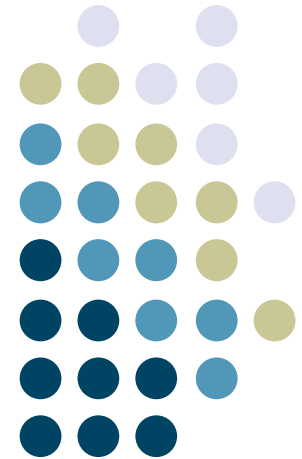


# Speech-Based Interaction

---

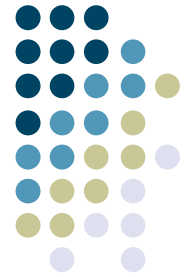


**Georgia  
Tech**

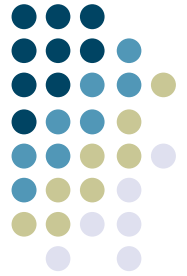


# Using Speech as a “Natural” Data Type

Georgia  
Tech

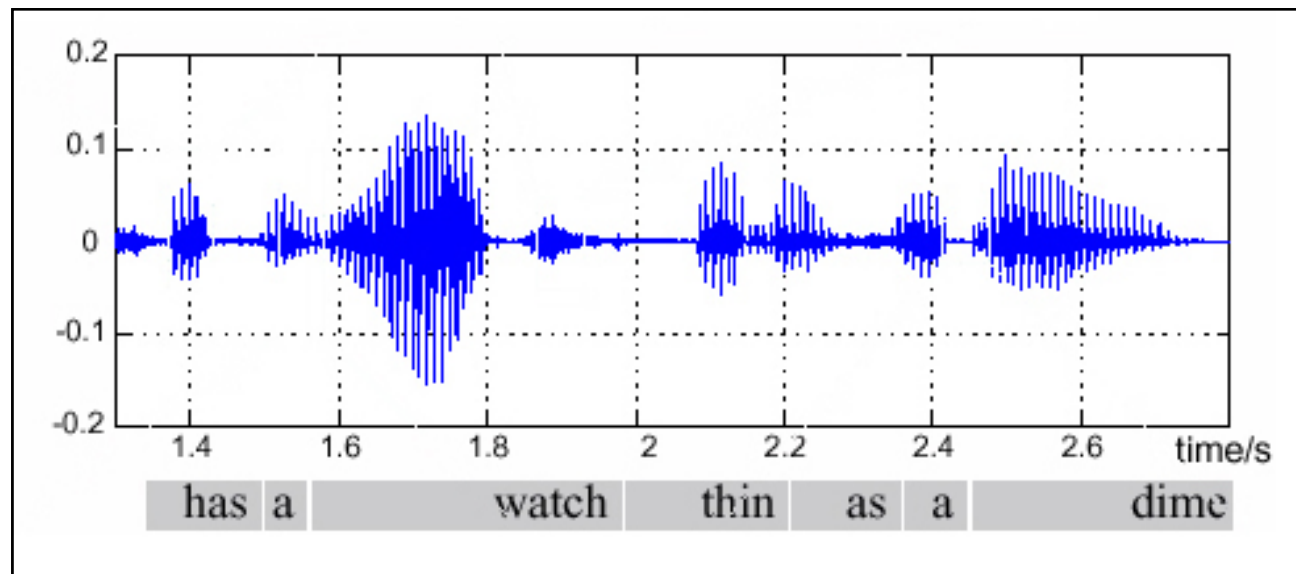


- Speech as Input
  - Chief decision: Recognition versus Raw Data
  - Recognition
    - Translate into other information (words)
    - Must deal with errors
    - Useful for either human *or* machine consumption of results
  - Raw Data
    - For use “as data” (not commands) for human consumption
    - Often linked with other context (time) in capture applications
- Speech as Output
  - Main issues: length of presentation time, lack of persistence, etc.



# Issues in Speech as Input

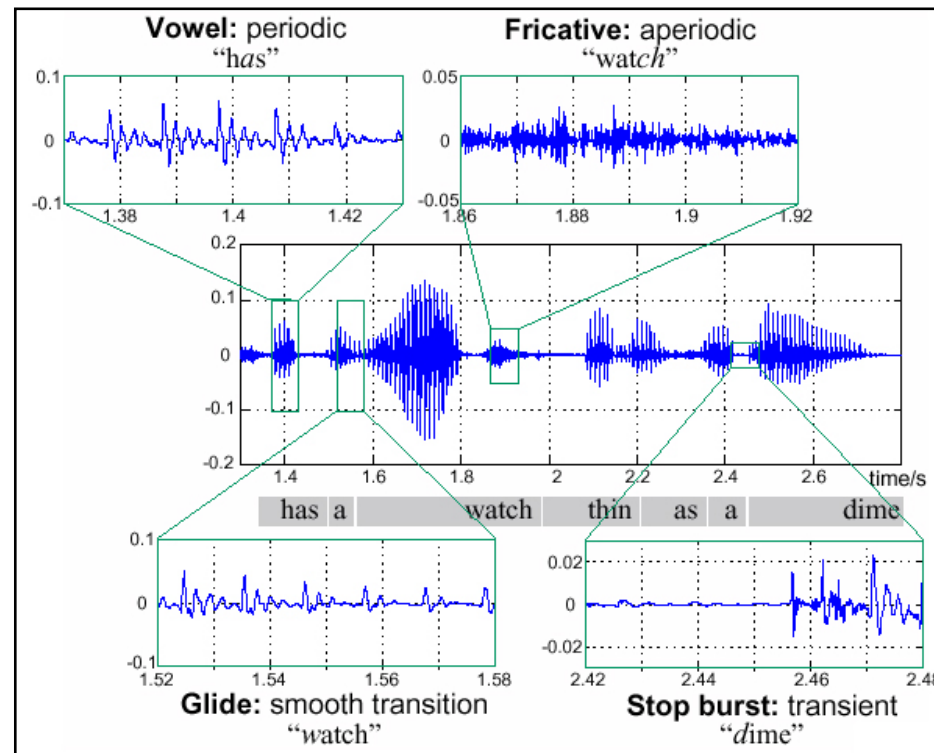
- Perfect recognition of speech (or semantic understanding of any kind of audio) is difficult to achieve



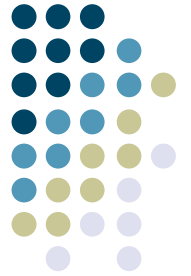
- Challenge: How would you begin?
  - Segmentation
  - Syntax



# Interesting features in speech

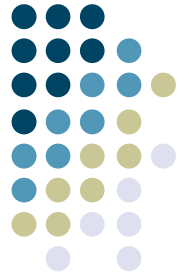


- Pauses between phrases as well...

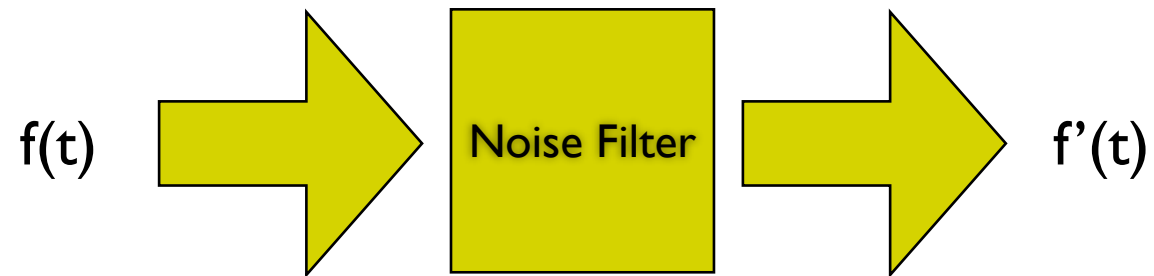


# Issues

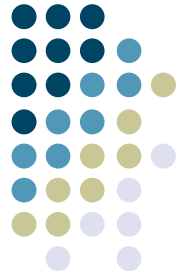
- Use of open air microphones & speakers can result in undesired audio
  - ambient noise
  - audio feedback
- Challenge: allow developers to easily add/use functions in their applications
  - Noise reduction
  - Enhance audio quality
  - Echo cancellation



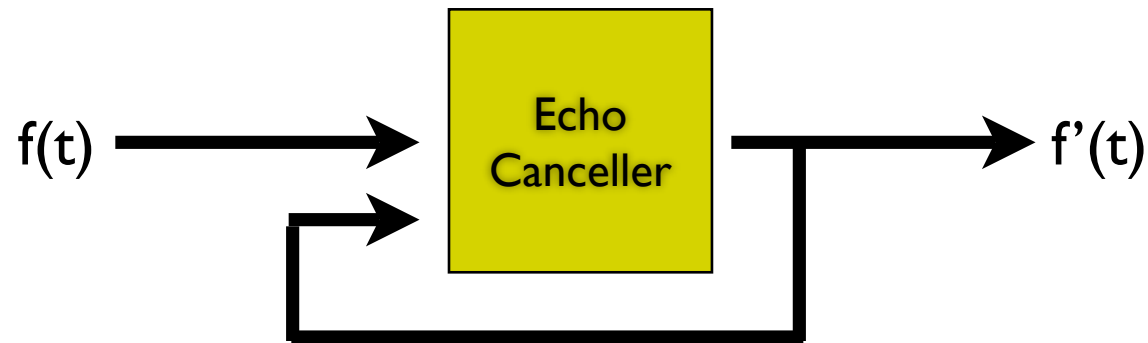
# Noise Reduction



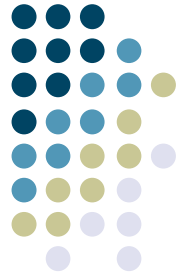
- Random noise is hard to predict



# Echo Cancellation



- Software and hardware exist, but are hard for developers to easily add to application
- Random noise is hard to predict, but echoes are not so random...

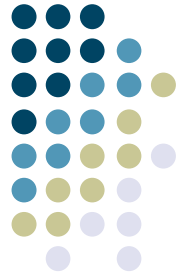


## More Issues

- It is still difficult to:
  - grab
  - chunk (segment)
  - store
  - search/index/grep
  - playback (think about the pain of automated phone menus...)
- Challenge: provide support for handling audio in manner similar to text



# Most Straightforward Speech Interface

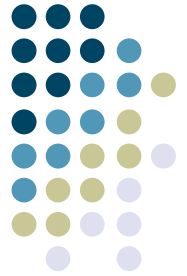


- Voice menu systems
- System speaks list of possibilities then waits for you to select one
  - Minor improvement: you can jump in whenever you hear the item you want
- Why are these so painful?

# Most Straightforward Speech Interface



- Voice menu systems
- System speaks list of possibilities then waits for you to select one
  - Minor improvement: you can jump in whenever you hear the item you want
- Why are these so painful?
  - Hierarchy -- very wide and deep makes for a big search space
  - Often no easy way to jump around in the tree
  - “Where you are” matters, but there’s no way to know “where you are” other than just hearing the menu again
  - Presentation time -- reading of long lists of options
- There are good points:
  - You know what you can do at any given time
- Triumph of ease of implementation over imagination



# Audio Features

- Think of as “degrees of freedom” of speech as an input device
- Pauses
  - Analogy to mouse up/down/drag?
- Who is speaking?
  - Turn-taking
- How is someone speaking?
  - Prosody, affect
- What is being said?
  - Recognition of words



# Case Study: Speech Acts

- Big idea: move away from voice as a replacement for menus (easy to implement but painful to use), toward more conversational interfaces

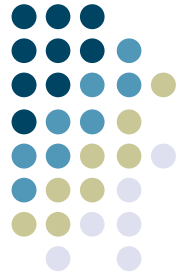
- “Designing SpeechActs: Issues in Speech User Interfaces,” Yankelovich, Levow, Marx, CHI’95

- Mail:

- SpeechActs: You have 14 new messages and 47 old messages. The first new message is from Eric Baatz regarding "report completed."
  - User: Let me hear it.
  - SpeechActs: "The first draft is ready for your comments. Eric."
  - User: Reply and include the current message.
  - SpeechActs: Preparing message to Eric Baatz. Begin recording after the tone. When finished, pause for several seconds.
  - User: Eric, I'll get those to you this afternoon.
  - SpeechActs: Say cancel, send, or review.
  - User: Send.
  - SpeechActs: Message sent. What now?
  - User: Next message.
  - SpeechActs: New message two from Stuart Adams, subject "Can we meet at 3:00 today?"
  - User: Switch to calendar...

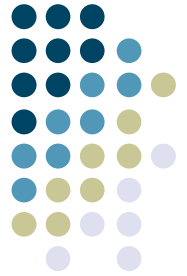
- Other commands:

- What do I have tomorrow?
  - What about Bob?
  - What did he have last Wednesday?
  - And next Thursday?
  - What was Paul doing three days after Labor Day?
  - What's the weather in Seattle?
  - How about Texas?
  - I'd like the extended forecast for Boston.



# Speech Acts

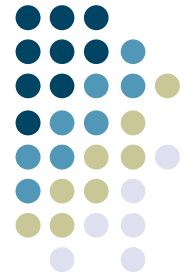
- How is this an improvement over voice menu systems?
  - No formal hierarchy -- so no need for commands to navigate it
  - “Where you are” doesn’t matter so much, so no need to fret over how to present it
  - Presentation time -- minimizes output from the system, focusing on *content* rather than *commands* or *context*
  - Conversational -- takes advantage of implicit contextual cues in the workflow, mimicking the way human conversation works
- Bad points?
  - You may not know what you have to say in order to control the system (not as explicit as in menus)



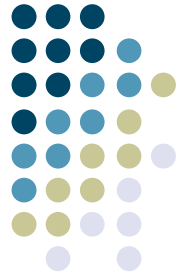
# Speech Acts Design Challenges

- Simulating Conversation
  - Avoid prompting wherever possible
  - Build context around subdialogs
  - Output prosodics: system asks “huh?”
  - Pacing: people often have to speak more slowly when talking to machines; need a way to “barge in” to machine output
- Transforming GUIs into SUIs
  - Vocabulary: need wide, domain-dependent vocabulary
  - Information organization: how to present content like email messages, flags, message numbers, etc., with consistency and w/o overwhelming the user
  - Information flow: speech “dialog boxes” (force users into a small set of choices) don’t fit well into conversational style (Users ignore or may produce unexpected answers: “Do you have the time?” not always answered by yes/no)

# Speech Acts Design Challenges (cont'd)



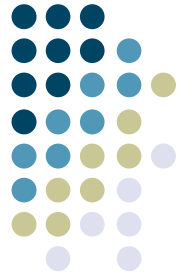
- Recognition errors
  - Rejection errors (utterance not recognized) are frustrating. Can yield “brick wall” of “I don’t understand” messages. Solution: provide progressive assistance
  - Substitution errors are damaging. Don’t want to verify every utterance. Approach: commands that present data are verified implicitly; commands that destroy data or are undoable are verified explicitly
  - Insertion errors (background audio picked up as commands or data). Solution: key to turn off recognizer
- The Nature of Speech
  - Lack of visual feedback. Users feel less in control; users can be faced with silence if they don’t do anything; long pauses in conversations are uncomfortable so users may feel a need to respond quickly; less information transmitted to the user at one time
  - Speed and persistence: although speech is easy for humans to produce it is hard to consume. Also not persistent: easy to forget, no on-screen reminder.



# Speech Acts Summary

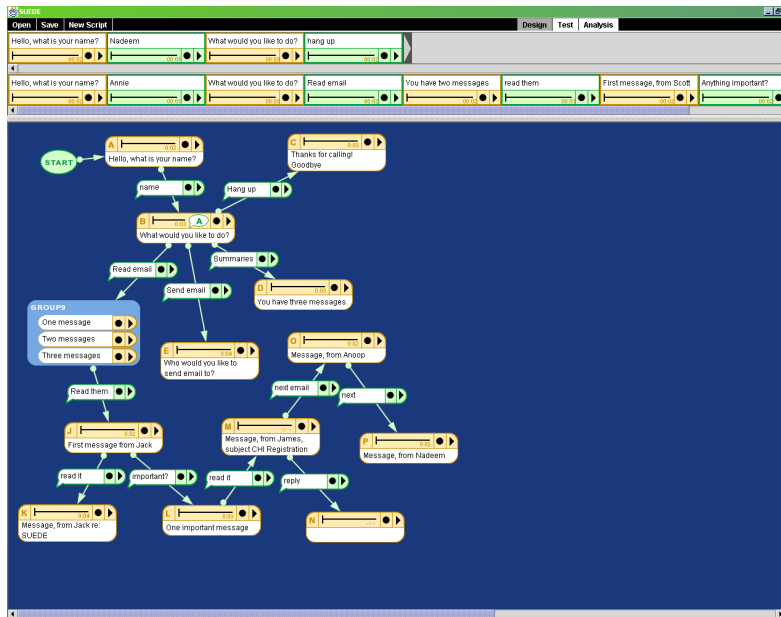
- SpeechActs shows the challenges in doing speech “right” (as opposed to just voice menus)
  - Speech as input
  - Speech as output
  - Real recognition
- Other systems that address the same set of challenges:
  - Voice Notes (MIT): speech as data, plus input and output
- There are other uses of speech that don’t involve so much hard (recognition and design) work though
  - Case studies:
    - Suede (Berkeley): faking “working” speech for UI design
    - Personal audio loop (GT): uninterpreted audio UI for human consumption
    - Family Intercom (GT): uninterpreted audio UI for human consumption



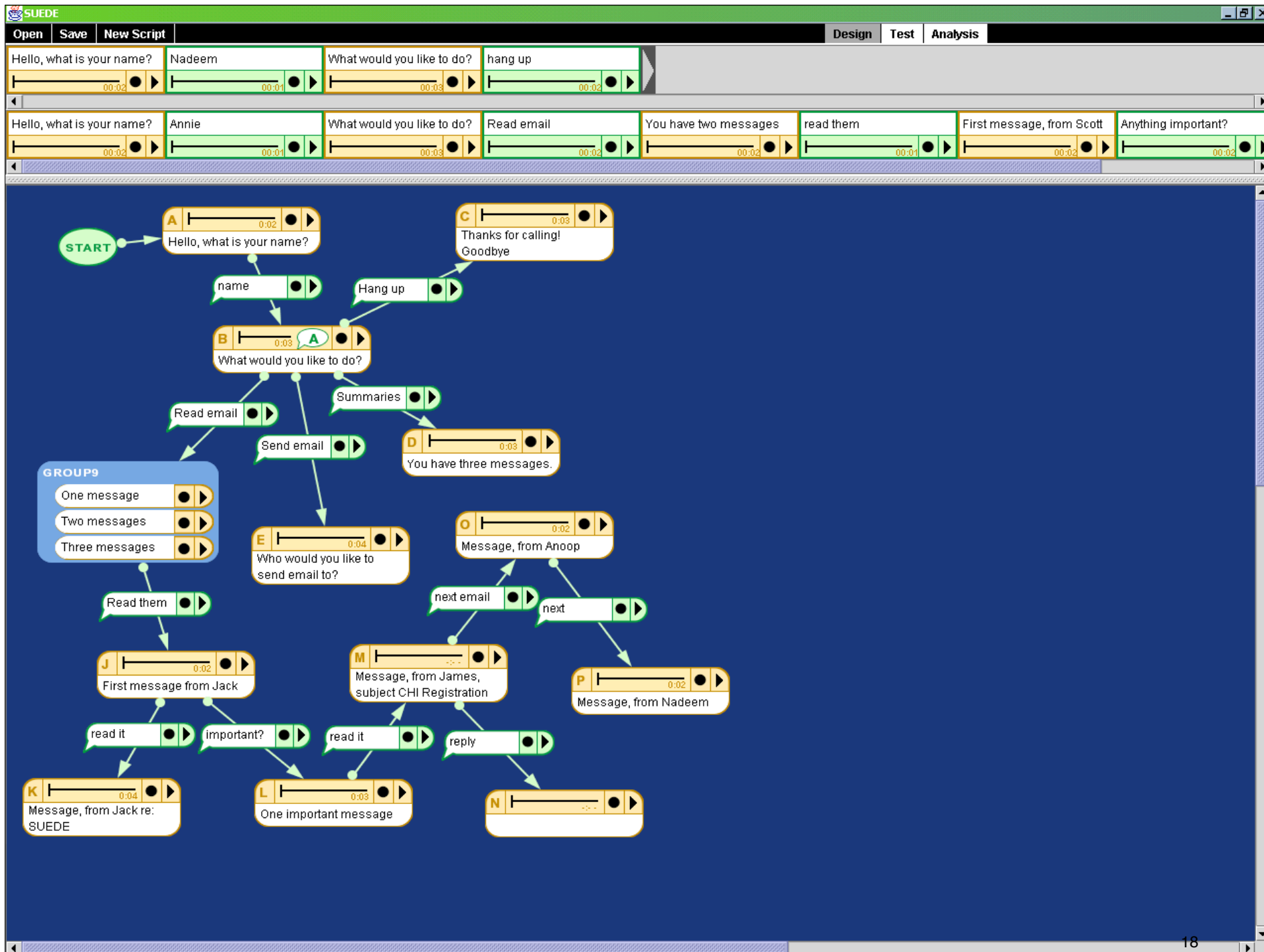


# Case Study: Suede

- Toolkit for prototyping speech interface



- <http://guir.berkeley.edu/projects/suede/>



SUEDE Test - User 0

Begin Test End Test Calibrate Silence % Errors 0 User ID: User 0

User 0

the movie you'd like to see? movie in what city or zip code?

03.2 03.5 02.2

Barge in Time out Not heard Not legal

**State: in what city or zip code?**

[Berkeley](#)

**option: San Francisco**

[Coronet](#)

[AMC 1000](#)

[Sony Metreon](#)

[10365](#)

**global:**

[hang up](#)

**SUEDE Analysis - movie**

Open Save

**User 0**

name | would you like to do? {name} | check movie times | the movie you'd like to see? | movie | in what city or zip code? |

02.7 | 02.0 | 02.0 | 03.2 | 03.5 | 02.2 |

**User 1**

me) | check movie times | the movie you'd like to see? | movie | in what city or zip code? | Berkeley | Shattuck Cinemas |

03.2 | 03.6 | 02.2 | 03.5 | 02.0 | 02.4 |

**User 2**

Hello what is your name? | name | would you like to do? {name} |

02.2 | 02.2 | 02.0 | 01.2 |

**User 3**

pick a new location | in what city or zip code? | San Francisco | Coronet | pick a new location | in what city or zip code? |

03.5 | 02.2 | 03.2 | 01.5 | 04.0 | 02.2 |

thank you for calling

could you like to do? {name}

movie times 3

movie 3

the movie you'd like to see?

Shattuck Cinemas

in what city or zip code?

User: 0 (-1 sec.)  
User: 1 (-1 sec.)  
User: 3 (-1 sec.)

Berkeley 1

San Francisco 2

10365 0

Lowe's

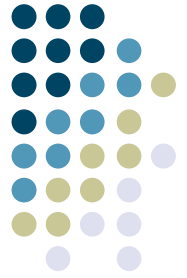
GROUP15

Coronet  
AMC 1000  
Sony Metreon

8, 10pm 0

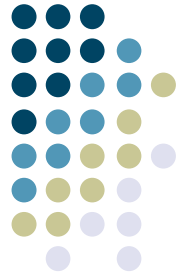
9pm 0

20



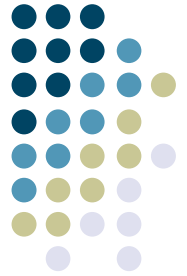
## Case Study: Personal Audio Loop

- Application which continuously buffers user's last 15 minutes of audio
  - "What were we talking about...?"
  - "What was that phone number I heard?"
- Features above are used to speed up audio playback when skimming for point of access
  - compressed or discarded in some cases

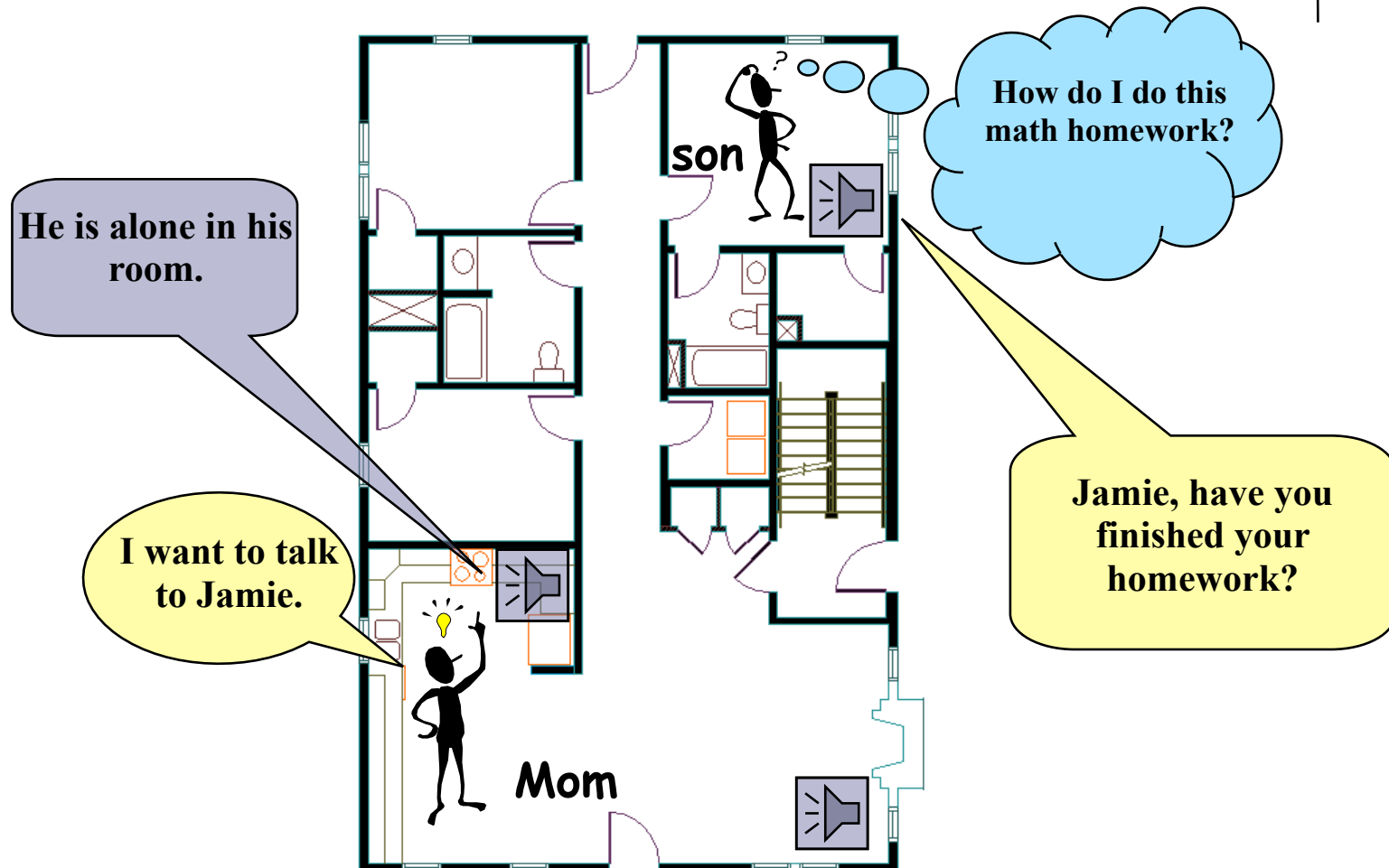


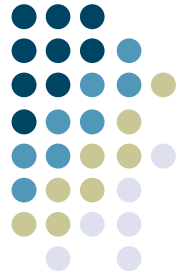
# Case Study: The Family Intercom

- Use location sensing in context-aware environment to connect people in different places in a conversation

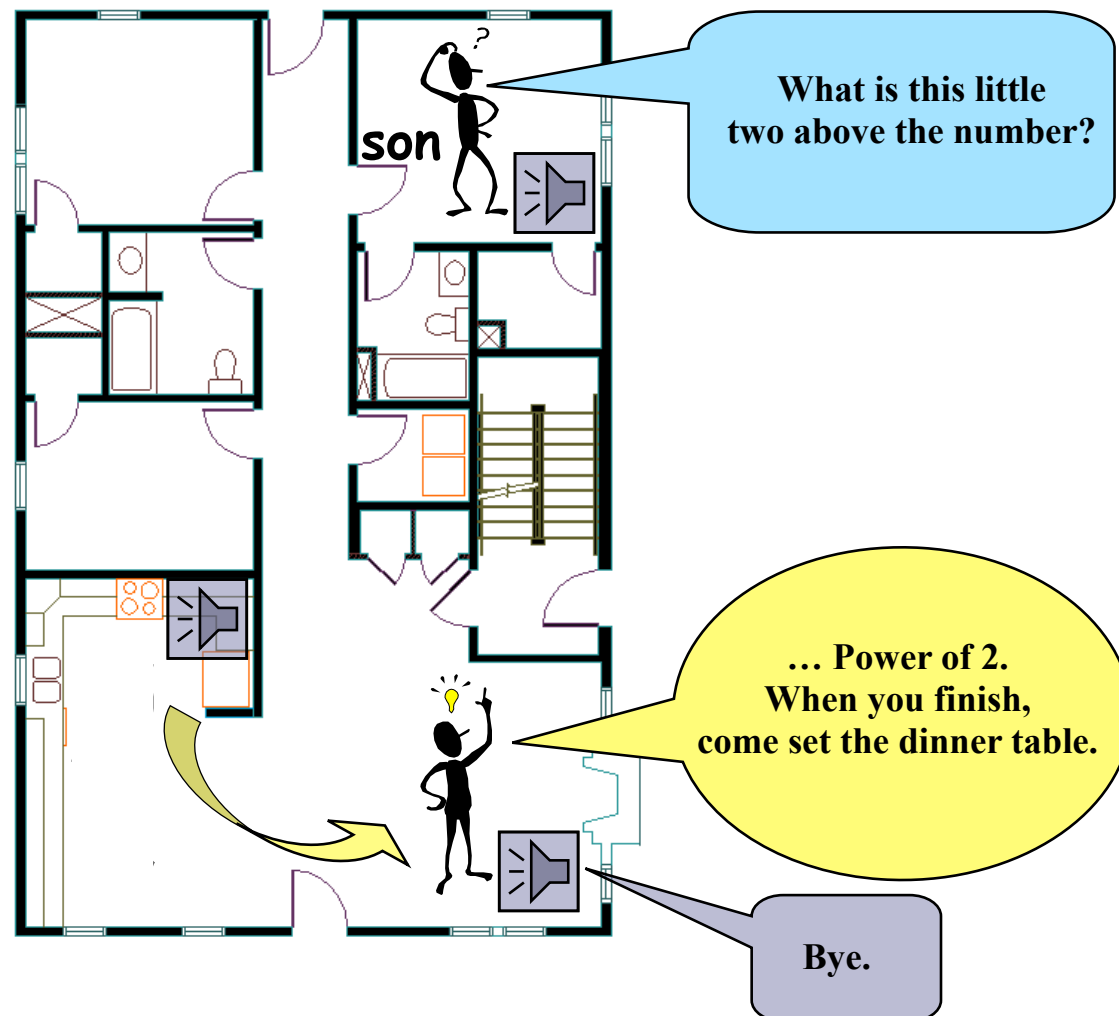


# The Family Intercom (UbiComp 2001)

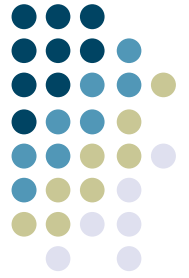




# The Family Intercom (UbiComp 2001)







# Resources

- Java Speech API:
  - Recognition and synthesis
  - <http://java.sun.com/products/java-media/speech/>
- FreeTTS:
  - A Java port of a very high quality speech synthesis package:
  - <http://freetts.sourceforge.net/docs/index.php>