A Time to Be Silent and a Time to Speak: Time-Sensitive Communicative Actions in a Reading Tutor that Listens

Gregory Aist and Jack Mostow

Project LISTEN
215 Cyert Hall, Language Technologies Institute
Carnegie Mellon University
4910 Forbes Avenue
Pittsburgh, PA 15213

aist+@andrew.cmu.edu, mostow@cs.cmu.edu

Abstract

Timing is important in discourse, and key in tutoring. Communicative actions that are too late or too early may be infelicitous. How can an agent engage in temporally appropriate behavior? We present a domain-independent architecture that models elapsed time as a critical factor in understanding the discourse. Our architecture also allows for "invisible experiments" where the agent varies its behavior and studies the effects of its behavior on the discourse. This architecture has been instantiated and is in use in an oral reading tutor that listens to children read aloud and helps them.

Introduction

As in comedy, one secret of good tutoring is timing. A response delivered too early or too late is infelicitous. If a tutor responds too quickly, students don't learn as well (Rowe 1972, Gambrell 1983, Stahl 1994, Tobin 1986, Tobin 1987). On the other side, we are all familiar with the frustration of waiting for a system that responds too slowly.

How can an agent be organized so as to generate turn-taking behaviors at appropriate times? That is, how can a tutor be "temporally correct"? And how can it gauge the effect of its actions on the discourse?

We present an architecture with the following characteristics. First, it is *time-sensitive* in that it models elapsed time as a critical aspect of understanding the discourse. Second, it is *domain-independent* because the rules for generating turn-taking behavior are prosodically driven and do not explicitly refer to domain objects. Finally, it *supports invisible experiments* where the agent varies its behavior and observes the effects of such variation.

We instantiate this architecture in the context of the Reading Tutor being developed by Carnegie Mellon University's Project LISTEN (Mostow et al. 1993, Mostow et al. 1994, Mostow et al. 1995, Mostow and Aist 1997). Project LISTEN is developing an automated tutor to help children learn to read. We adapt the Sphinx-II speaker-independent continuous speech recognition system (Huang et al. 1993) to listen to children read aloud.

The Reading Tutor runs on a single stand-alone PC. The child wears a headset microphone and has access to a mouse, but not a keyboard. Roughly speaking, the tutor displays a sentence, listens to the child read it, provides help in response to requests or on its own initiative based on student performance, and then displays the next sentence if the child has successfully read the sentence (Aist 1997).

The version of the Reading Tutor described here was deployed in a classroom setting in March of 1997, where it was used independently by students in need of remedial reading tutoring. Some sessions were videotaped for expert analysis. The Tutor was also used without the researchers present, and all sessions were conducted with limited supervision by the teacher.

Related Work

Various mechanisms have been proposed to allow an agent to take turns. Proposed solutions include dynamic constraint satisfaction (Donaldson and Cohen 1997), dialogue scripts (Ball 1997), and linearly combined feature vectors (Keim, Fulkerson, and Biermann 1997). Ward (1996) used prosodic rules to trigger backchanneling in a computerized "eavesdropper" that listened to converesations and interjected "mm" when its rule fired. We use turn-taking rules, which are similar in approach to

Ward's backchanneling rule but control the turn-taking behavior of a complete spoken language system.

Russell et al. (1996) describe another oral reading tutoring project, the Talking and Listening Book project, but they do not use continuous speech recognition. Their system either leads the child through the sentence one word at a time, or lets the child read through the sentence one word at a time. When the student is able to read the entire sentence, and the goal is to give help on fluent reading, a more sophisticated set of turn-taking behaviors is desirable. For example, our Reading Tutor can follow a student's continuous reading, detect (albeit with uncertainty) when and where the student is having trouble, and give the student encouragement or a hint to get her back on track.

Communicative Actions

Communicative Actions by the User

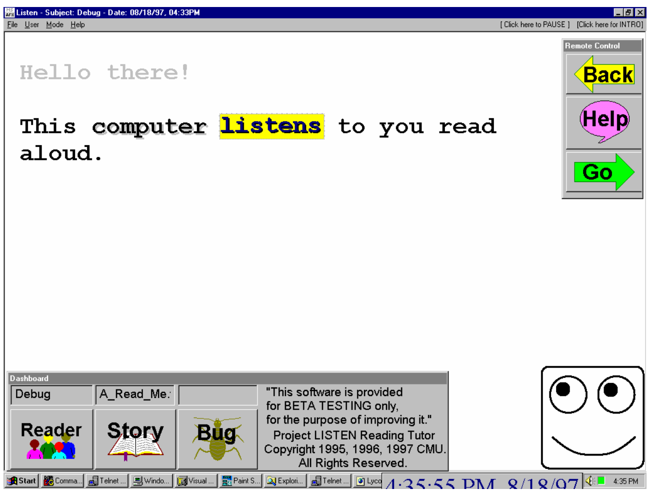


Figure 1. Screen layout for March 1997 Reading Tutor.

There are several communicative actions, both speech actions and other actions, available to the user (See Figure 1). The user can read a word aloud, read a sentence aloud, or read part of a sentence aloud. The user can click on a word for help on it. The user can click on buttons in the "Remote Control" window: *Back* (move to the previous sentence), *Help* (request help on the sentence), and *Go* (move to the next sentence). The user can also click on the buttons in the "Dashboard" window: *Reader* and *Story*. Students select their own subject codes using the Reader button, and select their own stories using the Story button.

Communicative Actions by the Reading Tutor

The Tutor can choose from several communicative actions, involving speech, graphics, and navigation. The Tutor can provide help on a word (e.g. by speaking the word), provide help on a sentence (e.g. by reading it aloud), backchannel ("mm-hmm"), provide just-in-time help on using the system, and navigate (e.g. displaying the next sentence). The Tutor uses digitized speech, synthesized speech, and graphics to provide assistance (Aist and Mostow 1997). These interventions employ synchronized audio and visual components, the importance of which is discussed in (Biermann and Long 1996).

A General Architecture for Understanding and Generating Time-Sensitive Communicative Actions

We describe an architecture with several important properties. First, it is *time-sensitive* because it uses elapsed time as a key component in processing student actions and in generating responses. Secondly, it is *domain-independent* (by design, if not yet by testing in multiple domains). Finally, it allows *invisible experiments* to be conducted, where the system varies its behavior and observes the effects of such variation on the dialogue. Our architecture uses explicit turn-taking rules to make decisions about when to take a turn.

We instantiated the architecture in the Reading Tutor. We specialized the user and Tutor actions and adapted the turn-taking rules to apply to the domain of oral reading tutoring.

Understanding Student Actions

Every student action that the Tutor agent needs to be aware of is classified as an event by the object that processes it. Direct manipulation, such as pressing a button, is processed by classifying the event (see below) and then responding to it. For example, the Remote Control classifies a click on the Help button as a request for help and notifies the Tutor that the user has requested help. User actions related to speaking, such as the onset of speech, the end of speech, partial speech recognition results, and the final recognition result, are handled by the Listener object, which is a custom-built interface to the continuous listening module and to the speech recognizer. The Listener notifies the Tutor when these events occur, and the Tutor records the event and responds if necessary.

Event Taxonomy

Each student action is classified as an abstract event. This hides from the Tutor the details of how the action was communicated. For example, clicking on the Back button is classified as a request to go back. Clicking on a previous sentence (displayed in gray above the current sentence) is also classified as a request to go back. This allows the Tutor agent to work with the logical communicative actions, not the specific interface details of how the user communicated them.

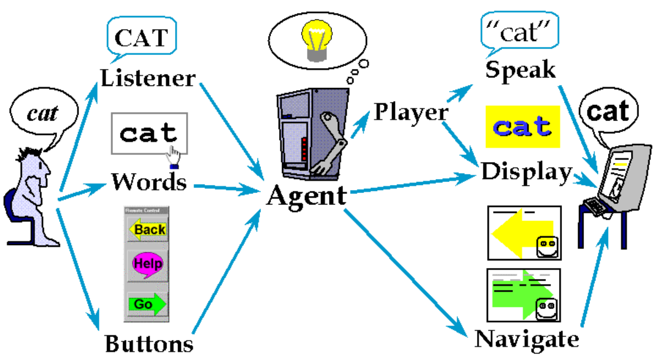
Each type of event recognized by the Tutor agent is assigned an event timer. An event timer is an object that measures the difference between now and when the event last occurred. There are also some special timers that represent classes of events. *Dead air time* is the elapsed time in which neither the user nor the agent has been speaking. *User turn time* is the elapsed time in the current user turn. *User idle time* is the elapsed time since any user event. *Total idle time* is the elapsed time since any event.

Generating Tutor Actions

The Tutor generates its own events autonomously, based on patterns detected in the discourse model. For example, the Tutor might decide that it is time to take a turn based on a period of silence after the user's turn. Once a Tutor event is generated, the Tutor responds to the event it generated by selecting a particular action, such as saying "mm-hmm", reading a word, or reading the sentence.

Some tutorial actions consist of several actions queued sequentially. For example, the Tutor can "recue" the student by reading part of a sentence and then highlighting the next word. These events are queued to a custom-built "multimedia player", which plays them in order. The Player notifies the Tutor when an item has been played, and when a sequence of items has been completed.

The information flow is shown in the diagram below.



Seven Rules of Turn-Taking

The Tutor uses seven turn-taking rules:

- Interrupt the student if the student's turn is very long
- Self-interrupt (stop speaking) if the student has overlapped
- Backchannel if the student pauses
- Take a "mini-turn" if the student continues to pause
- Take a turn if the student still continues to pause
- Take a turn if you hear the end of the student's turn
- Prompt the student if nothing has happened for a while

A mini-turn is a turn that leaves the discourse initiative with the student; it supplies content but is intended to encourage the student to continue. For example, if the student hesitates on a difficult word, the Tutor might supply the word to "unstick" the student.

How does the Tutor hear the end of the student's turn? In this domain, the Tutor recognizes the end of the student's turn (besides based on silence) when either (a) the last two words that the Tutor heard the student read are the last two words in the sentence, or (b) if the Tutor is expecting the student to read a single word, the Tutor heard the student read only that word.

These rules do not fire immediately upon entering the appropriate turn-taking state. Instead, each rule has a delay associated with it that indicates how long the turn-taking state must remain in the appropriate state before the rule will fire. This delay is compared with the appropriate event timer(s). For example, the rule for backchanneling compares the backchanneling delay against the dead air timer and the user action timer. Since results from the education literature indicate that delays of more than three seconds between teacher questions and teacher-supplied answers lead to increased student learning (Stahl 1994), we set most of the timing parameters to be greater than three seconds. The exception was backchanneling, since the Tutor's backchanneling was not intended to be perceived by the student as taking a turn. The delay for backchanneling was originally set to 1.5 seconds, but we increased it to 2 seconds because it seemed too fast for this task. At 1.5 seconds, despite being longer than normal conversational pauses, the Tutor seemed to interrupt students who were struggling with difficult words. Increasing the delay to 2 seconds made the Tutor seem more patient.

One indication of the generality of these rules is that we added only one rule (for self-interruption) when we expanded the Tutor to operate in full-duplex mode, so that it could talk and listen simultaneously. However, these turn-taking rules do not fully cover the space of possible turn-taking contexts. For example, there is no mechanism to generate intentional Tutor pauses, and there are no turn-taking rules that process student backchanneling.

Evaluating Communicative Actions

We can use this architecture to study timing in human-computer multimodal spoken dialogue. The results reported here are based on data collected March 18-19 and April 10, 1997, from a total of 18 subjects, in an urban elementary classroom.

Methodology

Besides interviewing students after they used the Tutor, we also analyzed the automatically generated transcripts of the dialogue to look for effects of Tutor behavior. In order to demonstrate the ability of this architecture to support invisible experiments, the Tutor backchanneled only half of the time (randomly selected) that its rule indicated backchanneling was appropriate. This allowed us to look for effects of Tutor backchanneling on the dialogue.

Results and Discussion

Students vary in what they find acceptable in terms of timing. With the exact same Tutor timing settings, some students found the Tutor too slow and others found it too fast. We are not sure exactly what aspect of the temporal behavior students had in mind. They could, for example, be thinking of how long the Tutor spent on each sentence, how fast it was reading, or how quickly it responded.

Backchanneling has a striking effect on the dialogue when compared to not backchanneling in identical contexts. Students are nearly twice as likely to continue reading after the Tutor backchannels (54.5%) as they are after the Tutor could have backchanneled but didn't (31.7%). These figures are based on 211 Tutor decisions to backchannel and 222 Tutor decisions not to backchannel.

Some caveats are in order. First, the volume of the backchanneling recordings was at the same level as the other Tutor phrases, and may have been too loud to be realistic. Second, the students may have perceived the Tutor's backchanneling as interruptioon, and begun speaking again to re-establsh context. Finally, the student may have actually finished reading the sentence, but the Tutor may have backchanneled instead of taking a turn because it failed to detect that the student was finished; the student may then have simply repeated the sentence.

Future Work

Since students vary in what timing characteristics they find acceptable, the Tutor should adapt its timing to individual students. We would like to extend this architecture to include intentional Tutor pauses within Tutor turns. We have recently completed a larger-scale study of the Reading Tutor in a summer reading clinic at an urban elementary school with over fifty students (with lower volumes on the backchannel recordings, among other changes); we would like to analyse these new data to further explore the effects of Tutor backchanneling. Finally, we intend to use the same methodology of 'invisible experiments' to test effects of other Tutor actions besides backchanneling.

Conclusion

What does this paper contribute? We have motivated our work on time-sensitive communicative actions in human-computer dialogue by looking at a domain, oral reading tutoring, where timing is especially important. We have given a set of seven turn-taking rules that govern how the Reading Tutor decides to take turns. Finally, we have given the first results on fully automatic analysis of the effect of communicative actions in full-duplex multimodal human-computer spoken tutorial dialogue.

Acknowledgments

This material is based upon work supported in part by the National Science Foundation under Grant No. IRI-9505156 and CDA-9616546, by the Defense Advanced Research Projects Agency under Grant Nos. F33615-93-1-1330 and N00014-93-1-2005, and by the first author's National Science Foundation Graduate Fellowship. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the official policies, either expressed or implied, of the sponsors or of the United States Government.

We thank Drs. Leslie Thyberg, Rollanda O'Connor, and Gayle Griffin for expertise on reading; Raj Reddy and the CMU Speech Group (especially Ravi Mosur) for Sphinx-II; Alex Hauptmann, Steven F. Roth, Morgan Hankins, and Maxine Eskenazi for helping evaluate the reading coach; Brian Milnes, Chetan Trikha, Stephen Reed, Scott Dworkis, Bryan Nagy, and David Sell for programming; Weekly Reader, Maxine Eskenazi, and Jennifer Gutwaks for text materials; and students, educators, and parents for Tutor tests in our lab and at Fort Pitt Elementary.

References

Aist, G. S. 1997. Challenges for a Mixed Initiative Spoken Dialog System for Oral Reading Tutoring. AAAI 1997 Spring Symposium on Computational Models for Mixed Initiative Interaction. To appear in a technical report.

Aist, G. S., and Mostow, J. 1997. Adapting Human Tutorial Interventions for a Reading Tutor that Listens: Using Continuous Speech Recognition in Interactive Educational Multimedia. To appear at CALL 97: Theory and Practice of Multimedia in Computer Assisted Language Learning. Exeter, UK.

Ball, G. 1997. Dialogue Initiative in a Web Assistant. AAAI 1997 Spring Symposium on Computational Models for Mixed Initiative Interaction. To appear in a technical report.

Biermann, A. W., and Long, P. M. 1996. The composition of messages in speech-graphics interactive systems. In Proceedings of the 1996 International Symposium on Spoken Dialogue, Philadelphia PA.

Donaldson, T. and Cohen, R. 1997. A Constraint Satisfaction Framework for Managing Mixed-Initiative Discourse. AAAI 1997 Spring Symposium on Computational Models for Mixed Initiative Interaction. To appear in a technical report.

Gambrell, L. B. 1983. The Occurrence of Think-Time During Reading Comprehension Instruction. *Journal of Educational Research* 77(2):77-80.

Huang, X. D., Alleva, F., Hon, H. W., Hwang, M. Y., Lee, K. F., and Rosenfeld, R. 1993. The Sphinx-II Speech Recognition System: An Overview. *Computer Speech and Language* 7(2):137-148.

Keim, G. A., Fulkerson, M. S., and Biermann, A. W. 1997. Initiative in Tutorial Dialogue Systems. AAAI 1997 Spring Symposium on Computational Models for Mixed Initiative Interaction. To appear in a technical report.

Mostow, J., Hauptmann, A. G., Chase, L. L., and Roth. S. 1993. Towards a Reading Coach that Listens: Automatic Detection of Oral Reading Errors. In Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93), 392-397. Washington DC: American Association for Artificial Intelligence.

Mostow, J., Roth, S. F., Hauptmann, A. G., and Kane, M. 1994. A Prototype Reading Coach that Listens. In Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), Seattle WA.

Mostow, J., Hauptmann, A., and Roth, S. F. 1995. Demonstration of a Reading Coach that Listens. In Proceedings of the Eighth Annual Symposium on User Interface Software and Technology, Pittsburgh PA. Sponsored by ACM SIGGRAPH and SIGCHI in cooperation with SIGSOFT.

Mostow, J., and Aist, G. S. 1997. The Sounds of Silence: Towards Automatic Evaluation of Student Learning in a Reading Tutor that Listens. To appear in the 1997 National Conference on Artificial Intelligence (AAAI 97).

Rowe, M. B. 1972. Wait-time and Rewards as Instructional Variables: Their influence in Language, Logic, and Fate Control. Presented to the National Association for Research in Science Teaching, Chicago IL.

Russell, M., Brown, C., Skilling, A., Series, R., Wallace, J., Bohnam, B., and Barker, P. 1996. Applications of Automatic Speech Recognition to Speech and Language Development in Young Children. In Proceedings of the Fourth International Conference on Spoken Language Processing, Philadelphia PA.

Stahl, R. J. 1994. Using 'Think-time' and 'Wait-time' Skillfully in the Classroom. *ERIC Abstracts*, report number EDO-SO-94-3.

Tobin, K. 1986. Effects of Teacher Wait Time on Discourse Characteristics in Mathematics and Language Arts Classes. *American Educational Research Journal* 23(2):191-200.

Tobin, K. 1987. The Role of Wait Time in Higher Cognitive Level Learning. *Review of Educational Research* 57(1):69-95.

Ward, N. 1996. Using Prosodic Clues to Decide When to Produce Back-channel Utterances. In Proceedings of the 1996 International Symposium on Spoken Dialogue, 1728-1731, Philadelphia PA.