A General Architecture for a Real-Time Discourse Agent and a Case Study in Computerized Oral Reading Tutoring

Gregory S. Aist

Computational Linguistics Program

Carnegie Mellon University

December 3, 1996

Jack Mostow, principal advisor

Nancy Green

Alex Rudnicky

Presentation Outline

- Problem Statement
- Significance to Computational Linguistics
- Highlight - Review of previous work
- Statement of proposed work
- Evaluation
- Timetable
- Additional notes
- Acknowledgements
- Conclusion

Problem Statement

- Spoken dialogue systems engage in strict turn-taking
- Human behavior is much more varied

How can spoken dialogue systems engage in more natural real-time interaction with users, and what software architecture can represent various discourse behavior, such as turn-taking, backchanneling, and overlap in a unified and consistent framework?

Most spoken dialog systems respond to user actions with a single conversational turn and only take such a turn after a period of silence following a user's turn. Humans engage in a richer set of dialogue behavior, including backchanneling, overlapping, and taking multiple turns.

How can spoken dialogue systems engage in more natural real-time interaction with users, and what software architecture can represent various discourse behavior, such as turn-taking, backchanneling, and overlap in a unified and consistent framework?

Significance to Computational Linguistics

- Time-sensitive behavior: a natural complement to other work on discourse
- General, transferable architecture
- Case study in oral reading tutoring
  - Worthwhile in its own right
  - Natural linguistic restrictions
  - 'Any-citizen' audience: the future of speech in a world of ubiquitous computing

Highlight – Review of previous work

Conversational acts (Traum & Hinkelman 1992)

Discourse Level	Act Type	Sample Act
Sub-utterance unit	Turn-taking	take-turn keep-turn release-turn assign-turn
Utterance unit	Grounding	initiate continue acknowledgement repair request-repair request-acknowledgement cancel
Discourse unit	Core speech acts	inform wh-question yn-question accept request reject suggest evaluate reqperm offer promise
Multiple discourse units	Argumentation	elaborate summarize clarify question-and-answer convince find-plan

Conversational analysis: human behavior as model.

Highlight – Review of previous work

- Spoken dialogue systems (ATIS, JANUS)
- Intelligent tutoring systems: reading
  - Commercial systems – help on demand
  - Research systems (DRA Malvern STAR, Project LISTEN)
- Instructional 'wait time' (Rowe 1972, Tobin 1987, Stahl 1994)
  - Ideal: 3 second interval
  - Reality: 0.7 to 1.4 seconds
  - Computers can be more patient

- Wait time is between people (teacher question/further speech) and many $\rightarrow$ 1

Research Context: Project LISTEN

Goal: Develop a computerized oral reading tutor that

- displays text on a computer screen
- uses speech recognition to listen to children read aloud
- offers assistance when needed

This M.S. project is intended to improve the range and flexibility of assistance that the tutor can offer.

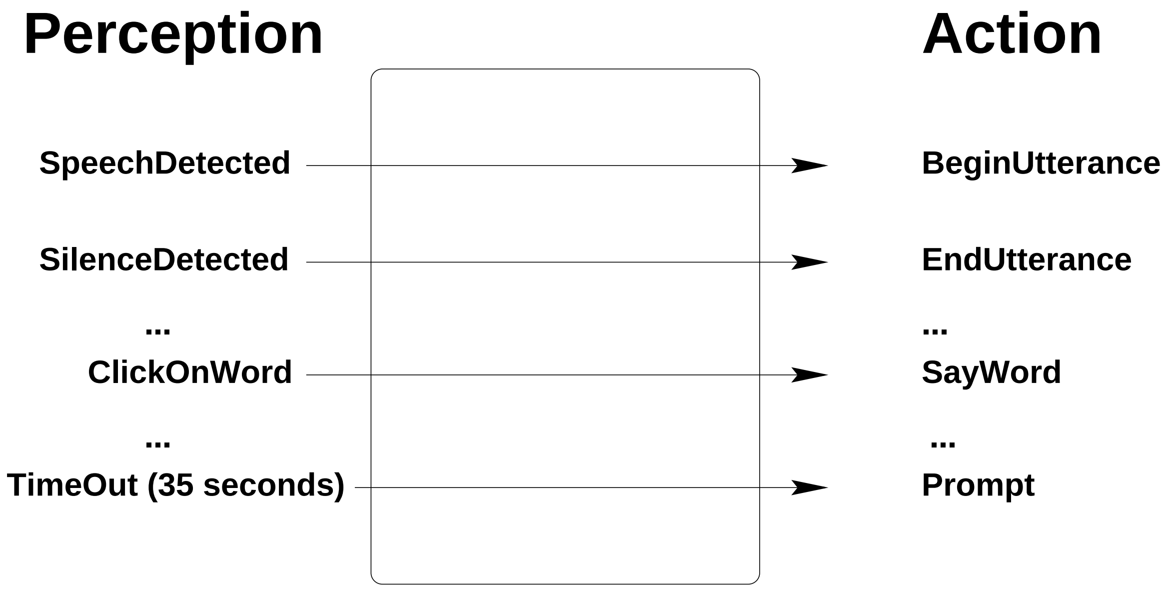
Statement of proposed work

- Input: Event stream
- Architecture
  - Event history
  - Event timers
  - System state
  - Task model
  - Decision procedure
- Output: Action

- Say one sentence about each architectural component

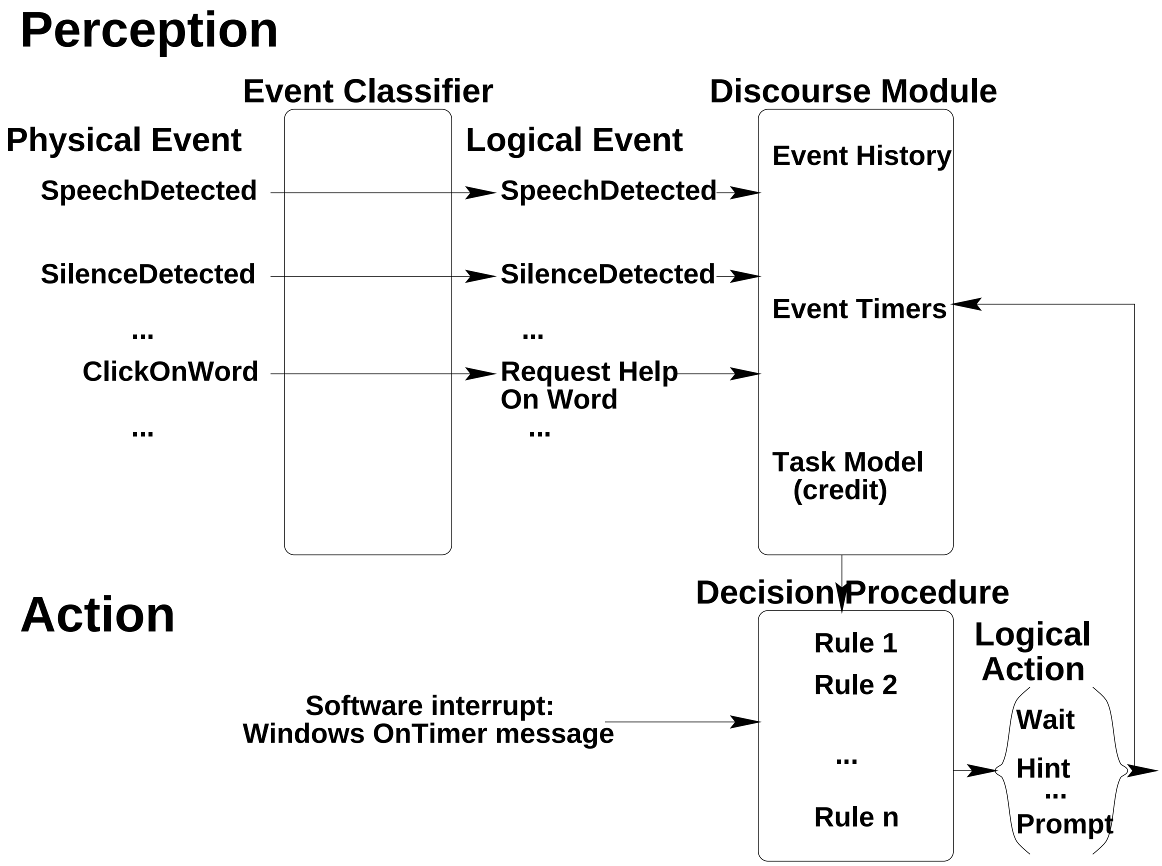
- Give examples of actions

Current Architecture



Immediate response, but behavioristic framework with ad hoc extensions

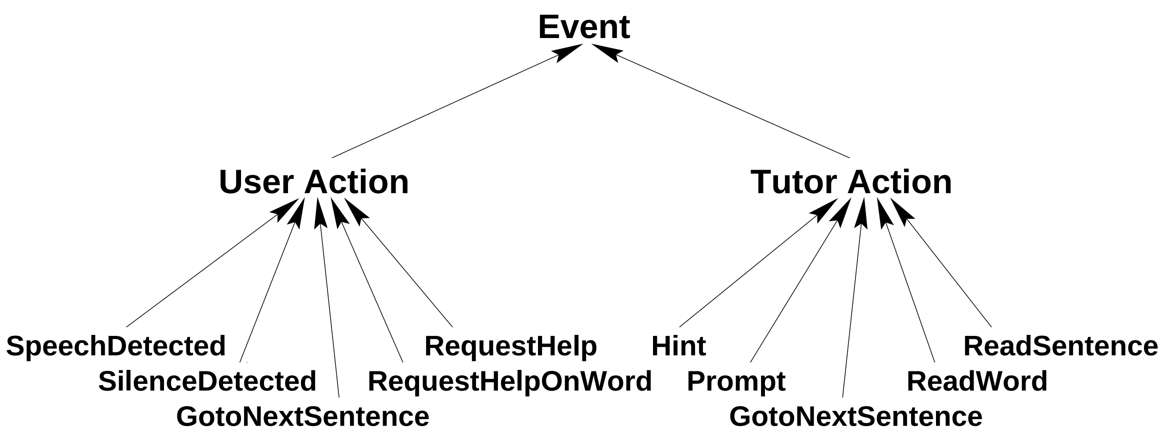
Proposed Architecture



Project LISTEN design specification will be used to guide implementation of specific actions and rules

Consistent use of internal state for richer behavior

Event Hierarchy



Discourse Module components

- Event History

  linear history of last n events

- Event Timers

  One timer for every node in the hierarchy, reset when an event of that type occurs

- Task Model

  Record of credit awarded for each word in a sentence during this interaction; used for deciding when to move on

Example rules

```
IF ((state == ready)
and (last_event == speech_detected))
THEN action = begin_utterance

IF ((state == hearing_sentence)
and (time_since_user_spoke > 3 seconds)
and (expected_word(W))
THEN action = give_hint(W)

IF (missed_word(W)
and (last_event == finalize_utterance)
THEN action = read_word(W)
```

Rules

- First order

- Conjunctions

Evaluation

- General architecture
  - Support for a wide range of discourse behavior
  - Inspectable and maintainable
  - Easy to instantiate in a new domain
- Case study: Oral reading tutoring
  - Participate in tutoring interactions
  - Acceptable to human experts – through evaluation of videotapes

Timetable

Dec 96	**Proposal**
Mar 97	**System completion**
Dec 96	Event classifier
15 Jan 97	Discourse module
31 Jan 97	Decision module
31 Jan 97	Replicate existing behavior
Feb 97	Add additional rules and responses
Mar 97	**Evaluation**
	– Package general architecture; internally publish
	– Expert tutor evaluate videotaped interactions
Apr 97	**Defense**
Apr 97	**Final write-up**

Additional notes

- Reliance on other studies
  - Project LISTEN pilot study at Fort Pitt Elementary School
  - Continued usability testing with CMU Role Models Program children
- Observer effects

  Experimenter disguised as programmer, out of peripheral vision
- Development resources: MS Visual C++, Perl scripts

Acknowledgements

- My advisor and committee members

- My fellow graduate students

- The participants in the two LISTEN studies

  ...and their parents!

This material is based on work supported under a 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.

Conclusion

- General architecture for a real-time discourse agent

- Case study in oral reading tutoring