A Multimodal Intelligent Tutoring System for Shipboard Damage Control

Brady Clark, John Fry, Matt Ginzton, Stanley Peters,
Heather Pon-Barry & Zach Thomsen-Gray
Stanford University
Center for the Study of Language Information
Stanford CA 94305-4115 USA
{bzack,fry,mginzton,peters,ponbarry,ztgray}@csli.stanford.edu

1 Abstract

We discuss a spoken dialogue system which engages students in multimodal tutorial conversations in order to evaluate their performance in a training simulator. The dialogue system is implemented using Nuance, Gemini, and Festival within the Open Agent Architecture. This paper focuses on our critique planner which implements the insights of the joint activity theory of dialogue (Clark 1996).

1.1 Keywords

tutorial dialogue, intelligent tutoring system, multimodal, joint activities, damage control

2 Introduction

Existing intelligent tutoring systems (e.g., AutoTutor [6], Andes-Atlas [10]) are based on the theoretical assumption that tutorial dialogue is different than conversational dialogue [9]. The tack taken by this approach has been to catalogue the dialogue acts (Socratic questions, hinting) found in tutorial dialogue. Dialogue acts are realized in tutoring systems (e.g., AutoTutor) as states in finite state automata which govern the dialog move option space for any input.

This approach does not explain why dialogue structure might vary across tasks and genres. It could be argued that tutorial dialogue is not really that different from other types of conversation. Rather, building on work of Herbert Clark [2, 3], apparent violations of conversational rules in tutorial dialogue (e.g., tutors ask questions they know the answer to) are really just a consequence of the activity— tutoring— that tutorial dialogue works in service of. Once one has a good model of the activity of tutoring, we can explain why conversation rules seem to be lifted or loosened at certain points in tutorial dialogue.

In this paper we describe a system we are developing for automated critiquing of student performance on a damage control training simulator (DC-Train, [1]) and show how it incorporates the theoretical insights of Clark's work on dialogue. In contrast to the text-based systems described above, our system is a *spoken* dialogue system and uses 'deep' processing techniques (our system utilizes Gemini, a symbolic grammar). These properties both jibe well with current research on dialogue and offer greater naturalness to the user.

3 Theoretical Background

Joint Activities *Joint activities* [2, 3] are those activities in which participants have to coordinate their individual actions to succeed; e.g. moving a desk. Joint activities like moving a desk can be divided into two types of actions: *basic* and *coordinating*. Basic actions for moving a desk include those illustrated in Figure 1 (example from [3]).

People manage joint activities by coordinating with each other using language and other signals. To illustrate, consider the following coordinating actions used when moving a desk:

Kim: Sandy, could you help me move the desk?

Sandy: Sure, be glad to.

Kim: (*Moves to one end of the desk*) Could you take that end (*pointing*)?

Sandy: Okay (*moving to other end of the desk*).

Kim: (*Looks at Sandy to see if she is ready.*)

A's actions	B's actions
A moves to end of the desk	B moves to other end of desk
A lifts her end	B lifts her end
A moves to second room	B moves to second room
A sets down her end	B sets down her end
A moves away from her end	B moves away from her end

Figure 1: Basic actions: moving a desk

Sandy: Okay ... now (*lifting her end at "now"*).

Kim: (*Lifts her end at "now".*)

Dialogue itself is also divided into basic and coordinating actions. Basic actions in dialogue are what people use to coordinate joint activities like moving a desk (*Could you help me move the desk?*). Coordinating actions in dialogue are what people use to coordinate the use of language itself.

For example, the utterance *How do you think you did?* in a debrief session between an instructor and a student consists of a basic action (the utterance is a question) and coordinating actions (the utterance establishes various preconditions; e.g. "Who is the speaker and who is the addressee?", "What is the joint activity to be carried out?", etc.).

Tutoring as a joint activity Empirical work by Barbara Fox [5] and others has shown that while tutorial dialogue shares certain discourse moves with other types of dialogues (e.g., questions and answers), tutorial dialogue has some special properties (e.g., tutors ask questions they know the answer to). The joint activity approach to dialogue argues that the content of tutorial dialogue is driven by both the problem space and the nature of tutorial help.

The joint activity approach treats tutoring as a type of joint activity falling somewhere between spontaneous conversation and classroom discourse. Like other joint activities, tutoring involves basic and coordinating actions. Basic actions in tutoring involve solving problems; e.g. physics problems. Non-linguistic and linguistic actions like hinting, asking questions and giving answers coordinate the basic actions in tutoring. To illustrate, consider how students verify understanding in a tutoring context [5]. Students verify understand via linguistic actions (e.g., using continuers like *mhm* or completing the tutor's current clause) or non-linguistic actions (e.g., nodding or staring blankly). These signals are also used by tutors to diagnose student understanding.

The joint activity approach to dialogue suggests that an intelligent tutoring system should include, at least, the following features:

- A multimodal interface which coordinates linguistic input and output (e.g., speech) with non-linguistic input and output (e.g., the user can indicate a point on a map with a mouse click or the system can illuminate a point on a map)
- Precise and reliable meaning representations which allow us to identify the discourse moves (e.g., an answer to a question) given linguistic or non-linguistic input

In the next section we describe a system which includes these two features.

4 Dialogue System for Training in Shipboard Damage Control

In this section we describe an intelligent tutoring system we are developing for automated critiquing of student performance on a damage control simulator. The simulator is DC-TRAIN [1], an immersive, multimedia training environment for damage control. DC-TRAIN's training scenarios simulate a mixture of physical phenomena (e.g., fire) and personnel issues (e.g., casualties). Our current tutoring system is restricted to fire damage scenarios only.

To facilitate the implementation of multimodal, mixed-initiative tutoring interactions, we implemented our system within the Open Agent Architecture (OAA) [8]. OAA is a framework for coordinating multiple asynchronous communicating processes. The core of OAA is a 'facilitator' which manages message passing between a number of software agents that specialize in certain tasks (e.g., speech recognition). Our system uses OAA to coordinate the following five agents:

1. The **Gemini** NLP system [4]. Gemini uses a single unification grammar both for *parsing* strings of words into logical forms (LFs) and for *generating* sentences from LF inputs. The use of a symbolic grammar enables us to provide precise and reliable meaning representations (in contrast to 'shallow' statistical approaches like Latent Semantic Analysis).

2. A **Nuance** speech recognition server, which converts spoken utterances to strings of words. The Nuance server relies on a language model, which is compiled directly from the Gemini grammar, ensuring that every recognized utterance is assigned an LF.

3. The **Festival** text-to-speech system, which 'speaks' word strings generated by Gemini.

4. A **Dialogue Manager** which coordinates inputs from the user, interprets the user's dialogue moves, updates the dialogue context, and delivers speech and graphical outputs to the user.

5. A **Critique Planner**, described below.

Agents 1-3 are 'off-the-shelf' dialogue system components (apart from the Gemini grammar, which must be modified for each application). We implemented agents 4 and 5 in Java specifically for this application. Variants of this OAA/Gemini/Nuance architecture have been deployed successfully in other dialogue systems; e.g., an unmanned helicopter interface [7].

Critique Planner Each session with DC-TRAIN produces a *session transcript*; i.e., a time-stamped record of every computer- and student-initiated event that occurred during the session. These transcripts are the input to our post-session Critique Planner (CP).

The CP plans a post-session tutorial dialogue in two steps.

- **Step 1**: An Expert Session Summary Graph (ESSG) is created from the Session Transcript. An ESSG is a tree whose parent nodes represent damage events and whose leaves represent the student-initiated actions in response to damage events. Each node falls into one of three classes: (i) correct actions, (ii) incorrect actions, and (iii) omitted actions.

- **Step 2**: A dialogue move graph (DMG) is generated from the ESSG. A DMG encodes all possible configurations of a dialogue structure that can be handled by the system.

Generating a DMG from an ESSG requires a tutoring strategy. The strategy we adopted (based on videotapes of 15 actual DC-TRAIN post-session critiques conducted by instructors at SWOS in Newport, RI) can be outlined as follows:

1. Summarize the results of the simulation (e.g., the final condition of the ship).

2. For each major damage event in the ESSG:

   (a) Ask the student to review his actions, correcting recollections if necessary.

   (b) Evaluate the correctness of each student action.

   (c) If the student committed errors, ask him how these could have been avoided, and evaluate the correctness of his responses.

3. Finally, review each type of error that arose in step (2c).

Figure 2 illustrates (part of) a sample dialogue with the system. Note that the multimodal interface (illustrated in Figure 3) coordinates linguistic output ("The first fire was in the engineering storeroom") with non-linguistic output (indicated in the dialogue in Figure 2 by "≪...≫"; e.g., "≪highlights fire 1≫").

The architecture of the critique planner incorporates the basic and coordinating action distinction of the joint activity theory of dialogue described in Section 2. Basic actions are given by ESSGs which organize the content of the tutorial dialogue. DMGs coordinate the basic tutoring actions (e.g., summarizing, asking questions, evaluating correctness of an action, etc.). The dialogue itself is coordinated by linguistic and non-linguistic actions; e.g., the computer tutor outputs *Yes* or *Okay* to convey that the system has understood the student's input.

Acknowledgments

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.

T=tutor/system, S=student

T: In this scenario, there were 4 fires. You put out 3 of them. Would you like to review this session or start another session?

S: Review this session.

T: ≪highlights fire 1≫ The first fire was in the engineering storeroom. This fire went out after 3 minutes. What actions did you take?

S: I sent a team to set fire boundaries.

T: Yes, that's correct. ≪shows fire 1 actions≫ You ordered Repair 5 to fight the fire and set fire boundaries. You handled this one well. Let's move on. ≪highlights fire 2≫ The second fire was in the ammo strikedown trunk. This fire went out after 4 minutes. What actions did you take?

S: I think I flooded the compartment.

T: No, that's incorrect. ≪shows fire 2 actions≫ You ordered Repair 5 to fight the fire and set fire boundaries. This was an error. . What did you do wrong?

S: I don't know.

T: Repair 5 was still fighting the earlier fire. A repair team can only be sent to one compartment at a time. What should you have done instead?

S: I should have sent repair 2 to set fire boundaries.

T: Yes, that's correct. Let's move on.
...

T: Now let's summarize. In this scenario, there were 4 fires. You put out 3 of them. There were two key points that arose in this session. First, a repair team can only be sent to one compartment at a time. Second, you must receive permission from the commanding officer before flooding a magazine. Would you like to review this session again, try another scenario, or exit the simulator?

S: Exit.

Figure 2: Sample dialogue

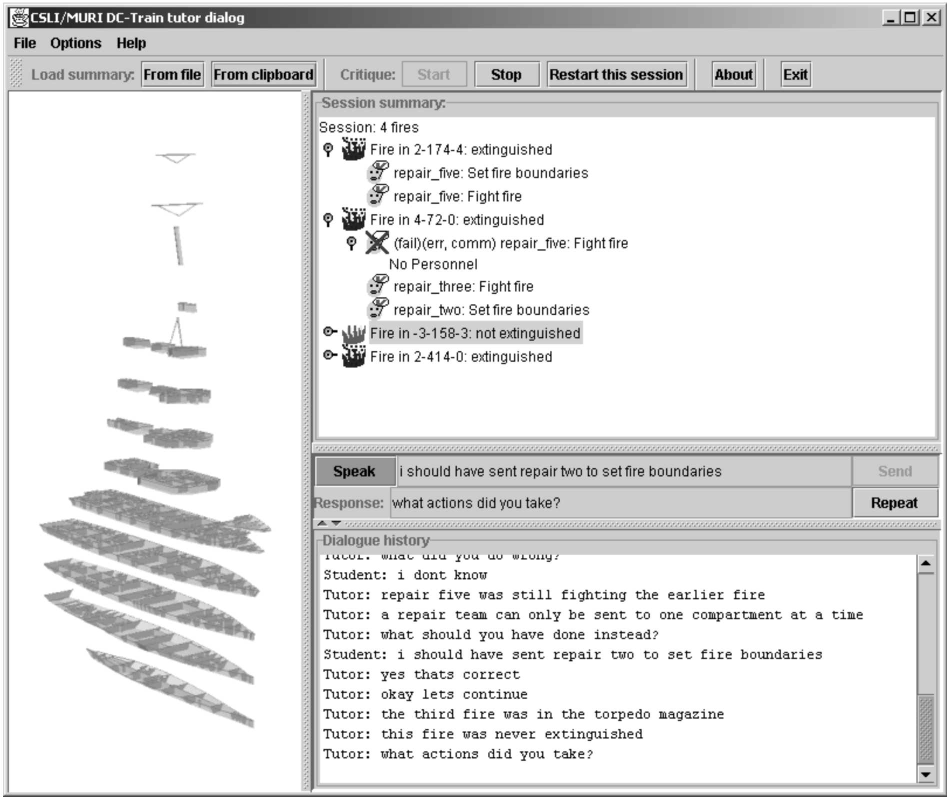

Figure 3: Screen shot of post-session tutorial dialogue system

References

[1] Bulitko, V.V. and D.C. Wilkins. Automated instructor assistant for ship damage control. *Proceedings of AAAI-99* (1999).

[2] Clark, H.H. Clark. *Using Language*. Cambridge University Press (1996).

[3] Clark, H.H. Clark. Pragmatics of Language Performance. *Handbook of Contemporary Pragmatics*(forthcoming).

[4] Dowding, J., J. Gawron, D. Appelt, J. Bear, L. Cherny, R.C. Moore and D. Moran. Gemini: A natural language system for spoken-language understanding. *Proceedings of the ARPA Workshop on Human Language Technology*(1993).

[5] Fox, B. *Human Tutorial Dialogue*. Lawrence Erlbaum (1993).

[6] Graesser, A., K. Wiemer-Hastings, P. Wiemer-Hastings, R. Kreuz and the Tutoring Research Group. AutoTutor: a simulation of a human tutor. *Journal of Cognitive Systems Research* 1 (2000), pp.35-51.

[7] Lemon, O., A. Bracy, A. Gruenstein and S. Peters. A multi-modal dialogue system for human-robot conversations. *Proceedings of NAACL* (2001).

[8] Martin, D., A. Cheyer and D. Moran. The Open Agent Architecture: a framework for building distributed software systems. *Applied Artificial Intelligence* 13, 1-2 (1999).

[9] Person, N., R. Kreuz, R. Zwaan and A.C. Graesser. Pragmatics and Pedagogy: Conversational Rules and Politeness Strategies May Inhibit Effective Tutoring. *Cognition and Instruction* 13,2 (1995), pp.161-188.

[10] Rosé, C., P. Jordan, M. Ringenberg, S. Siler, K. VanLehn, and A. Weinstein. Interactive Conceptual Tutoring in Atlas-Andes. *AI in Education* (2001).