Sydney, Australia, Nov. 30 – Dec. 4, 1998.

EXPANDING A TIME-SENSITIVE CONVERSATIONAL ARCHITECTURE FOR TURN-TAKING TO HANDLE CONTENT-DRIVEN INTERRUPTION

Gregory Aist
aist@cs.cmu.edu

Project LISTEN, http://www.cs.cmu.edu/~listen
Language Technologies Institute, Carnegie Mellon University
Pittsburgh PA 15213-3720 USA

ABSTRACT

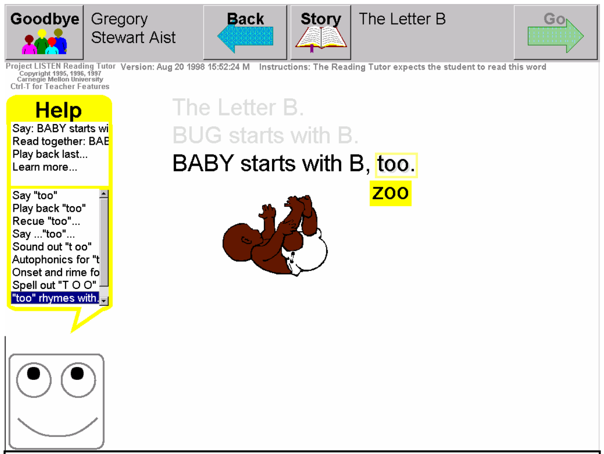
Turn taking in spoken language systems has generally been push-to-talk or strict alternation (user speaks, system speaks, user speaks, …) with some systems such as telephone-based systems handling barge-in (interruption by the user.) In this paper we describe our time sensitive conversational architecture for turn taking that not only allows alternating turns and barge in, but other conversational behaviors as well. This architecture allows backchanneling, prompting the user by taking more than one turn if necessary, and overlapping speech. The architecture is implemented in a Reading Tutor that listens to children read aloud, and helps them. We extended this architecture to allow the Reading Tutor to interrupt the student based on a non-self-corrected mistake – "content-driven interruption". To the best of our knowledge, the Reading Tutor is thus the first spoken language system to intentionally interrupt the user based on the content of the utterance.

1. MOTIVATION

Rich turn-taking is a ubiquitous feature of human-human spoken dialog. Rather than merely alternate between speakers, people backchannel, take multiple turns, interrupt each other, and finish each others' sentences (Fox 1993, Sacks et al. 1974, Duncan 1972). In tutorial dialog, rich turn-taking plays a substantial role in pedagogical effectiveness (Fox 1993). For example, the amoung of time that a teacher waits after asking a question before answering her own question affects student learning (Stahl 1994, Rowe 1972) – wait times of more than three seconds lead to better student learning (Tobin 1987, Tobin 1986). In order to expand the capabilities of spoken dialog systems to handle rich turn-taking, we started with a domain with a relatively simple content-based discourse model – oral reading tutoring – that is also interesting and important in its own right. Because of the simple content of the interaction, we are able to focus on other aspects of the dialog – specifically, turn-taking behavior.

2. A READING TUTOR THAT LISTENS

Project LISTEN's automated Reading Tutor (Mostow and Aist AAAI 1997, Aist and Mostow CALL 1997) builds on the speech analysis methods in (Mostow et al. 1994, Mostow et al. 1993) and the design recommendations in (Mostow et al. 1995). Unlike the reading coach in (Mostow et al. 1994), which required a NeXT machine for the student and a Unix workstation for the speech recognizer, the Reading Tutor runs in Windows™ 95 or NT 4.0 on a Pentium™, with a noise-cancelling headset microphone and a standard mouse. This platform is cheap enough to put in a school long enough to help children learn to read better. The Tutor incorporates materials adapted from Weekly Reader (a newsmagazine for children) and other sources. For other research related to using speech recognition to listen to oral reading, see (Bernstein and Rtischev 1991; Phillips, McCandless, and Zue 1992; Russell et al. 1996).



Project LISTEN Reading Tutor
Copyright 1995, 1996, 1997
Carnegie Mellon University
Ctrl-T for Teacher Features

Version: Aug 20 1998 15:52:24 M

Instructions: The Reading Tutor expects the student to read this word

The Letter B.
BUG starts with B.
BABY starts with B, too.

Figure 1. Reading Tutor, Fall 1998.

Roughly speaking, the Reading 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. (Aist 1997) describes how the Reading Tutor decides when to go on to the next sentence.

The student can read a word aloud, read a sentence aloud, or read part of a sentence aloud. The student can click on a word to get help on it. The student can click on Back to move to the previous sentence, Help to request help on the sentence, or Go to move to the next sentence (Figure 1). The student can click on Story to pick a different story, or on Goodbye to log out.

The Reading Tutor can choose from several communicative actions, involving digitized and synthesized speech, graphics, and navigation (Aist and Mostow 1997). The Reading 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. go on to the next sentence). With speech awareness central to its design, interaction can be natural, compelling, and effective (Mostow and Aist WPUI 1997).

3. CONVERSATIONAL ARCHITECTURE

There are several components to the Reading Tutor's conversational architecture.[1] First, a finite-state dialog model keeps track of the content state of the dialog. Second, an *event-time hierarchy* model keeps track of the elapsed time since various events, with events organized into a tree hierarchy. Thirdly, incoming events are recorded in the temporal model and may induce transitions in the content model. Finally, the conversational architecture has a "heartbeat" that evaluates a set of turn-taking rules five times per second and decides whether or not to take a turn based on those rules.

3.1. Finite-state content model

The conversational architecture separates the conventional content-based discourse model from the event-time temporal model. The content-based discourse model for the Reading Tutor is a simple finite-state machine. A portion of the model is shown in Figure 2.

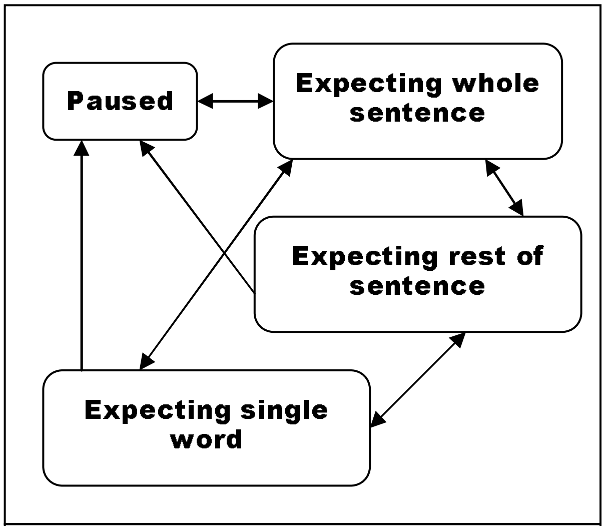


Figure 2. Partial finite-state dialog diagram for the Reading Tutor.

3.2. Event-time hierarchy temporal model

The temporal discourse model takes physical events such as button presses and translates them into logical events, such as a request for help. Each logical event has its own event timer, that measures the time since that event occurred. When a logical event occurs, the event timer for that event is reset to the current time. In addition, there are some event timers that are reset by multiple events. For example, there is a "dead air clock" that is reset when the student begins to speak, when the student stops speaking, when the Tutor begins to speak, and when the Tutor stops speaking. When neither the student nor the Tutor is speaking, the dead air clock measures the amount of time there has been silence in the interaction. The grouping of events in this manner forms the event-time hierarchy temporal model. Figure 3 shows some Reading Tutor events.

Student Request help on word Go back Go forward	**Tutor** Backchannel Prompt Take turn
Listener (speech detection and speech recognition) **Detect speech** **Detect silence** **Update hypothesis** **Final recognition**	Player (multimedia output) **Play sound** **Display text** **Highlight text** **Display picture**

Figure 3. Some Reading Tutor events.

3.3. Generating and Handling Events

Some events, such as clicking for help on a word, are generated by the student when clicking on objects. Other events are generated by the Listener, which interfaces with the speech recognizer and the speech/silence detection modules. Still other events are generated by the turn-taking rules as described below. All events are recorded in the temporal model (See 3.2 above). Some events may also have a side effect of causing a state transition in the content model.

3.4. Turn-taking rules

In addition to the event timers, four binary variables are used to represent the turn taking state of the interaction: is the Tutor speaking now? Is the Tutor taking a turn? Is the student speaking now? Is the student taking a turn? We distinguish "speaking now" from "taking a turn" because short utterances such as backchanneling should not constitute taking a turn. There are thus 16 turn taking states in this architecture, most of which represent transitions between participant turns. Five times per second, the architecture classifies the current turn taking state using the four variables and applies a list of rules specific to each turn taking state to decide whether to speak now or not. We use short-circuit evaluation: the first applicable rule fires, and the rest of the rules in the list are not

[1] A previous version of this architecture, with fewer rules and less explicit conversational state, but fewer conversational behaviors, is described in Aist, G. S., and Mostow, J. 1997. A Time to Be Silent and a Time to Speak: Time-Sensitive Communicative Actions in a Reading Tutor that Listens. AAAI Fall Symposium Series, Boston MA, USA.

checked. All thresholds in the architecture can be changed at runtime.

4. EXTENSION TO CONTENT-DRIVEN INTERRUPTION

In earlier versions of the architecture, the Reading Tutor interrupted the student only when the utterance was too long. Sometimes when students make mistakes they realize their mistake and correct themselves, but not always. A mistake on an early part of a sentence may prevent comprehension if uncorrected, so the Reading Tutor should interrupt an uncorrected mistake. In order to enable the Reading Tutor to interrupt based on a student's mistake, and to test the ability of the turn-taking architecture to generate a wide range of conversational behavior, we added a rule to the Reading Tutor that allows the Reading Tutor to interrupt the student in response to a non-self-corrected mistake. The full set of states and rules is as follows, with the interruption rules in boldface.

1. User turn, RT turn, user speaking, RT speaking
   1-a. If elapsed time for barge-in exceeds threshold, Tutor stops talking (this rule disabled to allow overlap)

2. User turn, RT turn, user speaking, RT not speaking
   2-a. If elapsed time for barge-in exceeds threshold, Tutor stops talking (this rule disabled to allow overlap)

3. User turn, RT turn, user not speaking, RT speaking

4. User turn, RT turn, user not speaking, RT not speaking

5. User turn, not RT turn, user speaking, RT speaking

6. User turn, not RT turn, user speaking, RT not speaking
6-a. If user's turn longer than one minute, interrupt
6-b. If heard an uncorrected error and elapsed time since last interruption exceeds "Interruption" threshold, interrupt the user.

7. User turn, not RT turn, user not speaking, RT speaking

8. User turn, not RT turn, user not speaking, RT not speaking
8-a. If heard an uncorrected error and elapsed time since last interruption exceeds "Interruption" threshold, interrupt the user.

9. Not user turn, RT turn, user speaking, RT speaking

10. Not user turn, RT turn, user speaking, RT not speaking

11. Not user turn, RT turn, user not speaking, RT speaking

12. Not user turn, RT turn, user not speaking, RT not speaking

13. Not user turn, not RT turn, user speaking, RT speaking

14. Not user turn, not RT turn, user speaking, RT not speaking

15. Not user turn, not RT turn, user not speaking, RT speaking

16. Not user turn, not RT turn, user speaking, RT not speaking
    16-1. Student took the previous turn.
    16-1-a. If the Listener detected the end of the sentence – that is, if the student does not seem to be stuck in the middle of the sentence but rather seems to have finished reading the sentence – take a turn immediately.
    16-1-b. If the user hasn't clicked on anything in a while, and the elapsed time on the dead air clock exceeds the "backchannel" threshold (~2 seconds), backchannel – say "mm-hmm" – and increase the intervention level – i.e. don't apply this rule again until some other turn-taking state has been entered.
    16-1-c. If the elapsed time on the dead air clock exceeds the "take turn" threshold, take a turn – e.g. read the entire sentence.
    16-2. Tutor took the previous turn.
    16-2-a. If the elapsed time on the dead air clock exceeds the "prompt" threshold, prompt the student – for example, say "Please read this sentence."

The rule for deciding when to interrupt the student is used in cases where the student is taking a turn and the Reading Tutor is not taking a turn, as shown above. In more detail, this rule is as follows: if the Tutor has heard an uncorrected error and has not recently interrupted, interrupt the student. An uncorrected reading error is defined as an error followed by at least the next word in the sentence. For example, if a student misreads a word and then reads the next word, the Reading Tutor would consider that an uncorrected reading error. A hypothetical but reasonable example is shown below:

Text: This computer listens to you read aloud.

Student: this com… copter listens

⇒ uncorrected reading error on 'computer'

5. DISCUSSION AND CONCLUSION

We have described an updated version of our time-sensitive conversational architecture that generates a wide range of rich turn-taking behavior, including backchanneling and taking multiple turns. We have further described the extension of this architecture to generate content-based interruption. In some sense the power of the model is shown by its easy extension to the new behavior of content-driven interruption. The model achieves some degree of simplicity by its factoring of state into content state (3 speech-handling states, excluding Paused) and temporal state (16 turn-taking states, plus additional continuous-variable information about elapsed time.)

Why use Boolean variables for the participants' turn-taking status? We could have chosen to use a single variable turn = <student, tutor>, but we would have had to add values <both> and <neither> in order to adequately describe situations where neither the student nor the Tutor is talking, or where both are talking. Thus we would have turn = <student, tutor, both, neither>. At that point we might as well use two Boolean variables student-turn and tutor-turn to represent those four values. As well, the use of separate variables allows us to express our confidence in the value of tutor-turn separately from the (less reliable) accuracy of speech endpointing reflected in student-turn. A single four-valued variable would confuse these naturally separate measures.

Why distinguish between "turn" and "speaking now"? The original distinction was motivated by the speech/silence detection API. However, transient sound should not be recorded as a student turn. Also, slight pauses should not be recorded as the end of a student's turn. We do however want to represent the immediate audio state ("speaking now"), in order to check for cases where factors other than extended silence – such as the student reaching the end of a sentence – indicate it is appropriate to take a turn.

There are currently sixteen turn-taking states in the architecture, but not all sixteen turn-taking states have discourse rules associated with them. Some turn-taking states have no rules at all. Others have several. The turn-taking states with no rules at all raise an interesting question: Do they actually correspond to other reasonable conversational behaviors, such as adjusting speaking rate when reading something together, or are they just artifacts of the architecture?

What does this paper contribute? We have described the current version of our time-sensitive architecture. We have described how a small modification allowed the architecture to generate content-based interruption, to allow the Reading Tutor to interrupt a student to catch a non-self-corrected reading error. To the best of our knowledge, the Reading Tutor is the first spoken language system to intentionally interrupt the user based on the content of the utterance.

5. ACKNOWLEDGEMENTS

This material is based upon work supported by NSF under IRI-9505156 and CDA-9616546. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Defense Advanced Research Projects Agency.

6. REFERENCES

1. 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. AAAI Technical Report SS-97-04.

2. 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. In Proceedings of CALL 97: Theory and Practice of Multimedia in Computer Assisted Language Learning. Exeter, UK.

3. Bernstein, J., and Rtischev, D. 1991. A voice interactive language instruction system. In Proceedings of the Second European Conference on Speech Communication and Technology (EUROSPEECH 91), Genova, Italy, vol. 2, pp. 981-984.

4. Duncan, S. 1972. Some signals and rules for taking speaking turns in conversations. Journal of Personality and Social Psychology 23(2):283-292.

5. Fox, B. A., *The Human Tutorial Dialogue Project: Issues in the Design of Instructional Systems,* Lawrence Erlbaum Associates, Hillsdale NJ, 1993.

6. 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.

7. 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. Selected as the AAAI-94 Outstanding Paper.

8. 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.

9. Mostow, J., and Aist, G. S. 1997. The Sounds of Silence: Towards Automatic Evaluation of Student Learning in a Reading Tutor that Listens. In Proceedings of the 1997 National Conference on Artificial Intelligence (AAAI 97), pages 355-361.

10. Mostow, J., and Aist, G. S. 1997. When Speech Input is Not an Afterthought: A Reading Tutor that Listens. Workshop on Perceptual User Interfaces, Banff, Alberta, Canada, October 1997.

11. Phillips, M., McCandless, M., and Zue, V. 1992. Literacy tutor: An interactive reading aid. Technical report, Spoken Language Systems Group, MIT, 545 Technology Square, NE43-601, Cambridge MA 02139 USA.

12. Rowe, M. B., "Wait-time and rewards as instructional variables: Their influence in language, logic and fate control," National Association for Research in Science Teaching, Chicago IL, 1972.

13. 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. ICSLP 96, Philadelphia.

14. Sacks, H., Schegloff, E. A., and Jefferson, G. 1974. A simplest systematics for the organization of turn-taking for conversation. Language 50(4): 696-735.

15. Stahl, R. J., “Using `Think-Time' and `Wait-Time' skillfully in the classroom”, ERIC Abstracts Report number EDO-SO-94-3, 1994.

16. Tobin, K., “Effects of teacher wait time on discourse characteristics in mathematics and language arts classes”, *American Educational Research Journal, Vol. 23, No. 2, 1986, pp. 191-200.*

17. Tobin, K., “The role of wait time in higher cognitive level learning”, *Review of Educational Research, Vol. 57, No.1, 1987, pp. 69-95.*