Multimodal Interaction

Lesson 7
Speech Interaction

Maria De Marsico
demarsico@di.uniroma1.it

Speech-based human-computer interaction

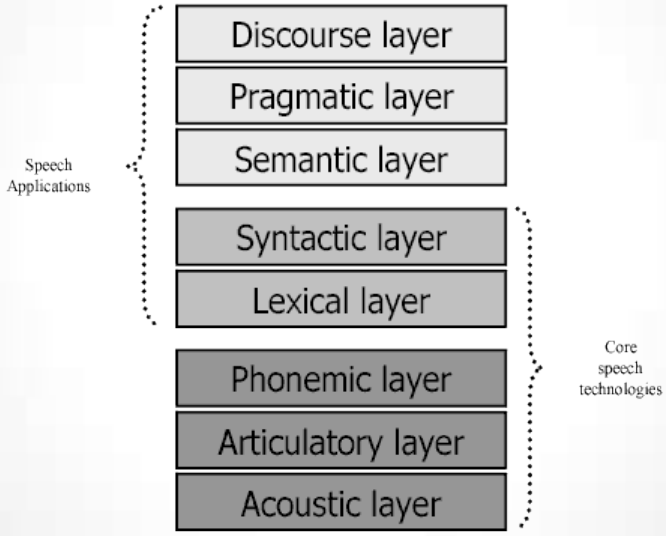
- From: **Design and Development of Speech Interfaces**
  http://www.cs.uta.fi/hci/spi/ddsi/

Levels of speech communication

- A useful categorization contains **eight** levels, which are organized into **three** layers.
- On the **bottom** layer, speech is considered as **sound** (acoustic), from human speech **production point of view** (*articulatory*) and on the level of single sounds the are the **basis of each spoken language** (*phonemic*).
- The **middle** layer considers language on the level of **words** (*lexical*) and how single words can be **inflected and combined** in sentences (*syntactic*).
- The **top** layer considers the **meaning** of words **independently** (*sematic*) and in **context** (*pragmatic*) and how single **utterances combine** into meaningful communication (*discourse*).

Maria De Marsico - demarsico@di.uniroma1.it

Levels of speech communication



Maria De Marsico - demarsico@di.uniroma1.it

Levels of speech communication

- The core speech technologies (speech **recognition**, speech **synthesis**) belong mainly to the **bottom** layer.
- Speech **application design** deal mainly with the **top** layer (discourse, pragmatic and semantic levels). The **middle** layers (syntactical and lexical levels) are **needed** both in technology and application areas.

Benefits of speech in hhi

- A classic study by Chapanis [1975] showed that speech was the **most efficient** communication modality in **interactive problem solving** tasks between humans.
- Compared to other modalities speech was superior: it was more than twice as efficient as writing, the next most efficient modality.
- Chapanis also found that unimodal speech communication was only slightly slower than multimodal communication in which all other modalities were used.

Benefits of speech in hci

- **Naturalness**.
- Ability to bring **more bandwidth** to the interaction when a multimodal interface is used = When multiple concurrent tasks are performed **simultaneously more efficiency** can be gained if **multiple modalities** can be used **concurrently**.
- Spoken utterances can be **very expressive** = By using speech it is possible to perform operations which may be hard to do with other modalities: many tasks which are difficult to do with the direct manipulation paradigm can be easier to do with spoken commands.
  - Example: selection of similar (e.g. color or shape) objects from a large set of objects that are scattered around the display area, or that cannot be displayed at once. By using (spoken) language homogeneous objects can be selected with a single utterance.

Restrictions of speech in hci

- We **cannot** use **all** human-human communication methods in speech interfaces = Some common phenomena, such as overlapping speech, cause **serious** problems in human-computer interaction.
- One major problem is the **memory limitations** of users. When no visual information is available, users need to memorize all the meaningful information = Users need to **remember** the **dialogue state** and **all relevant information** that the system has provided.
- Users also need to know **how to speak to the system**: there is **no spoken language interface** which can understand **unconstrained speech**.
- For non-native speakers it may be **easier to remember commands than to know how to speak "correctly" or "naturally" to the system.**

Restrictions of speech in hci

- Speech is a slow, temporal and serial medium, all of which make speech outputs hard to design.
- When used in **non-private** environments the **public** nature of speech should be taken into account = Speech outputs should be adapted to the particular users and usage situations in order to safeguard the user's **privacy**.
- Speech is an **error prone** communication method, requiring explicit error correction in interaction.
- Language is also often **ambiguous**.

Speech is sequential

- The listener receives information in the order in which the speaker decides to present it.
- Everything that is presented with speech must be **first linearized**.
  - Example: tables, which are two-dimensional visual entities. One possible table linearization repeats column labels for each entry.
- The linearity limits not only the **order** in which the listener receives the information but also the **speed** and **level of detail** = While it is possible to skim through a text, the rate of speech can be increased only so far

Speech is temporal/transient

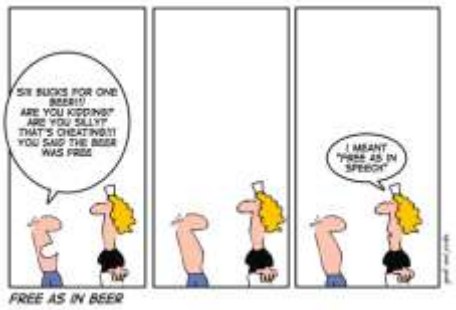
- The spoken message must be perceived **synchronously** with its production.
- When a word has been spoken, the speaker must **repeat** the word if the listener did not hear it. This requires spoken interaction to devote **reasonable resources** to **interactive error management** in which misheard and unheard information can be repeated and confirmed.
- Speech interaction also requires reasonable **mental resources**. While other modalities, such as vision and motor actions, can be used effectively while the user is speaking, this works only in **semiautomatic** tasks, such as walking, and, in most cases, driving a car. Things such as active reasoning interfere with the understanding and production of speech (we will return on this later)

Speech is error-prone

- Speech communication requires **error correction** since speech is transmitted over a noisy channel .
- In **face-to-face** dialogue, the distortion can be because of **external noises** and **insufficient sound volume**.
- In **machine-mediated** communication, the **technical components** contribute further to the degradation of signal = Microphones have their own limitations, and telephone lines, especially cellular phone connections, have rather low bandwidth.
- However, in most cases, the greatest variability in speech is caused by **speaker differences and variations within a single speaker**.
- This variability results in the need for **stochastic models in speech recognition**. The pronunciation of words differs greatly depending on the context - the surrounding words, the speaker's mental and physical state, etc.

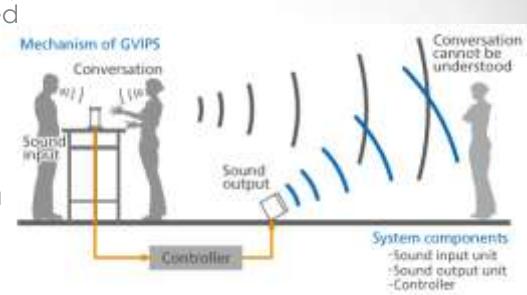
Speech is ambiguous

- Many words, phrases, and sentences can have **more than one meaning**.
- In human-computer dialogue, the ambiguities can be considered in the **design phase**.
- Since **speech recognition grammars** are often rather limited, the dialogue restriction can be designed so that ambiguities in user input remain minimal.
- Similarly, **system utterances** must be designed such that misunderstandings in the other direction should not arise.
- Because speech recognition errors can result in ambiguity, **longer user utterances** can easily result in enough misrecognized words to make determining the meaning of the utterance difficult or impossible.



Speech is public

- Speech sound spreads almost omnidirectionally, it can be heard by others who are present, even if the speech is not directed to them.
- When two people try to avoid being heard by others in public, they can **lower their voices** and whisper = This makes it **hard** also for one dialogue partner to hea what the other is saying.
- When machines are partners or mediators in dialogue, **headphones** can be used to listen to speech without the audio leaking to the ears of others. **However**, **one's own speech is still audible** to others. This limits the use of speech in confidential applications.



GVIPS (GLORY Voice Intelligent Protection System)
From: http://gloryeurope.com/en/groupinfo/technologies/others.html

Summary of pros and Cons of Speech Recognition

- Spoken language is effective for human-human interaction but often has **severe limitations** when applied to human-computer interaction.
- Speech is **slow** for presenting information, is **transient** and therefore **difficult to review or edit**, and interferes significantly with other cognitive tasks.
- **However**, speech has proved useful for store-and-forward messages, alerts in busy environments, and input-output for **blind** or **motor-impaired** users.

More on the Limits of Speech Recognition

- *Ben Shneiderman:* To improve speech recognition applications, designers must understand acoustic memory and prosody.
  (From: The Limits of Speech Recognition. COMMUNICATIONS OF THE ACM Vol. 43, No. 9, September 2000, pp. 63-65)
- **Prosody**: the rhythm, stress, and intonation of speech. Prosody may reflect various features of the speaker or the utterance: the **emotional** state of the speaker; the **form** of the utterance (statement, question, or command); the presence of **irony or sarcasm**; **emphasis, contrast, and focus**; or other elements of language that **may not be encoded** by grammar or choice of vocabulary.

Prosody

- Cons: The **emotive** aspects of prosody are **potent** for human-human interaction but may be **disruptive** for human-computer interaction
- Pros: The **syntactic** aspects of prosody, such as rising tone for questions, are important for a system's recognition and generation of sentences.

Affective computing and prosody

Human emotional expression is

- ***varied*** (across individuals),
- ***nuanced*** (subtly combining anger, frustration, impatience, and more),
- ***situated*** (contextually influenced in uncountable ways)

- Accurate simulation or recognition of emotional states is usually impractical.

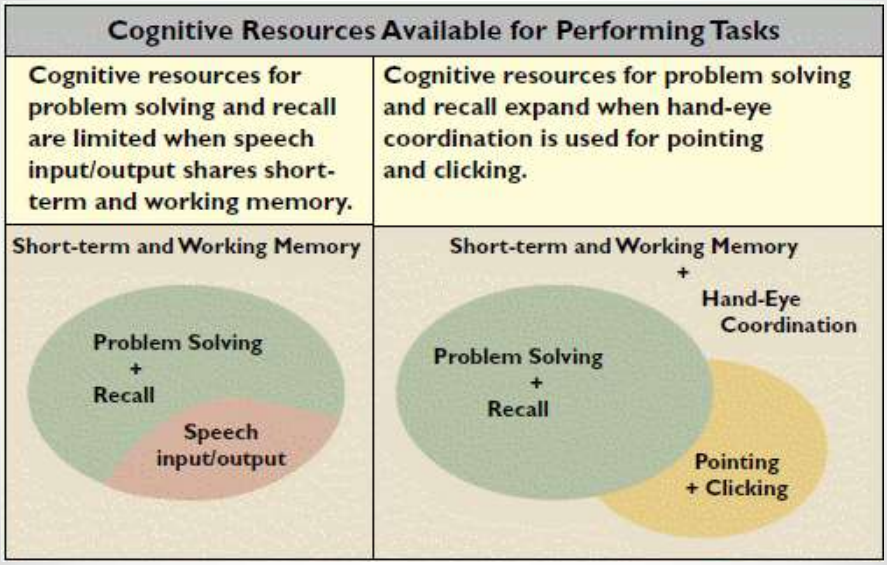
- Prosody control is provided in some markup-languages for speech synthesis , e.g. W3C Speech Synthesis Markup Language (**SSML**)(http://www.w3.org/TR/speech-synthesis/)

Speech and working memory

- Short-term and working memory are sometimes called **acoustic** or **verbal** memory.
- The part of the human brain that transiently holds chunks of information and solves problems **also** supports speaking and listening → Working on hard problems is best done without speaking or listening to someone.
- Physical activity is handled in another part of the brain → Problem solving is compatible with routine physical activities like walking and driving.
- Humans **speak and walk easily** but find it **more difficult** to **speak and think** at the same time

Maria De Marsico - demarsico@di.uniroma1.it

Speech and working memory



Maria De Marsico - demarsico@di.uniroma1.it

An experiment (Shneiderman 1993)

- 16 word-processor users were given the chance to issue voice commands for 18 tasks, including "page down", "boldface", "italic", and "superscript."
- For **most tasks**, this facility enabled a 12%–30% **speed-up**, since users could keep their hands on the keyboard and avoid mouse selections.
- One task required the **memorization** of mathematical symbols, followed by a "page down" command. The users then had to retype the symbols from memory.
- **Voice-command users had greater difficulty with this task than mouse users**. They repeatedly scrolled back to review the symbols, because speaking the commands appeared to interfere with their retention.

About dictation (see readings)

- In keyboarding, users can continue to think about how to refine their words **while** their fingers output an earlier version.
- In dictation, users may experience more **interference** between outputting their initial thought and elaborating on it (unless they read from a ready text).
- Developers of commercial speech-recognition software packages recognize this problem and often advise dictation of full paragraphs or documents, followed by a review or proofreading phase to correct errors.

Why to use speech?

- Speech may be **the only** possible modality …
- … it may be **the most efficient** modality …
- … or it may be **the most preferred** modality.
- Speech can be **supportive**, …
- … **alternative** …
- … or **substitutive** modality.

- In a **supportive** role speech can be used to achieve more bandwidth and make the interface more robust
- As an **alternative** modality speech may be suitable for some users, while other users may prefer other modalities. In this role speech can be used in situations when no other modalities are currently present.
- If speech **substitutes** some other modality (i.e. it is not the only possible modality for this situation), and it is not more efficient or preferred than the modality that it replaces, problems are likely and user satisfaction may decrease.

Speech applications

- Different classifications
- Main modality ←→ Multimodal applications
- Conventional speech applications (telephone applications, dictation system, voice portals) ←→ New application areas (pervasive computing, mobile applications).
- According to development and research questions:
  - Conventional
  - Multilingual
  - Multimodal
  - Pervasive.
- In addition, special user groups must be considered
- There is a wide range of research problems associated with speech applications beyond the speech recognition.

Conventional applications

- The first speech applications were telephone-based **interactive voice response** (**IVR**) systems, which used speech outputs and telephone keys for interaction to replace human operators.
- Based on **Dual-tone multi-frequency signaling** (**DTMF**)
- **DMTF** is used for telecommunication signaling over analog telephone lines in the voice-frequency band between telephone handsets and other communications devices and the switching center.
- The current trend is that DTMF inputs are replaced by **speech inputs.**
- Many applications are information services, such as timetable services, weather forecasting and banking services, e-mail applications and voice portals.

Conventional applications

- Desktop applications form another popular area of speech applications. Most desktop applications are either targeted at **dictation** or the **control** of existing graphical applications, such as word-processors or spreadsheet programs.
- **Dictation**: low efficacy of error correction in dictation. Error correction times have been reported to be up to three times as long as the actual text entry time
- **Command oriented** applications are getting more popular in **cars**. **Radio** and **cellular phone** control, among other non critical operations, can be performed via speech.
- The reason for car companies' interest is the **"driver's hands and eyes busy"** situation; speech technology is considered to enhance safety.
- **Speech output** is also common, especially in the popular GPS-based navigation aids that provide spoken guidance and enable the driver to keep the eyes on the road and traffic signs.
- **Aeronautic** applications form another area where the usage of speech has been studied for a long time.

Multilingual applications

- **Translation** systems are one example of multilingual speech applications.
- These systems translate spoken utterances between users. Typical applications include **booking systems** and other similar applications, where two human participants have a **common and known goal**, such as booking a hotel room or a trip, and the computer translates their spoken utterances between languages **in real-time**.
- Multilingual applications may also allow users to access information services using multiple languages.

Google brings real-time speech translation to your Android phone
http://www.youtube.com/watch?v=F-nMr886E44

Multimodal application

- From the historical perspective, multimodality offers promising opportunities, as presented by the famous **Put-That-There** system (speech + pointing)
- Later authors added **gaze direction tracking** to disambiguate other modalities
- Arguments favoring multimodality:
- it **prevent** errors, bring **robustness** to the interface, help the user to **correct** errors or **recover** from them = error-prone technologies can **compensate** each other, rather than bring redundancy to the interface, and reduce the need for error correction.
- Multimodality adds **alternative** communication methods to different situations and environments.
- Multimodal interfaces may also bring **more bandwidth** to the communication and it is possible to **simulate** other modalities, for example in the case of disabled users
- **However**, multiple modalities alone do not bring these benefits to the interface: the use of multiple modalities may be ineffective or even disadvantageous (**conflict**).

Multimodal application

- A multimedia information kiosk developed at University of Tampere

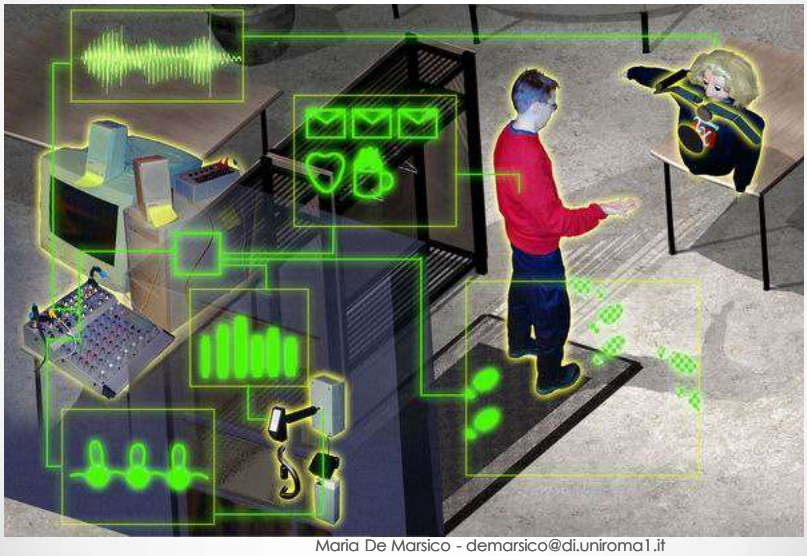
From http://www.cs.uta.fi/hci/spi/ddsi/speech_applications.html

Pervasive applications

- **Pervasive** applications = **ubiquitous** and **mobile** computing applications
- Pervasive applications **lack** large visual displays and familiar interaction devices, such as mice and keyboards
- By embedding systems into everyday environments (so-called "**intelligent environments**") the focus of speech applications shifts from **turn-based, user-initiative** conversational systems into a **more proactive and distributed** direction.

Pervasive applications

Example: Doorman from Tampere University



Pervasive applications

Example: Doorman from Tampere University

Dialogue example in the case of visitors:

- **User**: (Pushes the doorbell)
- **System**: I'm Doorman. Please say the name of a person or a place after the tone.
- **User**: Markku Turunen
- **System**: Welcome. The door is now unlocked.
- **User**: (Enters inside and faces the guidance puppet)
- **System**: The person that you are looking for, Markku Turunen, can be found from room 432. In order to get there turn left. Start near the meeting room. Turn right and head towards the sofa. Markku's office is in front left.

Dialogue example in the case of staff members:

- **System**: (Detects movement and starts recognition)
- **User**: This is Markku Turunen, good morning.
- **System**: (Verifies the speaker correctly.)
- **System**: Welcome Markku. The door is unlocked.
- **User**: (Enters inside and faces the guidance puppet)
- **System**: By the way, you have 12 unread messages in your inbox.

Special user groups

- User groups who have problems with common interfaces can benefit from speech technology. Examples include:
  - dyslexic people
  - motor impaired
  - illiterate people
  - visually impaired (speech input/output, e.g. screen readers, speech-based HTML readers)
- However, usage of speech can **exclude** some users. Example: those whose mother tongue is not the one in which the services are provided.
- Speech recognition can be very sensitive to nonstandard pronunciations common in non-native speech.

Speech applications

	Conventional Applications	Novel Applications
Dialogue participants	single user and single computer	multiple users and multiple computers
Environment	desktop and telephone	office and home environments, mobile settings
Dialogue characteristics	Single deterministic dialogue	open-ended, dynamically constructed concurrent dialogues
Activite participant	active user	user + (pro)active computer
Dialogue control	centralized dialogue management	distributed interaction management
Number of modalities	mostly unimodal	mostly multimodal
Fusion mechanism	alternative / exclusive (sequential) modalities	concurrent / synergistic (parallel) modalities
Modalities	speech, text, graphics	speech, sensors, haptics, gestures
Languages	monolingual	multilingual
Interaction paradigm	"natural" interaction methods based on human-human interaction	"innovative" interaction methods based on human-computer interaction
Usability metrics	efficiency, user satisfaction	privacy, non obstrusiveness, context-awareness
Information sources	explicit information (databases etc.)	fusion of sensed and learned information

From: http://www.cs.uta.fi/hci/spi/ddsi/speech_applications.html

Speech output (synthesis)

- In its basic form the synthesizer module receives a **text string** which is transformed into a **speech signal** (thus often called Text-To-Speech (TTS) systems).
- The text may contain **control codes**, which suggest that the synthesizer should alter its output.
- The most typical control codes control **prosody** (pauses, speaking rate, pitch etc.) and **switch** between different speaking voices and their parameters.
- The use of multiple synthesizers, necessary for example in **multilingual** applications, makes speech output design a little more complex, since we must control multiple synthesizers and in some cases also to decide which ones to use in which situations.

Speech synthesis process

- **First step**: to preprocess the given text into a list of phonemes and prosody control information, divided in phases
- **First preprocessing phase**: acronyms, special characters, numbers, e-mail and www addresses, etc. are **normalized** to plain text.
  - This is not trivial since many elements need to be read in **different** ways **depending on the context**: for example, the classic sentence "Dr. Jones lives on Palm Dr." should be read as "*Doctor Jones lives on Palm drive*".
- **Second preprocessing phase**: **prosodic** information is added. Prosody includes volume, speed, pitch variations, and pauses. If speech has no prosody, it sounds very monotonic and it is hard to understand.

Speech synthesis process

- **Last preprocessing step**: to **convert** the **written** text that consists of **graphemes** (~characters) to **phonemes**.
  - This process is based on a set of rules and an optional dictionary where mappings for the most common words can be found.
  - This process varies significantly between languages. For example, in Italian the mapping is almost one-to-one between graphemes and phonemes. On the other hand, in English and French the mapping is more challenging and complex rules and dictionaries are required.

Text	Price is $50.
Normalized	Price is fifty dollars.
Prosody markup	Price \\pitch -5\\ is \\pitch +5\\ fifty \\pitch -5..-10\\ dollars
Phoneme presentation (Truespell notation))	Pries \\pitch -5\\ iz \\pitch +5\\ fiftee \\pitch -5..-10\\ daalerz

Speech synthesis process

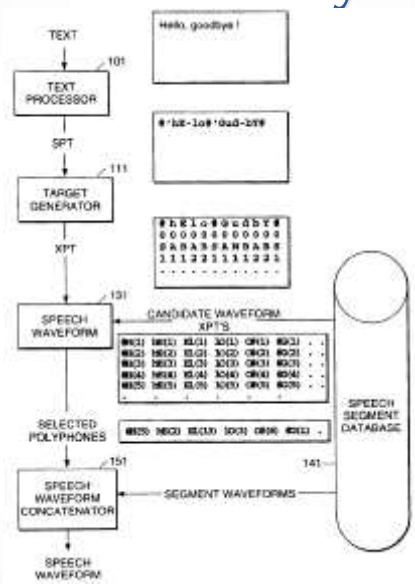
- The final step of text-to-speech synthesis is the generation of the actual waveform.
- There are three main approaches: ***concatenative*** *synthesis*, ***formant*** *synthesis*, and ***articulatory*** *synthesis*.

Concatenative synthesis

- Segments of recorded real speech are **concatenated** together with appropriate processing that creates the final waveform.
- Recorded material must come from one speaker, be of high quality and **cover** all the required units **preferably several times**.
- Signal processing methods, such as **MBROLA** (Multi-band re-synthesis overlap add) are used to combine the units together, and **prosodic information** is included after or during the concatenation process.
- The main challenge is modeling of **coarticulation**: the concatenated parts must match each other.
- One solution is to record units that start from the middle of the phoneme and continue to the middle of another phoneme. These units are called **diphones**.
- It is also possible to use three consecutive phonemes, i.e., **triphones**.
- N phonemes means at most N^2 diphones, and N^3 triphones. In English, this means 42 phonemes, 1500 diphones, and 30000 triphones.
- In principle, using **longer** units results in **better** speech quality. **However**, while the amount of recorded units increases the computing and memory requirements increase as well. In practice, the most advanced concatenative synthesizers have databases of hundreds of megabytes. Some devices, such as mobile phones, do not currently have such resources.

Maria De Marsico - demarsico@di.uniroma1.it

Concatenative synthesis



Maria De Marsico - demarsico@di.uniroma1.it

Formant synthesis

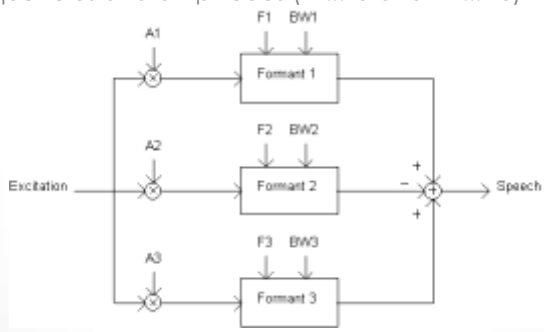
- **Formant:** a characteristic component of the quality of a speech sound; *specifically* **:** any of several **resonance bands** held to determine the phonetic quality of a vowel
- Formant synthesis uses a set of **phonological rules** to control audio signal generation.
- The signal is created by a simple model of human speech generation that includes a set of **filters** and a **source** of initial sound.
- Formant synthesis can be implemented with lower requirements than concatenative synthesis, most importantly the memory requirements are low as there is no prerecorded audio stored in the system
- The main problem of formant synthesis is **sound quality**. The speech is in general quite understandable but the **lack of naturalness** decreases its pleasantness.

Maria De Marsico - demarsico@di.uniroma1.it

Formant synthesis

Basic structure of cascade formant synthesizer.

Formant frequencies and amplitudes (F1...F3 and A1...A3)



Basic structure of a parallel formant synthesizer.

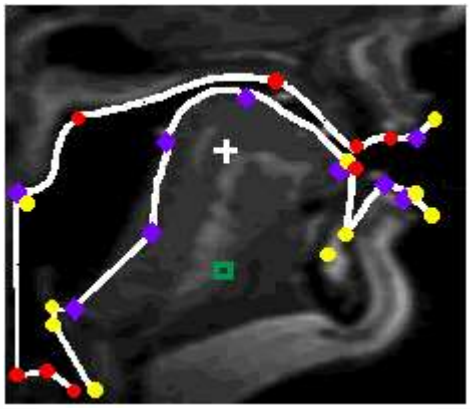
From: http://www.acoustics.hut.fi/publications/files/theses/lemmetty_mst/chap5.html

Maria De Marsico - demarsico@di.uniroma1.it

Articulatory synthesis

- Articulatory synthesis tries to model human speech production in much more concrete level than formant synthesis.
- It models the human vocal track in full 3D and simulates how it works.
- Such models are **very complex** to compute even when simplified
- Articulatory Synthesis generates speech by controlling the **speech articulators** (e.g. jaw, tongue, lips, etc.). Changing the position of the articulators results in a change in the shape of the vocal tract.
- In articulatory models this controllable vocal tract serves as a "filter" with its own resonance characteristics. Separate source models can be used to provide input for the system, resulting in sound being generated.

Articulatory synthesis



Non-speech audio

- Non-speech audio can be an **efficient** way to convey **information** in speech applications, especially in auditory-only interfaces where it can be used as an additional output channel.
- Non-speech audio can be categorized as:
  - **auditory icons,** related to natural sounds which are familiar to users; they convey meaning which is derived from their origin rather than acoustic properties (are sirens, sounds related to closing and opening of things etc.); their use is related to cultural conventions;
  - **earcons,** instead of conveying information on their origin, represent information in their acoustic parameters; earcons are usually not known to the users beforehand and they may be abstract: their use requires learning, but on the other hand they are free from the limitations the natural sounds often have and once learned, they can be easily recognized; earcons can be used, for example, for navigation and representing structural information;
  - **music** can be used similarly as auditory icons and earcons; for example, well-known compositions are widely recognized, so they are familiar to users in the same way as auditory icons.

Conversation techniques

- Turn-taking techniques, i.e., the ways to change speakers in a dialogue, can be difficult to implement in speech-only interfaces.
- In **natural** conversation, people use **non-verba**l cues for turn-taking, such as **gestures** and **eye movements**, as well as acoustic cues, such as pitch variations.
- Pauses, which are efficiently used for turn-taking in human-human communication, may be ambiguous in speech interfaces, since **there is no visual information** to resolve their meaning.
- Turn-taking may be especially problematic in interfaces where **barge-in** is not supported, since it is natural for humans to interrupt the other speaker. Overlapping speech is part of the turn-taking process in human dialogues.

Conversation techniques

- Turn-taking techniques, i.e., the ways to change speakers in a dialogue, can be difficult to implement in speech-only interfaces.
- In **natural** conversation, people use **non-verbal** cues for turn-taking, such as **gestures** and **eye movements**, as well as acoustic cues, such as pitch variations.
- Pauses, which are efficiently used for turn-taking in human-human communication, may be ambiguous in speech interfaces, since **there is no visual information** to resolve their meaning.
- Turn-taking may be especially problematic in interfaces where **barge-in** is not supported, since it is natural for humans to interrupt the other speaker. Overlapping speech is part of the turn-taking process in human dialogues.

Feedback

- Navigation and feedback are key issues in speech interface design.
- One of the problems in speech user interfaces is that users may easily become **disoriented** and not know what to say or what the system is doing.
- New users are often willing to **listen to guidance**, but it should be easy to get rid of it when a user does not need, or want, guidance.
- One form of feedback is **grounding**. In general sense, grounding refers to all actions in the dialogue that aim at keeping common knowledge between the participants. In speech applications, the system **should make sure** that the user is aware of what the system has understood.
- Grounding is particularly important **because of recognition errors**. For example, in a bus timetable system, without proper grounding, user might think that he received correct information while the system has been talking about another bus line because of a recognition error.

Dialogue initiative

- **System** initiative dialogue strategy
- **User** initiative dialogue strategy
- **Mixed** initiative dialogue strategy

System initiative dialogue strategy

- In a system-initiative dialogue system the computer **asks questions from the user**, and when the necessary information has been received, a solution is computed and a response is produced.
- System-initiative dialogue strategy can be highly efficient since the **paths** which the dialogue flow can take are **limited and predictable**. The most challenging issue for dialogue management is to **handle errors** successfully and **ask relevant questions** from the user. None of these tasks are trivial, however.
- Preferred by **novice users** who do not know how the system works. The system guides the user to reach his/her goal. Since the system asks questions, the user can be sure that **all necessary steps** will be performed. This makes the user **feel comfortable** with the system and prevents disorientation.

System initiative dialogue strategy

- The **disadvantages** of system-initiative dialogue strategy include the **clumsiness** of interaction with **experienced** users.
- Especially if the system assumes that only **single pieces of information** are exchanged in every dialogue turn, the dialogue may be long and advance slowly.
- Accept **multiple pieces** of information with a single utterance can support expert users = dialogue management more complicated + recognition grammars more complex + possiblitity of new kinds of errors.
- System-initiative dialogue strategy **is not suitable for all tasks**. It is most suitable for **well-defined, sequential tasks** where the system needs to know certain pieces of information in order to perform a database query or similar information retrieval tasks. Typical examples include database queries for bus timetables or flights.

User initiative dialogue strategy

- The user-initiative dialogue strategy assumes that the **user knows what to do** and **how to interact** with the system.
- The system waits for user inputs and reacts to these by performing corresponding operations.
- User initiative systems are often called **command and control systems**, although the language used may be rather sophisticated.
- The **advantage** of user-initiative systems is that experienced users are able to use the system freely and perform operations any way they like without the system getting in their way. This is also natural in open-ended tasks which have many independent subtasks.
- The main **weakness** of many user-initiative systems is that they require that users are familiar with the system and know how to speak.

Mixed initiative dialogue strategy

- The so-called mixed-initiative dialogue management strategy assumes that the initiative can be taken **either by the user or the system**.
- The user has freedom to take the initiative, but when there are problems in the communication, or the task requires it, the system takes the initiative and guides the interaction.
- Tasks may form a hierarchy in which **different subtasks** can use **different dialogue** strategies
- The system can adapt the style of the interaction **to suit particular users or situations** based on the success of the interaction. This can be done, for example, by using the system-initiative strategy at the beginning and letting the user take more initiative when she or he learns how to interact with the system. Similarly, if the user has problems with the user-initiative strategy, the system can take the lead when the interaction is not proceeding as well as expected.
- Mixed-initiative strategy is often a synonym for user-initiative strategy with **system-initiated error handling**.

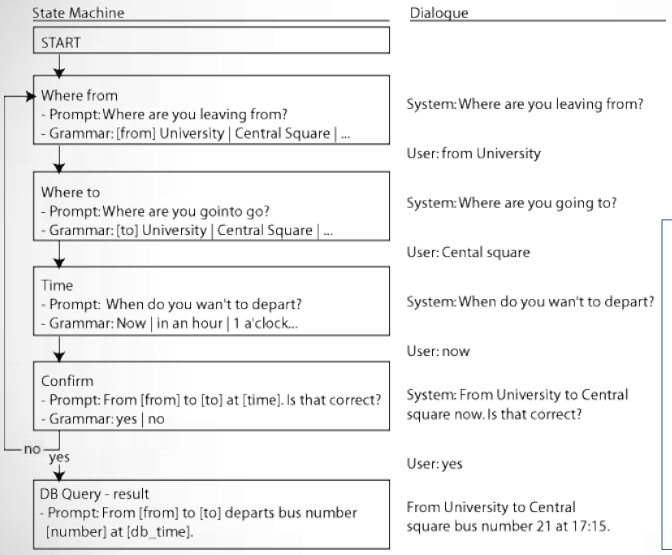
Dialogue control models

- **Finite state machines (state-based dialogues)**
- **Form-based systems**
- **Event-based systems**
- **Plan-based systems**
- …

Finite state machines

- A finite-state machine consists of a set of nodes representing dialogue states and a set of arcs between the nodes. Arcs represent transitions between states.
- The resulting network represents the whole dialogue structure, and paths through the network represent all the possible dialogues which the system is able to produce.
- Typically, nodes represent computer responses and arcs represent user inputs, which move the dialogue from one state to another.
- Finite-state machines are most suitable for **well-structured** and **compact** tasks. If there are **numerous states**, and more importantly, **a lot of transitions** between states, the **complexity** of the dialogue model increases rapidly.
- Common operations which can take place in most situations, such as error correction procedures, increase this complexity enormously. For these reasons, it is commonly argued that state-machines are mostly suitable for **small-scale applications**.

Finite state machines



In this figure there is only one point, the confirmation in the very end, where error correction is possible. If we want to enable error correction at any point, we need to add new states and at least one new transitions per existing state

Form-based systems

- Frame-based, or form-based systems use templates, i.e., collections of information (**sets of attribute-value pairs**), as the basis for dialogue management.
- Similarity can be found, e.g., from forms in web pages where a set of items must be filled in.
- The purpose of the dialogue is to fill necessary information slots, i.e., to find values for the required variables and then perform a query or similar operation on the basis of the frame.
- Frame-based solutions are commonly **mixed-initiative** systems so that user can fill in the form and system can ask for missing information.
- In contrast to the state-machine approach, frame-based systems are more open, since there is no predefined dialogue flow, but instead the **required information is fixed**.

Maria De Marsico - demarsico@di.uniroma1.it

Form-based systems

Form during the dialogue	Dialogue
From: To: Time:	System: Where are you leaving from?
	User: from University to Cental square
From: University To: Central Square Time:	System: When do you wan't to depart?
	User: now
From: University To: Central Square Time: now	System: From University to Central square now. Is that correct?
	User: yes
From: University (confirmed) To: Central Square (confirmed) Time: now (confirmed)	From University to Central square bus number 21 at 17:15.

Maria De Marsico - demarsico@di.uniroma1.it

Event-based systems

- The computer determines what to do on the basis of the interpreted **user utterance** and its own **state**.

Plan-based systems

- Plan-based systems focus on the interpretation of the utterance and on the intention that it conveys.
- This information is then matched to the **plan** which **models** the **dialogue**.
- Plans may be related to the domain, or to the discourse in general
- **Dialogue acts** play an important role in this approach.
- This approach sees both user and computer utterances as communicative acts, which are chained together to achieve the goal which the dialogue has.
- Since dialogue acts determine the success of the dialogue, it is important to recognize dialogue acts correctly. One of the current trends in research is to do this using machine learning techniques.

Error detection

The following is a categorization of different error types [Duff et al., 1996].

- Level 0: Missing input.
- Level 1: Recognition rejection.
- Level 2: Recognizer returns something that cannot be interpreted (makes no sense at all).
- Level 3: Recognizer returns something that is not semantically consistent.
- Level 4: Recognizer returns semantically well formed, but impossible to fulfill sentences.
- Level 5: Same as 4 with the exception that the impossibility is due to the dialogue context.
- Level 6: The backend system fails to fulfill the command (e.g. database connection is lost).

User can also detect many kinds of errors. In particularly, speech recognition errors that do not result in any of the above error types and cannot be identified by recognition scores and dialogue state, can only be detected by the user. The system designer must take care that the user can spot and correct every possible error that can occur, and also that the user know how to correct them.

- Level 7: User initiated error correction.

Error repair

- Error repair design takes place after the cause of the error has been deduced.
- The system decides how to handle the error, i.e., it selects a suitable error correction strategy.
- In an ideal case both error costs and error correction costs are calculated and compared. The type of error, dialogue state and information on past errors and repairs should be considered.
- Possible hypothesis: If the error correction costs are lower than the costs that the error causes, error correction is justified. Otherwise, the error correction should be omitted.

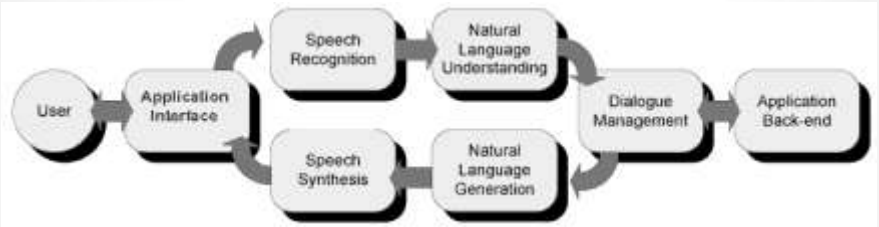
Error repair

Speaker	I. Explicit confirmations	II. Change of dialogue strategy & implicit confirmations	III. Selection from a list	IV. Ignoring the error correction
User:	*When does the next bus leave from Hervanta to Central Square?*	*When does the next bus leave from Hervanta to Central Square?*	*When does the next bus leave from Hervanta to Central Square?*	*When does the next bus leave from Hervanta to Central Square?*
System:	*Are you leaving from Hervanta?*	*I'm not sure if I got that.Lets try another way. Where are you leaving from?*	*Are you leaving from Hervanta, Hallila or Hospital?*	*Next bus from Hervanta to Central Station, bus number 33, leaving in fifteen minutes*
User:	*Yes.*	*Hervanta.*	*Hervanta.*	*No, next bus from Hervanta to Central Square!*
System:	*Is your destination Central Station?*	*Leaving from Hervanta. What is your destination?*	*Your destination is Central Station, Square, or Hospital?*	*Next bus from Hervanta to Central Square, bus number 13, leaving in two minutes*
User:	*No*	*Central Square.*	*Square.*	...
System:	*Please say your destination*	*Next bus from Hervanta to Central Square, bus number 13, leaving in two minutes*	*Next bus from Hervanta to Central Square, bus number 13, leaving in two minutes*	...
User:	*Central Square.*	...	...	...
System:	*Was that central square?*	...	...	...
User:	*Yes.*	...	...	...
System:	*Next bus from Hervanta to Central Square, bus number 13, leaving in two minutes*	...	...	...

Error correction

- When the actual error correction takes place, the system initiates an error correction dialogue, such as a confirmation, a yes/no question, rephrasing, respeaking, spelling or a selection from a list, to correct the error.
- It is important to realize, that error correction is a dialogue itself, and it is carried out like any other dialogue, and more errors can occur during the correction dialogue.
- **Confirmations** are used in most speech applications to correct errors. They can be seen as part of error handling, but they can also be seen as interface elements of normal dialogue to ensure that participants have a common understanding of the information (**grounding**)

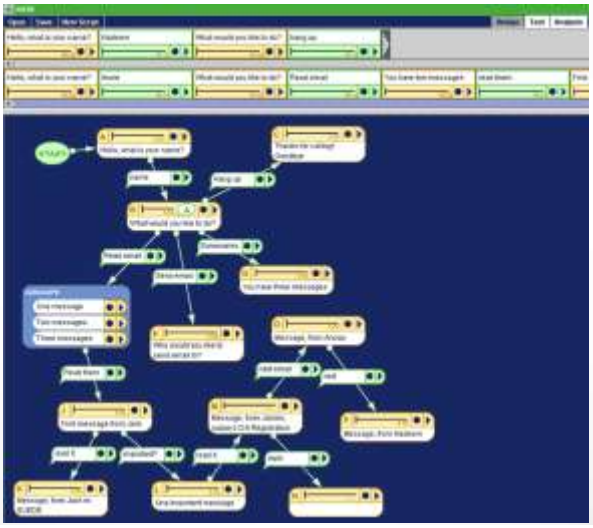
System components



Maria De Marsico - demarsico@di.uniroma1.it

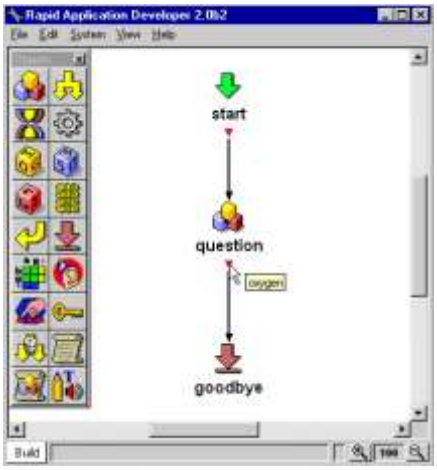
Development tools

- SUEDE (http://dub.washington.edu:2007/projects/suede/)



Development tools

- CLSU Toolkit (http://www.cslu.ogi.edu/toolkit/)

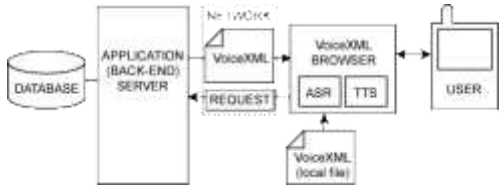


Development tools

- VoiceXML related tools (http://www.palowireless.com/voicexml/devtools.asp)

VoiceXML architecture

- VoiceXML environment consist of an **interpreter**, or VoiceXML browser, and a set of VoiceXML **documents**
- The **browser** includes speech recognition component, speech synthesis, audio and/or telephony interfaces and the dialogue manager that interpretes the VoiceXML documents.
- The **server** component generates the VoiceXML documents and possibly recognition grammars and takes care of the back-end processing.



- A VoiceXML application is a set of related **documents**.
- Each document defines a part of the dialogue flow. A document is a **finite state machine** consisting of dialogs and transitions to other documents (URIs).
- The building blocks of the dialogue in the documents are forms and menus.
- A **form** is a **dialog** containing **fields**, a **menu** is a **dialog** containing **choices**.
- While the **frame-approach** is used for **standard dialogue** control, an **event-based** approach is used for other situations, such as for **handling errors**.

Readings

- **Design and Development of Speech Interfaces** http://www.cs.uta.fi/hci/spi/ddsi/
- Karl, L., Pettey, M., and Shneiderman, B. Speech versus mouse commands for word processing applications: An empirical evaluation. *International Journal of Man-Machine Studies n. 39,* 4 (1993), 667–687.
- Danis, C., Comerford, L., Janke, E., Davies, K., DeVries, J., and Bertran, A. StoryWriter: A speech oriented editor. In *Proceedings of CHI'94: Human Factors in Computing Systems: Conference Companion* (Boston, Apr. 24–28). ACM Press, New York, 1994, 277–278.
- Sami Lemmetty Master's Thesis: **Review of Speech Synthesis Technology.** http://www.acoustics.hut.fi/publications/files/theses/lemmetty_mst/
- Keiichi TOKUDA, Heiga ZEN. **Fundamentals and recent advances**
- **in HMM-based speech synthesis.** http://hts.sp.nitech.ac.jp/?Tutorial