Spoken Dialogue for Simulation Control and Conversational Tutoring

Elizabeth Owen Bratt
CSLI,
Stanford University,
Stanford, CA 94305
ebratt@csli.stanford.edu

Karl Schultz
CSLI,
Stanford University,
Stanford, CA 94305
schultzk@csli.stanford.edu

Brady Clark
CSLI,
Stanford University,
Stanford, CA 94305
bzack@csli.stanford.edu

Abstract

This demonstration shows a flexible tutoring system for studying the effects of different tutoring strategies enhanced by a spoken language interface. The hypothesis is that spoken language increases the effectiveness of automated tutoring. The domain is Navy damage control.

1 Technical Content

This demonstration shows a flexible tutoring system for studying the effects of different tutoring strategies enhanced by a spoken language interface. The hypothesis is that spoken language increases the effectiveness of automated tutoring. Our focus is on the SCoT-DC spoken language tutor for Navy damage control; however, because SCoT-DC performs reflective tutoring on DC-Train simulator sessions, we have also developed a speech interface for the existing DC-Train damage control simulator, to promote ease of use as well as consistency of interface.

Our tutor is developed within the Architecture for Conversational Intelligence (Lemon et al. 2001). We use the Open Agent Architecture (Martin et al. 1999) for communication between agents based on the Nuance speech recognizer, the Gemini natural language system (Dowding et al. 1993), and Festival speech synthesis. Our tutor adds its own dialogue manager agent, for general principles of conversational intelligence, and a tutor agent, which uses tutoring strategies and tactics to plan out an appropriate review and react to the student's answers to questions and desired topics.

The SCoT-DC tutor, in Socratic style, asks questions rather than giving explanations. The tutor has a repertoire of hinting tactics to deploy in response to student answers to questions, and identifies and discusses repeated mistakes. The student is able to ask "why" questions after certain tutor explanations, and to alter the tutorial plan by requesting that the tutor skip discussion of certain topics. In DC-Train, the system uses several windows to provide information graphically, in addition to the spoken messages. In SCoT-DC, the Ship Display from DC-Train is used for both multimodal input and output.

Both DC-Train and SCoT-DC use the same overall Gemini grammar, with distinct top-level grammars producing appropriate subsets for each application. Our Gemini grammar currently has 166 grammar rules and 811 distinct words. In a Nuance language model compiled from the Gemini grammar (Moore 1998), different top-level grammars are used in SCoT-DC to enhance speech recognition based on expected answers.

2 Performance Assessment

Experiments to assess the effectiveness of SCoT-DC tutoring are underway in March 2004, with 15 subjects currently scheduled. In July 2003, students in the Repair Locker Head class at the Navy Fleet Training Center in San Diego ran 12 sessions with DC-Train. Sessions ranged from 1 to 65 user utterances, with an average of 21. The average utterance length was 7 words. In speech recognition, about 22% of utterances were rejected, and the sentences with a recognition hypothesis had a word error rate of 27%. The transcribed data, combined with developer test run data, gave us 327 unique out-of-grammar sentences. Of these, we found 79 examples where the automatic Nuance endpointing cut off an utterance too early, and 20 examples of disfluent speech. 118 sentences were determined to be potentially useful phrasings to add to the grammar, while 73 sentences were found to lie outside the scope of the application.

To address these issues, we have added new phrasings to the grammar. We also intend to use Nuance's Listen & Learn offline grammar adaptation tool, to give higher probabilities to likely sentences while retaining broad grammar-based coverage. We may also adjust endpointing time, based on partial speech recognition hypothesis, to give extra time to the kinds of sentences typically occurring with more internal pauses. Disfluencies may decrease as users become more familiar with DC-Train and SCoT-DC during the comparatively longer use expected from each user in a typical tutoring session

The graphical interface for the DC-Train simulator is shown in Figure 1.

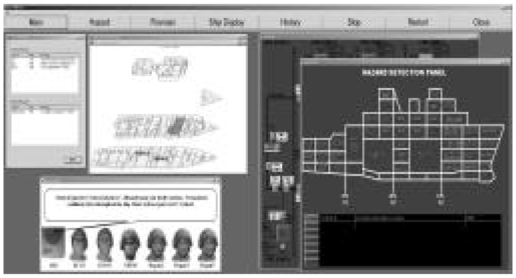

Figure 1: DC-Train simulator GUI

Each window on the screen is modeled on a source of information available to a real-life DCA on a ship, including as a detailed drawing of the several hundred compartments on the ship, a record of all communications to and from the DCA, a hazard detection panel showing the locations of alarms which have occurred, and a panel showing the firemain, i.e. the pipes carrying water throughout the ship, and the valves and pumps controlling the flow of the water. The window depicting heads represents the other personnel in the same room as the DCA, who are available to receive and transmit messages.

While in the original version of DC-Train, the DCA's orders and communications to other personnel on the ship took place through a menu system, this demo presents the newer spoken dialogue interface. Spoken commands take the form of actual Navy commands, thus enabling the Navy student to train in the same manner as they would perform these duties through radio communications on a ship.

The user clicks a button to begin speaking, and the speech is recognized by Nuance, using a grammar-based language model automatically derived from the Gemini grammar used for parsing and interpretation of the commands. A dialogue manager then maps the Gemini logical forms into DC-Train commands. To allow the student to monitor the success of the speech recognizer, the text of the utterance is displayed. Responses from the simulated personnel are spoken by Festival speech synthesis, and also displayed as text on the screen.

Most spoken interactions with DC-Train involve the student DCA giving single commands without any use of dialogue structure; however, the system will query the student for missing required parameters of commmands, such as the repair team who is to perform the action, or the number of the pump to start on the firemain. If the student does not respond to these queries, the system will provide the context of the command missing the parameter as part of a more informative request. The student retains the ability to issue other commands at this time, and need not respond to the system if there is a more pressing crisis elsewhere.

At the end of a DC-Train session, the student can then receive customized feedback and tutoring from SCoT-DC, based on a record of the student's actions compared to what an expert DCA would have done at each point, based on rules accounting for the state of the simulation. The goal of the tutorial interaction is to identify and remediate any gaps in the student's understanding of damage control doctrine, and to improve the student's performance in issuing the correct commands without hesitation.

The graphical interface to the SCoT-DC tutor is shown in Figure 2.

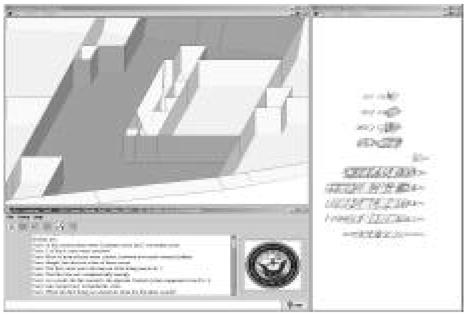

Figure 2: ScoT-DC tutor GUI

SCoT-DC uses two instances of the Ship Display from DC-Train, seen in Figure 3, one to give an overall view of the ship and one to zoom in on affected compartments, with color indicating the type of crisis in a compartment and the state of damage control there. The student can click on a compartment in the Ship Display as a way of indicating that compartment to the system. The automated tutor and the student communicate through speech, while the lower window displays the text of both sides of the interaction, and permits the user to scroll back through the entire tutorial session.

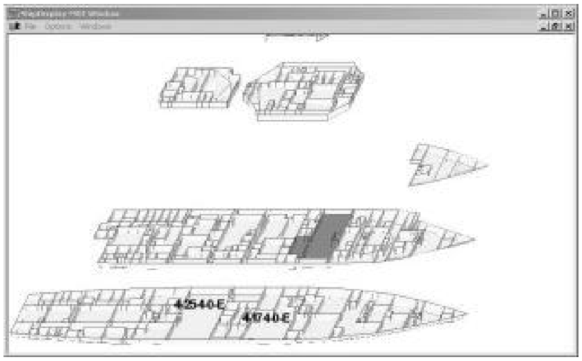

Figure 3: Highlighted Compartment

The tutor can also display bulkheads used to set boundaries for firefighting, as in Figure 4.

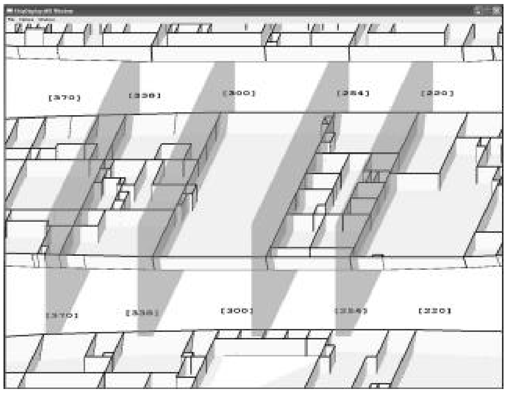

Figure 4: Highlighted Bulkhead Walls

A third kind of graphical information that the tutor may convey to the student involves regions of jurisdiction for repair teams, shown in Figure 5.

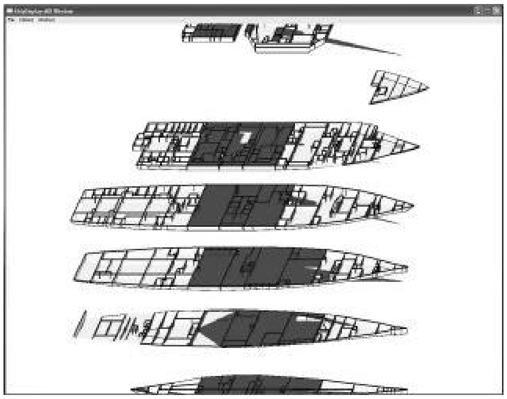

Figure 5: Repair Team Jurisdiction Regions

As in DC-Train, the student clicks to begin speaking, then Nuance speech recognition provides a string of words to be interpreted by a Gemini grammar. Also as in DC-Train, responses from the tutor are synthesized by Festival, although the tutor speaks with a more natural voice provided by FestVox limited domain synthesis, in which large units of the tutor's utterances may be taken from prompts recorded for this application.

Interpretation of the Gemini interpreted forms is handled by a more complex dialogue manager in SCoT-DC than in DC-Train, with a structured representation of the dialogue, which is used to guide the system's use of discourse markers, among other things. The dialogue is mainly driven by the tutor agent's strategies, though the student can request to move on to future topics without completing the current discussion, and also ask a "Why" question after some explanations.

Tutorial strategies generally guide the overall path of the conversation, such as choosing which crises to discuss based on the errors made by the student. Tutorial tactics apply at a lower-level throughout the dialogue, for example, when a student gives an incorrect answer, the tutor will give a general hint and repose the question. If the student answers incorrectly a second time the tutor will give a more specific hint and ask the question again. If the student fails a third time the tutor will give the correct answer, and proceed.

Running a full DC-Train scenario takes 20-40 minutes, and has the flavor of the following excerpt:

[buzzing alarm goes off, it is a fire alarm]

DCCO: Fire in compartment 2-78-01-L.
Student: Net80 to repair locker 2, investigate compartment 2-78-01-L.
Repair Locker 2: Reports, fire in compartment 2-78-01-L.
Repair Locker 2: Reports, smoke in compartment 2-78-01-L.
Student: Net80 to repair locker 2, fight the fire in compartment 2-78-01-L.
Student: Net80 to repair locker 2, set fire and smoke boundaries on primary forward 78, primary aft 126, secondary forward 42, secondary aft 174, above 1, below 2.

A reflective dialogue with the tutor will takes around 10 minutes. The following gives a sample of the kind of tutorial interaction.

Tutor: Hello, we are about to review your session from earlier today.

Tutor: All right, are you ready?
Student: Yes.
Tutor: In this session there were 5 primary crises and 1 secondary crisis.
Tutor: 5 of the 6 crises were resolved.
Tutor: Most of your actions were correct, however you made several mistakes.
Tutor: OK, let's discuss a few of those crises.
Tutor: The first crisis was a fire here in Crew Living space No. 1. [centers on this compartment]
Tutor: This fire was not contained early enough.
Tutor: As a result, the fire spread to the adjacent Combat system equipment room No. 2. [highlights 2nd compartment]
Tutor: OK, let's review how to handle this crisis. [removes highlight from 2nd compartment]
Tutor: What's the first thing you should do when the fire alarm sounds?
Student: Send repair two to investigate.
Tutor: Yes, that's right.
Tutor: However, you failed to do this during the session.
Tutor: This is a mistake you made more than once.
Tutor: Remember, alarms need to be verified first.
Tutor: OK, suppose that there is a fire.
Tutor: What's the next step?
Student: Set fire and smoke boundaries.
Tutor: Yes. That's what you did during the session.
Tutor: OK, what else needs to be done?
Student: The repair team should fight the fire.
Tutor: No, you missed a step.
Tutor: Remember, the crisis might interfere with other ship systems.
Tutor: Let's try again.
Tutor: What else needs to be done?
Student: I don't know.
Tutor: Well.
Tutor: You should order repair five to mechanically and electrically isolate the compartment.

A video clip of an older version of the ScoT-DC system is available at http://www-csli.stanford.edu/semlab/muri/November2002Demo.html

3 Acknowledgements

This work is supported by the Department of the Navy under research grant N000140010660, a multidisciplinary university research initiative on natural language interaction with intelligent tutoring systems.

3.1 References

A. Black and K. Lenzo, 1999. Building Voices in the Festival Speech Synthesis System (DRAFT) Available at http://www.cstr.ed.ac.uk/projects/festival/papers.html.

A. Black and P. Taylor. 1997. Festival speech synthesis system: system documentation (1.1.1). Technical Report Technical Report HCRC/TR-83, University of Edinburgh Human Communication Research Centre.

V. V. Bulitko and D. C. Wilkins. 1999. Automated instructor assistant for ship damage control. In Proceedings of AAAI-99.

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

Oliver Lemon, Alexander Gruenstein, and Stanley Peters. 2002. Collaborative Activities and Multi-tasking in Dialogue Systems , *Traitement Automatique des Langues* (TAL), 43(2):131-154, special issue on dialogue.

D. Martin, A. Cheyer, and D. Moran. 1999. ``The open agent architecture: A framework for building distributed software systems," Applied Artificial Intelligence, v.13:91-128.

Robert C. Moore. 1998. Using Natural Language Knowledge Sources in Speech Recognition." Proceedings of the NATO Advanced Study Institute.

Karl Schultz, Elizabeth Owen Bratt, Brady Clark, Stanley Peters, Heather Pon-Barry, and Pucktada Treeratpituk. 2003. *A Scalable, Reusable Spoken Conversational Tutor: SCoT.* In *AIED 2003 Supplementary Procdgs.* (V. Aleven et al eds). Univ. of Sydney. 367-377.