Advantages of Spoken Language Interaction in Dialogue-based Intelligent Tutoring Systems

Heather Pon-Barry, Brady Clark, Karl Schultz, Elizabeth Owen Bratt, and Stanley Peters

Center for the Study of Language and Information
Stanford University
210 Panama Street
Stanford, CA 94305-4115, USA
{ponbarry, bzack, schultzk, ebratt, peters}@csli.stanford.edu

Abstract. The ability to lead collaborative discussions and appropriately scaffold learning has been identified as one of the central advantages of human tutorial interaction [6]. In order to reproduce the effectiveness of human tutors, many developers of tutorial dialogue systems have taken the approach of identifying human tutorial tactics and then incorporating them into their systems. Equally important as understanding the tactics themselves is understanding how human tutors decide which tactics to use. We argue that these decisions are made based not only on student actions and the content of student utterances, but also on the meta-communicative information conveyed through spoken utterances (e.g. pauses, disfluencies, intonation). Since this information is less frequent or unavailable in typed input, tutorial dialogue systems with speech interfaces have the potential to be more effective than those without. This paper gives an overview of the Spoken Conversational Tutor (SCoT) that we have built and describes how we are beginning to make use of spoken language information in SCoT.

1 Introduction

Studies of human-to-human tutorial interaction have identified many dialogue tactics that human tutors use to facilitate student learning [13], [18], [11]. 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 between the category of a student utterance (e.g. partial answer, request for clarification) and the category of a tutor response (e.g. positive feedback, leading question) [23]. However, since the majority of dialogue-based ITS rely on typed student input, information from the student utterance is limited to the content of what the student typed. 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 [12]. This suggests that in order for a dia-

logue-based ITS to tailor its choice of tactics in the way that humans do, the student utterances must be spoken rather than typed.

Intelligent tutoring systems that have little to no natural language interaction have been deployed in public schools and have been shown to be more effective than classroom instruction alone [19]. However, the effectiveness of both expert and novice human tutors [3], [9] suggests that there is more room for improvement. Current results from dialogue-based tutoring systems are promising [22], [24] and 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 domains ranging from travel planning to in-car route navigation [1]. There is reason to believe that spoken dialogue tutorial systems can be just as successful.

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 [17]. In addition, a comparison of spoken vs. typed human tutorial dialogues showed that the spoken dialogues contained a higher proportion of student words to tutor words, which has been shown to correlate with student learning [25].

There are many ways an ITS can benefit from spoken interaction. One idea currently being explored is that prosodic information from the speech signal can be used to detect emotion, allowing developers to build more responsive tutoring systems [21]. Another advantage is that speech allows the student to use their hands to gesture while speaking (e.g. pointing to objects in the workspace). Finally, spoken input contains meta-communicative information such as hedges, pauses, and disfluencies, which can be used to make inferences about the student's understanding. These features of spoken language are all things that human tutors have access to when deciding which tactics to use, and that are also available to intelligent tutoring systems with spoken, multi-modal interfaces (although some are more feasible to detect than others). In this paper, we describe how an ITS can take advantage of spoken interaction, how we have begun to do this in SCoT, and the challenges we have faced.

2 Advantages of Spoken Dialogue

Spoken dialogue contains many features that human tutors use to gauge student understanding and student affect. These features include:

- hedges (e.g. "I guess I just thought that was right")
- disfluencies (e.g. "um", "uh", "What-what is in this space?")
- prosodic features (e.g. intonation, pitch, energy)
- temporal features (e.g. pauses, speech rate)

Studies in psycholinguistics have shown that when answering questions, speakers produce hedges, disfluencies, and rising intonation when they have a lower "feeling-of-knowing" [26] and that listeners are sensitive to these phenomena [4]. However, it is not entirely clear whether these generalizations apply to tutorial dialogue, and if they are present, how human tutors respond to them. In a Wizard-of-Oz style comparison of typed vs. spoken communication (to access an electronic mail system), the number of disfluencies was found to be significantly higher in speech than in typing [17]. There are no formal analyses comparing the relative frequencies of hedges, however, a rough comparison (by the author) of data from typed dialogues [2] and transcripts of spoken tutoring [10] suggests 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.

Human tutors may use the dialogue features listed above in assessing student confidence or uncertainty and in tailoring the discussion to the student's needs. In building an ITS, many of these features of spoken language can be detected, and used both in selecting the most appropriate tutoring tactic and in updating the student model.

Another benefit of spoken interaction is the ability to coordinate speech with gesture. Compared to keyboard input, spoken input has the advantage of allowing the student to use their hands to gesture (e.g. to point to objects in the workspace) while speaking. Studies have shown that speech and direct manipulation (i.e. mouse-driven input) have reciprocal strengths and weaknesses which can be leveraged in multimodal interfaces [14]. For certain types of tutoring (i.e. tutoring where the student is doing a lot of pointing and placing), spoken input and direct manipulation together may be better than just speech or just direct manipulation. Furthermore, allowing the student to explain their reasoning while pointing to objects in the GUI creates a *common workspace* between the participants [8] which helps contextualize the dialogue and facilitate a mutual understanding between the student and tutor, making it easier for the tutor to know if the student is understanding the problem correctly.

3 Overview of SCoT

Our approach is based on the assumption that the activity of tutoring is a joint activity[1] 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 [8]. 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 is developed within the Conversational Intelligence Architecture [20], a general purpose architecture which supports multi-modal, mixed-initiative dialogue.

SCoT-DC, the current instantiation of our tutoring system, is applied to the domain of shipboard damage control. Shipboard damage control refers to the task of contain-

[1] A joint activity is an activity where participants coordinate with one another to achieve both public and private goals [8]. Moving a desk, playing a duet, and shaking hands are all examples of joint activities.

ing the effects of fires, floods, 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 [5], a fast-paced, real-time, multimedia training environment for damage control. The fact that problem-solving in damage control occurs in real-time makes reflective tutorial dialogue more appropriate than tutorial dialogue during problem-solving. Because the student is not performing problem-solving steps during the dialogue, it is important for the tutor to get as much information as possible from the student's utterances. In other words, having access to both the meaning of an utterance as well as the manner in which it was spoken will help the tutor assess how well the student is understanding the material.

SCoT is composed of many separate components. The two most relevant for this discussion are the dialogue manager and the tutor. They are described in sections 3.1 and 3.2. A more detailed system description is available in [7].

3.1 Dialogue Manager

The dialogue manager handles aspects of conversational intelligence (e.g. turn management, construction of a structured dialogue history, use of discourse markers) in order to separate purely linguistic aspects of the interaction from tutorial aspects. It contains multiple dynamically updated components—the two main components are the *dialogue move tree*, a structured history of dialogue moves, and the *activity tree*, a hierarchical representation of the past, current, and planned activities initiated by either the tutor or the student. For SCoT, each activity initiated by the tutor corresponds to a tutorial goal; the decompositions of these goals are specified by activity recipes contained in the recipe library (see section 3.2).

3.2 Tutor

The tutor component contains the tutorial knowledge necessary to plan and carry out a flexible and coherent tutorial dialogue. The tutorial knowledge is divided between a *planning and execution system* and a *recipe library* (see Figure 1).

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 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 [15] allowing for automatic translation of the recipes into system activities. An example activity recipe will be shown in section 4.2.

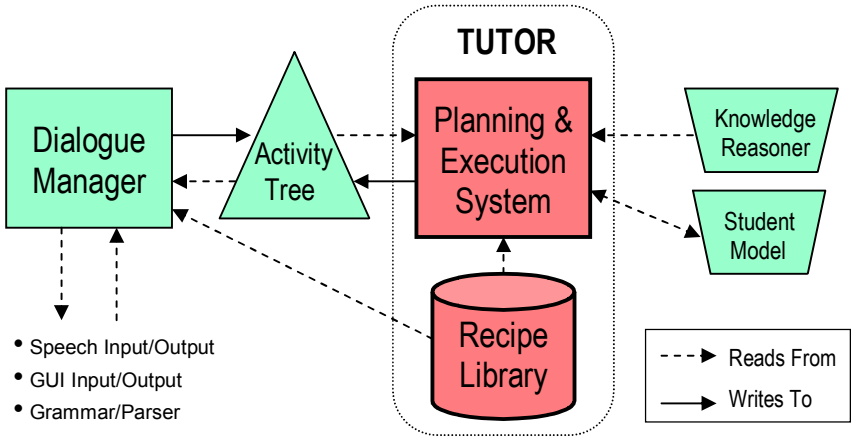

Fig. 1. Subset of SCoT architecture

Other components that the tutor makes use of are the **knowledge reasoner** and the **student model**. The knowledge reasoner provides a domain-general interface to domain-specific information; it provides the tutor with procedural, causal, and motivational explanations of domain-specific actions. The student model uses a Bayesian network to characterize the causal connections between pieces of target domain knowledge and observable student actions. It can be dynamically updated both during the problem solving session and during the dialogue.

4 Taking Advantage of Spoken Language in SCoT

4.1 Observations from human tutoring

Because spoken language interaction in tutoring systems is a relatively unexplored area, it is not clear which features of spoken language human tutors pay attention to in deciding when to use various tutorial tactics. As part of an ongoing study, we have been analyzing transcripts of human tutorial dialogue from multiple domains in order to make observations and form hypotheses about how human tutors use these features of spoken dialogue. Two such observations are described below.

One observation we have made is that if the student hedges a correct answer, the tutor will frequently paraphrase what the student said. This seems plausible because by paraphrasing, the tutor is grounding the conversation [8] while attempting to eliminate the student's uncertainty. An example of a hedged answer followed by paraphrasing is shown in Figure 2 below.

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.

Fig. 2. Excerpt from CIRCSIM corpus of human keyboard-to-keyboard dialogues [10]

Another observation we have made is that human tutors frequently refer back to past dialogue following an incorrect student answer with hedges or mid-sentence pauses. This seems plausible because referring back to past dialogue helps students integrate new information with existing knowledge, and promotes reflection, which has been shown to correlate with learning [6]. An example of an incorrect answer with mid-sentence pauses followed by a reference to past dialogue is shown in Figure 3 (each colon ':' represents a 0.5 sec pause).

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?

Fig. 3. Dialogue excerpt from Algebra corpus of spoken tutorial interaction [18]

4.2 Activity Recipes

The division of knowledge in the tutor component (between the recipe library and the planning and execution system) allows us to independently evaluate hypotheses such as the ones in section 4.1 (i.e. test whether their presence or absence affects the effectiveness of SCoT). Each hypothesis is realized by a combination of activity recipes, and the planning and execution system ensures that a coherent dialogue will be produced regardless of which activities are put on the activity tree.

An activity recipe corresponding to the tutorial goal ***discuss problem solving sequence*** is shown below. 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 activi-

ties (depending on whether the problem has already been discussed). The tutor places these activities on the activity tree, and the dialogue manager begins to execute their respective 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;
    }
}
```

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. Furthermore, the tutorial recipes are not particular to the domain of damage control; through our testing of various activity recipes we hope to get a better understanding of domain-independent tutoring principles.

4.3 Multi-modality

Another way that SCoT takes advantage of the spoken interface is through multi-modal interaction. Both the tutor and the student can interactively perform actions in an area of the graphical user interface called the *common workspace*. In the current version of SCoT-DC, the common workspace consists of a 3D representation of the ship which allows either party to zoom in or out and to select (i.e. point to) compartments, regions, and bulkheads (lateral walls of a ship). This is illustrated below in Figure 4, where the common workspace is the large window in the upper left corner.

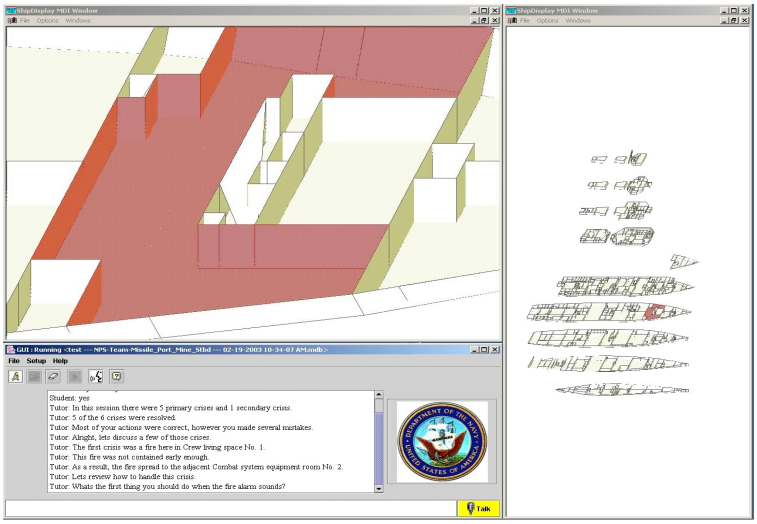

Fig. 4. Screen shot of SCoT-DC

The tutor can contextualize the problems being discussed by highlighting compartments in specific colors (e.g. red for fire, gray for smoke) to indicate the type and location of the crises. Because the dialogue in SCoT is spoken rather than typed, the student also has the ability to coordinate his/her speech with gesture. This latter coordination is an area we are currently working on, and we hope to soon support interchanges such the one in Figure 5 below, where both the tutor and student coordinate their speech with actions in the common workspace.

Tutor: If there is a fire here [highlights compartment], in compartment 1-126-0-A, which bulkheads should you set fire boundaries on?

Student: I should set primary boundaries here [selects bulkhead], and here [selects other bulkhead]...

Fig. 5. Example of coordinating speech with gesture

4.4 What we have learned

Although using spoken language in an intelligent tutoring system can bring about many of the benefits described above, it has also raised some challenges which ITS developers should be aware of.

Student affect. Maintaining student motivation is a challenge for all intelligent tutoring systems. We have observed issues relating to student affect, possibly stemming from the spoken nature of the dialogue. For example, in a previous version of SCoT, listeners remarked that repeated usage of phrases such as *You made this mistake more than once* and *We discussed this same mistake earlier* made the tutor seem overly

critical. Other (non-spoken) tutorial systems give similar types of feedback (e.g. [11]), yet none have reported this sort feedback to cause such negative affect. This suggests that users have different reactions when listening to, rather than reading, the tutor's output, and that further work is necessary to better understand this difference.

Improving Speech Recognition. We are currently running an evaluation of SCoT, and preliminary results show speech recognition accuracy to be fairly high (see section 5). However, we have learned that small recognition errors can greatly reduce the effectiveness of a tutoring session. Figure 6 shows one type of speech recognition error that occurred while evaluating SCoT-DC. The recognized phrases *ask repair two* and *the bridge can do* are sentence fragments which would never be appropriate answers to the question the tutor has just asked.

Tutor:	Which repair team is located here? [highlights compartment]
Student:	repair two (actual) / ask repair two (recognized)
Tutor:	I'm sorry, could you rephrase that?
Student:	repair team two (actual) / the bridge can do (recognized)
Tutor:	No, the correct answer is repair team two.

Fig. 6. Example of speech recognition errors

We have addressed this problem by defining distinct speech recognition language models for different tutorial contexts. If the tutor has just asked about a repair team, then the possible answers are restricted to personnel on the ship. If the tutor has just asked about what action should be taken, then the language model is restricted to verb phrase fragments describing actions. In both cases, if there is no successful recognition in the small, tailored grammar, we then back off to the whole grammar. Adapting the language model to the dialogue context in this way appears to be aiding our recognition performance significantly, in line with an 11.5% error rate reduction found in other dialogue systems [27]. Misrecognitions not only prevent the tutor from properly assessing the student's knowledge, they also cause the student to distrust information coming from the tutor, which makes it difficult to facilitate learning. Thus, taking advantage of the tutorial and dialogue context to constrain the language model can substantially benefit the overall system.

5 Conclusions & Current Evaluation of SCoT

In this paper, we argued that spoken language interaction is an integral part of human tutorial dialogue and that information from spoken utterances is very useful in building dialogue-based intelligent tutors that understand and respond to students as effectively as human tutors. We described the Spoken Conversational Tutor we have built, and described how SCoT is beginning to take advantage of features of spoken language. We do not yet understand exactly how human tutors make use of spoken language features such as disfluencies and pauses, but we are building a tutorial frame-

work that allows us to test various hypotheses, and in time reach a better understanding of how to take advantage of spoken language in intelligent tutoring systems.

We are currently evaluating the effectiveness of SCoT-DC (a version that does not yet make use of meta-communicative information or include a student model) with students at Stanford University. Preliminary quantitative results suggest that interacting with SCoT improves student learning (measured by performance in DC-Train and on a written test). Qualitatively, naïve users have found the system fairly easy to interact with, and speech recognition has not been a significant problem—preliminary results show very high recognition accuracies. Excluding out-of-grammar utterances (e.g. "request the, uh...shoot" or "oops my bad"), which account for approximately 12% of the total utterances, recognition accuracy has been approximately 0.79, or approximately 0.98 ignoring minor misrecognitions (i.e. singular vs. plural forms and *a* ↔ *the*) that do not affect the tutor's classification of the utterance. Further results will be available by the time of the conference. In addition, we are planning on running evaluations of the new version of SCoT in the near future to test the effectiveness of hypotheses about spoken language along the lines of those described in section 4.1.

Acknowledgements. This work is supported by the Office of Naval Research under research grant N000140010660, a multidisciplinary university research initiative on natural language interaction with intelligent tutoring systems. Further information is available at http://www-csli.stanford.edu/semlab/muri.

References

1. 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
2. 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.
3. Bloom, B.S. (1984). The 2 sigma problem: The search for methods of group instruction as effective one-on-one tutoring. *Educational Researcher, 13,* 4-16.
4. 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.
5. Bulitko, V., & Wilkins., D. C. (1999). Automated instructor assistant for ship damage control. In *Proceedings of AAAI-99*.
6. Chi, M.T.H., Siler, S., Jeong, H., Yamauchi, T., & Hausmann, R.G. (2001). Learning from tutoring. Cognitive Science, 25:471-533.
7. 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.
8. Clark, H.H. (1996). *Using Language.* Cambridge: University Press.
9. Cohen, P.A., Kulik, J.A., & Kulik, C.C. (1982). Educational outcomes of tutoring: A meta-analysis of findings. *American Educational Research Journal, 19,* 237-248.

10. Transcripts of face-to-face and keyboard-to-keyboard tutorial dialogues, between physiology professors and first-year students at Rush Medical College (received from M. Evens).
11. Evens, M., & Michael, J. (Unpublished manuscript). *One-on-One Tutoring by Humans and Machines*. Computer Science Department, Illinois Institute of Technology.
12. Fox, B. (1993). *Human Tutorial Dialogue*. New Jersey: Lawrence Erlbaum.
13. 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.
14. Grasso, M.A., & Finin, T.W. (1997). Task Integration in Multimodal Speech Recognition Environments. *Crossroads, 3(3),* 19-22.
15. Gruenstein, A. (2002). Conversational Interfaces: A Domain-Independent Architecture for Task-Oriented Dialogues. Unpublished M.S. Thesis, Stanford University.
16. Hausmann, R. & Chi, M.T.H. (2002). Can a computer interface support self-explaining? *Cognitive Technology, 7(1),* 4-15.
17. Hauptmann, A.G. & Rudnicky, A.I. (1988). Talking to Computers: An Empirical Investigation. *International Journal of Man-Machine Studies 28(6),* 583-604
18. 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.
19. 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.
20. 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.
21. Litman, D., & Forbes, K. (2003). Recognizing Emotions from Student Speech in Tutoring Dialogues. In *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU).*
22. Person, N.K., Graesser, A.C., Bautista, L., Mathews, E., & the Tutoring Reasearch 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.
23. Person, N.K., & Graesser, A.C. (2003). Fourteen facts about human tutoring: Food for thought for ITS developers. In *Proceedings of the AIED 2003 Workshop on Tutorial Dialogue Systems: With a View Towards the Classroom.*
24. Rosé, C., Jordan, P., Ringenberg, M., Siler, S., VanLehn, K., & Weinstein, A. (2001). Interactive Conceptual Tutoring in Atlas-Andes. In *Proc. of AI in Education 2001*.
25. Rosé, C.P., Litman, D., Bhembe, D., Forbes, K., Silliman, S., Srivastava, R., & VanLehn, K. (2003). A Comparison on Tutor and Student Behavior in Speech Versus Text Based Tutoring. In *Proc. of the HLT-NAACL 03 Workshop on Educational Applications of NLP.*
26. Smith, V. L., & Clark, H. H. (1993). On the course of answering questions. *Journal of Memory and Language, 32,* 25-38.
27. Xu, W. & Rudnicky, A. (2000). Language modeling for dialog system. In *Proceedings of ICSLP 2000*. Paper B1-06.