressionistically, coaching was extremely effective in boosting performance. All novices improved significantly, and found the coaching helpful and easy to understand, even amidst all the activity of a typical DC-Train scenario. At times, the coach had to speak over other spoken system output, indicating that an automated coach will also likely have to interrupt the system to speak to the student, balancing the negative effect of the disruption against the need for coaching (McFarlane & Latorella, 2002). In the Navy DCA sessions, most interruptions (17) were to explain student

error. Explaining what to do next (12) and alerting the student to anticipate upcoming events (10) followed in frequency. As the student improved, and the coaching became more advanced, there was less need to interrupt the system. In the Navy DCA sessions, the later sessions had many fewer coaching events in general, compared to the initial session, as shown in Table 2.

Table 2. Navy DCA Coaching Events Per Session

Session	Subject 1	Subject 2
1	79	108
2	48	70
3	55	71

Stages of human coaching

The type of coaching changed over the course of the DC-Train sessions. The initial stage of coaching often focused on how to say commands fluently, for improved recognition and understanding by the system. In the two Navy DCA sessions, the first half of each student's session had 6 questions about how to say a command, while one student asked 2 such questions in the second half, and the other student asked 3. The requirement of fluency, though imposed by the state of the technology, correlates with a need for fluent communication over the radio in real-life damage control.

A related initial stage area for coaching was the communication protocol needed by the DCA. For example, the DCA is expected to verbally acknowledge incoming messages and to engage in *interrogative* dialogues to supply information missing from a DCA's initial order.

Also in the initial phase, the student learned where to find information from the various displays, such as checking the Message Blanks to see if investigators had reported back, checking the ship display to find the location of the nearest bulkheads to set fire boundaries on, and finding the compartment number within the reports in the Message Blanks. The spoken interface puts more emphasis on the student being able to produce the correct information, compared to a menu interface, where the student would be presented with valid candidates to choose from.

The intermediate stage of coaching focused on recognizing connections between events, and groupings of events. This layer of understanding helped the student learn the best time to take actions, or to expect further input. Learning the general structure of a scenario helped the students feel more comfortable and helped them recognize and incorporate the incoming messages better. Before receiving coaching on this issue, students might plunge into issuing a command in the midst of a flurry of incoming system messages, and thus neglect to affirm the incoming messages, leading to those messages being repeated and distracting the student in the subsequent minutes. Recognizing groupings of events allowed the student to work within the overall situation in a more effective way. A second area of coaching in the intermediate stage was boundary setting, helping the student understand how to select appropriate bulkheads and decks to stop the spread of fires.

The final stage of coaching focused on strategies and priorities of damage control. The communication competence and conceptual awareness taught in the earlier stages appeared to be prerequisites to understanding how to assess situations and make choices among valid options.

The human coaching in these sessions was highly successful, based on this gradual deepening of content, from the basics of issuing commands, to recognizing hierarchy and connections, and finally to issues of strategy and prioritization.

Style of Coaching Language

The linguistic style of an automated coach will be a prominent feature of the student's experience. (Student questionnaires indicate strong reactions toward certain phrases and spoken intonations of the SCoT tutor.) Thus, modeling the coach's style on an experienced Navy DCA coach should help the coach interact in the most useful way.

In the two Navy DCA sessions, the coach always took more initiative than the students. For Student 1, the ratio of coach-to-student initiative was 63-18, and for Student 2 it was 38-26. The form of the coach's language was mostly direct commands or explanations (~50 events per session), followed by hints (~30 per session). There were instances of increasing directness, specifically, of a hint, followed by a stronger hint, followed by a direct command, as the student failed to grasp the issue. Similarly, statements were roughly twice as common as questions (~60 vs. ~30) in the coach's speech in each session. The coach delivered much more praise than criticism (~20 vs. ~5) when providing a direct assessment of the student's actions. The more frequent human characteristics of coach initiative and direct statements are simpler to develop in an automated coach, which facilitates the course of system development close to the human model.

STUDENT ATTITUDE TOWARD SYSTEM

One basic starting question is whether students would want to use an automated spoken language coaching system, while using a speech-enabled simulator. While students need not like a system to learn, if they enjoy the interaction, they may be more highly motivated to learn, and to spend more time learning (Lepper & Henderlong, 2000).

Many USNA students gave responses to the questionnaire item "What did you like the most about interacting with this system?" that indicated interest and engagement with the spoken tutor. Some selected examples are: "It gave the feel of working with a person.", "I did not have to type anything.", "I saw what I was saying pop up.", "Hands-on learning", "Seems like talking to a real person.", "Speaking with the tutor." These feelings toward the spoken tutor would likely extend to a spoken coach during the simulation as well.

Another question might be whether students would feel comfortable making spoken requests for help to an automated coach. Many other automated coaches (e.g. Constantino-Gonzalez & Suthers, 2003) provide guidance to the student when the student requests it, though the request often takes the form of pushing a button. Students are demonstrably likely to make help requests of a spoken coach, because students made such requests of Voice-Enabled DC-Train during times when DC-Train had no coach or help facility. Table 4 presents data from the Stanford experiments and some exploratory DC-Train sessions from a Repair Locker Head class at Fleet Training Center in San Diego, as well as from the currently transcribed DC-Train data from USNA. While these amounts are lower than 3% of each corpus, the figures show that some students are aware of their problems and are verbally requesting guidance even with no encouragement at all from the system to do so. Once the system can answer such requests, the success of student help requests and awareness of the help capability would likely increase the number of help requests.

Table 4. Help Requests in DC-Train Corpora

Type of question	# instances, Stanford/San Diego data	# instances, partial USNA data
What to do	5	30
How to phrase command	3	2
How to use the interface	3	0
What is happening	3	0
Request info	3	0
Express distress	2	4
Total guidance requests	20	36
Total DC-Train utterances	10,880	1,596

EFFECTIVENESS OF SYSTEM

In order for a spoken coach to be effective, its speech recognition and natural language understanding need to perform well. For the USNA experiment, the newly added trigram language model for speech recognition was not fine-tuned well enough to give a high level of accuracy for many subjects, leading to frustration. Earlier experiments with grammar-based language models had enjoyed greater accuracy. For the coach, speech recognition accuracy is particularly critical because the student is under time pressure to issue correct commands to DC-Train at the same time the student may be asking questions of the coach, or responding to the coach.

One particular characteristic of speech to the simulator that is challenging for automatic recognition is the number of word fragments; that is, words which are partially spoken, as in "pri- primary". The recognition accuracy on sentences with disfluencies is significantly lower than sentences without them (Stolcke & Shriberg, 1996). Figures on the proportions of DC-Train spoken commands with word fragments and pauses in the USNA experiment are not yet available, but in an August 2004 experiment with Stanford students 451 (7%) of a total 6734 commands to the DC-Train simulator involved word fragments, and 180 (3%) involved pauses noted by the human transcriber. This makes encouraging smooth delivery of commands important for success with the system, and a useful skill for a future DCA to have for radio communication. Human coaching was successful in focusing novices' attention on their spoken delivery; an automated coach should aim to help similarly.

One bright side for developers of spoken systems is that the current state of the art in speech recognition and natural language understanding appears to be advanced enough that spoken tutoring techniques can be studied, without the effects of speech recognition performance dwarfing the contrasts between tutoring techniques. A Stanford experiment showed no correlation between

learning gains as measured by pre- and post-tests and a subject's speech recognition performance (Pon-Barry et al. 2004b). Diane Litman (p.c.) reports similar results with the tutor ITSPOKE.

DETECTING COACHABLE PROBLEMS

Given the complexities of the damage control domain, and the limitations of any automated coach, the question arises of how tractable it is to detect problem areas for coaching in a complicated simulator like DC-Train. There are some indications from the logs of USNA cadet DC-Train sessions that coachable problems are apparent.

In the typical 15-minute long DC-Train session of these experiments, there should never be a case where the same command should be given more than once. Any instance of a repeated command likely indicates that the student is confused in some way. Either they do not realize that they have previously issued this command, or they are not aware of how long to expect the commanded action to take to complete, so they are incorrectly assuming that the command was not followed. The coach can notice when the student repeats a command, and attempt to address whatever misconception caused it. The USNA data had 164 instances of repeated commands, with a maximum of the same command repeated 3 times from 6 different subjects' sessions. *Investigate* commands were the most frequently repeated, indicating that the students may need coaching in what response to expect to this command.

Table 5. Types and Frequencies of Repeated Actions

Action repeated	Number of sessions with this action repeated
investigate	117
isolate	29
Fight fire	8
Dewater	3
Desmoke	2
Set flood boundaries	1
Set smoke boundaries	1

The student's spoken input may provide distinctive evidence of a particular problem addressed in the human coaching. If the student is looking for information to complete a command, there may be a detectable pause within the utterance before the information, as in "Send repair [pause] three." Examination of Sphinx recognizer (Huang et al., 1993) forced alignments on Stanford experiment data using the transcripts from human transcribers indicated that the pauses detected automatically by Sphinx appeared to match the pauses detected by humans fairly well. This pause detection capability can be integrated into the coaching and tutoring system, based on the speech recognition string from the standard system speech recognizer, Nuance.

Another issue from the human coaching sessions which can be detected in USNA sessions is dealing with the communication protocol a DCA should follow. USNA cadets acknowledge incoming messages 2408 times, for an average of 4 per session. Almost half the sessions have no acknowledgements at all, indicating that the cadets needed to acknowledge more messages. In terms of the *interrogative* protocol, in which some other agent requests more information of the DCA, there were 4323 instances of USNA cadets receiving *interrogative* requests, for an average of 8 per DC-Train simulator session. 37% of the sessions involved no *interrogative* requests. The transcripts will reveal how frequently the cadets responded successfully.

Finally, another issue in deciding whether the coach should act based on seeing some error, compared to its knowledge of appropriate actions for the situation, is whether the student is likely to persist with an error, or to figure it out due to interacting with the simulator and noticing the effects of the actions. Data from the USNA experiment indicate that students' performance in their first DC-Train session in a particular area predicts their performance in that same area in the second session. Since many of these subjects had tutoring between the two DC-Train sessions, looking at the correlation between these separate sessions may not be as helpful as looking at DC-Train actions with no intervening tutoring.

Within a single session, the issue is whether an initial small number of actions can determine that a student is likely to have many future problems in a certain area. If a single action of each type could predict the later percentage correct in actions of that type, the coach would be able to work with very little data. If, say, three actions of a certain type are necessary for predictive value, the coach can still diagnose the student's problem areas fairly quickly. Results from the USNA data are given in the table below, indicating some success in predicting later problem areas. In this domain, errors appear likely to persist in students' interactions with a simulator, if they are not corrected. An automated coach and tutor can boost the effectiveness of the simulator in providing a rich array of learning experiences for the student DCA.

Table 6. Initial Actions Predicting Later Performance in USNA data

Num. of actions as predictor	Pearson corr. w/ later % correct	Signifi cance	Sig. Level (2-tail)	N
1	.116	.030	.05	348
3	.232	.001	.01	202

RELATED WORK

The domain of the verbal coach in Roberts, Pioch & Ferguson (2000), maneuvering unmanned underwater vehicles, involves less speech, and, unlike in DC-Train, the opportunity to correct continuous actions, like steering too far left.

Johnson, Rickel, & Lester (2000) argue for the value of animated pedagogical agents for interactive learning environments. As reviewed by Gulz (2004), these characters may have benefits in the areas of motivation, communication, personal relationships, and stimulating certain activities, though Gulz characterizes the evidence for these benefits as scattered and ambiguous. SCoT focuses on the role of the natural language interface, separate from the role of a sympathetic virtual character, and aims to support the functions of a mentor in being responsive and providing information, without situating the coach within a visual character.

CONCLUSION

The domain of damage control, as experienced with the Voice-Enabled DC-Train simulator, provides an important test case for the effectiveness of spoken natural language coaching in a fast-paced, intense, highly verbal situation. Human spoken coaching for DC-Train is effective, and provides a structure of gradually deepening content that can be replicated in an automated coach. A coach that analyzes the characteristics of a student's speech to the simulator has particular advantages in detecting areas of hesitation and guiding the student toward using the correct communication protocol, which should contribute to the student's situational awareness.

ACKNOWLEDGEMENTS

This work is supported by the Department of the Navy under research grants N000140010660, a multi-disciplinary university research initiative on natural language interaction with intelligent tutoring systems, and N000140510144, on Spoken Language Coaching During Dynamic Problem Solving.

REFERENCES

Aleven, V. (Ed.) (2001). *Workshop on Tutorial Dialogue Systems*, San Antonio, TX, May. International Society of Artificial Intelligence in Education.

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

Black, A. W. and Taylor, P. A. (1997). *The Festival Speech Synthesis System: System documentation.* Technical Report TR-83, Human Communication Research Centre, Univ. of Edinburgh, Scotland.

Bulitko, V., & Wilkins., D. C. (1999). Automated instructor assistant for ship damage control. In *Proceedings of the Eleventh Conference on Innovative Applications of Artificial Intelligence, IAAI-99,* (pp. 778-785).

Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science, 13*, 145-182.

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

Collins, A., Brown, J.S., & Newman, S.E. (1987). Cognitive Apprenticeship: Teaching the Craft of Reading, Writing and Mathematics. In L.B. Resnick (Ed.), *Cognition and Instruction: Issues and Agendas.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Constantino-Gonzalez, M. A. and Suthers, D. D. (2003). Automated Coaching of Collaboration based on Workspace Analysis: Evaluation and Implications for Future Learning Environments. In *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03).* IEEE.

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 the 31st Annual Meeting of the Association for Computational Linguistics* (pp. 54-61).

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

Evens, M., & Michael, J. (in press, 2005). *One-on-One Tutoring by Humans and Computers*. Lawrence Erlbaum Associates. Mahwah, NJ.

Gertner, A. & VanLehn, K. (2000). Andes: A Coached Problem Solving Environment for Physics. In G. Gauthier, C. Frasson & K. VanLehn (Eds), *Intelligent Tutoring Systems: 5th International Conference*, (pp. 133-142). Berlin: Springer (Lecture Notes in Computer Science, 1839).

Graesser, A.C., VanLehn, K., Rose, C., Jordan, P., & Harter, D. (2001). Intelligent tutoring systems with conversational dialogue. *AI Magazine*, *22*, 39-51.

Gruenstein, A. (2002). *Conversational interfaces: A domain-independent architecture for task-oriented dialogues.* Unpublished M.S. Thesis, Stanford.

Gulz, A. (2004). Benefits of Virtual Characters in Computer Based Learning Environments: Claims and Evidences. *International Journal of Artificial Intelligence in Education, 14*, 313-334.

Huang, D., Alleva, F., Hon, H. W., Hwang, M. Y., Lee, K. F., & Rosenfeld, R. (1993). The Sphinx-II Speech Recognition System: An Overview. *Computer, Speech and Language*, 7, 137-148.

Johnson, W. L., Rickel, J. W., and Lester, J. C. (2000). Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. *International Journal of Artificial Intelligence in Education, 11*, 47-78.

Katz, S., O'Donnell, G., and Kay, H. (2000). An Approach to Analyzing the Role and Structure of Reflective Dialogue. *International Journal of Artificial Intelligence in Education, 11*, 320-343.

Koedinger, K.R., Anderson, J.R., Hadley, W.H. and 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., Gruenstein, A., & Peters, S. (2002). Collaborative activities and multitasking in dialogue systems. In C. Gardent (Ed.), *Traitement Automatique des Langues (TAL, special issue on dialogue), 43(2)*, 131-154.

Lepper, M. R. & Henderlong, J. (2000). Turning "play" into "work" and "work" into "play": 25 years of research in intrinsic versus extrinsic motivation. In C. Sansone & J. M. Harackiewicz (Eds.), *Intrinsic and extrinsic motivation: The search for optimal motivation and performance*, 257-307. San Diego: Academic Press.

Lesgold, A. LaJoie, S., Bunzo, M., and Eggan, G. (1988). *SHERLOCK: A Coached Practice Environment for an Electronics Troubleshooting Job.* Univ. of Pittsburgh Learning Research and Development Center, Pittsburgh, PA.

Litman, D. J. & Forbes-Riley, K. (2004). Predicting Student Emotions in Computer-Human Tutoring Dialogues. In *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Barcelona, Spain.

McFarlane, D. C. & Latorella, K. A. (2002) The Scope and Importance of Human Interruption in Human-Computer Interaction Design. *Human-Computer Interaction*, *17,* 1-61.

Peters, S., Bratt, E. O., Clark, B., Pon-Barry, H., & Schultz, K. (2004) Intelligent Systems for Training Damage Control Assistants. In *Proceedings of I/ITSEC 2004*. Orlando, Florida.

Pon-Barry, H. (2004). In search of Bloom's missing sigma: Adding the conversational intelligence of human tutors to an intelligent tutoring system. Unpublished M.S. Thesis, Stanford University.

Pon-Barry, H., Clark, B., Schultz, K., Bratt, E. O. & Peters, S. (2004a) Advantages of Spoken Language in Dialogue-based Tutoring Systems. In J. Lester, R. M. Vicari, & F. Paraguaçu (Eds.)., *Proc. of ITS 2004, 7th International Conference on Intelligent Tutoring Systems.* (pp. 390-400.) Maceiò, Brazil. Lecture Notes in Computer Science 3220. Springer.

Pon-Barry, H., Clark, B., Bratt, E. O., Schultz, K. & Peters, S. (2004b) Evaluating the Effectiveness of SCoT: a Spoken Conversational Tutor. In *Proc. of ITS 2004 Workshop on Dialogue-based Intelligent Tutoring Systems: State of the Art and New Research Directions (pp. 23-32).* Maceio, Brazil.

Roberts, B., Pioch, N., & Ferguson, W. (2000). Verbal Coaching During a Real-time Task. *International Journal of Artificial Intelligence in Education , 11*, 377-388.

Sanders, G. A. (1995). *Generation of Explanations and Multi-Turn Discourse Structures in Tutorial Dialog, Based on Transcript Analysis.* Ph.D. thesis, Illinois Institute of Technology.

Schultz, K., Bratt, E., Clark, B., Peters, S., Pon-Barry, H., & Treeratpituk, P. (2003). A scalable, reusable spoken conversational tutor: SCoT. In *AIED 2003 Supplementary Proceedings: 2003 Workshop on Tutorial Dialogue Systems: With a View Towards the Classroom* (pp. 367-377).

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

Stolcke, A. & Shriberg, E. (1996). Statistical language modeling for speech disfluencies. In *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, *vol. 1* (pp. 405-408). Atlanta, GA.