

Speech Recognition and Conversational Interfaces

Virtues of Spoken Language

Natural:	Requires no special training
Flexible:	Leaves hands and eyes free
Efficient:	Has high data rate
Economical:	Can be transmitted and received inexpensively

Speech interfaces are ideal for information access and management when:

- The information space is broad and complex,
- The users are technically naive, or
- Speech is the only available modality.

Information Access via Speech



Future UIs for Information Access

- **Star Trek style UI**
 - verbally ask the computer for info or services
 - may be common in mobile/hands-free situations
 - hard to get to work well since it requires perfect speech recognition & unambiguous language understanding



Human Factors in Speech

- High Error Rates
 - Speech recognition
 - Background noise, intonation, pitch, volume
 - Grammars (missing words, size limitations)
- *“When speech recognition becomes genuinely reliable, this will cause another big change in operating systems.”* (Bill Gates, The Road Ahead 1995)

The Space of Recognition

		Domain	
Speaker	Dependent	Independent	
Dependent	not interesting	Transcription (training)	
Independent	We are here	Ultimate Goal (requires knowledge)	

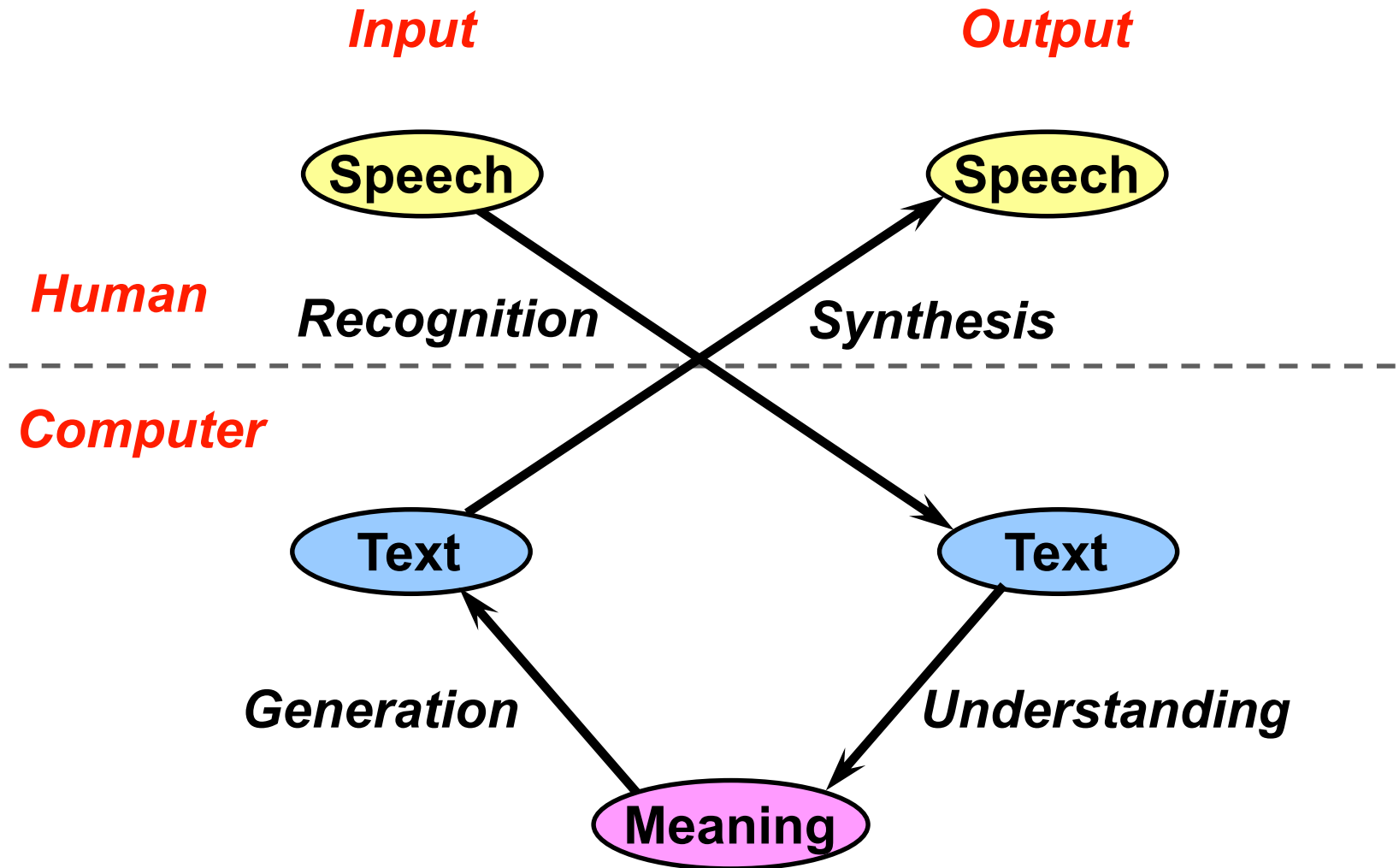
The Space of Recognition

- Speaker dependent
 - First train the system to recognize your speaking
 - Better recognition rates
- Domain dependent:
 - Only recognize what is in the domain
 - Better recognition rates
 - Domain can be large. How is it specified?

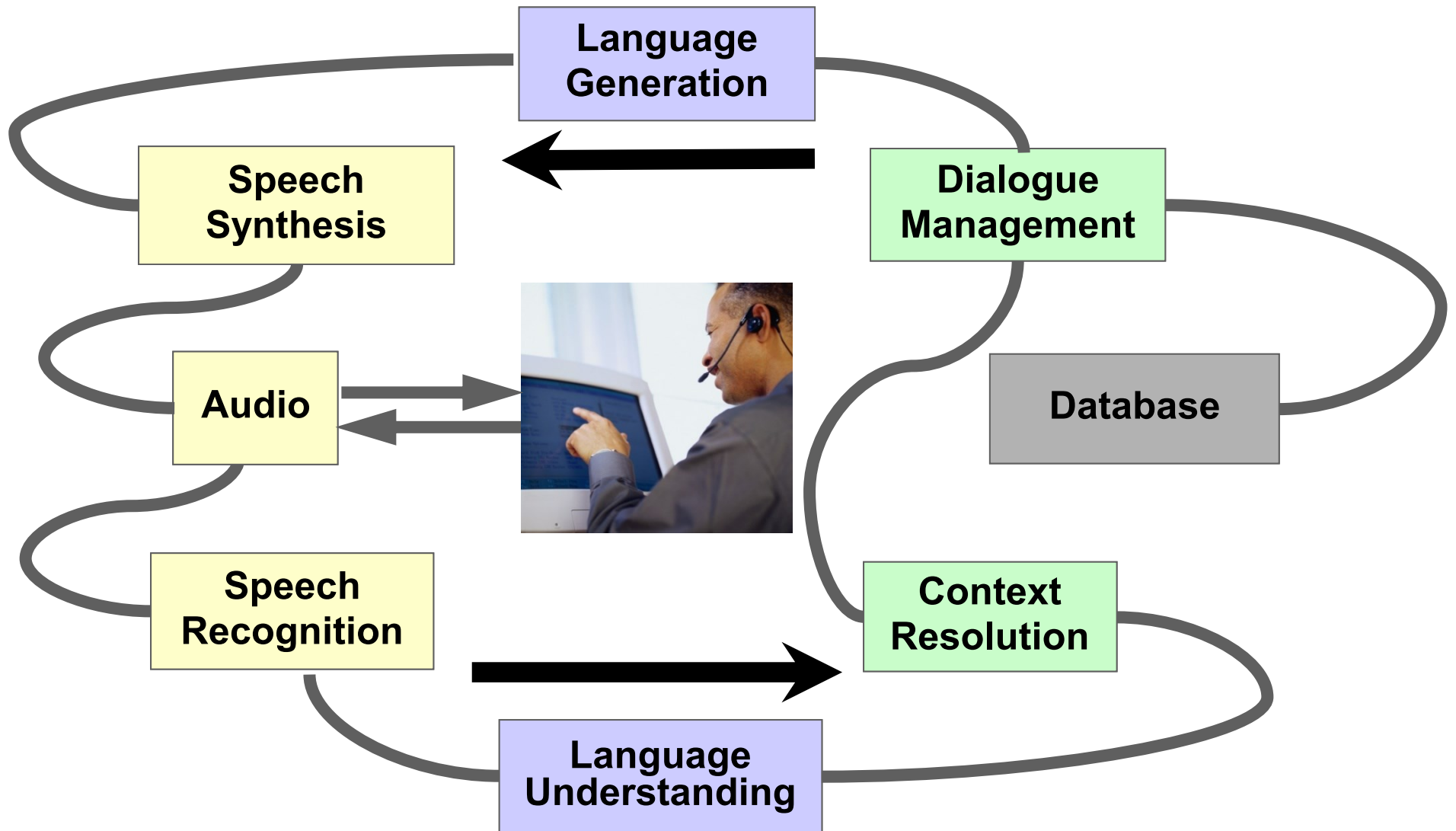
Why Speech?

- No special training -- naive users
- Leaves hands and eyes free -- but must know when to start recognition
- High data rate -- assuming low errors
- Inexpensive I/O -- microphone, speaker, button
 - speaker needed for feedback
- Some things are easier to specify with speech

Communication via Spoken Language



Components of Conversational Systems



Overview

- **Components of a speech based interface**
 - ASR: Automated Speech Recognition
 - NLU: Natural Language Understanding
 - Generation
 - Dialogue Manager

Conversational Agents

- **Also known as:**
 - Spoken Language Systems
 - Dialogue Systems
 - Speech Dialogue Systems
- **Applications:**
 - Travel arrangements (Train, hotel, flight)
 - Telephone call routing
 - Tutoring
 - Communicating with robots
 - Anything with limited screen/keyboard

A travel dialog: Communicator

S1: Hello. You've reached the [Communicator]. Tell me your full name
U2: *hi i'd like to fly to seattle tuesday morning*
S3: Traveling to Seattle on Tuesday, August eleventh in the morning. Your full name?
U4: *[John Doe]*
S5: Hello, [John], I'm glad to hear from you again. You can now ask me for help.
S6: OK. Let's see I have a United flight departing Pittsburgh at ten forty five a.m. arriving into Seattle at one fifty seven p.m. Is that OK?
U7: *Yes*
S8: Will you return to Pittsburgh from Seattle?
U9: *no i will go to Los Angeles*
S10: Traveling to Los Angeles. on what date
U11: *Sorry can we go back to talk about the previous leg again*
S12: For the first leg, you are on a United flight 1446 departing Pittsburgh at ten forty five A.M. tomorrow, arriving in Seattle at one fifty seven P.M. Do you want to take that?
U13: *actually i want to leave on wednesday*
S14: a flight on wednesday.
S15: I have a flight departing Pittsburgh at ten forty five a.m., arrives Seattle at one fifty seven p.m. Is that OK?
U16: *Yes*

Figure 19.1 The travel domain: a fragment from a successful conversation between a user (U) and the Communicator system (S) of Xu and Rudnicky (2000).

Call routing: ATT HMIHY

S: How may I help you?

U: *can you tell me how much it is to tokyo?*

S: You want to know the cost of a call?

U: *yes that's right*

S: Please hold on for rate information.

S: How may I help you?

U: *what's the area code for chicago?*

S: You want an area code.

U: *(silence)*

S: The area code for Chicago is 312.

Figure 19.2 The call routing/telephone information domain: two actual conversations between a user (U) and the AT&T How May I Help You (HMIHY) system (S) (Gorin et al., 1997).

A tutorial dialogue: ITSPOKE

S: I didn't understand. What force acts on all objects within close proximity to earth?

U: *uh the force of gravity*

S: Fine. Besides the Earth's gravitational force, are there any other forces acting on the person, keys, and elevator after the release?

U: *no*

S: Yep. Thus, keep in mind that after the release, the only force on the person, keys, and elevator is the force of gravity. If gravity is the only force acting on an object, will it be moving or staying still?

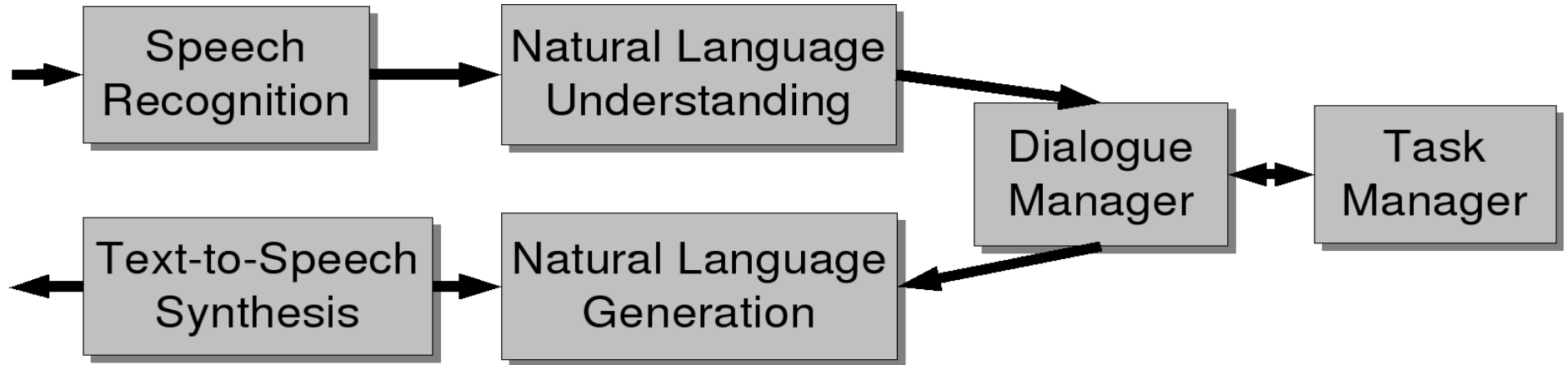
U: *it will be moving*

S: Fine. Not only are the person, keys, and elevator moving, they have only gravitational forces on them. When an object is falling and has only gravitational force on it, it is said to be in what?

U: *freefall*

Figure 19.3 The tutoring domain: part of an actual conversation between a student user (U) and the ITSPOKE system (S) of (Litman and Silliman, 2004), based on the Why2-Atlas text-based tutoring system (?).

Dialogue System Architecture



Speech Recognition

- **Refers to the technologies that enable computing devices to identify the sound of human voice.**

List all the Auburn University orders.



Automated Speech Recognition engine

- **Standard ASR engines**
 - Speech to words
- **But specific characteristics for dialogue**
 - Language models could depend on where we are in the dialogue
 - Could make use of the fact that we are talking to the same human over time.
 - Barge-in (human will talk over the computer)
 - Confidence values

Speech Recognition

- **Continuous Recognition**

- Allows a user to speak to the system in an everyday manner without using specific, learned commands.

- **Discrete Recognition**

- Recognizes a limited vocabulary of individual words and phrases spoken by a person.

Speech Recognition

- **Word Spotting**
 - Recognizes predefined words or phrases.
 - Used by discrete recognition applications.
 - * **“Computer I want to surf the Web”**
 - * **“Hey, I would like to surf the Web”**

Speech Recognition

- **Voice Verification or Speaker Identification**

- Voice verification is the science of verifying a person's identity on the basis of their voice characteristics.
- Unique features of a person's voice are digitized and compared with the individual's pre-recorded "voiceprint" sample stored in the database for identity verification.
- It is different from speech recognition because the technology does not recognize the spoken word itself.

Speech Synthesis

- Refers to the technologies that enable computing devices to output simulated human speech.

James, here are the Auburn University orders.



Speech Synthesis

- **Formant Synthesis**

- Uses a set of phonological rules to control an audio waveform that simulates human speech.
- Sounds like a robot, very synthetic, but getting better.

Speech Synthesis

- **Concatenated Synthesis**

- Uses computer assembly of recorded voice sounds to create meaningful speech output.
- Sounds very human, most people can't tell the difference.

Concatenative Speech Synthesis

- Output waveform generated by concatenating segments of pre-recorded speech corpus.
- Concatenation at phrase, word or sub-word level.

Synthesis Examples

The **third** ad is a **1996 black Acura Integra** with **45380** miles.
The price is **8970** dollars. Please call **(404) 399-7682**. 

labyrinth
abracadabra
obligatory → **laboratory** 

compassion
disputed
cedar city
since
giant
since → **computer**
science 

Continental flight **4695** from **Greensboro** is expected in
Halifax at **10:08 pm** local time. 

Uses of Speech Technologies

- **Interactive Voice Response Systems**
 - Call centers
- **Medical, Legal, Business, Commercial, Warehouse**
- **Handheld Devices**
- **Toys and Education**
- **Automobile Industry**
- **Universal Access (visual/physical impaired)**

Segment-Based Speech Recognition

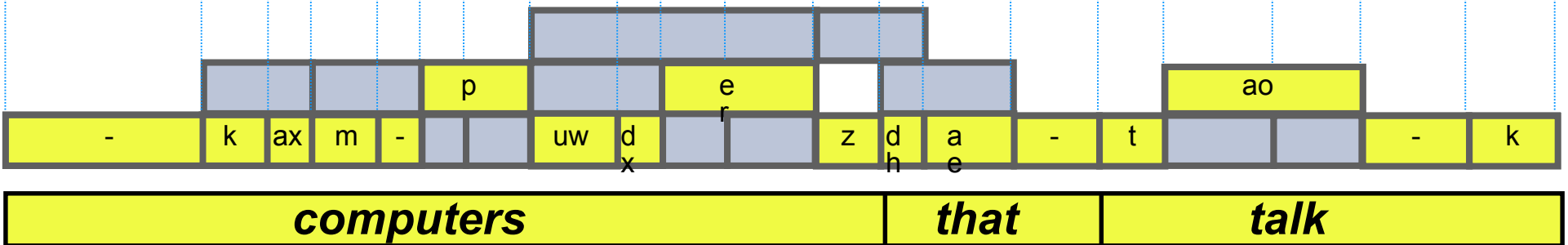
Waveform



Frame-based measurements (every 5ms)

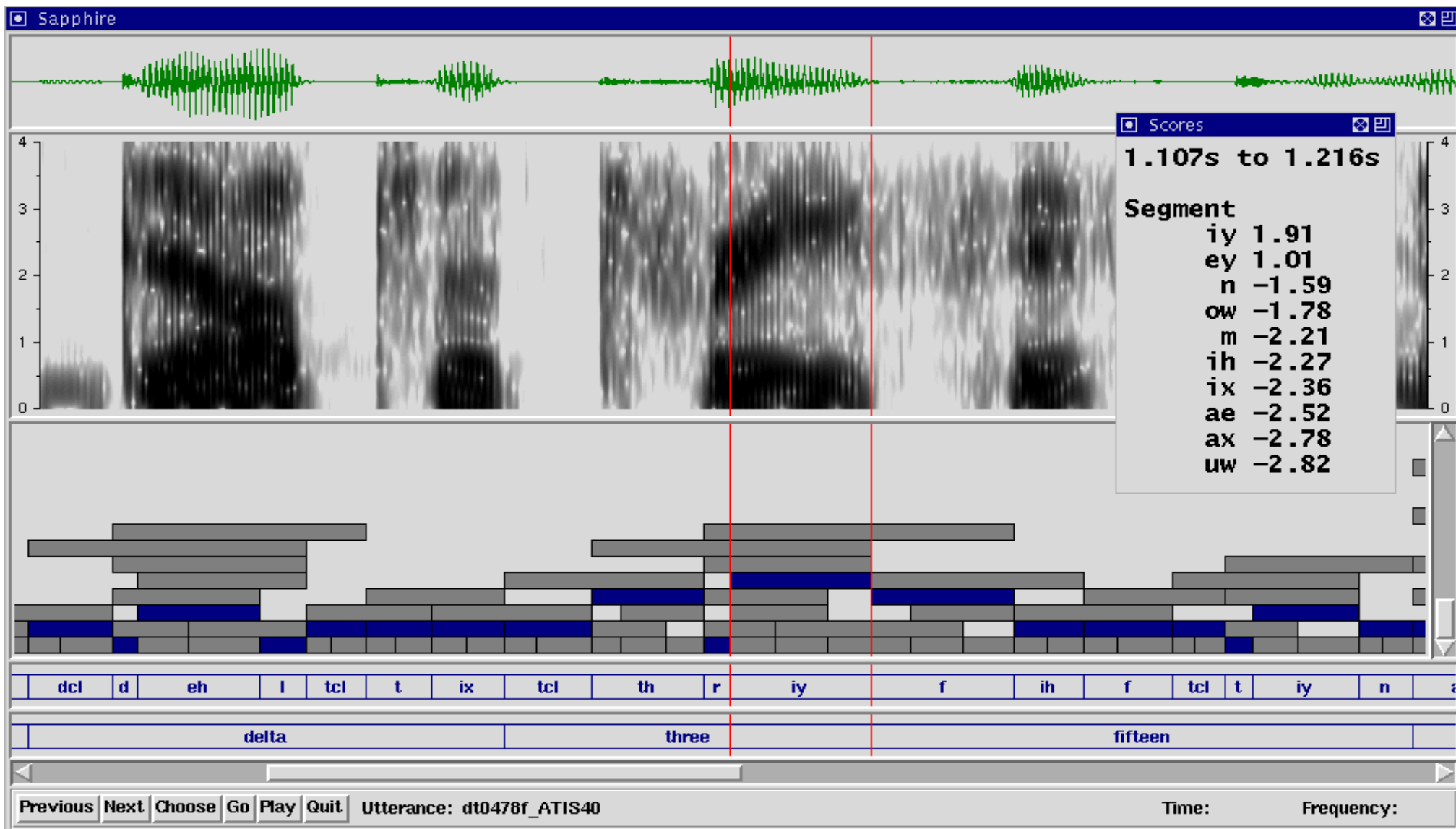


Segment network created by interconnecting spectral landmarks



Probabilistic search finds most likely phone & word strings

Segment-Based Speech Recognition



Speech Recognition: Dimensions of Success

- **Size of vocabulary: A few words to tens of thousands**
- **Accuracy, recognition percentage**
 - >99%, >95%, >75%
- **Repeatability of performance**
- **Cost**
- **Speaker-dependent vs. speaker-independent**
- **Training required or not required**
- **Location of microphone**
- **Acoustic environment, quiet or noisy environment**
- **Discrete words or continuous speech**

Language Model

- **Language models for dialogue are often based on hand-written Context-Free or finite-state grammars.**
- **Why? Because of need for understanding; we need to constrain user to say things that we know what to do with.**

Language Models for Dialogue (2)

- **We can have LM specific to a dialogue state**
- **If system just asked “What city are you departing from?”**
- **LM can be**
 - City names only
 - FSA: (I want to (leave|depart)) (from) [CITYNAME]
- **A LM that is constrained in this way is technically called a “restricted grammar” or “restricted LM”**

Barge-in

- **Speakers barge-in**
- **Need to deal properly with this via speech-detection, etc.**

Natural Language Understanding

- Or “NLU”
- There are many ways to represent the meaning of sentences
- For speech dialogue systems, most common is “Frame and slot semantics”.

An example of a frame

- Show me morning flights from Boston to SF on Tuesday.
- **SHOW:**
- **FLIGHTS:**
- **ORIGIN:**
- **CITY: Boston**
- **DATE: Tuesday**
- **TIME: morning**
- **DEST:**
- **CITY: San Francisco**

How to generate this semantics?

- Many methods,
- Simplest: “**semantic grammars**”
- **CFG in which the left hand side of rules is a semantic category:**
 - LIST -> show me | I want | can I see|...
 - DEPARTTIME -> (after|around|before) HOUR | morning | afternoon | evening
 - HOUR -> one|two|three...|twelve (am|pm)
 - FLIGHTS -> (a) flight|flights
 - ORIGIN -> from CITY
 - DESTINATION -> to CITY
 - CITY -> Boston | San Francisco | Denver | Washington

Semantics for a sentence

- **LIST FLIGHTS ORIGIN**
- **Show me flights from Boston**

- **DESTINATION DEPARTDATE**
- **to San Francisco on Tuesday**

- **DEPARTTIME**
- **morning**

Frame-filling

- **We use a parser to take these rules and apply them to the sentence.**
- **Resulting in a semantics for the sentence**
- **We can then write some simple code**
- **That takes the semantically labeled sentence**
- **And fills in the frame.**

Problems with any of these semantic grammars

- **Relies on hand-written grammar**
 - Expensive
 - May miss possible ways of saying something if the grammar-writer just doesn't think about them
- **Not probabilistic**
 - In practice, every sentence is ambiguous
 - Probabilities are best way to resolve ambiguities
 - Good statistical models can be learned and built using existing techniques.

Generation and TTS (Text-to-speech)

- **Generation component**
 - Chooses concepts to express to user
 - Plans out how to express these concepts in words
 - Assigns any necessary prosody to the words
- **TTS component**
 - Takes words and prosodic annotations
 - Synthesizes a waveform

Generation Component

- **Content Planner**

- Decides what content to express to user
 - * **(ask a question, present an answer, etc)**
- Often merged with dialogue manager

- **Language Generation**

- Chooses syntactic structures and words to express meaning.
- Simplest method
 - * **All words in sentence are prespecified!**
 - * **“Template-based generation”**
 - * **Can have variables:**
 - What time do you want to leave CITY-ORIG?
 - Will you return to CITY-ORIG from CITY-DEST?

HCI constraints on generation for dialogue: “Coherence”

- Discourse markers and pronouns (“Coherence”):

- (1) Please say the date.

- ...

- Please say the start time.

- ...

- Please say the duration...

- ...

- Please say the subject...

Bad!

- (2) First, tell me the date.

- ...

- Next, I’ll need the time it starts.

- ...

- Thanks. <pause> Now, how long is it supposed to last?

- ...

- Last of all, I just need a brief description

Good!

Dialogue Manager

- **Controls the architecture and structure of dialogue**
 - Takes input from Automated Speech Recognition/Natural Language Understanding components
 - Maintains some sort of state
 - Interfaces with Task Manager
 - Passes output to Natural Language Generation/Text-to-speech modules

User-centered dialogue system design

1. Early focus on users and task:

- interviews, study of human-human task, etc.

2. Build prototypes:

- Wizard of Oz systems

3. Iterative Design:

- iterative design cycle with embedded user testing

Dialogue system Evaluation

- **Whenever we design a new algorithm or build a new application, need to evaluate it**
- **How to evaluate a dialogue system?**
- **What constitutes success or failure for a dialogue system?**

Task Completion Success

- **% of subtasks completed**
- **Correctness of each questions/answer/error msg**
- **Correctness of total solution**

Task Completion Cost

- **Completion time in turns/seconds**
- **Number of queries**
- **Turn correction ration: number of system or user turns used solely to correct errors, divided by total number of turns**
- **Inappropriateness (verbose, ambiguous) of system's questions, answers, error messages**

User Satisfaction

- **Were answers provided quickly enough?**
- **Did the system understand your requests the first time?**
- **Do you think a person unfamiliar with computers could use the system easily?**

Speech interface guidelines

- **Speech recognition is errorful**
- **System state is often opaque to the user**
- **<http://www.speech.cs.cmu.edu/air/papers/SplnGuidelines/SplnGuidelines.html>**

1. Errors

- **Speech based interfaces are error-prone due to speech recognition.**
 - Humans aren't perfect speech recognizers, therefore, machines aren't either.
- **Goal: Reduce the number and severity of errors.**

1. Errors

- **Use Specific Error Messages**
- **Limit Background Noise**
- **Allow the User to Turn Off the Input Device**
- **Provide an Undo Capability**
- **Use Auditory Icons**
- **Use Multi-Modal Cues for Errors If Applicable**
- **Don't Assume People Hear Everything**

Use Specific Error Messages

- **Bad Example**

- System: “Say the departure date.”
- User: “Tomorrow.”
- System: “Say the departure date.”
- User: “I want to travel tomorrow.”
- System: “Say the departure date.”

Use Specific Error Messages

- **Good Example**

- System: “Say the departure date.”
- User: “Tomorrow.”
- System: “I don’t understand that date. Say the month, date and year. For example, say October 13th, 2003.”
- User: “July 1st, 2003.”

Limit Background Noise

- **Background noise is input.**
- **Computer hears the background, not the user.**

Allow the User to Turn Off the Input Device

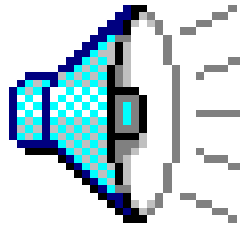
- **This reduces background noise errors.**
- **For Speech Based Interfaces, allow the user to place the system in an ignore mode**
 - System ignores input until a keyword is spoken, i.e. “I am back”.

Provide an Undo Capability

- **Build in ways for users to cancel out, go back and undo actions.**

Use an Auditory Icon

- **Auditory Icons are sound clips with a message.**
- **When errors occur, play an auditory icon to notify the user.**



Use Multi-Modal Cues for Errors If Applicable

- **Use more than one mode to signal an error, if possible.**
- **Play an auditory icon, display a message and speak a message.**

Don't Assume People Hear Everything

- **Just because the system spoke it, doesn't mean the user heard it.**
- **Say important information first or last to improve the likelihood of it being heard.**

2. Feedback

- **During HCI, the user needs feedback from the computer.**
- **When a user issues a command, the system should acknowledge that the user has been heard.**
- **Users also want feedback when the system is busy.**

2. Feedback

- **Supply Alternative Guesses**
- **Acknowledge the User's Speech**
- **Show When It Is the User's Turn to Talk**
- **Allow for Verification**
- **Use Non-Speech Audio for Transitions**
- **Use In-Progress Messages**

Supply Alternative Guesses

- **Users may say one word, but the computer hears a different word. (IDEAL SOLUTION)**
 - i.e. User says “Boston” and the computer hears “Austin”.
 - The computer should respond “Did you say Austin or Boston?”
- **This is easier said than done because you have to know all the words that sound alike in order to accomplish this for a large vocabulary.**

Supply Alternative Guesses

- **Repeat what the user said and allow the user to correct what was recognized. (REAL SOLUTION)**
 - i.e. User says “Boston” and the computer hears “Austin”.
 - The computer should respond “You said Boston, is that correct?”

Acknowledge the User's Speech

- **When the user speaks, provide feedback that she was heard.**
 - Auditory Icon
 - Go to the next option
 - If the next option is time consuming, let the user know in advance. i.e. “I heard you, let me process your request”

Show When It Is the User's Turn to Talk

- In a multi-modal user interface, provide the user with a visual cue that the computer is listening.



Show When It Is the User's Turn to Talk

- **In a speech based interface, provide the user with:**
 - Prompt
 - Auditory Icon

Allow for Verification

- **Users tend to verify more when using a speech interface versus a visual interface.**
- **Speech based interfaces should allow the user to verify what is happening and what has happened.**

Use Non-Speech Audio for Transitions

- **When the user issues a command that requires a transition, play an auditory icon to acknowledge the transition is under way.**
- **Avoid non-speech feedback that sounds like equipment noise.**

Use In-Progress Messages

- **If there is more than a 3 seconds delay between when the user issues a command and the system responds, issue an in-progress message.**
- **For best results, your in-progress messages should be informative.**
 - i.e. tell the user their position in the wait queue when it changes.

Use In-Progress Messages

- **Playing a musical auditory icon in the background doesn't work alone, but it is better than nothing.**
- **Combine the verbal message with music to have the best effect.**

3. Confirmations

- **Confirmations are questions you ask of the user to be sure that the user has been heard correctly.**



3. Confirmations

- **Use Confirmations Appropriately**
- **Ask for Clarifying Information**
- **Use Confirmations for Destructive or Predictable Actions**
- **Be Specific**

Use Confirmations Appropriately

- **Don't over confirm**
 - You could overdo the confirmations by asking for a confirmation for every input.
- **You have to balance the cost of making an error with the extra time and annoyance in requiring the user to confirm a lot of statements.**

Ask for Clarifying Information

- **If the expected response has more than one known response, then you may want to clarify what the user said.**
- **i.e. “Do you want to set up an appointment or contact the person by phone”**

Use Confirmations for Destructive or Predictable Actions

- **If the user's action is destructive, delete files, require a confirmation.**
- **If the user's input prone to errors, require a confirmation.**
 - i.e. the grammar has a lot of sound alike words.

Be Specific

- **If the system doesn't recognize what was spoken, be specific about what you need.**
- **i.e.**
 - “Please repeat the date again” vs. “Please repeat”
 - “Do you mean December 3rd?” is not a good example, unless you are fairly confident.

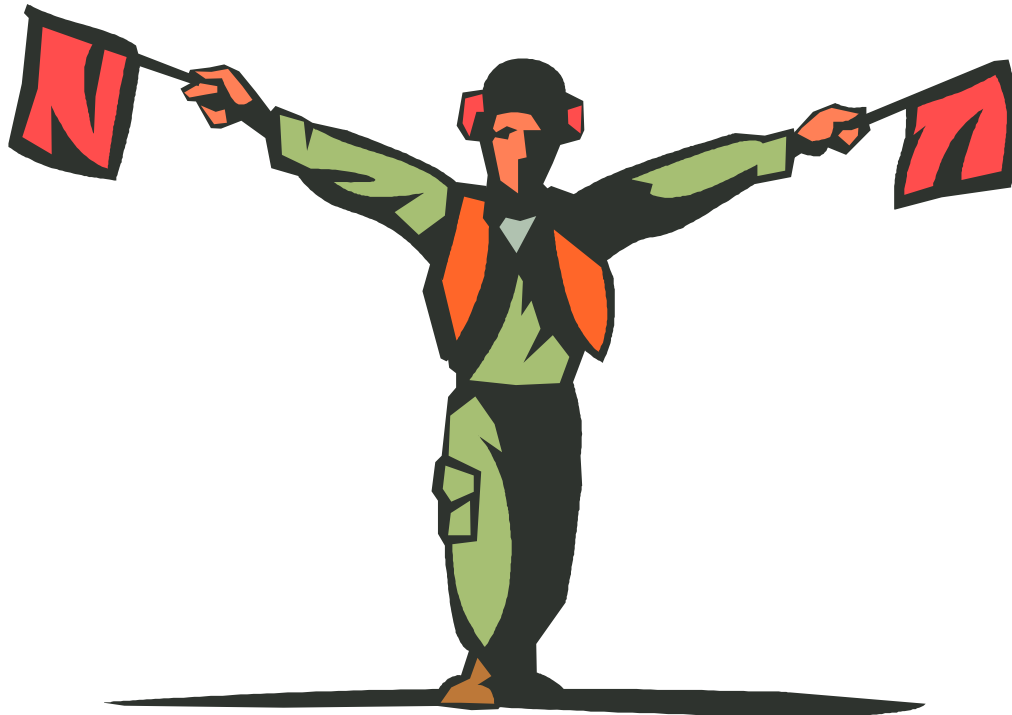
4. Command-and-Control

- **Speech based interfaces that recognize a limited vocabulary of individual words and phrases spoken by the user.**



5. Command-and-Control

- **User Constraints**
- **Be Brief**



User Constraints

- **Limit the user's input through specific prompts.**



User Constraints

- **Bad dialouge:**
 - System: “Welcome to the XYZ Company. We look forward to servicing your travel needs. What are the dates of travel that you would like me to check for?”
 - User: “We are interested in traveling the first week of July, say July 1st to July 5th”.
- **The system’s statement is too open. This is a natural dialogue that humans understand.**

User Constraints

- **Good dialogue:**

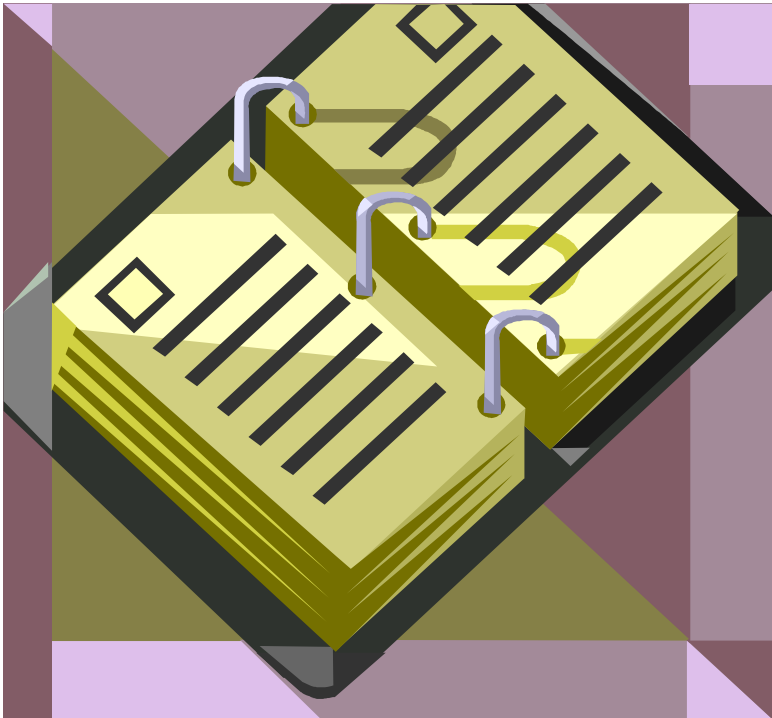
- System: “Welcome to the XYZ Company. Say the departure date of travel. For example, say October 1st, 2003.”
- User: “July 4th, 2003”
- System: “Thank you. Say the return date.”

Be Brief

- **People model the length of system speech.**
 - If the system is lengthy, then the user will tend to be lengthy.
- **The length of user speech is directly proportional to the number of recognition errors.**
 - The longer you speak, the chances of errors increases.

Allow Relative Dates

- **For example:**
 - next Friday, yesterday, tomorrow, next week, next month, etc.



Avoid Long Pauses

- **People don't like dead air in conversation.**
- **Use auditory icons or speech to avoid long pauses.**

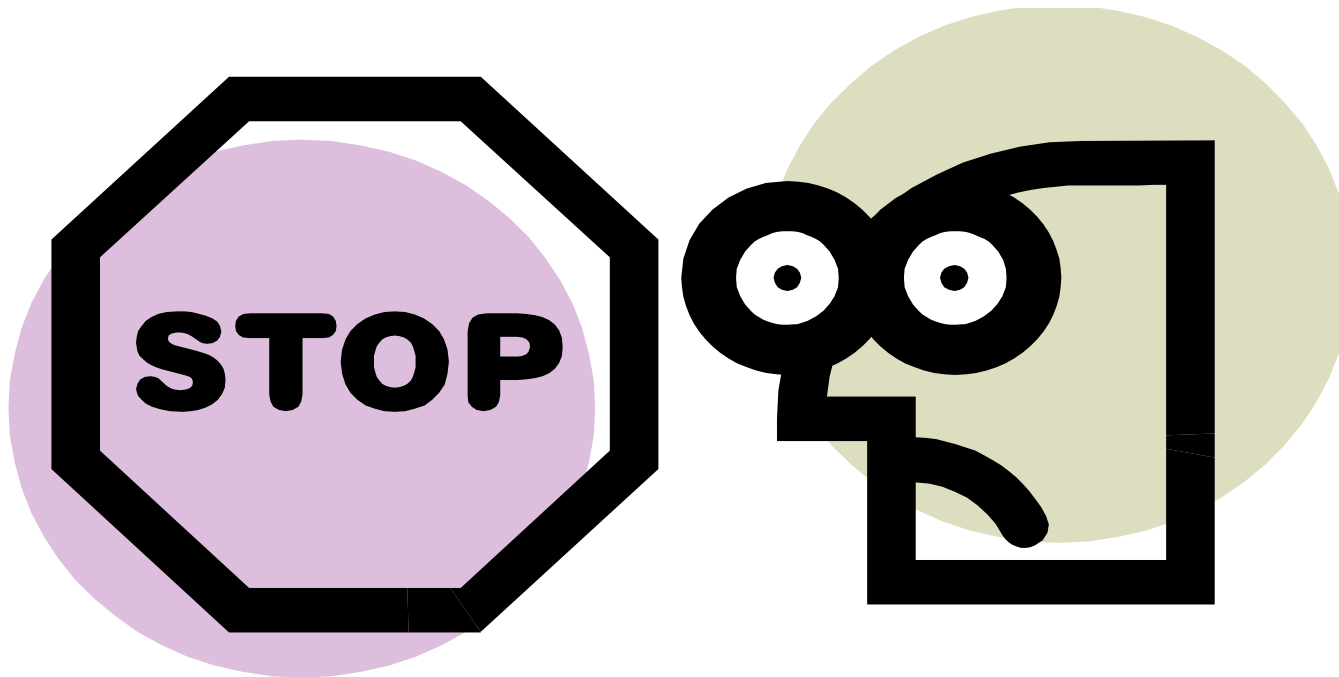


Choose an Appropriate Speed

- **If the systems speaks fast, then the users will speak fast.**
- **Users will mimic the speed of the computer.**

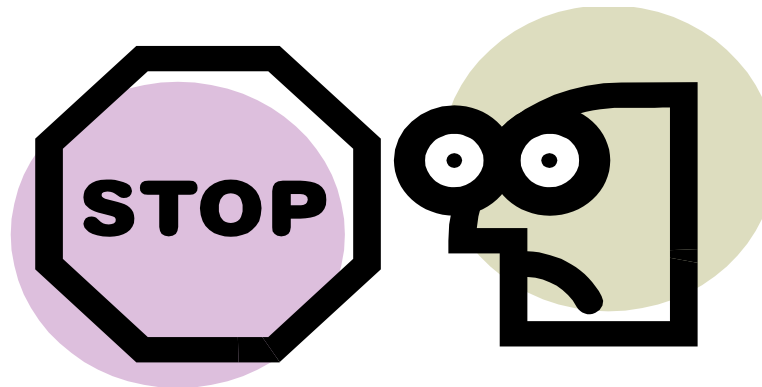
Use Barge-In

- **Allow users to interrupt the computer's speech. This is barge-in.**

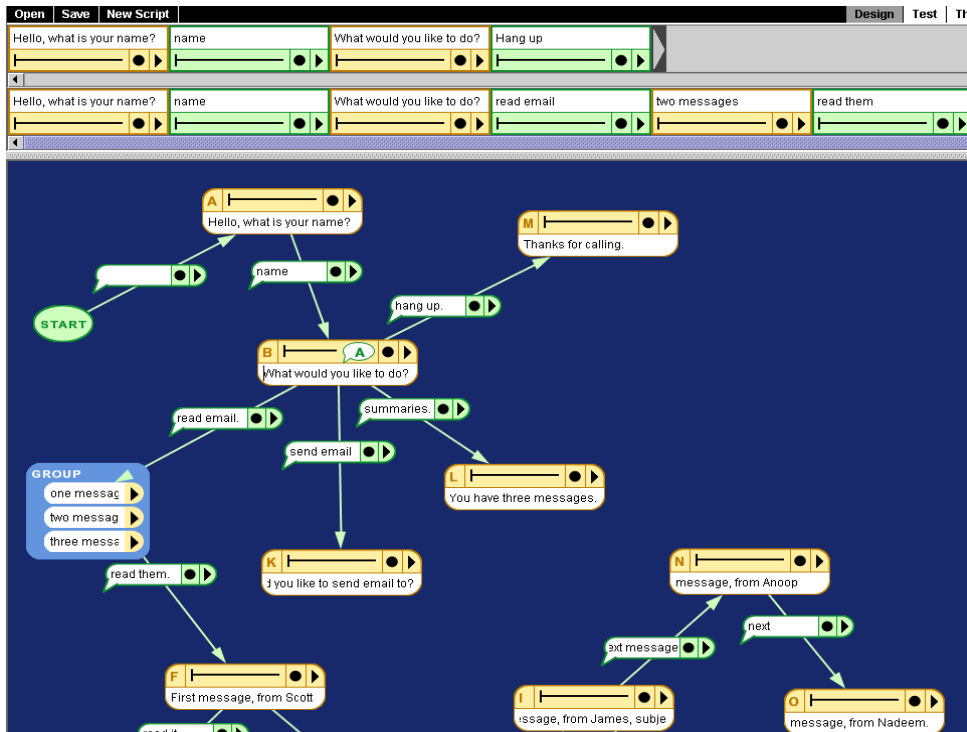


Use Barge-In

- **When a group of users have adapted to a speech based interface and they barge-in, they barge-in with 2 seconds of the introduction (maybe less).**
- **So, if you have to change the options of the speech based interface, how do you notify the users if they barge-in within 1 second?**



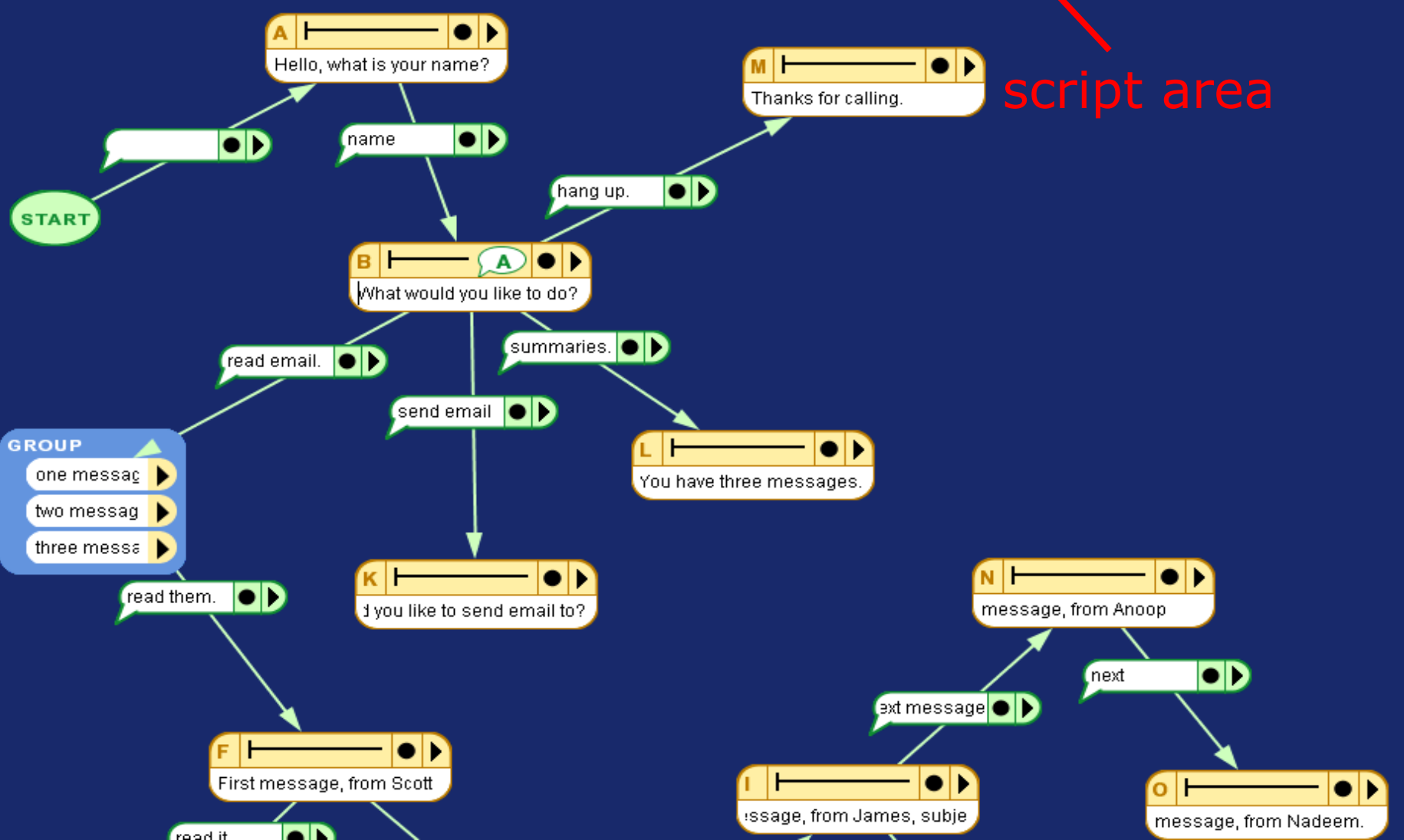
SUEDE: Low-fi Prototyping for Speech-based UIs



- **Built-in iterative design**
 - design – test – analysis
 - fast -> no real recognition
- **Support design practice**
 - example scripts
 - Wizard of Oz (WoZ)
- **Handle needs of real UIs**
 - error simulation

Open Save New Script Design Test Think

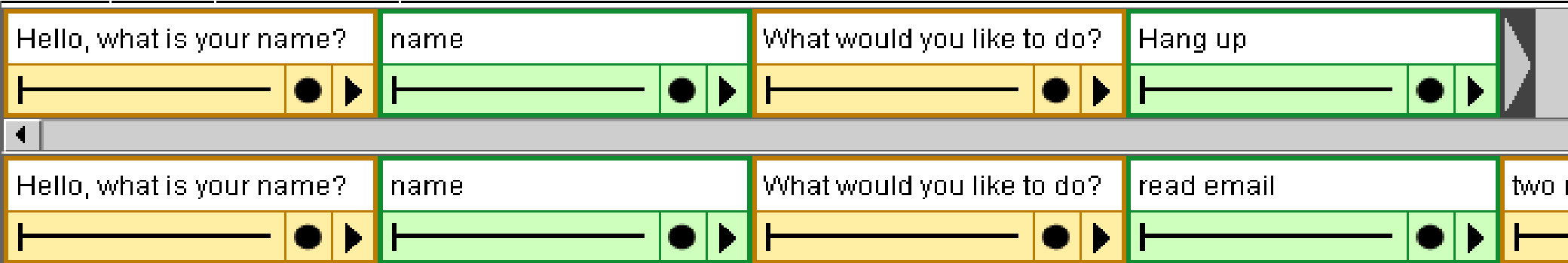
Hello, what is your name?	name	What would you like to do?	Hang up
Hello, what is your name?	name	What would you like to do?	read email
			two messages
			read them
			First message

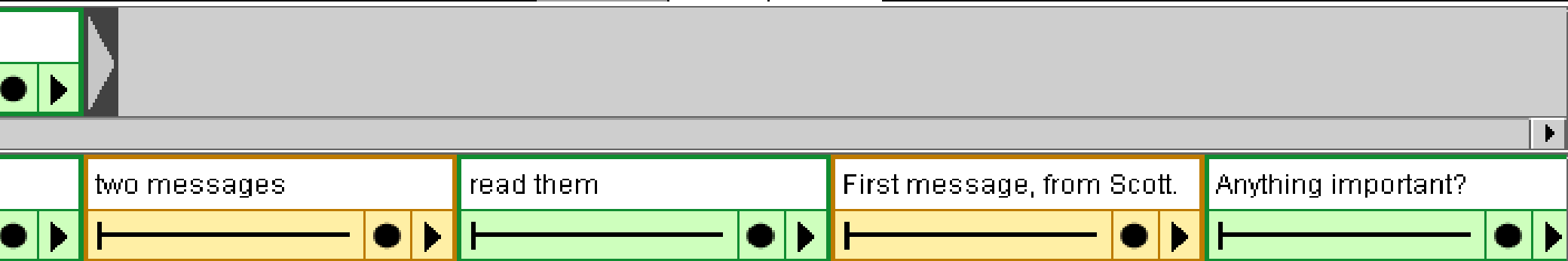


script area

machine prompt

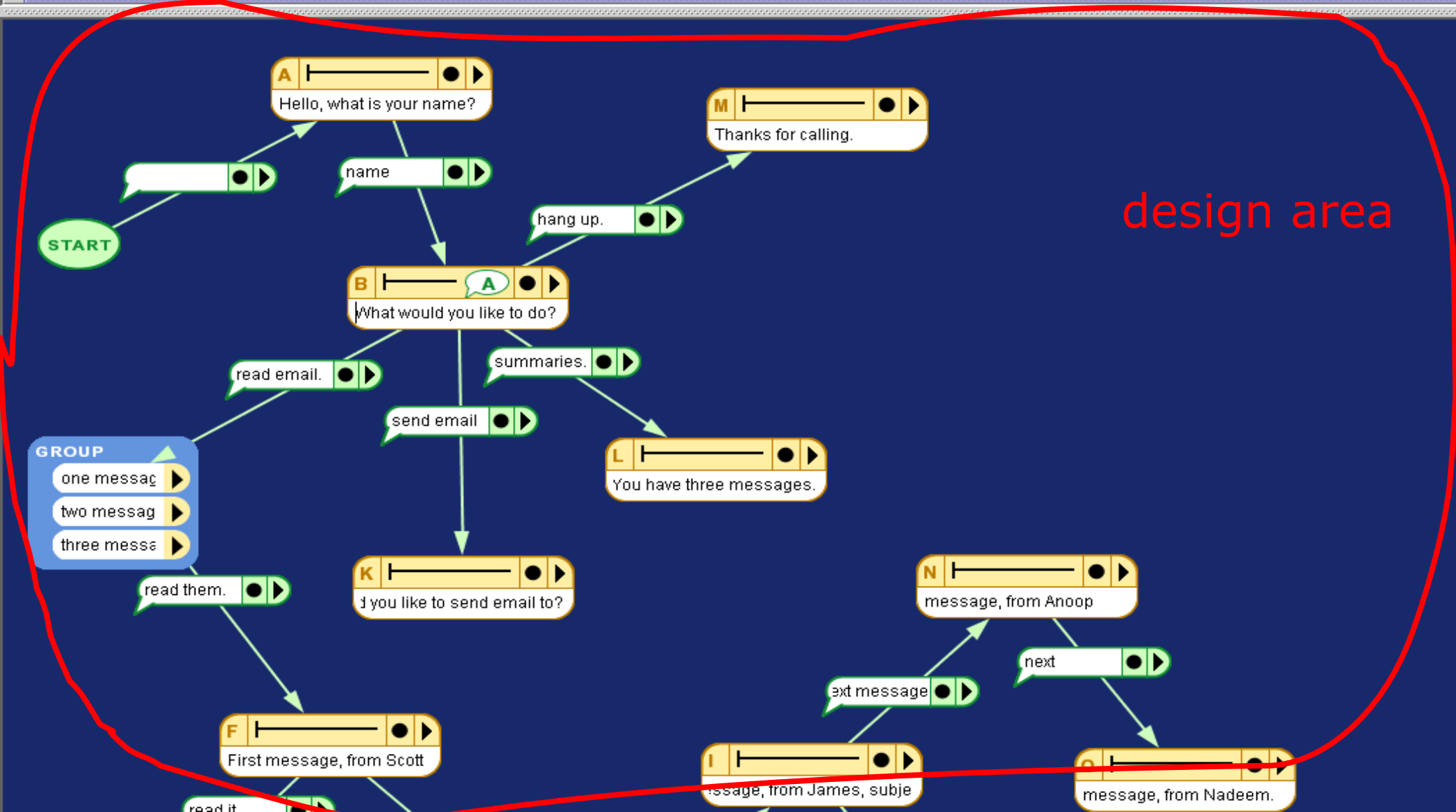
user response



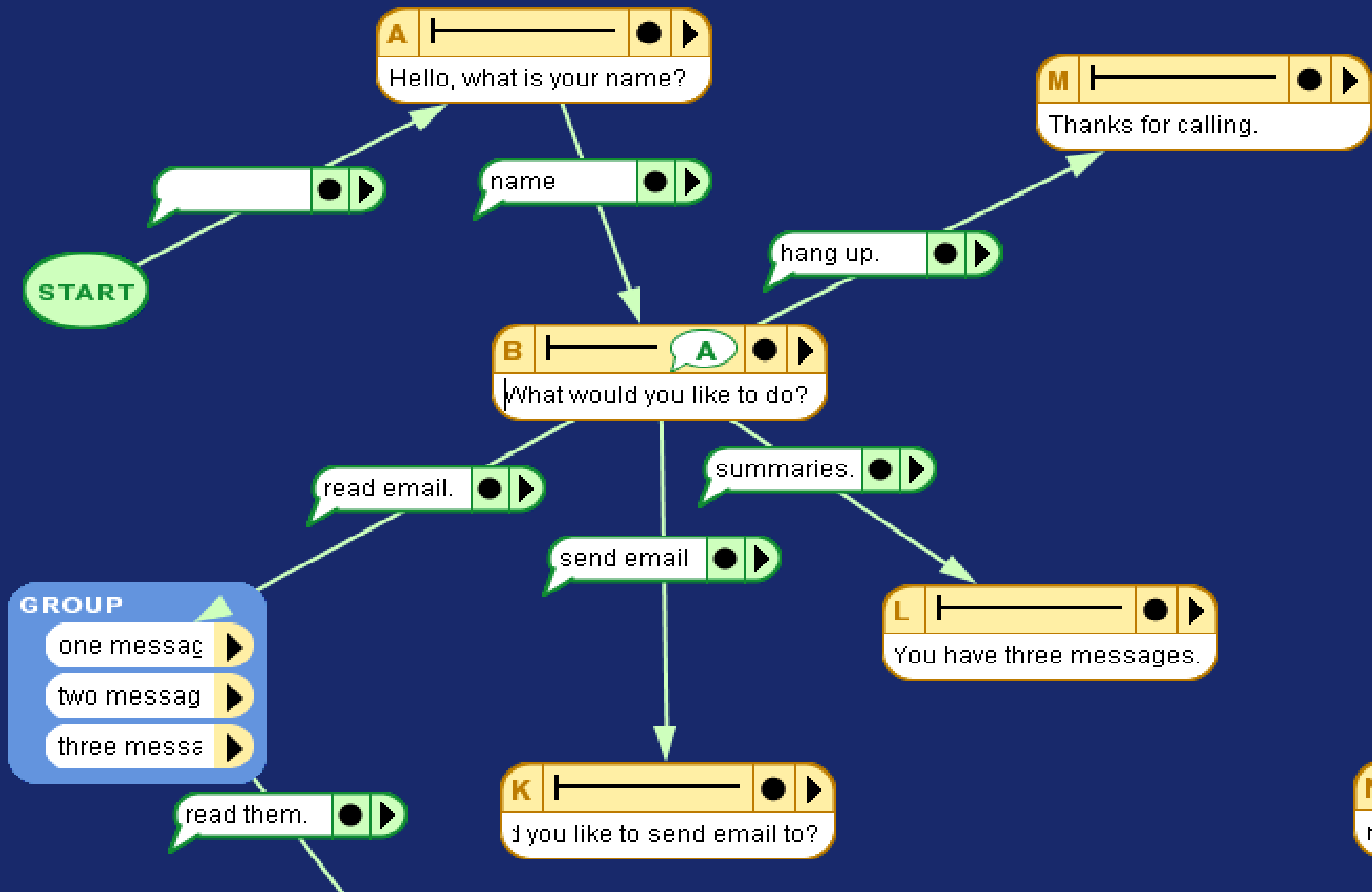


Hello, what is your name?	name	What would you like to do?	Hang up
▶	▶	▶	▶

Hello, what is your name?	name	What would you like to do?	read email	two messages	read them	First message,
▶	▶	▶	▶	▶	▶	▶



design area



SUEDE Summary

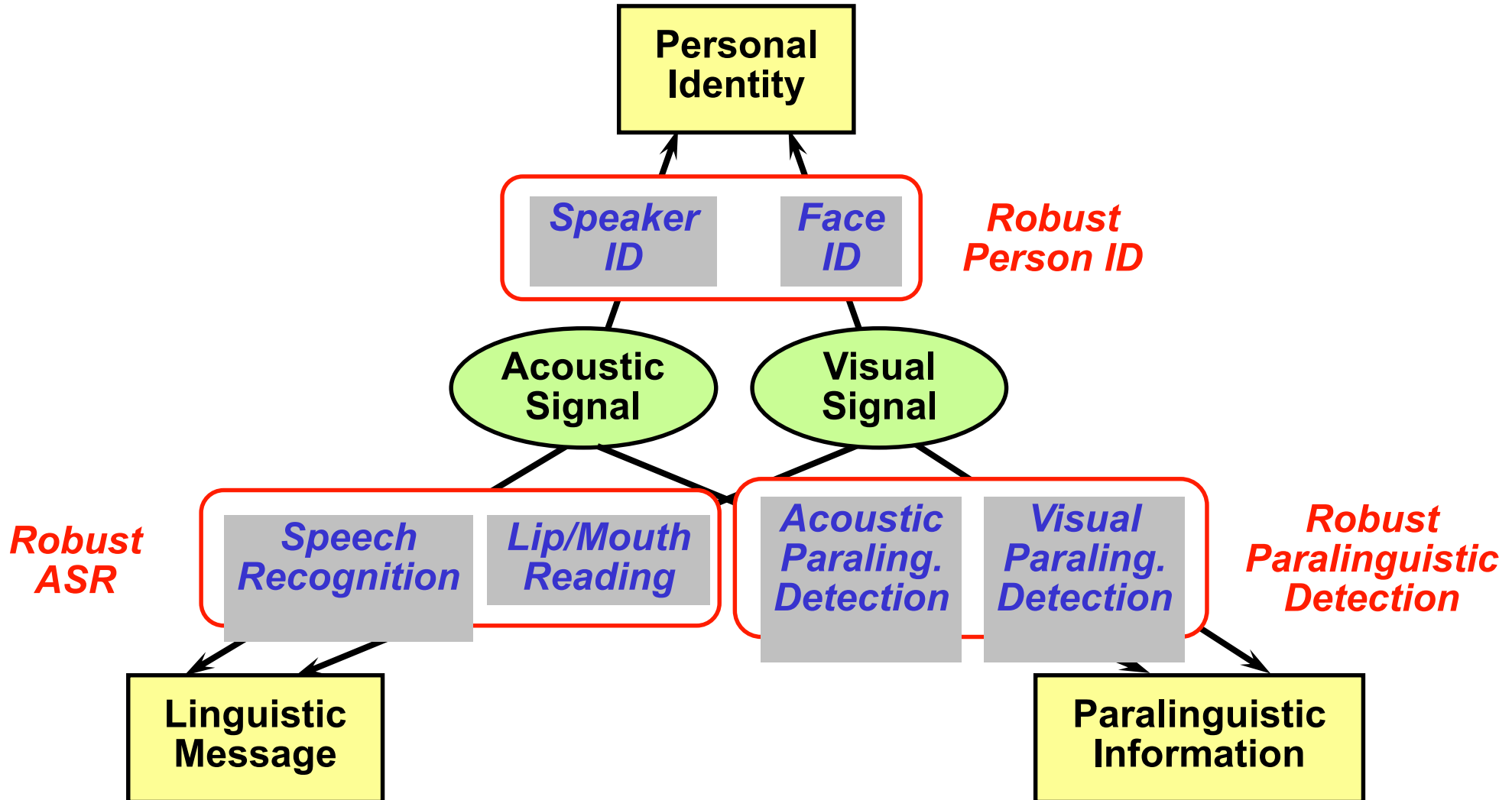
- **Speech is an important mode for info access in the field**
- **SUEDE supports speech-based UI design**
 - moving from concrete examples to abstractions
 - embeds iterative design w/ design-test-analyze
- **Designers using SUEDE need not be experts in speech recognition technology**

Audio Visual Integration

- **Audio and visual signals both contain information about:**
 - Identity of the person: *Who is talking?*
 - Linguistic message: *What's (s)he saying?*
 - Emotion, mood, stress, etc.: *How does (s)he feel?*
- **The two channels of information**
 - Are often inter-related
 - Are often complementary
- – Must be consistent

Integration of these cues can lead to enhanced capabilities for future human computer interfaces

Audio Visual Symbiosis



Assignment #7 (Reading #6)

- **JUPITER: A Telephone-Based Conversational Interface for Weather Information**
by Zue et al.
- **Appeared in IEEE Transactions on Speech and Audio Processing, Vol. 8, No. 1, January 2000**