

An Introduction to Spoken Dialogue Systems. Fundamentals, Standards and Tools

16th ESSLLI
August 9-13, 2004
Nancy, France

Prof. Ramón López-Cózar Delgado
Dept. Languages and Computer Systems
Computer Science Faculty
University of Granada, Spain
E-mail: rlopezc@ugr.es
<http://www.ugr.es/~rlopezc>



Course outline

- Introduction
- Automatic Speech Recognition (ASR)
- Natural Language Processing (NLP)
- Dialogue management
- Natural Language Generation (NLG) and Speech synthesis
- Standards
- Tools

2

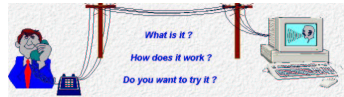
R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Outline

- Historic antecedents
- Current applications of speech processing
- Introduction to spoken dialogue systems
- Main problems of implementation
- Examples of spoken dialogue systems
- Implementation recommendations
- References



3

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Historic antecedents

- Development of talking machines has been a human challenge for centuries
 - Statues of divine God could speak ...
 - In our times: acoustic models (vocal tract) models and electro-acoustic models
- **C. G. Kratzenstein (1779)**
 - reed + resonance chamber
 - Vowel sounds
- **W. Von Kempelen (1791)**
 - blowers + reeds + variable-form resonance chambers
 - Vowels and consonants sounds
 - Complete sentences

4

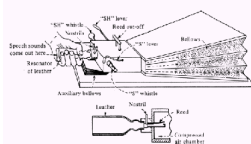
R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Historic antecedents

- **Faber (1846)**
 - Talking organ
- **Sir Charles Wheatstone (1891)**
 - enhanced version of Von Kempelen machine



5

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Historic antecedents

- **R. Riesz (1937)**
 - Variable-length reed
 - Sentence intonation
- **J. Q. Stewart (early 20th century)**
 - Electronic devices
 - Two coupled resonators excited by electric pulse
 - Different frequencies generate vowel sounds

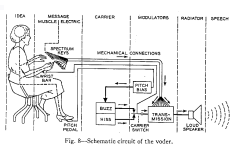
6

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

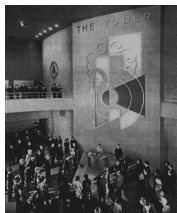
Historic antecedents



• H. Dudley, R. Reiz and S. Watkins (1930)

- voder : first electric machine able to generate sentences
- Controlled by a human operator
 - Keyboard to control several parameters
 - pedal for pitch control
- Exhibited at the 1939 World's Fair in New York

Sample sentence 

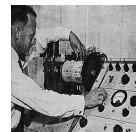


ESSLLI 2004

1. Introduction

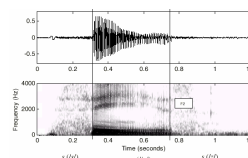
R. López-Cózar, UGR

Historic antecedents



• F. S. Cooper, A. M. Liberman y J. M. Borst (1951)

- spectrogram = visual representation of voice signal



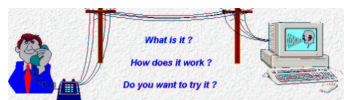
- Electric device generated voice using a light ray and a wheel 50 concentric circles of variable densities
- The machine transformed into voice the information coded in the spectrogram

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Outline



- Historic antecedents
- • Current applications of speech processing
- Introduction to spoken dialogue systems
- Main problems of implementation
- Examples of spoken dialogue systems
- Implementation recommendations
- References

ESSLLI 2004

1. Introduction

R. López-Cózar, UGR

Current applications of speech processing



• Some features of speech

- Can be used in the distant (e.g. appropriate for vehicles)
- Great expressivity
- Omni-directional (can provide information to several people simultaneously)
- Temporal and sequential (once uttered is not available)

• Current applications

- Control of computer programs
- Telephone-based applications
- Dictation systems
- Vehicle systems control
- Speaker recognition systems
- Dialogue systems

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Current applications of speech processing



• Control of computer programs

- Navigation in menu systems
 - Application initialisation (e.g. "calculator", etc.)
 - Document editing ("italics", "bold", "font change", etc.)
 - Hand-free operation

• Telephone-based applications

- Application navigation using numeric menus
- Initially navigation using **DTMF (Dial Tone Multi Frequency)**
 - Each number is associated to a function in the application
 - "Press 1 to accept, press 2 to deny and press 3 to finish"
 - Problem: users may forget the associations number-function

ESSLLI 2004

1. Introduction

R. López-Cózar, UGR

Current applications of speech processing



• Telephone-based applications

- Later, **IVR (Interactive Voice Response)** systems
 - Isolated words "yes" / "no" are used instead of buttons
 - Appropriate for applications with small vocabulary and little number of functions
 - Speech is just an alternative way of communication, it does not allow for more communication power
 - Users are very limited in the interaction
- Generally pre-recorded messages are used (no speech synthesis techniques to generate messages dynamically)

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Current applications of speech processing

- **Dictation systems**
 - Speaking is faster than writing
 - Input modes: text / command execution
 - Typically user adaptation is needed (training)
 - Vocabulary $\approx$ 60.000 words
 - Addition of new words by user
 - Examples:
 - ViaVoice (IBM)
 - Voice Xpress (Lernout & Hauspie)

13

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Current applications of speech processing

- **Control of vehicle systems**
 - Control of hand-free devices (e.g. telephone)
 - Increase security while driving
 - Speech recognition is a very difficult task
 - Tremendous noise
 - Robust speech recognition techniques needed
 - Example AutoPC (Clarion)
 - Vocabulary $\approx$ 1.200 words
 - Radio station selection, e-mail reader, GPS car navigation, etc.

14

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Current applications of speech processing

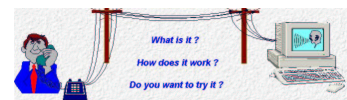
- **Speaker recognition**
 - **Speaker identification:** determine who is the speaker among a set of possible speakers
 - **Speaker detection:** detect if there is change of speaker
 - **Speaker verification:** check the speaker identity
 - Applications
 - Tele-bank (e.g. check status of bank account)
 - Database access
 - Forensic applications
 - Voice-based login (e.g. iMAC, Apple)

15

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Outline

- Historic antecedents
- Current applications of speech processing
- ➔ • Introduction to spoken dialogue systems
- Main problems of implementation
- Examples of spoken dialogue systems
- Implementation recommendations
- References

16

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Introduction to spoken dialogue systems

- Technology started in the 70s aprox.
- Objective: developing systems able to communicate with humans using spontaneous speech
- Initial projects
 - **DARPA SLS** (Spoken Language Systems) in USA
 - **SUNDIAL** (Speech UNDERstanding and DIALog) in Europe
- **SLS**
 - Only for English
 - Restricted domains (mainly airplane travels)
 - Focused on the input interface (speech recognition + understanding)
 - Periodic competitions of developed systems

17

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Introduction to spoken dialogue systems

- **SUNDIAL**
 - Several languages (English, French, German and Italian)
 - Main application domains: airplane and train travels
 - Focused on input interface, dialogue management and output interface
 - Stimulates cooperation instead of competition (e.g. integration of components developed in different labs)
- **Main application areas**
 - Travel information and reservation
 - Weather forecast
 - Bank information
 - Product sell
 - Directory information, etc.

18

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Introduction to spoken dialogue systems

- Example: **directory information system**

S: Welcome to the directory information system. To speak in English say "English"; to speak in French, say "French"

U: English

S: City please

U: Paris

S: Say the family name and first name please

U: Smith John

S: The number is 91 408 7654, ...

19

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Introduction to spoken dialogue systems

- Example: **train timetable information system**

S: Good morning, welcome to the train timetable information system. How may I help you?

U: Good morning, I need to travel from Paris to Nancy

S: When would you like to travel from Paris to Nancy?

U: this Friday, at eight a.m.

S: Do you want to travel this Sunday, September 25th, at eight a.m.

U: No, this Friday

S: Do you want to travel this Friday, September 23th, at eight a.m.?

U: Correct

S: There are the following trains ... Would you like another train?

U: No thanks, that one is ok

S: We wish you a good travel. Thanks for your call. Goodbye.

20

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Introduction to spoken dialogue systems

- **Advantages of spoken dialogue systems**

- Economic savings for companies
- Telephone use: comfortable y flexible
- Natural communication
- Users have a greater level of expression (in comparison with DTMF systems)
- Users are not limited to utter a few commands

- **Some current challenges**

- Greater robustness
- More natural dialogue
- Enhance multimodal interaction
- Enhance multilingual interaction

21

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Introduction to spoken dialogue systems

- **Application domain**

- Task to be performed by the dialogue system. Examples:
 - ATIS (Air Travel Information Service)
 - TRAINS
 - etc.

- **Communication types**

- visual / written / oral
- unimodal / multimodal

- **Interaction strategies**

- System directed
- User directed
- Mixed initiative
- Advantages / Disadvantages

22

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Introduction to spoken dialogue systems

- **Discourse type**

- How can users speak?
- Directly related to the speech recogniser used
- **IWR**, Isolated Word Recognition
 - “Please say ‘one’ if you wish the weather forecast, say ‘two’ if you want to make a reservation, and say ‘three’ if you want other type of service”.
- **CWR**, Connected Word Recognition
- **CSR**, Continuous Speech Recognition
- **Word Spotting**

23

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Introduction to spoken dialogue systems

- **Technologies used in current systems**

- **ASR** (Automatic Speech Recognition)
- **NLP** (Natural Language Processing)
- **Dialogue management**
- **NLG** (Natural Language Generation)
- **Speech synthesis**

- **Classification of dialogue systems**

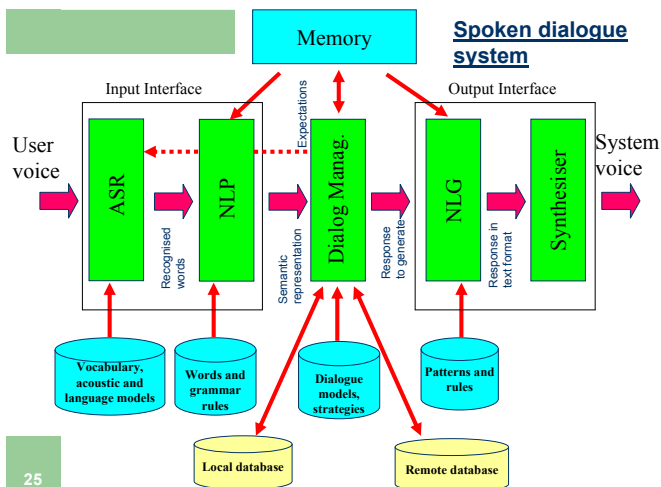
- **Oral (unimodal)**: only use speech for input/output
- **Multimodal**: use several communication channels
 - **Input**: speech, lip movements, user gestures, etc.
 - **Output**: speech, images, graphics, sounds, etc.
- **Multilingual**: allow users speak in several languages (e.g. English, French, Spanish, etc.)

24

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Modular structure of a dialogue system

- **Input interface**
 - Speech recogniser: provides sequence(s) of recognised words
 - Semantic analyser: obtains the meaning of utterances (typically using **frames**)
 - Additional input modalities (**multimodal systems**)
 - User gestures
 - User lip movements
 - User gaze direction
 - Facial expressions of user
 - Text written by user (e.g. using a PDA)
 - etc.

Modular structure of a dialogue system

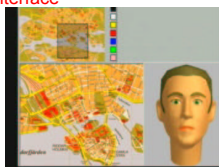
- **Dialogue manager**
 - Decides the system response
 - Asks for additional data and confirmations
 - Generates expectations to enhance ASR
- **Memory module**
 - Store data obtained from user interaction
 - Provides information about the previous turns to the dialogue manager
- **Database**
 - Contains the information the user needs

Modular structure of a dialogue system

- **Output interface**
 - Natural language generator (or response generator)
 - Generates responses in text format
 - Speech synthesizer
 - Word concatenation (little flexible, pretty good intelligibility)
 - Diphone concatenation (great flexibility, less intelligibility)
 - Additional output modalities (**multimodal system**)
 - Graphics
 - Animated agent (face, or face and body), synchronised movements
 - Other sounds
 - etc.

Introduction to spoken dialogue systems

- **Multimodal dialogue system**
 - **Additional information in the input interface**
 - User lip movements, user gaze,
 - User gestures, text written by user, etc.
 - **Additional information in the output interface**
 - Graphics
 - Other sounds
 - Animated agent
 - Facial expressions
 - Gestures and lip movements
 - Synchronisation with oral output

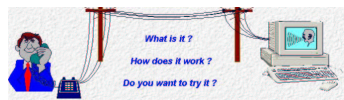


Introduction to spoken dialogue systems

- **Multilingual dialogue system**
 - Strategies
 - Language selection at the interaction start
 - Automatic language detection
 - **Foundations**
 - Extraction of the **same semantic representation** independently of the language used (**interlingua**)
 - System components designed to be **language independent**
 - Language-dependent stored externally in configuration files
 - Dialogue management strategies do not change
 - Dialogue manager decides which files to use
 - Use of a multilingual TTS (text-to-speech) system (e.g. Festival)

Outline

- Historic antecedents
- Current applications of speech processing
- Introduction to spoken dialogue systems
- ➔ • Main problems of implementation
- Examples of spoken dialogue systems
- Implementation recommendations
- References



Main problems of implementation

- In terms of ASR
 - Types
 - **Coarticulation**: phonemes are affected by neighbour phonemes
 - **Variability**: differences in pronunciation, distinct accents, ...
 - **Vocabulary size**: the likely of errors increases with the v. size
 - **Noise and interferences**: e.g. traffic, machines, etc.
 - **Out-of-vocabulary words (OOV)**: cannot be recognised
 - **Cross-talk**: other people talking closely to the microphone
 - **Other factors**: speaker state of mind, illness, etc.
 - In consequence, three types of error may occur
 - **Insertions**: the recognised sentence contains more words
 - **Deletions**: the recognised sentence contains less words
 - **Substitutions**: some words in the recognised sentence are changed

Main problems of implementation

- Problem of **coarticulation**
 - Speech = sequence of sounds (phonemes)
 - Number of phonemes limited for each language
 - Sequence of phonemes make syllables, and these make words
 - Each phoneme has characteristic articulatory and acoustic features
 - These features allow distinguish words ("bit" / "pit")
 - **Problem**: connection of phonemes to make bigger units change the acoustic features of the phonemes (i.e. they are affected by their neighbours)
 - Coarticulation may happen
 - Inside a word (e.g. /t/ in "tea", "tree", "city", "steep", etc.)
 - Between words (e.g. "this" in "this car" or "this ship")

Main problems of implementation

- In terms of NLP
 - **Ellipsis** (omission of words)
 - E.g. "one bacon with cheese" (What does "one" refers to?)
 - **Anaphora**
 - E.g. "I want it big" (What does "it" refers to?)
 - **Ambiguity** (lexical and structural)
 - E.g. "right" (What does it mean? correct, privilege, ...)
 - E.g. "My sister saw a boy playing in the park with a telescope" (What does it mean? Two possible interpretations)
 - **Derived from ASR**
 - insertions, deletions and substitutions can provoke errors in the linguistic analysis of sentences

Main problems of implementation

- In terms of dialogue management
 - **Interaction management**
 - Who has the initiative in the dialogue?
 - **Management of ASR/NLP errors**
 - What to do when there were errors?
 - **Management of confirmations**
 - What data provided by the user must be confirmed?
 - When / how confirm the data?
 - **Try to avoid errors**
 - How generate expectations to avoid ASR/NLP errors?
 - **In multimodal systems**
 - Input information processing (analysis and fusion)
 - Decide what output modalities to use to generate the outputs
 - Synchronize output modalities

Main problems of implementation

- In terms of NLG
 - **Generate syntactic and semantically correct sentences**
 - I want two ham sandwiches
 - Did you say you want a lemon beer?
 - **Choose the most adequate words and expressions in a given context**
 - **Anaphora**: Did you say you want it big?
 - **Ellipsis**: The red car is mine in the blue (...) is yours

Main problems of implementation

- In terms of speech synthesis
 - Achieve great **intelligibility**
 - Sentences must be easily understood by users
 - Achieve great **naturalness**
 - Artificial voice must be similar to the human one
 - Election of linguistic units

Less complexity and versatility



More naturalness

- Phonemes
- Diphonemes
- Triphonemes
- Words
- Sentences

37

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Main problems of implementation

- Related with response time
 - Users **do not accept** slow systems, specially in telephone applications
- Related with the expectations of users
 - Users rarely are aware of the technical limitations
 - They typically expect a perfect performance !

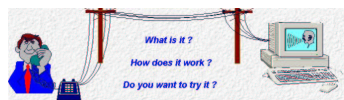
38

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Outline



- Historic antecedents
- Current applications of speech processing
- Introduction to spoken dialogue systems
- Main problems of implementation
- • Examples of spoken dialogue systems
- Implementation recommendations
- References

39

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Example of spoken dialogue systems

- **Pegasus** - airplane travel information, USA
- **Tosburg** - fast food ordering, Japan
- **Tarsan** - travel information, Japan
- **ToyTrains** - organisation of train routes, USA
- **Railtel** - train travel information, Italy
- **Dialogos** - train travel information, Italy
- **Jupiter** - weather forecast information, USA
- **Atos** - call routing, Spain
- **Stacc** - exam marks, Spain
- **Saplen** - fast food ordering, Spain

40

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Pegasus

- (Zue et al. 1994)
- Application domain: **ATIS** (Air Travel Information Service)
- Query module EAASY SABRE to obtain information from the Official Airline Guide database, USA
- Speech recogniser: **n-best** (n=10)
- Recognition hypothesis are the inputs to the NLP module
- Semantic representations obtained are stored using **frames**
- Vocabulary: 1,300 words
- Unimodal input: speech
- Multimodal output: speech + text

41

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Tosburg

- (Seto et al. 1994)
- TOSBURG = Task-Oriented Speech dialogue System Based on speech Understanding and Response Generation
- Application domain: fast food ordering
- TOSBURG II: allows cancellation of speech output (**barge-in**)
- NLP based on **word-spotting**
- Interaction strategy: **user-guided**
- Modules:
 - Keyword detector
 - Keyword analyser
 - Dialogue manager
 - Multimodal response generator: speech + graphics

42

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Tarsan

- (Yamada et al. 1994)
- TARSAN = Travel Assistant system composed of information-Retrieval, Speech conversation Access and Natural-language processing
- Application domain: travel information (addresses, telephone numbers, museums, hotels, ...)
- Provides speech-based access to information in commercial CD-ROMs
- Multimodal output: speech + text
- Modules:
 - Continuous-speech recogniser
 - NL analyser
 - Query module to CD-ROMs
 - Speech synthesiser (text-to-speech)

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



ToyTrains

- (Ferguson et al. 1996)
- Application domain: organisation of train routes
- Modules
 - ASR: SPHINX II
 - NLP:
 - Speech-act types (e.g. “greeting”, “query”, “inform”, “confirm”, etc.)
 - Speech-act type unification allows processing partial information provided by the user
 - Dialogue manager
 - Graphic interface
 - allows highlighting screen areas, draw/delete routes, etc.
 - Speech synthesiser

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

RailTel

- (Billi et al. 1997)
- Application domain: train travel (information and reservation)
- Modules:
 - Isolated word recognition module: FLEXUS
 - Database: Infotransporti
 - Timetables, train types
 - Interaction strategy: system-based
 - Speech synthesiser: ELOQUENS
 - Vocabulary: 533 words
 - 476 station names
 - Hours, week days, months, etc.

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



Dialogos

- (Danielli, 1996)
- Extension of RAILTEL system: continuous-speech recognition
- Modules: voice interface, linguistic processor, dialogue manager, response generator, database query, speech synthesiser
- Language model: bigram trained with 30,000 sentences
- Vocabulary: 2,983 city names, 33 station names, 76 numbers, months, week days, etc.
- NLP: generation of **frames** that are sent to the dialogue manager
- Dialogue manager: generates expectations that are sent to the voice interface to enhance ASR
- Sentence generation: instantiation of **patterns**

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Jupiter

- (Polifroni al. 1998)
- Application domain: weather forecasts
- Obtains forecasts from Internet sites several times a day
 - Forecasts are analysed and transformed into **frames**
- Multilingual: English, German, Chinese Mandarin, Spanish
- Developed using the **Wizard of Oz** technique (simulation of system functions by a human operator)
- Evaluation after three weeks working:
 - 487 sentences
 - 51.3% completely understood
 - 2.5% partially understood
 - 5% out-of-domain
 - Analysis of Internet forecasts: 99%

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



ATOS

- (López Soto et al. 1997)
- ATOS = Automatic Telephone Operating Service
- Application domain: telephone configuration, creation & modification of personal agenda, directory query, place calls, etc.
- Modules: continuous-speech recogniser, semantic analyser, dialogue manger, text-to-speech
- Semantic analyser creates **frames**
- Vocabulary ≈ 2,000 words
- Speech recognition 95% in lab conditions
- ASR recognition errors caused mainly due to:
 - Gender / number discordances
 - Insertions and deletions

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



STACC

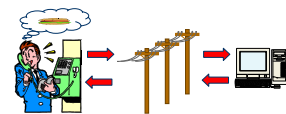
- (Rubio et al. 1997)
- STACC = Automatic Telephone System to Access Exam marks
- Modules: continuous-speech recogniser, telephone line interface, flow control
- Telephone line interface: call reception and finish, message generation
- Flow control: decides the message to generate as well as the system next action
- Evaluation
 - 248 spontaneous queries, 216 students of UGR, 1191 sentences
 - 84.3% obtained their mark in the first trial
 - 100% obtained their mark in the third trial
 - The system never provided wrong information
 - ASR 85.7%, Speech understanding: 89.6%

49

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



SAPLEN

- (López-Cózar et al. 1997, 1998, 1999)
- Fast food information & ordering
- Vocabulary $\approx$ 2,000 words
- Mixed interaction strategy
- Language models: bigrams
- $\approx$ 500 dialogue obtained in a fast food restaurant allowed to know
 - Vocabulary
 - Syntactic and semantic structures used by clients
 - Goals pursued by clients and restaurant assistants
 - Set of sentences the systems should generate as responses

50

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



SAPLEN: Module structure

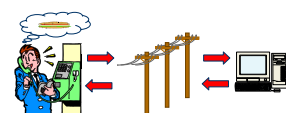
- **Input interface**
 - Continuous-speech recogniser
 - Semantic analyser
- **Memory**
 - User interactions stored as frames
 - Provides dialogue history to the dialogue manager
- **Database**
 - Restaurant products, known clients, addresses
 - Database handler
- **Dialogue manager**
 - Controls de interaction between the other modules as well as the system-user interaction
- **Output interface**
 - Sentence generator (text format)
 - Festival speech synthesiser (<http://www.cstr.ed.ac.uk/projects/festival/>)

51

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



SAPLEN: Sentence analysis

- Frames are obtained from keywords (words really important from a semantic point of view)
- Semantic representation are obtained form semantic constituents (related words from the semantic point of view)
- **Implicit recovery**: robustness against some ASR errors
 - Example:

User: I want a ham sandwich
Recogniser output: I want ham sandwich
Semantic representation:
 (action=desire, amount=1, product=sandwich, type=ham)

52

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



SAPLEN: Sentence generation

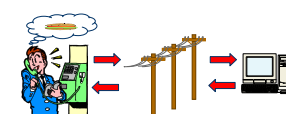
- 41 patterns were used do generate several types of sentence
- Each pattern has
 - Expressions
 - Gaps
- **Generation**
 - Expansion of expressions
 - Gap filling with appropriate words
- **Grammatical rules were used to**
 - Decide gender / number correspondences
 - Decide anaphoric references (i.e. pronouns) and ellipsis (using contextual information)

53

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



SAPLEN: Sentence generation

- Pattern used to ask the user for "drink taste"

```
[ "WHICH" ["OTHER"] "TASTE" ]
[ <desire> ]
[ "FOR THE" ]
[<drink_size> <temperature> <drink_name> <complements> ]
"?"
```

Dialogue example

- Example of sentences this pattern can generate:

WHICH OTHER TASTE WOULD YOU LIKE FOR THE BIG FANTA?
 WHICH TASTE?
 WHICH TASTE DO YOU WANT FOR THE SMALL MILKSHAKE?

54

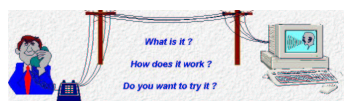
R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Outline

- Historic antecedents
- Current applications of speech processing
- Introduction to spoken dialogue systems
- Main problems of implementation
- Examples of spoken dialogue systems
- Implementation recommendations
- References



Implementation recommendations

- **Setting up a dialogue system requires**
 - Knowledge about the words the users will likely utter
 - Interpret them in a given context
 - Generate appropriate responses in the dialogue context
- **Very difficult task**
 - The systems do not have knowledge about the world nor cultural inheritance
 - Voice signal may be corrupted by a variety of causes
 - Background noise, features of microphones, telephone bandwidth limitation
 - Differences in vocal tract of users
 - Differences in user pronunciations
 - Grammatically incorrect sentences
 - Ambiguous sentences
 - Coarticulatory effects

Implementation recommendations

- **Use indications of system understanding (feedback)**
 - Users must always be aware of the dialogue status
- **Use “system busy” indicators**
 - To avoid users think the system is malfunctioning
 - Use background music if the systems takes more than 3 sec to generate the output
- **Use “turn control” indications if barge-in is not possible**
 - GUI interface: a microphone can suggest the user turns
 - Telephone applications: phone tones
- **In some occasions, restrict user utterances to avoid errors**
 - “Did you say Sevilla or Melilla?”

Implementation recommendations

- **Use explicit / implicit confirmations**
 - “Did you say you want to travel to Sevilla?”
 - “Ok, destination Sevilla, where do you want to depart from?”
- **Ask for user help**
 - “Please can you spell the destination city?”
- **Use confidence measures**
 - Scores associated to the recognised words
 - System confidence on the correct recognition of the words
- **Use implicit recovery**
 - Robustness against some ASR errors (these system errors may be unnoticed by users)
- **Generate error responses with sufficient information to allow the user correction**
 - “I didn’t get the destination city. Please say it again clearly”

Implementation recommendations

- **Detect user error indicators**
 - “no, I said to Paris”
- **In multimodal systems use the input modalities available**
 - PDA stylus to write unknown words
 - Cameras to detect the user is really talking
- **Take into account user expectations**
 - Make users aware they are talking to a machine
 - Notice them about technological limitations
- **Use a correct prompt strategy**
 - Brief and concise system messages
 - Allow for user interruptions if possible (barge-in)

Implementation recommendations

- **Use appropriate prosody**
 - pitch, volume, speed, pauses
 - change pitch to emphasise determined words
 - Speed: 150 -170 words / min (English)

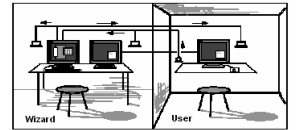


Implementation recommendations

- **Evaluate prototype before setting up in the real world**
 - Evaluation measures: subjective and objective
 - **Subjective:** typically used to assess the whole system
 - Opinion of test users
 - System naturalness
 - User satisfaction
 - Ease to carry out tasks
 - Quality of system messages, etc.
 - **Objective:** typically used to assess components individually
 - Statistic measures
 - WA (Word Accuracy), SR (Sentence Recognition)
 - SU (Sentence Understanding), IR (Implicit Recovery)



Implementation recommendations



- **Wizard of Oz technique (WoZ)**
 - Allows evaluating system performance and system-user interaction
 - Also used to collect dialogue corpus in initial development stages
 - **Goal:** test users are made to believe that they are interacting with a real dialogue system, but they interact with a human operator (Wizard) who simulates the system functions
 - Users interact using a set of scenarios
 - Costly technique
 - Wizard training
 - User selection
 - Session transcription
 - Result analysis, etc.



References

- Billi, R., Castagneri, G., Danielli, M. 1997. Field trial evaluations of two different information inquiry systems. *Speech Communication*
- Danielli, M. 1996. On the use of expectations for detecting and repairing human-machine miscommunication. *Working notes of the AAAI-96*. 87-93
- Ferguson, G. M., Allen, J. F., Miller, B. W., Ringger, E. K. 1996. The Design and Implementation of the TRAINS-96 system: A Prototype-Mixed Planning Assistant. *TRAINS Technical Note 96-5*, Computer Science Dept., Universidad de Rochester
- López-Cózar, R., García, P., Díaz, J., Rubio, A. J. 1997. A Voice Activated Dialog System for Fast-Food Restaurant Applications. *Eurospeech*, pp. 1783-1786
- López-Cózar, R., Rubio, A. J., García, P., Segura, J. C. 1998. A Spoken Dialogue System Based on a Dialogue Corpus Analysis. *First International Conference on Language Resources and Evaluation*, pp. 55-58



References

- López-Cózar, R., Rubio, A. J., García, P., Segura, J. C. 1999. Use of confidence values and expectations in the SAPLEN dialogue system (*In Spanish*). *Natural Language Processing*, 24, pp. 37-41
- López Soto, M. T., Quesada, J. F., Álvarez, J. 1997. Application of LEKTA to the ATOS environment (*In Spanish*). *Natural Language Processing*, 21
- Polifroni, J., Seneff, S., Glass, J., Hazen, T. J. 1998. Evaluating Methodology for a Telephone-Based Conversational System. *First International Conference on Language Resources and Evaluation*, Granada, pp. 43-49
- Rubio, A. J., García, P., De la Torre, A., Segura, J. C., Díaz-Verdejo, J., Benítez, M. C., Sánchez, V., Peinado, A. M., López-Soler, J. M., Pérez-Córdoba, J. L. 1997. STACC: an automatic service for information access using continuous speech recognition through telephone line. *Eurospeech '97*, vol. 4, pp. 1779-1782
- Seto, S., Kanazawa, H., Shinchi, H., Takebayashi, Y. 1994. Spontaneous Speech dialogue system TOSBURG II and its evaluation. *Speech Communication* 15, pp. 341-353



References

- Yamada, M., Itoh, F., Sakai, K., Komori, Y., Ohora, Y., Fujita, M. 1994. A spoken dialogue system with active/non-active word control for CD-ROM information retrieval. *Speech Communication* 15, pp. 355-365
- Zue, V., Seneff, S., Polifroni, J., Phillips, M., Pao, C., Goodine, D., Goddeau, D., Glass, J. 1994. PEGASUS: A spoken dialogue Interface for on-line air travel planning. *Speech Communication* 15, pp. 331-340

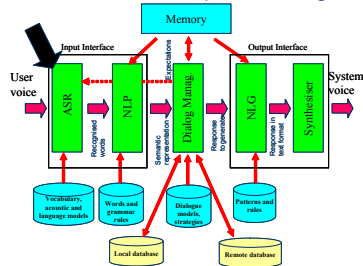


Course outline

- Introduction
- ➔ • Automatic Speech Recognition (ASR)
- Natural Language Processing (NLP)
- Dialogue management
- Natural Language Generation (NLG) and Speech synthesis
- Standards
- Tools

Localisation

- Module structure of a spoken dialogue system



67

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Outline

- Introduction
- Historic antecedents
- Stochastic approach
 - Acoustic modelling
 - Language modelling
 - Search of word sequence
- Other approaches to ASR
 - N-best recognition
 - Keyword spotting
- Measures
- References

68

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Introduction

- Automatic Speech Recognition (ASR)** (Rabiner et al. 1996)
 - Goal:** Transforming words uttered by a speaker (voice signal) into the sequence of words in text format
 - Several technologies involved:
 - Acoustics
 - Computer science
 - Linguistics
 - Psychology
 - Signal processing
 - Pattern recognition
 - Information theory, etc.

69

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Introduction

- Classification of ASR systems: Speakers**
 - Single speaker**
 - Only recognises voice a speaker (who makes the training)
 - High recognition rates
 - Necessary for speakers with pronunciation problems
 - Speaker independent**
 - Can recognise voice of other users (in addition to the training ones)
 - Lower recognition rates
 - Necessary for telephony applications
 - Adapted to the speaker**
 - Several speakers make the training
 - Later adaptation to a single speaker

70

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Introduction

- Classification of ASR systems: Speech style**
 - Isolated Word Recognition (IWR)**
 - Speakers must make long pauses between words
 - Easy detection of word start/end
 - Connected Word Recognition (CWR)**
 - Speakers must make short pauses between words
 - Continuous Speech Recognition (CSR)**
 - Speakers do not need make pauses, they can speak normally
 - Coarticulation problem
 - No clear detection of word frontiers (e.g. "I scream for ice-cream")

71

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Introduction

- Classification of ASR systems: Speech style**
 - Read speech**
 - Speakers must speak as if they were reading
 - Spontaneous speech**
 - Speakers can speak naturally as if they were speaking to other person
 - Typical spontaneous speech problems (hesitations, repetitions, ...)
 - Keyword spotting**
 - The system goal is detecting determined words, not the accurate sequence of words uttered by the speaker

72

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Introduction

- **Classification of ASR systems: Vocabulary size**
 - Increasing the vocabulary increases recognition difficulty, memory requirements and computation
 - **Small:** 1-99 words
 - **Medium:** 100-999 words
 - **Large:** > 1,000 words
 - Currently up to 70,000 words

73

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Introduction

- **Difficulties of the ASR**
 - **Acoustic variability:** each person pronounces sounds differently
 - **Acoustic confusion:** similarity between words with similar sounds (e.g. wall / war, word / world)
 - Homophones: two words that sound identical but have different meaning
 - E.g. "The results of the first **period** are explained in the next **paragraph**"
 - **Linguistic variability:** distinct dialects of a language
 - **Coarticulation:** concatenation of neighbour phonemes affect their characteristics
 - **Vocabulary size:** the bigger, the greater difficulty
 - **Out-Of-Vocabulary words (OOV):** words not included in the recogniser vocabulary cannot be recognised

74

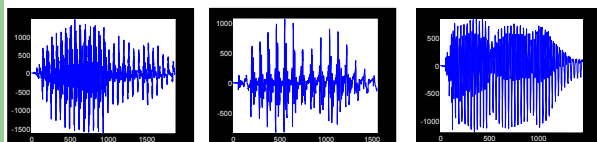
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Introduction

- **Difficulties of the ASR**
 - **Sentences not permitted by the grammar:** cannot be recognised
 - **Acoustic variability**
 - E.g. three pronunciations of the word "el" uttered by three different speakers

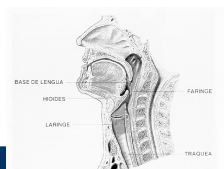


75

R. López-Cózar, UGR

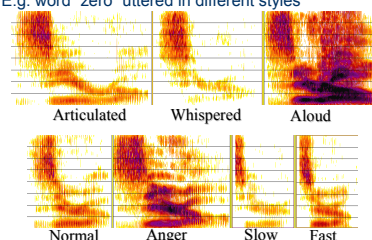
ESSLI 2004

1. Introduction



Introduction

- **Difficulties of the ASR**
 - **Acoustic variability**
 - E.g. word "zero" uttered in different styles



Spectrograms

76

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Introduction

- **Difficulties of the ASR**
 - **Environment conditions**
 - **Laboratory:** high quality voice, without noise, reverberations or echoes
 - **Real-world:**
 - Environment (music, air conditioning, ...)
 - Channel distortions (microphones used, telephone line bandwidth limitation, ...)
 - Breath, coughs, etc.

77

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Introduction

- **Difficulties of the ASR**
 - **ASR systems success if can provide high recognition rates**
 - Cannot transcribe voice without restrictions
 - **Appropriate tasks**
 - Keyboard and mouse are not available
 - Tasks require free hands/eyes
 - Users cannot speak or are not accustomed to use a keyboard
 - Disabled users (e.g. cannot move hands)
 - **Not appropriate tasks**
 - Tasks require users speak with other people while interacting
 - Users are in very noise environments

78

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Introduction

- **Activation of ASR systems**
 - **Manual activation**
 - User indicates that s/he is going to speak
 - Reliable and robust
 - **Automatic activation**
 - In spoken dialogue systems
 - In “rigid” turns
 - In turns with interruption (barge-in)
 - Activation words (*Star Trek*)
 - “Computer, ...”
 - “Mother, ...”

79

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Introduction

- **ASR is an error-prone technology**
 - **Insertions**: recognised sentence contains additional words
 - **Deletions**: some words are missed in the recognised sentence
 - **Substitution**: some words of the uttered sentence are substituted by others in the recognised sentence
 - **Rejection**: the user starts talking but the system detects nothing
 - **False alarms**: the user says nothing but the recogniser outputs words

80

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Introduction

- **Rejections or substitutions**
 - User utters words not contained in the system vocabulary
 - Uttered sentences is not parsed by any active grammar
 - User starts talking before the system is ready to recognise
 - User utters words that sound very similarly
 - User makes long pauses during uttering the sentence
 - User hesitates when speaking (false starts, “um”, “eh”, ...)
 - Microphone is not correctly set
- **False alarms**
 - Laughs, coughs, other people's voice, etc.

Current error rates: 5-15%

81

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Introduction

- **Main applications of ASR**
 - Device control for disabled people
 - Computer data and command input
 - Database access
 - Telephone control
 - Credit card validation
 - Dictation systems
 - Spoken dialogue systems
 - Etc.

82

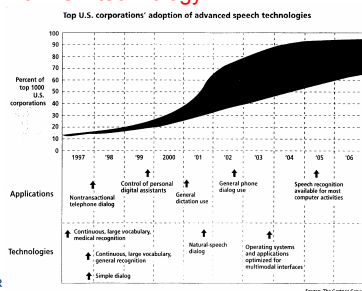
R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Introduction

- **Evolution of ASR technology**



83

R. López-Cózar, UGR

Outline

- Introduction
- Historic antecedents
- Stochastic approach
 - Acoustic modelling
 - Language modelling
 - Search of word sequence
- Other approaches to ASR
 - N-best recognition
 - Keyword spotting
- Measures
- References

84

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Historic antecedents

- **Beginning, 50s**
 - 1952 Davis, Biddulph and Balashek (Bell Labs, USA)
 - Single speaker isolated digits
 - 1959 Fry and Denes (University College, UK)
 - Several words of one or more phonemes
 - 1959 Forgie and Forgie (MIT)
 - 10 words /b/ - vocal - /t/, speaker independent
- **Fundaments, 60s**
 - 1968 Vintsyuk (Kiev University, Soviet Union)
 - Dynamic Time Warping, DTW

85

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Historic antecedents

- **First solutions, 70s**
 - 1975 Itakura (USA)
 - LPC for ASR
 - 1979 Sakoe (NEC Labs, Japan)
 - Two-level Dynamic programming for continuous speech
 - 1979 Bridle (JSRU, UK)
 - One-step algorithm
- **Expansion, 80s**
 - Continuous-speech recognition for large size vocabularies
 - 1980 Ferguson
 - Hidden Markov Models (HMMs)
 - Statistical models
 - 1985 Jelinek (IBM)
 - Tangora, isolated word dictation (language model)

86

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Historic antecedents

- 1987 K.-F. Lee (Carnegie-Mellon)
 - SPHINX (continuous speech with HMM)
- 1989 Waibel (CMU, USA)
 - Time-Delay Neural Networks (TDNN)
- **Start of business, 90s**
 - Cheap and powerful computers
 - IBM: VoiceType (connected words) and ViaVoice (continuous speech)
 - Learnout & Hauspie: NaturallySpeaking
 - HTK (Hidden Markov Model Toolkit)
- **A reality, 2000s** (but with important limitations)

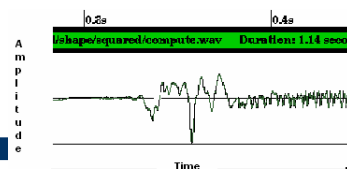
87

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Historic antecedents



- **Approaches to ASR** (Rabiner et al. 1996)
 - **Expert systems (rule-based)**: not used currently for this task
 - **Pattern matching**: efficient for isolated words, single speaker, small vocabulary
 - **Stochastic (probabilistic)**: allows using information about multiple pronunciations, most used nowadays
 - **Conexionist (artificial neural networks)**: do not provide better results than the stochastic approach

88

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Outline

- Introduction
- Historic antecedents
- **Stochastic approach**
 - Acoustic modelling
 - Language modelling
 - Search of word sequence
- Other approaches to ASR
 - N-best recognition
 - Keyword spotting
- Measures
- References

89

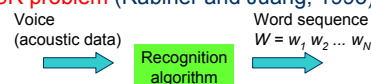
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Stochastic approach

- **The ASR problem** (Rabiner and Juang, 1993)



- **Definition of the ASR problem**
 - Given a sequence of acoustic data A , which is the sequence of words W uttered?
 - Maximise probability: $W = \max_W P(W | A)$
 - It is not easy to compute $P(W | A)$, thus the **Bayes** rule is used:

$$P(W | A) = \frac{P(A | W)P(W)}{P(A)}$$

90

R. López-Cózar, UGR

ESSLI 2004

Stochastic approach

- **Bayes rule**

$$P(W | A) = \frac{P(A | W)P(W)}{P(A)}$$

- $P(A | W)$ is the **acoustic model**. Probability of obtaining A when W is uttered
- $P(W)$ is the **language model**. Probability of uttering W (it only depends on the word sequence)
- $P(A)$ is not necessary since is independent from the word sequence. Thus, the above equation can be rewritten as follows:

$$W = \max_W P(A | W)P(W)$$

ASR fundamental equation

Stochastic approach

- **ASR stages**

- **Training** (Macherey et al. 2003)
 - The system learns, using voice samples and text, the acoustic $P(A | W)$ and language $P(W)$ models
 - Allows reducing ASR errors (e.g. IBM ViaVoice)
- **Recognition**
 - Input acoustic data (speaker sentence) is transformed into a word sequence by applying the ASR fundamental equation

Outline

- Introduction
- Historic antecedents
- Stochastic approach
 - Acoustic modelling
 - Language modelling
 - Search of word sequence
- Other approaches to ASR
 - N-best recognition
 - Keyword spotting
- Measures
- References

Acoustic modelling

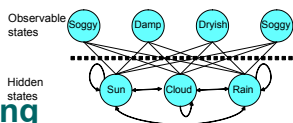
- What are the recognition basic units? (Sasou et al. 2003)
 - Choosing one or other influences strongly the performance of the ASR system
 - **Word models**
 - Appropriate for isolated word recognition or when the vocabulary is very small
 - **Subword models**
 - Appropriate for continuous-speech recognition or when the vocabulary is very large
 - Phonemes, syllables
 - Diphonemes, triphonemes, etc. (contextual, allow modelling coarticulation)

Acoustic modelling

- How to model these units?
 - **Hidden Markov Models (HMMs)**
 - Word models: A HMM is used for each word
 - Subword models: A HMM is used for each acoustic unit

Acoustic modelling

- Acoustic modelling: $P(W|X)$
 - For each acoustic unit a model is built
 - A sentence is modelled as the concatenation of the models corresponding to these units
 - Usual current acoustic models: **Hidden Markov Models**, HMMs
 - Simple mathematical formulation: probabilistic theory
 - Each state emits symbols or vectors with a given probability
 - "Hidden": state transitions not directly related to the emitted symbols



Acoustic modelling

• Hidden Markov Models

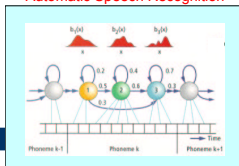
- Sometimes it is not possible to know the states of a process
 - E.g. A hermit cannot know the weather state but can infer it from the state of seaweeds
 - He uses an “algorithm” to make an assumption considering the seaweeds
 - Weather state is likely related to the seaweed state
- There are two types of state
 - Observable (seaweed)
 - Hidden (weather)
- The number of observable and hidden states can be different
 - 3 weather states (sun, cloud, rain) and 4 seaweed states (soggy, damp, dryish, dry)

Acoustic modelling

• Hidden Markov Models

- 3 probability types:
- **Vector π** (initial probabilities) $\rightarrow$
- **Transition matrix** (probs. of hidden states given a previous hidden state) $\rightarrow$
- **Confusion matrix** (probs. of observable states given a hidden state) $\rightarrow$

Automatic Speech Recognition



	Sun	Cloud	Rain
Weather today	1.0	0.0	0.0

	Sun	Cloud	Rain
Weather yesterday	0.5	0.25	0.25
Cloud	0.375	0.125	0.375
Rain	0.125	0.625	0.375

		Dry	Dryish	Damp	Soggy
Seaweed	Sun	0.50	0.20	0.15	0.05
Cloud	Sun	0.25	0.25	0.25	0.25
Rain	Sun	0.05	0.10	0.35	0.50

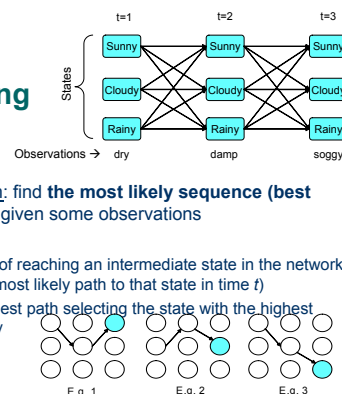
Acoustic modelling

- Problems that can be addressed with HMMs (Rabiner and Juang, 1993)
 - Evaluation (**Forward Algorithm**)
 - Find the probability of a sequence of observations given a HMM
 - Decodification (**Viterbi Algorithm**)
 - Find the most likely sequence of hidden states given determined observations
 - Learning (**Forward-Backward algorithm**)
 - Generate a HMM from a sequence of observations

Acoustic modelling

• Viterbi algorithm

- **Decodification problem**: find the **most likely sequence (best path)** of hidden states given some observations
 - **Forward algorithm**
 - δ = partial prob. of reaching an intermediate state in the network (represents the most likely path to that state in time t)
 - Creation of the best path selecting the state with the highest partial probability
 - **Backward algorithm**
 - A backward pointer, Ψ , is used to indicate the best current state in the path



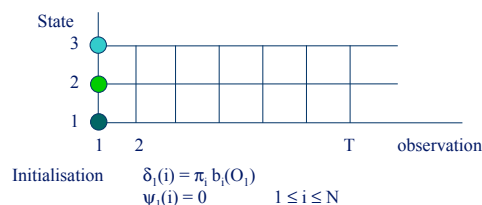
Acoustic modelling

• Viterbi algorithm. Initial definitions:

- Set of states: $S = \{s_1, s_2, \dots, s_N\}$
- Set of observations: $O = \{v_1, v_2, \dots, v_n\}$
- Probs. of the initial state: $\Pi = (\pi_i)$
- Transition probabilities: $A = (a_{ij})$
- Generation probabilities: $B = (b_i(v))$
- The HMM is: $\lambda = (\Pi, A, B)$

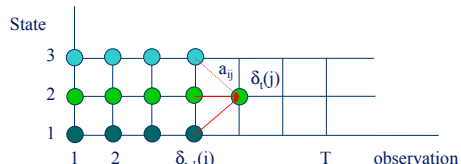
Acoustic modelling

• Viterbi algorithm:



Acoustic modelling

Viterbi algorithm:



$$\begin{aligned} \delta_t(j) &= \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t) & 2 \leq t \leq T \\ \psi_t(j) &= \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t) & 1 \leq j \leq N \end{aligned}$$

103

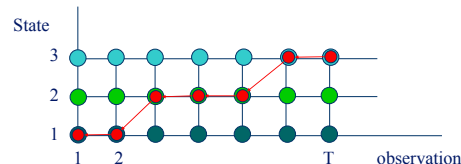
R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Acoustic modelling

Viterbi algorithm:



$$\begin{aligned} \text{Backtracking} \quad P^* &= \max_{1 \leq i \leq N} [\delta_T(i)] \\ q_T^* &= \arg \max_{1 \leq i \leq N} [\psi_T(i)] \\ q_t^* &= \psi_{t+1}(q_{t+1}^*) \quad t = T-1, T-2, \dots, 1 \end{aligned}$$

104

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Outline

- Introduction
- Historic antecedents
- Stochastic approach
 - Acoustic modelling
 - Language modelling
 - Search of word sequence
- Other approaches to ASR
 - N-best recognition
 - Keyword spotting
- Measures
- References

105

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Language modelling

- **Only** sentences permitted by the grammar can be recognised (Mori et al. 2003; Zitouni et al. 2003)
- Basic grammar types
 - Stochastic (for dictation)
 - Rule-based (or finite-state)

106

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Acoustic modelling: Stochastic grammars

- Probabilistic point of view
- Computation of $P(W)$ can be approximated:

$$P(W) = P(w_1, w_2, \dots, w_q) = P(w_1) P(w_2|w_1) P(w_3|w_1, w_2) \dots P(w_q|w_1 \dots w_{q-1})$$

- Calculating these probabilities is difficult
 - It is typical to use histories of n words (n-grammars)

$$P(w_i | w_1, w_2, \dots, w_{i-1}) = P(w_i | w_{i-n+1} \dots w_{i-1})$$

- Generally, $n=2$ (**bigrams**) or $n=3$ (**trigrams**)
- E.g.: **bigram**: $P(w_i | w_{i-1})$

- Assumes statistical independence from the oldest part

$$P(W) = P(w_1, w_2, \dots, w_q) = P(w_1) P(w_2|w_1) P(w_3|w_2) \dots P(w_q|w_{q-1})$$

107

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Language modelling: Rule-based grammars

- The grammar indicates the recogniser the expected user sentences
- Fast and “accurate” recognition
- It requires careful design to allow for sufficient expressivity

```
#JSFG v1.0
// Define de grammar name
grammar SimpleCommands;
// Define de rules
public <Command> = [<Polite>] <Action> <Object> (and <Object>)*;
<Action> = open | close | delete ;
<Object> = the window | the file;
<Polite> = please;
```

JSFG grammar (Java Speech Grammar Format)

Please open the window
Open the window
Please open the file and close the window
Delete the window

Some sentences allowed
by this grammar

108

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Language modelling: Rule-based grammars

- **Alphabet** V = finite set of symbols
- **Sentence** = finite sequence of symbols of an alphabet
- V^* = set of all the possible sentences generated from V , including the empty sentence
- V^+ = similar to V^* but not including the empty sentence
- **Language** = set (finite or not) of sentences
- **Grammar** = model to describe the language
- Formal grammar, G : $G = \{V_T, V_N, P, S\}$
 - V_T : terminal vocabulary
 - V_N : non-terminal vocabulary
 - P : productions
 - S : initial symbol

109

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Language modelling: Rule-based grammars

- Classification of formal grammars (Chomsky)
 - **Type 0: Unrestricted**
 $\alpha \rightarrow \beta$ where $\alpha \in V^+$ and $\beta \in V^*$
 Too complicated for ASR
 - **Type 1: Context-sensitive**
 $\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$ where $\alpha_1, \alpha_2 \in V^*$, $\beta \in V^+$ and $A \in V_N$
 Too complicated for ASR
 - **Type 2: Context-free**
 $\alpha \rightarrow \beta$ where $A \in V_N$ and $\beta \in V^+$
 Used in some ASR systems
 - **Type 3: Regular (or finite-state)**
 $A \rightarrow a B$ (or $A \rightarrow B a$) where $A, B \in V_N$ and $a \in V_T$
 $A \rightarrow a$

Generally used for ASR

110

R. López-Cózar, UGR

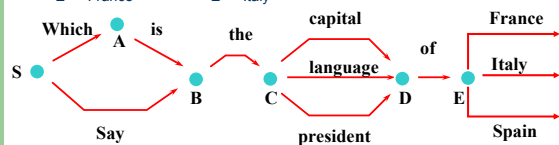
ESSLLI 2004

1. Introduction

Language modelling: Rule-based grammars (regular)

- $V_T = \{\text{Which, is, Say, the, capital, language, president, of, France, Italy, Spain}\}$
- $V_N = \{A, B, C, D, E\}$ – Initial symbol: S
- P :

$S \rightarrow \text{Which } A$	$S \rightarrow \text{Say } B$	$A \rightarrow \text{is } B$
$B \rightarrow \text{the } C$	$C \rightarrow \text{capital } D$	$D \rightarrow \text{of } E$
$C \rightarrow \text{language } D$	$C \rightarrow \text{president } D$	$E \rightarrow \text{Spain}$
$E \rightarrow \text{France}$		



111

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Outline

- Introduction
- Historic antecedents
- Stochastic approach
 - Acoustic modelling
 - Language modelling
 - Search of word sequence
- Other approaches to ASR
 - N-best recognition
 - Keyword spotting
- Measures
- References

112

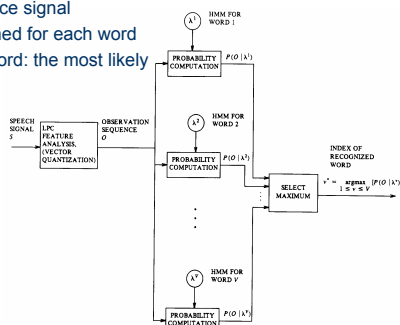
R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Search of word sequence

- **Isolated word recognition** (Astrov and Andrassy, 2003)
 - Analysis of voice signal
 - A HMM is trained for each word
 - Recognised word: the most likely



113

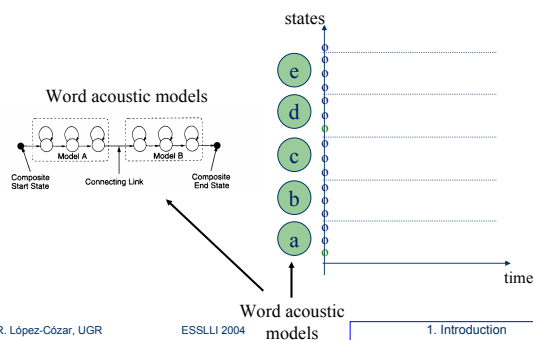
R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Search of word sequence

- **Continuous speech recognition**
 - Viterbi algorithm



114

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

Search of word sequence

- Forward path:** intra-word transitions

Word acoustic models

115 R. López-Cózar, UGR ESSLII 2004 1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

Search of word sequence

- Forward path:** Inter-word transitions

Word acoustic models

Language model

116 R. López-Cózar, UGR ESSLII 2004 1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

Search of word sequence

- Forward path:** Construction of the best path

Word acoustic models

117 R. López-Cózar, UGR ESSLII 2004 1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

Search of word sequence

- Backward path (Backtracking)**

Word acoustic models

118 R. López-Cózar, UGR ESSLII 2004 1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

Outline

- Introduction
- Historic antecedents
- Stochastic approach
 - Acoustic modelling
 - Language modelling
 - Search of word sequence
- ➔ • Other approaches to ASR
 - N-best recognition
 - Keyword spotting
- Measures
- References

119 R. López-Cózar, UGR ESSLII 2004 1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

N-best recognition

- The recogniser provides several outputs (one optimal and other suboptimal), i.e., the most likely **N paths** (Stolcke et al. 1997)
 - Sometimes the correct word sequence is in the N best paths
- Requires incorporating new mechanisms in the Viterbi algorithm
- Increment of computation and memory requirements
- Posterior linguistic analysis to determine the most likely hypothesis from a syntactic/semantic point of view

120 R. López-Cózar, UGR ESSLII 2004 1. Introduction



Keyword spotting

- The objective is recognising specific words, not all the words uttered by the user (Heracleous and Shimizu, 2003)
 - E.g. "I would like to make a **transfer** from uhm my account to uhm this new account"
 - Only **transfer** is detected (the other words are ignored)
- Initial systems from the 70s
- Useful in some applications but not for others (semantic content is not considered), e.g.
 - "Please cancel the **transfer** I made yesterday"
 - "Please confirm if a **transfer** has been made to my account"
 - "Please make a **transfer** from my investment account to my current account"

Outline

- Introduction
- Historic antecedents
- Stochastic approach
 - Acoustic modelling
 - Language modelling
 - Search of word sequence
- Other approaches to ASR
 - N-best recognition
 - Keyword spotting
- Measures
- References



Measures

- Perplexity (PP)**
 - Measures the difficulty of the recognition task
 - It depends on the flexibility of the permitted language
 - Average number of possible transitions from any word to any other word
 - Currently up to 250
 - Perplexity of a system with 35,000 words without a language model?

Language model is very important for reducing the perplexity

Measures

- Confidence scores** (Liu et al. 2003; Cerisara, 2003)
 - How sure is the recogniser about the recognised words?
 - Confidence score** = high-medium-low value (depending on the confidence) assigned to each recognised word
 - Very used in spoken dialogue systems
 - Speech understanding based on high-confidence words
 - Confirmations asked for low-confidence words
 - Simple way to compute a confidence score:**
 - Using N-best list:** word W appearing in many recognition hypothesis should have a high-confidence score
 - $P(h_i)$ = probability of hypothesis i in N-best list

Sum of probabilities of the hypothesis containing W

Sum of probabilities of the N recognition hypothesis

$$MC_{nbest} = \frac{\sum_{i=1, W \in h_i}^N P(h_i)}{\sum_{i=1}^N P(h_i)}$$

Measures

- Word Accuracy (WA)**

$$WA = \frac{W_t - (W_i + W_d + W_s)}{W_t} \times 100$$

- % of recognised words in all words uttered
- W_t = total number of words in the corpus
- W_i = number of inserted words
- W_d = number of deleted words
- W_s = number of substituted words

Measures

- Keyword Accuracy (KWA)**

$$KWA = \frac{K_t - (K_i + K_d + K_s)}{K_t} \times 100$$

- % of recognised keywords in all keywords uttered
- K_t = total number of keywords in the corpus
- K_i = number of inserted keywords
- K_d = number of deleted keywords
- K_s = number of substituted keywords



Measures

• Sentence Recognition (SR)

$$SR = \frac{S_r}{S_t} \times 100$$

- % of recognised sentences (a sentence is recognised if **sequence of recognised words = sequence of uttered words**) in all sentences uttered
- S_r = number of recognised sentences
- S_t = total number of sentences in the corpus

127

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



References

- Astrov, S., Andrassy, B. 2003. Large vocabulary speaker independent isolated word recognition for embedded systems. Proc. Eurospeech, pp. 1137-1140
- Cerisara, C. 2003. The use of confidence values in vector-based call routing. Eurospeech '03, pp. 633-636
- Heracleous, P., Shimizu, T. 2003. An efficient keyword spotting technique using a complementary language for filler models training. Proc. Eurospeech, pp. 921-924
- Liu, Y. 2003. Automatic phone set extension with confidence measure for spontaneous speech. Eurospeech '03, pp. 2741-2744
- Macherey, W., Ney, H. 2003. A comparative study on maximum entropy and discriminative training for acoustic modeling in automatic speech recognition, Eurospeech '03, pp. 793-796
- Mori, S., Nishimura, M., Itoh, N. 2003. Language model adaptation using word clustering. Eurospeech '03, pp. 425-428

128

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



References

- Rabiner, L. R., Juang, B. H. 1993. Fundamentals of Speech Recognition. Prentice-Hall
- Rabiner, L. R., Juang, B. H., Lee, C. H. 1996. An Overview of Automatic Speech Recognition, Automatic Speech and Speaker Recognition: Advanced Topics, Kluwer Academic Publisher, pp. 1-30
- Sasou, A., Asano, F., Tanaka, K., Nakamura, S. 2003. Adaptation of acoustic model using the gain-adapted HMM decomposition method. Proc. Eurospeech, pp. 29-32
- Stolcke, A., König, Y., Weintraub, M. 1997. Explicit word error minimization in N-best list rescoring. Proc. Eurospeech, pp. 163-166
- Zitouni, I., Siohan, O., Lee, C. H. 2003. Hierarchical class n-gram language models: Toward better estimation of unseen events in speech recognition. Proc. Eurospeech, pp. 237-240

129

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



Course outline

- Introduction
- Automatic Speech Recognition (ASR)
- Natural Language Processing (NLP)
- Dialogue analysis and management
- Natural Language Generation (NLG) and Speech synthesis
- Standards
- Tools

130

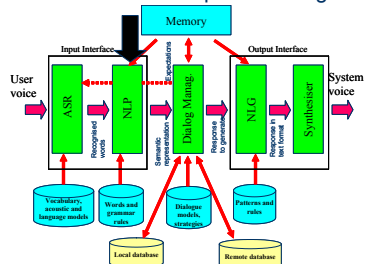
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Localisation

• Module structure of a spoken dialogue system



131

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



Outline

- Types of NLP analyses
- Main problems of NLP
- Some grammars for NLP
- Implicit recovery
- Methods for knowledge representation
- Evaluation measures

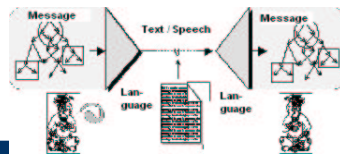
132

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

NLP



- Formalisation to automate the human ability to communicate using natural languages
- **Natural language** = language used by human beings to transmit knowledge, emotions, etc.
- Human ability to process language is extremely complex
 - Codification of message using chains of sounds or letters
 - Decoding of chains to obtain the original message
 - Process carried out building complex structures from initial simple structures (e.g. letters → words → sentences → texts)

133

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

NLP



- Some applications of NLP (Dale et al. 2000)
 - Text processing and stylistic correction
 - Intelligent writing assistance
 - Natural language analysis and generation
 - Database interfaces
 - Information extraction
 - Generation of reports from databases
 - Machine translation, etc.
 - **Dialogue systems**

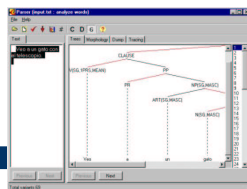
134

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Types of NLP analyses



- **Lexical (or morphologic) analysis**
 - Detect word types and assign them to lexical categories
 - Possibly, lexical ambiguity (e.g. "right")
- **Syntactic analysis**
 - Build structures that represent the grammatical functions of words
 - Transformation of a lineal sequence into a hierarchical structure
 - Possibly, syntactic (or structural) ambiguity
 - Some NLP systems omit this level of analysis

135

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Types of NLP analyses



- **Semantic analysis**
 - Provides meaning to the syntactic constituents
 - What is the meaning of words? Necessary to maintain a dialogue
- **Pragmatic analysis**
 - What is the meaning of words in a given context?
 - E.g. "Can you tell me the hour?"

136

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Outline



- Types of NLP analyses
- • Main problems of NLP
- Some grammars for NLP
- Implicit recovery
- Methods for knowledge representation
- Evaluation measures

137

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Main problems of NLP



- **Ellipsis** (omission of words)
 - E.g. "one ... with cheese" ("one" what?)
- **Anaphora**
 - E.g. "I'd like it big" (what does "it" refer to?)
- **Ambiguity** (lexical and structural)
 - **Lexical**: "right" (what does it mean? privilege, right-side, agreement ...)
 - **Structural**: "My sister saw a boy playing in the park with a telescope" (two possible interpretations)
- **Derived from ASR errors (dialogue systems)**
 - insertions, deletions and substitutions can provoke errors in NLP
- **Derived from spontaneous speech (dialogue systems)**
 - Hesitations: "uhm ..."
 - False starts: "On Mond - Tuesday"

138

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Outline

- Types of NLP analyses
- Main problems of NLP
- • Some grammars for NLP
- Implicit recovery
- Methods for knowledge representation
- Evaluation measures

139

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Some grammars for NLP

- **Phrase-structure grammars** (Borsley, 1996)
 - Great number of rules define the structure of a variety of sentences
 - Efficient for some kinds of applications, but
 - Difficult to maintain and enlarge
 - Difficult to port to other applications
- **Context-Free Grammars** (Satta, 2000)
 - Very used to describe well-formed sentences
 - Very efficient algorithms are available
 - Not very appropriate to handle spontaneous speech

140

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Some grammars for NLP

- **Semantic Grammars** (Su et al. 2001)
 - Rules are applied considering syntactic and semantic considerations simultaneously
 - Advantages
 - Useful to avoid some possible ambiguities (if the semantic interpretations have no meaning)
 - Useful to ignore syntactic aspects that do not affect the semantics
 - Very useful for restricted domains
 - Disadvantages
 - A huge number of rules may be necessary
 - Sentence analysis can be costly

141

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Outline

- Types of NLP analyses
- Main problems of NLP
- Some grammars for NLP
- • Implicit recovery
- Methods for knowledge representation
- Evaluation measures

142

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Implicit recovery

- Used in the NLP component to make dialogue systems more robust against some ASR errors (Billi et al. 1997)
- **Example 1: word deletions**

User: "I want a sandwich"

Recogniser: "want sandwich"

SR OK: (action=order, amount=1, product=sandwich)
- **Example 2: word insertion**

User: "I want two beers"

Recogniser: "I want two two beers"

SR OK: (action=order, amount=2, product=beer)

143

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Outline

- Types of NLP analyses
- Main problems of NLP
- Some grammars for NLP
- Implicit recovery
- • Methods for knowledge representation
- Evaluation measures

144

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Methods for Knowledge representation

- Propositional logic
- First-order predicate calculus
- Conceptual networks
- **Frames**

145

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Methods for Knowledge representation

- **Frames** (Allen, 1995)
 - Abstract representations of real-world entities and their relationships
 - Each entity if constituted of a set of attributes (**slots**)

```
[PERSON] →
[Supertype] → (mammalian)
[NAME] → ()
[GENDER] → ()
[AGE] → ()
[HEIGHT] → ()
```

Frame representing a PERSON entity

```
[PERSON] →
[Supertype] → (mammalian)
[NAME] → (JOHN)
[GENDER] → (MALE)
[AGE] → (33)
[HEIGHT] → (1,70)
```

Instance of the PERSON entity

- Slot filling: incremental / heritage / specific functions
- Obligatory / optional slots. Filling restrictions

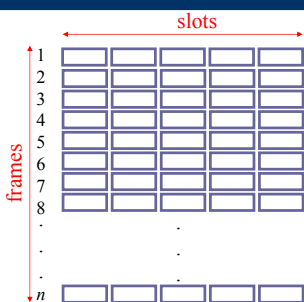
146

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Methods for Knowledge representation



Frame unification procedure: combine values of two frames to obtain a new frame

147

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Example of frame unification

(Viajero system, López-Cózar et al. 2000)

S: Welcome to the service ... # FR STAT SRC DEST TYPE DEPT

U: I want to go to Sevilla → 1 ☒ ☐ ☐ Sevilla ☐ ☐

S: From where do you want to leave?

U: from Granada → 2 ☒ ☐ Granada ☐ ☐ ☐

Unification of frames 1 and 2

→ 3 ☒ ☐ Granada Sevilla ☐ ☐

S: Do you want a direct bus?

U: Yes direct → 4 ☒ ☐ ☐ ☐ Direct ☐

Unification of frames 3 and 4

→ 5 ☒ ☐ Granada Sevilla Direct ☐

S: There are direct buses from Granada to Sevilla at 9, 10, 11 and 12. Which one do you want?

U: The one at 12 → 6 ☒ ☐ ☐ ☐ 12

Unification of frames 5 and 6

→ 7 ☐ Granada Sevilla Direct 12

148

R. López-Cózar, UGR

ESSLLI 2004

Outline

- Types of NLP analyses
- Main problems of NLP
- Some grammars for NLP
- Implicit recovery
- Methods for knowledge representation
- **Evaluation measures**

149

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Evaluation measures

- **Sentence Understanding, SU** (Billi et al. 1997)

$$SU = \frac{S_u}{S_t} \times 100$$

- % correctly understood sentences (a sentence is correctly understood if the **semantic representation obtained = correct semantic representation**)
- S_u = number of sentences correctly understood
- S_t = total number of sentences in the corpus

150

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Evaluation measures

- **Implicit Recovery**, IR (Billi et al. 1997)

$$IR = \frac{S_{IR}}{S_e} \times 100$$

- % sentences **with recognition errors that can be correctly understood** by the system without additional correction turns
- S_{IR} = number of erroneous sentences that are correctly understood
- S_e = total number of erroneous sentences

151

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



References

- Allen, J. 1995. Natural language understanding. The Benjamin/Cummings Publishing Company Inc.
- Billi, R., Castagneri, G., Danielli, M. 1997. Field trial evaluations of two different information inquiry systems". Speech Communication
- Borsley, R. 1996. Modern phrase structure grammar. Blackwell Publishing
- Dale, R., Moisl, H., Somers (eds). 2000. Handbook of natural language processing. Dekker Publishers
- López-Cózar, R., Rubio, A. J., García, P., Díaz-Verdejo, J. E., López-Soler, J. M. 2000. Telephone-based system to assist bus travellers. (*In Spanish*) Proc. First Spanish Meeting on Speech Technologies, CD-ROM ISBN 84-95118-58-0
- Neumann, G. A Uniform Computational Model for Natural Language Parsing and Generation, Ph.D. Thesis, Saarlandes University, Germany, November 1994
- Satta, G. 2000. Parsing techniques for lexicalized context-free grammars. Proc. of the 6th IWPT, Trento, Italy
- Su, Y., Zheng, F., Huang, Y. 2001. Design of a semantic parser with support to ellipsis resolution in a Chinese spoken dialogue system. Proc. Eurospeech, pp. 2161-2164

152

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Course outline

- Introduction
- Automatic Speech Recognition (ASR)
- Natural Language Processing (NLP)
- **Dialogue management**
- Natural Language Generation (NLG) and speech synthesis
- Standards
- Tools

153

R. López-Cózar, UGR

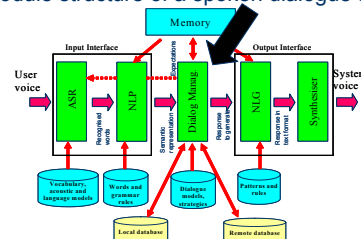
ESSLI 2004

1. Introduction



Localisation

- Module structure of a spoken dialogue system



154

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Outline

- **Dialogue modelling**
- Confirmation strategies
- Interaction strategies
- Multimodal dialogue management
- Multilingual dialogue management
- Evaluation measures
- References

155

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



Dialogue modelling

- Two approaches
 - **Based on finite states**
 - Grammars
 - State-transition networks
 - **Self-organising (or locally handled)**
 - Plans
 - Frames
 - Object oriented
 - Event driven

156

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Dialogue modelling: Grammars

- Rules express sequential regularities in dialogues (**adjacency pairs**) (Traum and Hinkelman, 1992)
 - question → answer
 - question → clarification
 - etc.
- Rules decide acceptable dialogues (analogy with syntactic grammars)
- Grammar terminal elements
 - Speech act types (e.g. “ask”, “confirm”, “accept”, “reject”, etc.)

157

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Dialogue modelling: Grammars

- Advantages**
 - Simple to represent dialogue flow
 - Easy computation
 - Each prompt sets expectations
 - Activate language models, confidence thresholds, ...
- Disadvantages**
 - Grammar construction can be difficult
 - Little flexible interaction, deviation from expected derivations not allowed (problems if the user is informative)
 - Not appropriate to setup mixed-initiative interaction strategy

158

R. López-Cózar, UGR

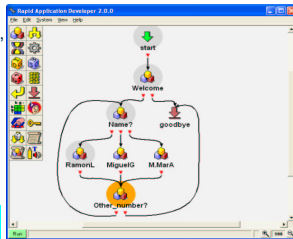
ESSLLI 2004

1. Introduction

Dialogue modelling: State-transition networks

- Model dialogue flow in a very structured way (Aust and Oerder, 1995)
- Very simple
 - States = system prompts
 - Transitions = user utterances

<http://csli.cse.ogi.edu/toolkit/>
 © Oregon Health & Science University
 © University of Colorado
 © University of Edinburgh



159

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Dialogue modelling: State-transition networks

- Main disadvantages**
 - Lack of flexibility
 - User cannot deviate from the predetermined path in the network (problems with over informative utterances)
 - Inefficient to model complex dialogues
- Possible solutions**
 - Restrict user utterances
 - Incorporate additional transitions
 - The network can become intractable !
 - E.g. Philips system: 1,000 questions, several thousands of transitions

160

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Dialogue modelling: Plans

- Developed to solve the lack of flexibility of finite-state approaches (Pollack, 1992)
- Based on inference, rules and axioms
- The system must infer user goals and activate plans
- The system interacts to achieve facts
 - Fact achievement triggers rules that may generate new facts
- Dialogue “states” are generated dynamically
- Advantages**
 - Allow more flexible interaction than finite-state approaches
- Disadvantages**
 - Plan design and utilisation may be complex
 - It may be difficult to create inference rules for complex application domains

161

R. López-Cózar, UGR

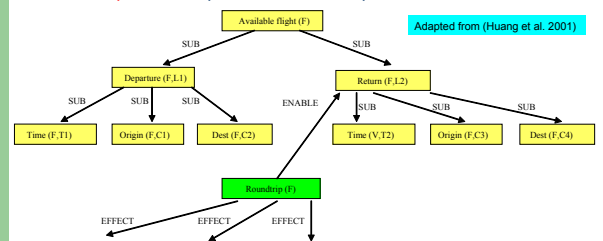
ESSLLI 2004

1. Introduction



Dialogue modelling: Plans

- Example.** Partial plan to make an airplane travel reservation



162

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Outline

- Dialogue modelling
- • Confirmation strategies
- Interaction strategies
- Multimodal dialogue management
- Multilingual dialogue management
- Evaluation measures
- References

163

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Confirmation strategies

- Input interface does not always works correctly
- Confirmations are needed. Techniques:
 - **Ask again**
 - Additional dialogue turns are generated to confirm data
 - Dialogues are less efficient
 - Easy for the user to correct errors
 - **Direct (or explicit) confirmation**
 - Idem to the former, but dialogues seem more natural
 - **Indirect (or implicit) confirmation**
 - Dialogue turns used to obtain new data are also used to confirm previous data
 - Dialogues are more efficient
 - Users may have problems to correct data

164

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Confirmation strategies

- Examples of **explicit** confirmations

S1: Please say the name of the destination city
 U1: Madrid
 S2: did you say Madrid?
 → U2: yes
 S3: ok, destination Madrid. Please say the name of the departure city
 U3: Granada
 S4: Did you say Granada?
 → U4: yes
 S5: ok, departure from Granada. Please say the day of the week you want to travel
 U5: Monday
 ...

165

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Confirmation strategies

- Examples of **implicit** confirmations

S1: Please say the name of the destination city
 U1: Madrid
 S2: ok, destination Madrid. Please say the name of the departure city
 → U2: Granada
 S3: ok, departure from Granada. Please say the day of the week you want to travel
 → U3: Monday ...

- Example of **ask again**

S1: Please say the name of the destination city
 U1: Madrid
 S2: Sorry, I didn't get that. Please say the name of the destination city
 U2: Madrid

166

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Confirmation strategies

- **Confidence scores**
 - Used to decide which data must be confirmed and how
 - Based on acoustic and/or language models
 - Confidence thresholds: $U_1, U_1, U_2, \dots$
 - A possible method to assign confidence scores
 - Use **N-best** list
 - Words that are in many hypothesis "must" have more confidence: assign them a great score

167

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Confirmation strategies

- **Confidence scores**

$$C(A) = \frac{\sum_{i \in I_A} e^{-\lambda \cdot sc_i}}{\sum_{i=1}^N e^{-\lambda \cdot sc_i}}$$

(Souvignier et al. 2000)

- I_A set of hypothesis in which A appears
- sc_i probability of hypothesis i
- λ arbitrary value (probabilities are here logarithmic values)

168

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Interaction strategies: Dialogue systems

- **User-directed strategy**
 - User has the initiative in dialogue
 - Little used in current systems
- **System-directed strategy**
 - System has the initiative in dialogue
 - Restriction of user possible sentences
 - Lack of flexibility
 - Usually some navigation commands are allowed (e.g. "go back", "start over", "help", "remove last entry", "remove all", etc.)

175

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Interaction strategies: Dialogue systems

- **Mixed-initiative strategy**
 - Combines advantages of the two previous strategies
 - User and system can take the initiative in the dialogue
 - User can provide additional data, interrupt the system and correct data
 - System can restrict user sentences and ask for confirmations
 - Requires to handle
 - Larger vocabulary at any dialogue state
 - Greater number of possible user sentences and intentions

176

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Interaction strategies: Selection of the best strategy

- **Mixed-initiative is always the best interaction strategy?**
- (Hone and Baber, 1995)
 - Relationship between restrictions in interaction strategies and time to carry out tasks
 - Time depends mainly on the performance of the speech recogniser
 - In principle, restricted strategy should increment the number of necessary turns and then the time required, however
 - Little restricted interaction strategy
 - Incremented errors, and then incremented time
 - Restricted strategy
 - Decreased errors, and then reduced time

177

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Interaction strategies: Selection of the best strategy

- **Mixed-initiative is always the best interaction strategy?**
- (Potjer et al. 1996)
 - Studied two versions of a dialogue system
 - (1) **System-directed strategy, isolated ASR and keyword detection**
 - (2) **Mixed-initiative strategy, continuous ASR and complex NLP**
 - In theory, less number of turns in version (2)
 - However, number of turns was incremented due to correction of ASR errors
 - Version (1) not seemed to be slower than (2) for users
 - They said to be satisfied with both versions of the system

178

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Interaction strategies: Adaptive strategies

- The performance of a dialogue system can vary notably for different users
 - **IDEA: Adapt strategies according to the dialogue evolution**
- (Litman and Pan, 1999)
 - TOOT system (information about train travel)
 - Users can order the system to change its confirmation and interaction strategies when the dialogue becomes problematic
 - Requires previous training of users

179

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Interaction strategies: Adaptive strategies

- (Litman and Pan, 2000)
 - **Adaptive strategy controlled by the system**
 - **Adaptation**
 - Interaction strategy (system-directed / user-directed / mixed)
 - Confirmation strategy (explicit, implicit, none)
 - **Adaptive module**
 - Use more restrictive strategies when for problematic situations
 - RIPPER program used to infer classification rules
 - Training using tagged dialogue corpora
 - Parameters: efficiency, etc.

180

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Outline

- Dialogue modelling
- Confirmation strategies
- Interaction strategies
- ➔ • Multimodal dialogue management
- Multilingual dialogue management
- Evaluation measures
- References

181

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



Multimodal dialogue management

- Speech is the most natural communication means for human beings, but speech-based interaction with computers has some limitations
 - ASR technology is **error-prone**
 - Users can have **problems to understand** the information provided by the system
 - Users may have problems to **know the state** of the dialogue
- Human communication relies on additional communication modalities
 - Hand/body gestures, facial expressions, lip movement, etc.
- These additional modalities allow
 - Add/modify information in speech-based communication
 - Increase speech recognition rates

182

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Input modalities

- Automatic speech recognition
- Pointing
- Gesture recognition
- Handwriting recognition
- Eye / gaze tracking
- Keyboard
- Object recognition
- Speaker verification
- Face recognition



SmartKom system



183

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



Output modalities

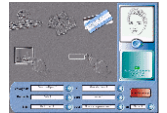
- Speech synthesis
- Graphic user interface (GUI)
- Face synthesis
- Sounds
- Text
- Animated agent



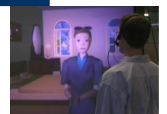
AdApt system



Olga system



Mask system



Rea system



August system

184

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Advantages of multimodal dialogue

- Using several input/output interaction modalities
 - User and system can **obtain more information** from each other
 - Error reduction
 - Greater **interaction freedom** for users
 - Speech, keyboard, mouse, pointing, etc.
 - Users have different preferred modalities
 - For disabled people some modalities are not available
 - Input modalities **can compensate** their own limitations
 - Speech can compensate limitations of GUIs
 - GUIs can compensate speech limitations
 - More **expressive** output
 - Greater user satisfaction

185

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



Cooperation between interaction modalities

- **complementary**: **different** informations are transmitted through one or more modalities (e.g. point with the finger and say "put that there")
- **redundancy**: the **same** information is transmitted through several modalities
 - E.g. point and say "I want the second option on the right"
- **equivalency**: the same information can be transmitted through several modalities
 - E.g. selection of menu option using mouse **or** voice command

186

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Face recognition and tracking

- **Recognition** (detection of user face)
 - Colour-based methods
 - Methods based on facial features
 - Hybrid methods
- **Tracking**
 - Track the face movement in scenes

187

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Face recognition

- **Colour-based methods**
 - Colour distribution of face is restricted to a small region of the chromatic space
 - **problem:** depends on the illumination and camera
 - More useful with additional information channels
 - e. g. image analysis to detect mobile objects (head and hands)



188

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Face recognition

- **Methods based on facial features**
 - Information about face shapes and texture
 - Image processing techniques to detect important facial features:
 - Detection algorithm based on deformable eye patterns (Yuille et al. 1989)
- **Hybrid methods**
 - Robust for a wider range of conditions (lighting, camera, head orientation, ...)
 - Combine information about several analysis channels (e.g. shape, texture, colour and movement)

189

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Face tracking

- After the face can be recognised it can be tracked
- **Methods**
 - Yang and Waibel (1997)
 - Stiefelbogen et al. (1997)
- **Yang and Waibel (1997)**
 - Based on colour, geometry and movement information
 - Independent from lighting, camera, cloths and background
 - Adaptive model of skin colour (using a temporal filter)
 - Great processing speed (using predictions about face movements based on current position and speed)

190

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Face tracking

- **Stiefelbogen et al. (1997)**
 - Better results
 - Based on statistical measures about colour and information about facial geometry (e.g. eyes, nostrils and lip corners)



191

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Recognition of lip movements

- Provides information about word pronunciations
- (Agelfors et al. 1998)
 - Using just lip movement information
 - 100% recognition of three vocals
 - 70% recognition of ten digits
 - 25% recognition of some sentences
- **Training:** Analysis of a real people lip movements
 - Learning parameters associated to the pronunciation of phonemes
- **Recognition**
 - Face detection in images
 - Lip detection in face
 - Comparison of lip shape with previously obtained patterns

192

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Recognition of user gestures

- Gesture types (Scherer and Ekman, 1982)
 - **symbolic**: mean something (e.g. move head to affirm / negate)
 - **deictic**: point to objects (e.g. "put that there")
 - **iconic**: visual description of objects, relations or actions
 - **rhythmical**: rhythm of speaking (e.g. arm movements)

193

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Recognition of user gestures



- **Intrusive devices**
 - Touch-sensitive gloves, movement trackers, etc.
 - Acceptable if the application allows using this kind of devices (e.g. virtual reality)
- **Not intrusive devices**
 - Cameras and artificial vision techniques to segment and classify images
 - Tasks
 - User location
 - Hands and arms location
 - Gesture classification
 - Worse results

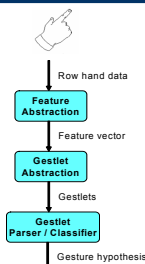
194

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Intrusive algorithm for gesture recognition



Steps

1. **Feature abstraction**: obtains data from the glove (fingers position, orientation and hand movement)
2. **Gestlets abstraction**: constructs of triplets **Gestlet**=(posture, orientation, movement) from the glove data
3. **Gestlets analysis**: decides the user gesture from the set of gestlets

Adapted from (Koons et al. 1993)

195

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Fusion of multimodal input

- **Fusion** = union of information provided by different input modalities (Faconti et al. 1996)
- Can be carried out at several levels
 - **Signal level (lexical fusion)**: combination at the signal level. Some current applications:
 - Audio-visual speech recognition: combination of acoustic information with lip movement information
 - Technologies: ANN, HMM, FFT, etc.
 - **Semantic level**: combination of informations considering their meanings
 - Steps: Fusion in low level module + Interpretation at a higher level
 - E.g. Fusion of sentence "put that there" with hand pointing

196

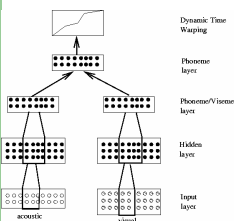
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Fusion of multimodal input: Signal level

- **Artificial Neural Networks (ANNs)** (Meier et al. 1997)
 - Some ANNs are used for acoustic recognition and other for visual recognition. Separately trained and integrated for recognition



Test type	Only visual information	Only acoustic information	Combined information
Clean data	55 %	98.4 %	99.5 %
SNR 16 dB	55 %	59.6 %	73.4 %
SNR 8 dB	55 %	36.2 %	66.5 %

SNR = Signal to Noise Ratio (signal quality)

197

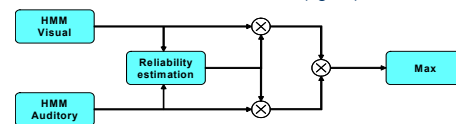
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Fusion of multimodal input: Signal level

- **Hidden Markov Models (HMMs)** (Adjoudani et al. 1997)
 - Initial integration: recognition considering both kind of signals
 - Posterior integration: initial decision based on separated signals and final decision base on both results (figure)



198

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



Representation of multimodal information

- **Melting pots** (Nigay and Coutaz, 1995)
 - Chunks of multimodal Informations are encapsulated with temporal stamps
 - Different fusion mechanisms are used: microtemporal, macrotemporal and contextual
 - **microtemporal**: combination of informations produced simultaneously or very closely in time
 - **macrotemporal**: combination of sequential informations near in time or complementary to each other
 - **contextual**: combination of informations considering semantic restrictions

199

R. López-Cózar, UGR

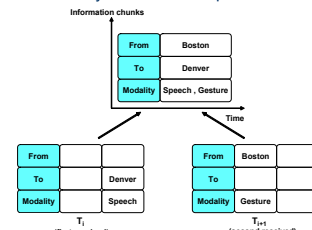
ESSLLI 2004

1. Introduction



Representation of multimodal information

- Example. **Macrotemporal fusion using melting pots**
 - System prompt: "To Denver from Boston?"
 - The user says "Denver" and points "Boston" in a map



Adapted from (Nigay and Coutaz, 1995)

200

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Animated agents

- **Animated agent** = Image of human face/body that makes gestures and facial expressions synchronised with the oral output
 - Greater expressivity and understanding of system messages and status
 - Very useful in noise environments and when there is cross-talk
- (House et al. 2001)
 - Recognition of VCV stimuli in noisy environments:
 - 30% → only speech synthesiser
 - 47% → speech synthesiser + animated agent

201

R. López-Cózar, UGR

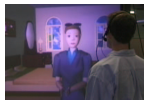
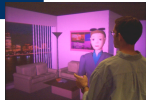
ESSLLI 2004

1. Introduction



Animated agents

- **REA system**
 - Developed at MIT Media Lab (USA)
 - Real Estate Agent
 - Find out user needs
 - Show available apartments
 - Try to sell/rent apartments
 - Input modalities
 - User voice and gestures
 - Output modalities
 - Animated agent, apartment images

<http://gn.www.media.mit.edu/groups/gn/projects/humanoid/>


202

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Animated agents

- **SmartKom** <http://www.smartkom.org/>
 - Developed at the Research Center for Artificial Intelligence (DFKI GmbH, Germany)
 - Three application domains: public, mobile, home/office
 - Input: user speech and gestures
 - Output: animated agent, graphs
 - Representation of multimodal information using XML
- **Olga** <http://www.speech.kth.se/~beskow/papers/fon97olga.html>
 - Developed at KTH Lab (Sweden)
 - Information about microwave ovens
 - Input: speech
 - Output: animated agent + graphics



203

R. López-Cózar, UGR

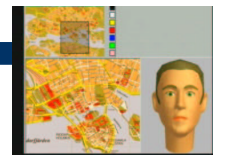
ESSLLI 2004

1. Introduction

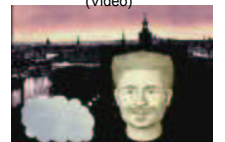


Animated agents

- **AdApt** <http://www.speech.kth.se/ctt/proj/adapt/>
 - Developed at KTH Lab (Sweden)
 - Information about apartments in Stockholm
 - Input: speech + mouse
 - Output: animated agent + map + text
- **August** <http://www.speech.kth.se/august/>
 - Developed at KTH Lab (Sweden)
 - Information about itself, Swedish author August Strindberg, restaurants and other services in Stockholm
 - Input: speech
 - Output: animated agent + thought balloons



(Video)



(Video)

204

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Animated agents

- **Mask** <http://www.limsi.fr/lip/kiosk-sncf.html>
 - Developed at LIMSI Lab (France)
 - Multimodal-Multimedia Automated Service Kiosk
 - Information about train travels, reservations, prices, etc.
 - Input: speech, touch-sensitive screen
 - Output: speech, graphs, videos, sounds
 - Set up in St-Lazare station, Paris

S: The MASK system is listening.
 U: Uh ... I'd like to go from Paris to Bordeaux tomorrow ...
 S: What time do you want to leave?
 U: around 12:30
 S: There is a train at 12:43, do you want to make a reservation?
 U: Yes in first class

205

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Outline

- Dialogue modelling
- Confirmation strategies
- Interaction strategies
- Multimodal dialogue management
- • Multilingual dialogue management
- Evaluation measures
- References

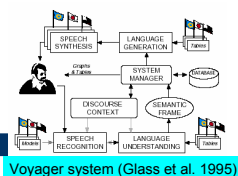
206

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Multilingual dialogue management



Voyager system (Glass et al. 1995)

- Multilingual systems can interact using several languages (e.g. English, French, Spanish, etc.)
- **Fundament**: obtain the same semantic representation, independently of the language
- Characteristics:
 - **ASR**: several vocabularies, training acoustic and language model for each available language
 - **Dialogue management**: is not affected
 - **Response generation**: patterns specific for each language
 - **Text-to-Speech**: multilingual synthesiser (e.g. Festival)

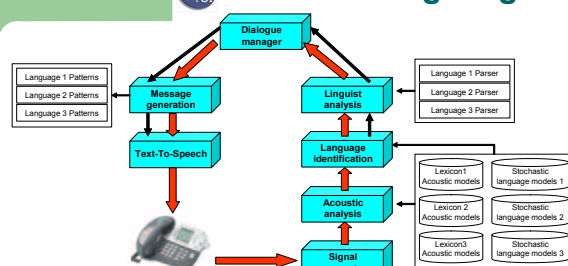
207

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Processing stages

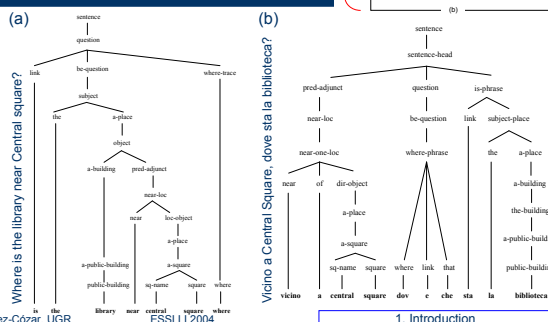


- **Signal processing**
- **Acoustic analysis**
 - Acoustic models
- **Language identification**
 - Monolingual recognisers for the available languages
 - Lexicons, acoustic models, language models for each available language
- **Linguistic analysis**
 - Parsers for each language
- **Dialogue management**
- **Message generation**
 - Depending on the language
- **Text-to-Speech conversion**
 - Depending on each language

208

Multilingual Speech understanding

The **same** semantic representation is obtained for both languages



209

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Multilingual response generation

flight_leg → airline_flight leaves_from
 airline_flight → :airline "flight" :flight_number
 leaves_from → "leaves" from_source at_dpt_time
 at_dpt_time → "at" :depart_time
 from_source → "from" :source

Patterns for response generation in English

United flight 94 leaves from Boston at 3:00 p.m.

flight_leg → airline_flight leaves_from
 airline_flight → :airline "vuelo" :flight_number
 leaves_from → "salida" from_source at_dpt_time
 at_dpt_time → "a las" :depart_time
 from_source → "desde" :source

Patterns for response generation in Spanish

United vuelo 94 salida desde Boston a las 3:00 p.m.

210

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Multilingual management

- **Jupiter system** (Zue et al. 2000)
 - Languages: English, German, Mandarin Chinese, Spanish
 - Weather forecast information (obtained from Internet, *National Weather Service*)
 - Developed using the **Wizard of Oz** technique
- **Voyager system** (Glass et al. 1995)
 - Languages: English, Japanese, Italian
 - Geographic information about Boston (USA)
 - Maps, aerial photographs, distances between places, etc.

211

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Voyager system

U1: Show me the map of the universities in Cambridge
(Voyager shows a map of the MIT and Harvard)
U2: How do I get to Harvard?
(Voyager asks "Where are you starting from?")
U3: MIT
(Voyager gives directions from Harvard to MIT)
U4: show me an aerial photo of Harvard
(Voyager displays an aerial photo of Harvard)
U5: What museums are near there?
(Voyager provides a list of museums near Harvard)
U6: How is the traffic on Memorial Drive?
(Voyager reads the traffic report for Memorial Drive)
U7: Is there any construction on the Central Artery?
(Voyager gives info about current construction on the Central Artery)
U8: Are there any delays?
(Voyager reports about any delays on the Central Artery)

(Glass et al. 1995)

<http://www.sls.csail.mit.edu/sls/whatwedo/applications/voyager.html>

212

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Outline

- Dialogue modelling
- Confirmation strategies
- Interaction strategies
- Multimodal dialogue management
- Multilingual dialogue management
- • Evaluation measures
- References

213

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Evaluation measures

- **Turn Correction Ratio** (TCR)

$$TCR = \frac{t_c}{t_t} \times 100$$

- % of turns used to repair errors
 - User Correction Turns (UCT)
 - System Correction Turns (SCT)
- t_c = number of turns used to repair errors (UCT + SCT)
- t_t = total number of turns in the dialogue corpus

214

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Evaluation measures

- **Contextual Appropriateness** (CA)

$$CA = \frac{t_{sa}}{t_{st}} \times 100$$

- How appropriate/correct is the system performance?
- Little appropriateness if the system prompts the user for data s/he previously provided (perhaps unexpectedly)
- t_{sa} = number of turns in which the system has an appropriate behaviour (responds as logically expected)
- t_{st} = total number of system turns

215

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Evaluation measures

- **Transaction Success** (TS)

$$TS = \frac{t_{si}}{t_{ur}} \times 100$$

- How successful is the system in providing the information requested by the user?
- t_{si} = system turns in which it gives the correct information
- t_{ur} = total number of turns in which the users requests information

216

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



References

- Adjoudani, A., Guiard-Marigny, T., LeGoff, B., Reveret, L., Benoit, C. 1997. A multimedia platform for audio-visual speech processing. Proc. Eurospeech, pp. 1671-1674
- Agelfors, E., Beskow, J., Dahlquist, M., Granström, M., Lundeberg, K.-E., Spens, K.-E., Ohman, T. 1998. Synthetic faces as a lipreading support. Proc. ICSLP '98, Sydney, Australia
- Aust, H., Oerder, M. 1995. Dialogue control in automatic inquiry systems. ESCA Workshop on Spoken Dialogue Systems, pp. 121-124
- Ekman, P., Friesen, W. 1978. Facial Action Coding System. Consulting Psychologists Press, Inc.
- Faconti, G., Bordegoni, M., Kansy, K., Trahanias, P., Rist, T., Wilson, M. 1996. Formal framework and necessary properties of the fusion of input modes in user interfaces. Interacting with computers, vol. 8(2), pp. 134-161
- Glass, J., Flammia, G., Goodine, D., Phillips, M., Polifroni, J., Sakai, S., Seneff, S., Zue, V. 1995. Multilingual spoken-language understanding in the MIT Voyager system. Speech Communication, 17 (1-2), pp. 1-18

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



References

- House, D., Beskow, J., Granström, B. 2001. Timing and interactions of visual cues for prominence in audiovisual speech perception. Proc. Eurospeech , pp. 387-390
- Hone, K. S., Baber, C. 1995. Using a simulation method to predict the transaction time effects of applying alternative levels of constraint to utterances within speech interactive dialogues. Proc. ESCA Workshop on Spoken Dialogue Systems, pp. 209-212
- Huang, X., Acero, A., Hon, H. W. 2001. Spoken language processing: A guide to theory, algorithm, and system development. Prentice Hall
- Koons, D., Sparrell, C., Thorisson, K. 1993. Integrating simultaneous input from speech, gaze and hand gestures. M. Maybury, ed., Intelligent multimedia interfaces, pp. 257-275, Morgan Kaufmann
- Litman, D. J., Pan, S. 1999. Empirically evaluation an adaptable spoken dialogue system. Proc. 7th International Conference on User Modeling, pp. 55-64
- Meier, U., Stiefelhagen, R., Yang, Y. 1997. Processing of visual speech under real world conditions. Proc. AVSP '97 workshop, Rhodes, Greece

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



References

- Nigay, L., Coutaz, J. 1995. A generic platform for addressing the multimodal challenge. Proc. International Conference on Human-Computer Interaction, ACM, pp. 98-105
- Pollack, M. E. 1992. The uses of plans. Artificial Intelligence, 57(1), pp. 43-68
- Potjer, J., Russel, A., Boves, L., Den Ons, E. 1996. Subjective and objective evaluation of two types of dialogues in a call assistance service. Proc. IVTTA, pp. 121-124
- Scherer, K., Ekman, P. 1982. Handbook of Methods in Nonverbal Behavior Research. Cambridge University Press
- Sidner, C. L., Forlines, C. 2002. Subset languages for conversing with collaborative interface agents. Proc. ICSLP, pp. 281-284
- Souvignier, B., Kellner, A., Rueber, B., Schramm, H., Seide, F. 2000. The thoughtful elephant: Strategies for spoken dialog systems. IEEE Transactions on Speech and Audio Processing, vol. 8, no. 1
- Stiefelhagen, R., Yang, J., Waibel, A. 1997. A model-based gaze tracking system. Int. Journal of Artificial Intelligence, 6(2), pp. 193-209

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



References

- Traum, D., Hinkelman, E. A. 1992. Conversation acts in task-oriented spoken dialogue. Computational Intelligence, 8(3). Special Issue on Non-literal Language
- Yang, J., Waibel, A. 1997. A real-time face tracker. Int. Journal of Artificial Intelligence Tools, 6, pp. 193-209
- Yuille, A., Cohen, D., Hallinan, P. 1989. Feature extraction from faces using deformable templates. Conference on vision and pattern recognition, pp. 104-109
- Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T., Hetherington, L. 2000. Jupiter: A telephone-based conversational interface for weather information. IEEE Trans on Speech and Audio Proc., 8(1), pp. 85-89

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Course outline

- Introduction
- Automatic Speech Recognition (ASR)
- Natural Language Processing (NLP)
- Dialogue management
- Natural Language Generation (NLG) and speech synthesis
- Standards
- Tools

221

R. López-Cózar, UGR

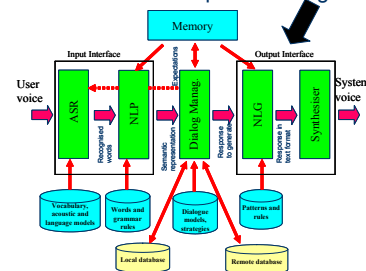
ESSLLI 2004

1. Introduction



Localisation

- Module structure of a spoken dialogue system



222

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Outline

- Natural Language Generation (NLG)
 - Objective
 - Stages
 - Main problems
 - NLG systems
- Speech synthesis
 - Voice coding
 - Parametric synthesis
 - Formant synthesis
 - Text-To-Speech (TTS)

223

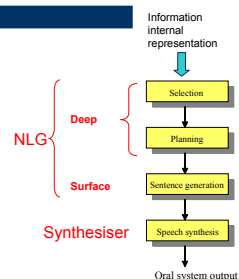
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Natural Language Generation (NLG)

- **Objective:** generation of text message from the internal representation of information handled by the computer
- Stages (Neumann, 1994; Hovy, 1993)
 - **Deep generation:** selection of information to provide to the user and organisation and sentence-like structures
 - **Syntactic (or surface) generation:** creation of syntactically correct sentences from the sentence-like structures



224

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Natural Language Generation (NLG)

- Main problems
 - Keep syntactic (gender/number correspondences) and semantic correction, avoid wrong sentences:
 - Did you say you want **one** sandwiches?
 - Did you say you want a **lemon beer**?
 - Avoid redundancy (using anaphora and ellipsis)
 - **Anaphora:** Did you say you want **it** big?
 - **Ellipsis:** The ham sandwich costs 2,30 euros and the bacon (...) costs 2 cents
 - Election of the most appropriate words to express a sentence: John's car, John's sport car, The car, The red vehicle, The red Mazda, etc.

225

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

NLG systems

- **Canned systems**
 - Used by commercial software products (e.g. generation of error messages)
- **Template-based systems**
 - Generation of the same types of sentence, but with some variations
 - Pattern = concepts + expressions + gaps
 - **Planning stage:** selection of patterns to generate the message
 - **Generation stage:** expansion of concepts and expressions, and gap filling with the appropriate words

226

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

NLG systems

- **Example.** Pattern to ask the user for "drink taste" (SAPLEN system, López-Cózar et al. 1997)

```
[ "WHICH" ["OTHER"] "TASTE" ]
[ <desire> ]
[ "FOR THE" ]
[<drink_size> <temperature> <drink_name> <complements> ]
"?"
```

- Example of sentences this pattern can generate:
 - WHICH OTHER TASTE WOULD YOU LIKE FOR THE BIG FANTA?
 - WHICH TASTE?
 - WHICH TASTE DO YOU WANT FOR THE SMALL MILKSHAKE?

227

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

NLG systems

- **Phrase-based systems**
 - Use generalised patterns (constituted of other patterns)
 - Each pattern is divided recursively into simpler patterns until all concepts, expressions and gaps are transformed into words

```
[ PATTERN ] →
[ SUBJECT PREDICATE ] →
[ [ ARTICLE ADJECTIVE NAME ] [ VERB COMPLEMENT ] ] →
The white dog is in the courtyard
```

- Very flexible and robust but difficult to build if the application is very complex

228

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

NLG systems

- **Feature-based systems**
 - Expressions are represented using a set of simple features
 - Mode: affirmative, negative, imperative, interrogative, etc.
 - Tense: present, past, future
 - Etc.
 - Sentence = set of expressions = set of features
 - NLG = election of the most appropriate features to build each information segment

Sofisticación

229

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Outline

- Natural Language Generation (NLG)
 - Objective
 - Stages
 - Main problems
 - NLG systems
- • Speech synthesis
 - Voice coding
 - Parametric synthesis
 - Formant synthesis
 - Text-To-Speech (TTS)

230

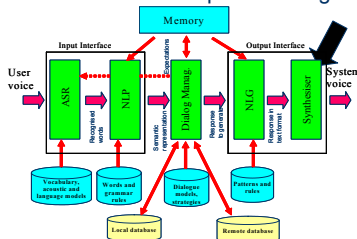
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Speech synthesis

- Module structure of a spoken dialogue system



231

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Outline

- Natural Language Generation (NLG)
 - Objective
 - Stages
 - Main problems
 - NLG systems
- • Speech synthesis
 - Voice coding
 - Parametric synthesis
 - Formant synthesis
 - Text-To-Speech (TTS)

232

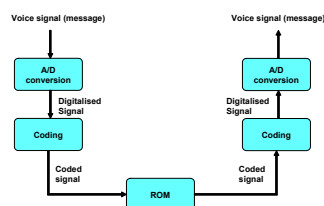
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Speech synthesis

- **Voice coding** (Barnwell et al. 1996)
 - Sentences to synthesise are previously recorded and stored
 - Recording
 - A/D signal conversion → digitalised voice (bit sequence)
 - Bit coding (to reduce the number of necessary bits)
 - Storage in ROM
 - Play back (speech synthesis)
 - Decoding
 - D/A conversion
 - Obtained signal is sent to the speaker



233

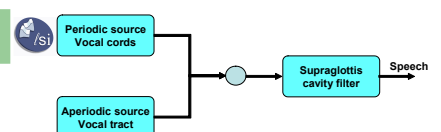
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Speech synthesis

- **Parametric synthesis** (Beskow, 1998)
 - Uses signal elements that can be easily modified using the **source / filter model**
 - **Source**: models excitation of vocal tract
 - **Filter**: models articulation (e.g. position and movement of articulators)
 - Source/filter model is controlled by several hundreds of phonetic rules based on the human vocal tract
 - Air flows from the lungs
 - Noise is produced when the air crashes with obstacles



234

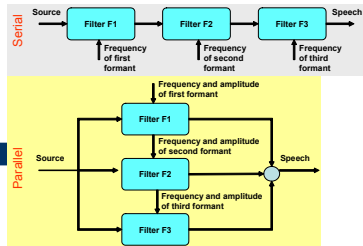
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Speech synthesis

- **Formant-based synthesis** (Puckette, 1995)
 - Based on units called **formants**
 - **Formant** = group of harmonics of the periodic waveform produced by the vibration of the vocal cords
 - Electronic resonators generate the formants
 - **Phoneme** = configuration of several formants
 - Speech synthesis carried out combining properly the frequencies of several electronic resonators
 - Architectures: **serial** and **parallel**



235

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Outline

- Natural Language Generation (NLG)
 - Objective
 - Stages
 - Main problems
 - NLG systems
- Speech synthesis
 - Voice coding
 - Parametric synthesis
 - Formant synthesis
 - Text-To-Speech (TTS)

236

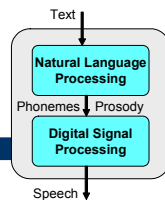
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Text-To-Speech Synthesis (TTS)

- **TTS synthesiser** (Dutoit, 1996)
 - Computer program that reads any text aloud
 - Automatic production of speech through a letter-to-sound transcription of the sentences to utter
 - Limitations of the word/sentence concatenation systems
 - Concatenate isolated words or parts of sentences
 - Only applicable when a limited vocabulary is required (generally a few hundreds of words)
 - Sentences to generate respect a very restricted structure (e.g. announcements of train arrivals/departures)
 - TTS is a very complex task, although very simple for humans



237

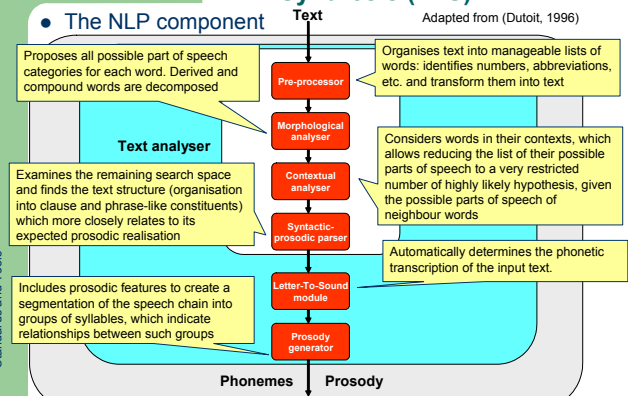
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Text-To-Speech Synthesis (TTS)

- The NLP component



238

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Text-To-Speech Synthesis (TTS)

- The DSP component
 - Operates analogously to dynamically controlling articulatory muscles and vibration frequency of the vocal cords, taking articulatory constraints into account
 - This tasks can be carried out
 - **Explicitly**: a series of rules that formally describe the influence of phonemes on one another. **Synthesis-by-rule**
 - **Implicitly**: storing examples of phonetic transitions and co-articulations into a speech segment database and using them just as they are (in place of phonemes). **Synthesis-by-concatenation**

239

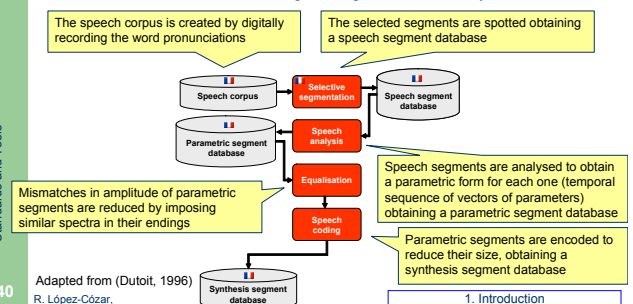
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Text-To-Speech Synthesis (TTS)

- The DSP component (synthesis-by-concatenation)
 - **Database preparation**
 - Segments are chosen to minimise concatenation problems (often diphones and triphones)
 - A list of words containing the segments is carefully created



240

R. López-Cózar,

1. Introduction



Text-To-Speech Synthesis (TTS)

- The DSP component (synthesis-by-concatenation)
 - Task of the synthesiser**
 - Obtain an **adequate sequence of concatenated segments** from the parametric segment database
 - Adjust prosody** (intonation and duration)
 - From the original intonation and duration stored in the original speech corpus
 - To the ones imposed by the natural language processing module

241

R. López-Cózar, UGR

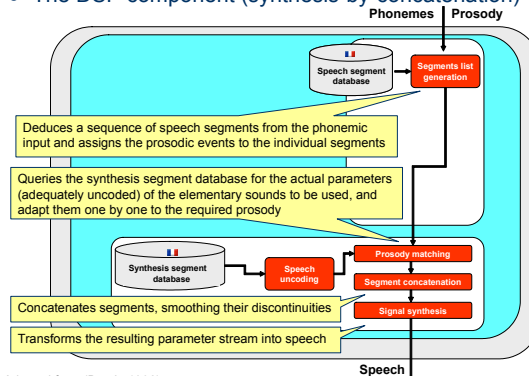
ESSLLI 2004

1. Introduction



Text-To-Speech Synthesis (TTS)

- The DSP component (synthesis-by-concatenation)



242

Adapted from (Dutoit, 1996)

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Text-To-Speech Synthesis (TTS)

- Quality of the concatenative synthesis depends on
 - Type of segments chosen**
 - Allow for as many coarticulation effects as possible
 - Easily connectable
 - Number and length kept as small as possible (but longer units decrease concatenation problems. Some trade-off is necessary)
 - Model of speech signal used for speech analysis and synthesis**
 - Production models**
 - Mathematical models of the human phonation process
 - LPC (Linear Predictive Coding) (Markel and Gray, 1976)
 - Formant (MITTALK, JSRU, INFOVOX, etc.).
 - Phenomenological models**
 - No consideration about human production mechanism
 - H/S (hybrid Harmonic/Stochastic) (Abrantes et al. 1991)
 - TD-PSOLA (Time-Domain Pitch-Synchronous-Overlap-Add) (Moulines and Charpentier, 1990)

243

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Text-To-Speech synthesis

- Examples in English
 - Festival** (University of Edinburgh)
 - Diphone concatenation
 - English (British and American), Spanish, Welsh
 - American English (Kal)
 - American English (Ked)
 - American English (female voice, MBROLA)
 - British English (mail voice)

<http://www.cstr.ed.ac.uk/projects/festival/>

Hello friend, how are you?
I am speaking in English
This is a simple example



244

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Text-To-Speech synthesis

- Examples in English
 - ViaVoice** (IBM)
 - American English (mail voice)
 - American English (female voice)
 - American English (child voice)
 - American English (elderly woman)
 - American English (elderly woman, **faster**)

<http://www-306.ibm.com/software/voice/viavoce/dev/index.shtml>

Hello friend, how are you?
I am speaking in English
This is a simple example



245

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Text-To-Speech synthesis

- Examples in Spanish
 - Zaragoza University, Spain**
 - Diphone concatenation
 - Madrid Polytechnic University, GTH, Spain**
 - Diphone concatenation
 - Festival**
 - ViaVoice**
 - Adult man
 - Adult woman
 - Child
 - Elderly woman

Hola amigo, ¿Qué tal te encuentras?
¿Cómo estás hoy?
Esto es un pequeño ejemplo



246

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



References

- Abrantes, A. J., Marques, J. S., Trancoso, I. M. 1991. Hybrid sinusoidal modeling of speech without voicing decision. *Proc. of Eurospeech*, pp. 231-234
- Baptist, I., Seneff S. 2000. GENESIS-II: A versatile system for language generation in conversational system applications. *Proc. of ICSLP*
- Baptist, L. 2000. GENESIS-II: A language generation module for conversational systems. SM. Thesis, MIT
- Barnwell, T. P., Nayebi, K., Richardson, C. H. 1996. Speech coding: A computer laboratory textbook. Georgia Tech Digital Signal Processing Laboratory Series
- Beskow, J. 1998. A tool for teaching and development of parametric speech synthesis. *Proc. Fonetik 98*, Stockholm, Sweden
- Dutoit, T. 1996. An introduction to text-to-speech synthesis. Kluwer Academic Publishers
- Hovy E. H. 1993. Automated discourse generation using discourse relations. *Artificial Intelligence*, 63, pp. 341-385
- López-Cózar, R., García, P., Díaz, J., Rubio, A. J. 1997. A voice activated dialog system for fast-food restaurant applications. *Proc. Eurospeech*, pp. 1783-1786
- Markel, J. D., Gray Jr, A. H. 1976. Linear prediction of speech. Springer Verlag, New York, pp. 10-42

247

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

References

- Moulines, E., Charpentier, F. 1990. Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*, Vol. 9, 5-6
- Neumann, G. 1994. A Uniform Computational Model for Natural Language Parsing and Generation. PhD Thesis. Universidad de Saarlandes, Germany
- Puckette, M. 1995. Formant-based audio synthesis using nonlinear distortion. *Audio Ingeneering Society*, Reprinted from JAES 43/1, pp. 40-47

248

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



Course outline

- Introduction
- Automatic Speech Recognition (ASR)
- Natural Language Processing (NLP)
- Dialogue management
- Natural Language Generation (NLG) and Speech synthesis
- Standards
- Tools

249

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Outline

- Standards
 - VoiceXML
 - CCXML
 - CallXML

250

R. López-Cózar, UGR

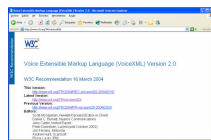
ESSLLI 2004

1. Introduction



VoiceXML

- In the **80s and 90s**, developers of spoken dialogue systems set up systems programming at low level
- In the **90s** appeared web browsers that supported speech
 - Developers could concentrate on the application logic, letting aside low-level issues
- **VoiceXML (or VXML)**
 - Standard based on XML developed by the W3C (World Wide Web Consortium) to access web information using speech
 - **System** → **User** interaction: synthesised speech, pre-recorded voice files
 - **User** → **System** interaction: speech, DTMF
- **VoiceXML versions**
 - v1.0 (2000) <http://www.w3.org/TR/2000/NOTE-voicexml-20000505/>
 - v2.0 (2001) <http://www.w3.org/TR/voicexml20/>



251

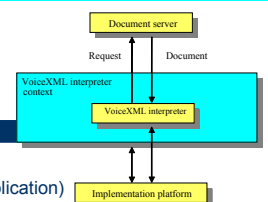
R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

VoiceXML

- Architecture
 - **VoiceXML interpreter** (client application)
 - Execute application logic
 - Play prompts and process user requests
 - Search for information in web servers and provide it to the user
 - **Document server** (Web server)
 - Process requests sent by the VoiceXML interpreter
 - Provide VoiceXML documents
 - **VoiceXML interpreter context**
 - Process VoiceXML document
 - Answer user calls
 - Supervise user input (help, no answer, etc.) and play predefined messages

<http://www.w3.org/TR/2000/NOTE-voicexml-20000505/> © W3C


252

R. López-Cózar, UGR

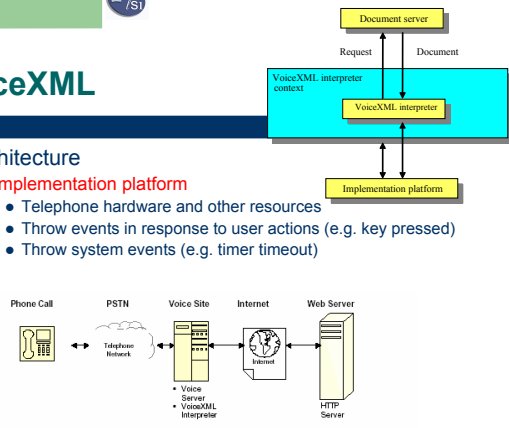
ESSLLI 2004

1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Architecture
 - Implementation platform
 - Telephone hardware and other resources
 - Throw events in response to user actions (e.g. key pressed)
 - Throw system events (e.g. timer timeout)

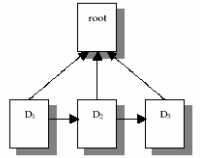


253 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Application
 - Set of VoiceXML documents D_i that share a root document of the application
 - When the user interacts with a document, its root document is always in the system memory
 - Root document is removed from memory when a transition is made to another document that does not belong to the application
 - Variables in root document are visible from any other document of the application
 - Grammars of the root document are always active, independently of the application document in which the user is interacting



254 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Session: a session begins when the user starts to interact with the VoiceXML interpreter
- Grammar: vocabulary and sentences allowed in each dialogue state. A state can have one or more grammars associated.
- Events: can be thrown by the implementation platform due to a variety of reasons (e.g. no response from user, user response not understood, user asks for help, errors in document, etc.)
- Links: specify transitions to
 - other parts of the same document
 - other document of the same application
 - other document of other application

255 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Declaration of variables


```
<var name="telephone"/>
<var name="telephone" expr="6305551212"/>
<var name="y" expr="document.z+1"/>
<var name="city" expr="Nancy"/>
```

telephone has the special value undefined
- Value assignation to variables


```
<assign name="flavour" expr="chocolate"/>
<assign name="document.mycost" expr="document.mycost+14"/>
```
- Clear variables


```
<clear namelist="city state zip"/>
```

If variables are not specified, all the variable in the form are cleared

256 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Executable content
 - Procedural logic that can be in:
 - <block> (section of a form)
 - <filled> (section of a form)
 - <filled> (section of a field)
 - Event handlers (e.g. <catch>, <help>, etc.)
 - Elements that can be inside executable content
 - <var ...>
 - <assign ...>
 - <clear ...>
 - <if ...> ... <elseif ...> ... <else> ...
 - <prompt ...>
 - <reprompt ...>
 - <goto ...>
 - <submit ...>
 - <exit>
 - <return>
 - <disconnect>
 - <script> ...

257 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Conditional logic
 - <if> ... <elseif> ... <else> ... </if> elements are used to create conditional sections in VoiceXML documents. <elseif> and <else> are optional

```
<if cond="total > 1000">
  <prompt>You spent too much money.</prompt>
</if>

<if cond="amount < 29.95">
  <assign name="x" expr="amount"/>
<else/>
  <assign name="x" expr="29.95"/>
</if>

<if cond="taste == 'vanilla'">
  <assign name="taste_code" expr="'v'"/>
<elseif cond="taste == 'chocolate'">
  <assign name="taste_code" expr="'c'"/>
<elseif cond="taste == 'strawberry'">
  <assign name="taste_code" expr="'s'"/>
<else/>
  <assign name="taste_code" expr="'?'"/>
</if>
```

258 1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools


<http://www.w3.org/TR/2000/NOTE-voicexml-20000505/>
© W3C

VoiceXML

- Variables of form items and conditions
 - Each form item has associated a variable that by default is set to **undefined**
 - This variable is given a name using the **name** attribute (e.g. <field name="destinationCity" ...>)
 - Each item has associated a **guard condition** that determines if the item can be selected by the FIA to be visited
 - Generally, initial values are not assigned to the variables associated to the form fields, and no explicit guard conditions are used ...
 - ... but sometimes it is necessary a **more detailed flow control**
 - One way to do this is to assign guard conditions to the items

destinationCityReservation is visited only if destinationCity == undefined

```

<!-- destination city for reservation -->
<field name="destinationCityReservation" cond="destinationCity == undefined">
  <prompt>Please say the name of the destination city</prompt>
  <grammar src="cities.jsgf"/>
</field>

```

265
R. López-Cózar, UGR
ESSLI 2004
1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools


<http://www.w3.org/TR/2000/NOTE-voicexml-20000505/>
© W3C

VoiceXML

- FIA phases
 - Initialisation
 - 1.1 Initialisation of prompt counter variables (set to 1)
 - 1.2 Each variable is set to **undefined** or to the value of the **expr** attribute
 - Loop
 - 2.1 **Selection phase**: determine next item to visit
 - 2.2 **Collection phase**: obtain user input
 - 2.3 **Process phase**: process user input or event collected during the collection phase

266
R. López-Cózar, UGR
ESSLI 2004
1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools


<http://www.w3.org/TR/2000/NOTE-voicexml-20000505/>
© W3C

VoiceXML

- Several techniques can be employed to control the order in which form items are visited by the FIA:
 - Assign a value to the form item , then it is not visited
 - E.g. <assign name="sourceCity" expr="true"/>
 - Use <clear> to set the item to **undefined**, which makes it eligible to be visited
 - E.g. <clear namelist="sourceCity destinationCity"/>
 - Use <goto ...> to explicitly specify the next item to visit
 - E.g. <goto nextitem="confirmExit"/>

267
R. López-Cózar, UGR
ESSLI 2004
1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools


<http://www.w3.org/TR/2000/NOTE-voicexml-20000505/>
© W3C

VoiceXML

- Interaction strategy
 - System-directed
 - The simplest one: form fields are visited one by one in sequential order (only one field is filled in each interaction)
 - Only the voice and/or DTMF grammars of the visited field are active
 - Mixed
 - Grammars of other states can be active when the user is in a given state of the document or of the application
 - If the user utters a sentence allowed by other state's grammar then the execution continues in the other state
 - Great flexibility

268
R. López-Cózar, UGR
ESSLI 2004
1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools


<http://www.w3.org/TR/2000/NOTE-voicexml-20000505/>
© W3C

VoiceXML

Example. System-directed strategy

```

<form id="weather-information">
  <block>Welcome to this automatic service for weather information.</block>
  <field name="province">
    <prompt>In which province?</prompt>
    <grammar src="province.gram" type="application/x-jsgf"/>
    <catch event="help">
      Please say the name of the province for which you want to know the weather information.
    </catch>
  </field>
  <field name="city">
    <prompt>In which city?</prompt>
    <grammar src="city.gram" type="application/x-jsgf"/>
    <catch event="help">
      Please say the name of the city for which you want to know the weather information.
    </catch>
  </field>
  <block>
    <submit next="/servlet/weather_info" namelist="city province"/>
  </block>
</form>

```

```

S: Welcome to our service... In which province?
U: help
S: Please say the name of the province ...
U: Granada
S: In which city?
U: Madrid
S: I didn't understand. In which city?
U: Loja
S: The weather in Loja Granada is sunny at 12 AM ...

```

269
ESSLI 2004
1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools


<http://www.w3.org/TR/2000/NOTE-voicexml-20000505/>
© W3C

VoiceXML

- Mixed initiative
- System and user can take the initiative in the dialogue (i.e. both can decide what to do next)
- One or more **<initial>** elements and one or more form-level grammars must be in the document
- When there are form-level grammars:
 - Fields can be filled in any order
 - More than just a field can be filled by the same user sentence
- Form-level grammars can be active when the user is in other dialogs
- Example:
 - A document has two forms: car rent and hotel reservation
 - Both forms have active grammars for the document
 - User can provide hotel reservation information when the system asks for car rent information

270
R. López-Cózar, UGR
ESSLI 2004
1. Introduction

271

ESLLI 2004

1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

Example. Mixed initiative

Allows recognising e.g. "from Paris to Nancy"

The initial message cannot be interrupted by the user

The variable associated to the field **initial_prompt** is set to **true**, thus the field is not visited again by the FIA

Necessary to hear the next prompt

Allows recognising e.g. "Paris"

```

<form id="get_from_and_to_cities">
  <grammar src="http://www.directions.example/grammars/from_to.gram"/>
  <block>
    <prompt bargain="false">
      Welcome to the Driving Directions by phone.
    </prompt>
  </block>
  <initial name="initial_prompt">
    <prompt>Where do you want to drive from and to?</prompt>
    <nomatch count="1">
      For example, say from Paris to Nancy.
    </nomatch>
    <nomatch count="2">
      I'm sorry, I still don't understand.
      I'll ask you for information one piece at a time.
      <assign name="initial_prompt" expr="true"/>
      <reprompt>
        Necessary to hear the next prompt
      </reprompt>
    </nomatch>
  </initial>
  <field name="sourceCity">
    <grammar src="http://www.directions.example/grammars/city.gram"/>
    <prompt>From which city are you leaving?</prompt>
    ... etc. ...
  </field>
  ... etc. ...
</form>

```

272

ESLLI 2004

1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Links: **<link>**
 - Can have one or more grammars associated
 - Are activated when the user input is accepted by any of these grammars
 - Scope of links
 - If inside of **<vxml>** → grammars active throughout the document
 - If inside of **<form>** → grammars active in the form
 - If in root document at document level → grammars active in any document of the application
 - They allow
 - Transiting to a new document or dialog (like **<goto>**)
- throw an event (like **<throw>**)

This link is activated if the user utters these words or presses the "2" key in the telephone

If the user utters these words, a "help" event is thrown

```

<link next="http://www.voicexml.org/books/main.vxml">
  <grammar type="application/x-jsgf"> books | VoiceXML books </grammar>
  <dtmf> 2 </dtmf>
</link>

<link event="help">
  <grammar type="application/x-jsgf">
    I don't understand | help | explain to me
  </grammar>
</link>

```

273

ESLLI 2004

1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- The **<grammar>** element is used to specify a grammar that is used to:
 - Decide the set of recognisable sentences (i.e. what the user can say to carry out actions or provide data)
 - Obtain either a character string (**field level grammar**) or a pair attribute-value (**form level grammar**)
- Types of grammars
 - Internal
 - External

Internal grammar

Form-level external grammar

Field-level external grammar

```

<link next="#exit">
  <grammar>goodbye | bye | finish </grammar>
</link>

<form id="information_management">
  <grammar src="travel_from_to.jsgf"/>
  <initial>
    <prompt>How may I help you? </prompt>
  </initial>
  <!-- destination city -->
  <field name="destinationCity">
    <prompt>Which city you want to travel?</prompt>
    <grammar src="cities.jsgf">
      Field-level external grammar
    </grammar>
  </field>
  ...
</form>

```

274

ESLLI 2004

1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- E.g. JSGF grammar (Java Speech Grammar Format)

Form level grammar

Field level grammar

```

#JSGF V1.0;
grammar mainElection;
public <mainElection> = [<start>] <sandwich> [<final>];

<start> = [please] I want | I would like to help | what about;
<final> = please | would be fine;

<sandwich> =
  [<amount>] <ingredient>
  [<sandwich> | sandwiches ];
<amount> = a | one | two | three | four;
<ingredient> = ham | cheese | bacon ;

#JSGF V1.0;
grammar travel_from_to;
public <travel_from_to> =
  [<desire>]
  [<travel> <city> (this.destinationCity=$city) ]
  [<from> <city> (this.sourceCity=$city) ];

<desire> = I want | I would like | I need ;
<travel> = [ to get | to go ] to;
<city> = Paris | Nancy | Madrid | London |
  Lisbon | Edinburgh | Granada | Barcelona;
<from> = from | leaving from;

```

If the sentence contains "<travel> <city>" the city is assigned to the **destinationCity** field

If the sentence contains "<from> <city>" the city is assigned to the **sourceCity** field

275

ESLLI 2004

1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Scope of grammars
 - Field grammar**: only active when the FIA visits the field. Does not have "scope" attribute
 - Link grammar**: the scope on the element that contains the link. Does not have "scope" attribute
 - Form grammar**:
 - scope="dialog"**: only active when the user is in the form (the default)
 - scope="document"**: active when the user is in any dialog of the document
 - scope="document"** and the document is the application **root document**, active when the user is in any dialog of any document of the application
 - Menu grammar**: by default **dialog**. Only active when the user is in the menu

276

ESLLI 2004

1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Event handling
 - Generated by the **implementation platform** if the user does not respond, asks for help, etc.
 - Generated by the **VoiceXML interpreter** if it finds errors or a **<throw>** element in the document
 - Elements related to events:
 - <throw>**
 - <catch>**
 - <error>**
 - <help>**
 - <noinput>**
 - <no match>**
 - <throw>** is used to throw an event

```

<throw event="nomatch">
<throw event="telephone.disconnect.hangup">

```

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Event handling
 - <catch>** is used to respond to events (contains executable code)


```
<form id="missile_launch">
  <field name="user_id" type="digits">
    <prompt>What is your username?</prompt>
  </field>
  <field name="password">
    <prompt>Password?</prompt>
    <grammar>letlucce</grammar>
    <help>It is the name of a vegetable</help>
    <catch event="nomatch noinput" count="3">
      <prompt>Security violation!</prompt>
      <submit next="http://www.example.com/appehend_felon.xml" name="user_id">
    </catch>
  </field>
  <block>
    <goto next="#get_city"/>
  </block>
</form>
```

The third time one of these events occur, the code in the **catch** section is executed
 - Abbreviated notation

abbreviated notation of **<catch event="error">** (analogous for the other cases)

```
<error>An error occurred. Please call later<exit></error>
<help>Help is not available.</help>
<noinput>I didn't hear anything. Please try again.</noinput>
<nomatch>I heard something, but it is not a known city.</nomatch>
```

ESSLI 2004 1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Some other events
 - telephone.disconnect.hangup** → user hang up
 - telephone.disconnect.transfer** → call transfer to another phone number
 - error.badfetch** → e.g. a document is missing, badly written URI, error when accessing a resource in Internet, etc.
 - error.semantic** → e.g. zero division, variable not defined, etc.
 - Etc.

ESSLI 2004 1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Resource fetching
 - Access to web resources is determined by three attributes:
 - catching**
 - "safe"**: use the most recent version of the resource
 - "fast"**: use the cached version of the resource
 - fetchtimeout** → time interval to wait for the resource arrival before throwing an **error.badfetch** event
 - fetchhint** → specify when the interpreter context must obtain the resource from the server:
 - "prefetch"**: download resource when document is loaded
 - "safe"**: download resource when it is really necessary
 - "stream"**: used for big resources. Start processing the resource as it is arriving, no wait until complete arrival

```
<property name="caching" value="fast">
  <form id="test">
    <block>
      <!-- The following welcome message rarely changes, thus caching fast is ok -->
      <audio src="http://www.weather.example/vxml/welcome.wav"/>
      <!-- Other messages change frequently, thus caching safe is better -->
      <audio caching="safe" src="http://www.online_additions.example/forecast/ad17"/>
    </block>
  </form>
```

The elements of this document will use **caching="fast"** by default

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Message recording
 - <record>** is used to record user messages
 - Recordings can be played back or sent to a web server

```
<?xml version="1.0"?>
<vxml version="1.0">
  <form>
    <record name="greeting" beep="true" maxtime="10s" type="audio/wav" finalsilence="4000ms" dtmfterm="true">
      <prompt> Please record your message after the tone.</prompt>
      <noinput> I didn't here anything. Please try again.</noinput>
    </record>
    <field name="confirmation" type="boolean">
      <prompt> You recorded <value expr="greeting"/>.</prompt>
      <prompt> To keep it, say yes. To discard it, say no.</prompt>
      <filled>
        <if cond="confirmation">
          <submit next="save_greeting.pl" method="post" name="greeting"/>
        </if>
      </filled>
      <clear>
      </clear>
    </field>
  </form>
</vxml>
```

A beep is played to signal the recording start

The recording stops when the user presses a telephone key

The message is sent to a web server

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Call transfer
 - <transfer>** is used to transfer the call to another telephone number. Two transfer types:
 - bridging**: the original caller resumes his session with the interpreter (**bridge="true"**)
 - blind transfer**: no resumption is possible, as soon as the call connects, the event **telephone.disconnect.transfer** is generated (**bridge="false"**)

```
<form name="transfer">
  <var name="duration" expr="0"/>
  <block>
    <audio src="chopin12.wav"/>
  </block>
  <transfer name="myCall" dest="phone://18005551234" connecttimeout="30s" bridge="true">
    <filled>
      <assign name="duration" expr="myCall$.duration">
        <if cond="myCall == 'busy'">
          <prompt> Sorry, all our operators are busy at this time. Please try again later.</prompt>
        <else cond="myCall == 'noanswer'">
          <prompt> Sorry, the timetable for our operators is from 9 A.M. to 7 P.M., Monday through Saturday.</prompt>
        </if>
      </filled>
    </transfer>
  </block>
  <submit name="myCall duration" next="cgi-bin/report"/>
</form>
```

The call duration is assigned to **duration**

Transfer result (stored in field name):

 - "busy"** - endpoint refused the call.
 - "noanswer"** - no response from endpoint.
 - "network_busy"** - Intermediate network refused the call.
 - "near_end_disconnect"** - Call completed & finished by the caller.
 - "far_end_disconnect"** - Call completed & finished by endpoint.
 - "network_disconnect"** - Call completed & finished by the network.

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Execution of actions when fields are filled
 - <filled>** is used to decide what to do when the user input fills field(s). Two possibilities:
 - Child of <form> element**: action when one or more fields are filled
 - Child of <field> element**: action when one field is filled

```
<form id="get_source_and_destination_cities">
  <field name="sourceCity">
    <grammar src="http://www.grammars.example/voicexml/city.gram"/>
    <prompt>From which city are you planning to leave?</prompt>
  </field>
  <field name="destinationCity">
    <grammar src="http://www.grammars.example/voicexml/city.gram"/>
    <prompt>Which city are you planning to go to?</prompt>
  </field>
  <filled mode="all" name="sourceCity destinationCity">
    <if cond="sourceCity == destinationCity">
      <prompt>The source and destination cities cannot be the same.</prompt>
    </if>
  </filled>
</form>
```

Values of mode attribute:

 - "all"** (default) – the action is executed when all fields are filled and, at least, one has been filled by the last user input
 - "any"** – the action is executed when any field is filled by the last user input

<filled> child of form

```
<form id="get_city">
  <field name="city">
    <grammar src="http://www.grammars.example/city.gram"/>
    <prompt>City name?</prompt>
  </field>
  <filled>
    <if cond="city == 'Nancy'">
      <prompt> Nancy is a great city ....</prompt>
    </if>
  </filled>
</form>
```

<filled> child of field

1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- SUBMIT**
 - `<submit>` allows to send a list of variables to a document server via HTTP Get or Post

```
<submit next="mywwwserver.ugr.es" method="post" namelist="name model serial_number" fetchtimeout="100s" fetchaudio="audio/brahms2.wav"/>
```
- EXIT**
 - Time to wait for the server answer
 - Audio file to play while the answer arrives
 - `<exit/>` returns control to the interpreter context, which decides what to do next, e.g.:
 - Execute a higher level menu
 - End call
 - Transfer call to a human operator
 - Etc.
- DISCONNECT**
 - `<disconnect/>` makes the interpreter to disconnect the user call, throwing the event **telephone.disconnected.hangup**

289 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- SCRIPT**
 - `<script>` allows to specify a block of server-side code that performs a determined function (analogous to HTML `<script>`)
 - It can be inside a `<vxml>` element or in executable code

```
<?xml version="1.0"?>
<vxml version="1.0"?>
  <script><![CDATA[ function factorial(n) { return (n <= 1)? 1 : n * factorial(n-1); } ]]> </script>
  <form id="factorial">
    <field name="fact_number" type="number">
      <prompt>Say a number and I will say its factorial.</prompt>
      <filled>
        <prompt>
          Factorial of <value expr="fact_number"> is <value expr="factorial(fact_number)"/>
        </prompt>
      </filled>
    </field>
  </form>
</vxml>
```

Inside vxml

```
...
<block>
  <script>
    var f = new Date();
    hours = f.getHours();
    minutes = f.getMinutes();
    seconds = f.getSeconds();
  </script>
</block>
...
<prompt> Current time, <value expr="hours"/> hours, <value expr="minutes"/>
minutes and <value expr="seconds"/> seconds. </prompt>
```

Inside executable code

290 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://www.w3.org/TR/2000/NOTE-voicexml-20000505/ © W3C

VoiceXML

- Execution in multiple documents**

```
<?xml version="1.0"?>
<vxml version="1.0"?>
  <var name="farewell" expr="adios"/>
  <link next="operator_xfer.vxml"> <grammar> operator </grammar> </link>
</vxml>
```

Root document: **app-root.vxml**

```
<?xml version="1.0"?>
<vxml version="1.0" application="app-root.vxml">
  <form id="say_goodbye">
    <field name="answer" type="boolean">
      <prompt> Shall we say <value expr="application.farewell"/>? </prompt>
      <filled>
        <if cond="answer">
          <exit/>
        </if>
        <clear namelist="answer"/>
      </filled>
    </field>
  </form>
</vxml>
```

Leaf document specifies URI of root document **app-root.vxml**

Leaf document: **main.vxml**

291 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://www-106.ibm.com/developerworks/wireless/edu/wi-dw-wivoice2-1.html © IBM

Dynamic VoiceXML

- Fundamentals**
 - Use **JSP**, **ASP**, **Perl**, etc. to generate dynamically VoiceXML documents (e.g. containing information obtained from a database query)
 - A VoiceXML document can have ASP, JSP, etc. code embedded

```
<?xml version="1.0"?>
<vxml version="1.0"?>
  <form id="greeting">
    <block>
      <%
        out.println("Hello friend")
      %>
    </block>
  </form>
</vxml>
```

ASP

```
<?xml version="1.0"?>
<vxml version="1.0"?>
  <form id="greeting">
    <block>
      <%
        response.write("Hello friend")
      %>
    </block>
  </form>
</vxml>
```

JSP

292 R. López-Cózar, UGR

http://www-106.ibm.com/developerworks/wireless/edu/wi-dw-wivoice2-1.html © IBM

Dynamic VoiceXML: "news.asp" document

```
<?xml version="1.0"?>
<vxml version="1.0"?>
  <!-- give the news -->
  <form id="news_report">
    <block>
      <%
        'Set connection to database
        Set Con =Server.CreateObject("ADODB.Connection")
        Con.Open "PROVIDER=MICROSOFT.JET.OLEDB.4.0;DATA SOURCE=C:\news.mdb"

        'Execute query to get detail
        NewsTableSQL = "SELECT * FROM NewsTable"
        Set QueryResult = Con.Execute( NewsTableSQL )

        'User will hear the output of the query
        response.write(QueryResult("Detail"))

      %>
    </block>
  </form>
</vxml>
```

news.mdb: MS Access database with a field named **Detail** that stores news

Database in news.mdb

293 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://www-106.ibm.com/developerworks/wireless/edu/wi-dw-wivoice2-1.html © IBM

Dynamic VoiceXML: Perl script

```
<?xml version="1.0"?>
<vxml version="1.0"?>
  <form id="login">
    <field name="pin" type="digits">
      <prompt>Please say the 4 digits of your PIN.
    </prompt>
    <filled>
      <submit next="http://www.webreference.com/cgi-bin/perl/20/pin.pl/" />
    </filled>
    <noinput>No PIN has been entered.
    </noinput>
    <nomatch count="1">PIN code invalid.
    </nomatch>
    <nomatch count="3">Too many attempts. Please call later.
    </nomatch>
  </form>
</vxml>
```

VoiceXML document

```
#/usr/bin/perl -w

use strict;
use CGI;

my %users = (
  '1234' => 'Ramón',
  '1357' => 'Juan',
  '2468' => 'Pedro'
);

my $q = new CGI;
my $pin = $q->param("pin");

print "Content-Type: text/xml\n\n";

print "<<VXML:
  <?xml version='1.0'?>
  <vxml version='1.0'?>
    <form>
      <block>
        VXML

        print "Good morning Mister $users{$pin}.
        How may I help you?" || "$pin is a wrong PIN.";

        print "<<VXML:
          </block>
        </form>
      </vxml>
    </vxml>
```

Perl script

294 R. López-Cózar, UGR ESSLII 2004 1. Introduction

Outline

- Standards
 - VoiceXML
 - CCXML
 - CallXML

295

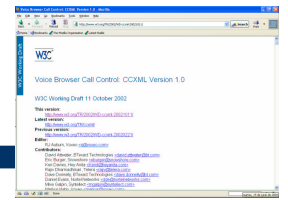
R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

CCXML

- Standard developed by the W3C to provide the functionality VoiceXML lacks in terms of call control
- Major features in comparison with VoiceXML
 - Synchronous event handling
 - Multi-conference handling
- Designed to be used with VoiceXML, not separately
- CCXML version: 1.0 (2002) <http://www.w3.org/TR/2002/WD-ccxml-20021011/> © W3C



296

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

CCXML: Fundamentals

- Variable definition: `<var name="..." expr="...">`
- Generation of **debug messages**: `<log expr="...">`
- `<exit/>` tag used to avoid "dead" applications

```
<?xml version="1.0" encoding="UTF-8"?>
<ccxml version="1.0">
  <var name="MyVariable" expr="This is a CCXML Variable"/>
  <log expr="Hello World. I just made a variable: ' + MyVariable"/>
  <exit/>
</ccxml>
```

<http://docs.voxeo.com/ccxml/1.0/home.htm>

297

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

CCXML: Basic event handling

- <eventhandler>**: code that executes when determined events are thrown
- Events are thrown **asynchronously** and are handled **sequentially** (following the order of handlers in the document)
- <transition>**: events can provoke transitions
- Some event types
 - connection.CONNECTION_ALERTING** (call detection)
 - connection.CONNECTION_CONNECTED** (call is being attended)
 - connection.CONNECTION_FAILED** (call could not be established)
 - call.CALL_INVALID** (call ended)

```
<?xml version="1.0" encoding="UTF-8"?>
<ccxml version="1.0">
  <eventhandler>
    <transition event="connection.CONNECTION_ALERTING" name="evt">
      <log expr="The called ID is ' + evt.calledid + '."/>
    </transition>
  </eventhandler>
</ccxml>
```

<http://docs.voxeo.com/ccxml/1.0/home.htm>

298

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

CCXML: Basic event handling

- if ... then ... else**
 - With calls
 - Accept
 - Reject
 - Disconnect

```
<?xml version="1.0" encoding="UTF-8"?>
<ccxml version="1.0">
  <eventhandler>
    <transition event="connection.CONNECTION_ALERTING" name="evt">
      <log expr="The called ID is ' + evt.calledid + '."/>
      <if cond="evt.calledid == '8315551234'">
        <reject/>
      </if>
      <accept/>
    </transition>
    <transition event="connection.CONNECTION_CONNECTED">
      <log expr="Call was answered."/>
      <disconnect/>
    </transition>
    <transition event="call.CALL_INVALID">
      <exit/>
    </transition>
  </eventhandler>
  <log expr="Application is starting."/>
</ccxml>
```

Rejected call

Call detection

Accept call

Call is being attended

Disconnect call

Finish session

This code always executes since is out of the event handler

<http://docs.voxeo.com/ccxml/1.0/home.htm>

299

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

CCXML: State machine

- CCXML works like a "state machine"
 - Events change the application "state"
 - Each **<eventhandler>** can have associated a state variable (statevariable)
 - Creation of the state machine**

```
<?xml version="1.0" encoding="UTF-8"?>
<ccxml version="1.0">
  <var name="app_state" expr="initial"/>
  <eventhandler statevariable="app_state">
    <eventhandler>
    </eventhandler>
  </ccxml>
```

The state variable is assigned the initial value

The state variable is associated to the event handler

<http://docs.voxeo.com/ccxml/1.0/home.htm>

300

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

http://docs.voxeo.com/ccxml/1.0/home.htm

CCXML: State machine

- Call answering using the state variable

```

<eventhandler statevariable="app_state">
  <transition state="initial" event="connection.CONNECTION_ALERTING" name="evt">
    <assign name="app_state" expr="accepting"/>
    <accept/>
    <var name="in_callid" expr="evt.callid"/>
  </transition>

  <transition state="accepting" event="connection.CONNECTION_CONNECTED">
    <log expr="app_state" expr="incall_0"/>
    <send event="continue" target="session.id" delay="3000"/>
  </transition>
</eventhandler>

```

Two conditions in the same <transition> tag

The value of the state variable is changed

The value of the state variable is changed

We throw an event called continue to our session with 3 sec of delay

301 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://docs.voxeo.com/ccxml/1.0/home.htm

CCXML: State machine

- Handling user-defined events

```

<eventhandler statevariable="app_state">
  <transition state="incall_0" event="user.continue">
    <assign name="app_state" expr="incall_1"/>
    <send event="continue" target="session.id" delay="1000"/>
  </transition>

  <transition state="incall_1" event="user.continue">
    <disconnect callid="in_callid"/>
    <assign name="app_state" expr="disconnecting"/>
  </transition>
</eventhandler>

```

It is necessary to prefix user to the event name to differentiate user-events from predefined-events

We throw a continue event with 1 sec of delay

Execution result: 4 sec after accepting the call it is disconnected

302 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://docs.voxeo.com/ccxml/1.0/home.htm

CCXML: State machine

- Handling several disconnection types

```

<eventhandler statevariable="app_state">
  <transition event="call.CALL_INVALID" cond="(incall_0 == app_state) || (incall_1 == app_state)">
    <log expr="premature disconnection detected in state [" + app_state + "]">
    <exit/>
  </transition>

  <transition event="call.CALL_INVALID">
    <log expr="not premature... wise, learned, mature disconnection"/>
    <exit/>
  </transition>

  <transition event="call."">
    <log expr="unhandled call. " event detected"/>
  </transition>
</eventhandler>

```

What should happen if this transition was the first in the event handler?

- The order of transitions in the event handler is important

303 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://docs.voxeo.com/ccxml/1.0/home.htm

CCXML: Error handling

- Case 1: invoking an action that cannot be carried out
- Case 2: referencing a non existing object

```

<eventhandler>
  <transition event="connection.CONNECTION_ALERTING">
    <accept/>
  </transition>

  <transition event="connection.CONNECTION_CONNECTED">
    <accept/>
    <log expr="second answer invoked, awaiting error event...">
  </transition>
</eventhandler>

```

This element provokes throwing the event: error.stateismatch since an accepted call cannot be accepted again

```

<eventhandler>
  <transition event="call.CALL_ACTIVE">
    <log expr="making a document mistake now ..."/>
    <log expr="and the temperature is " + below.zero"/>
  </transition>
</eventhandler>

```

below.zero has not been previously defined

304 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://www.w3.org/TR/2002/WD-ccxml-20021011/ © W3C

CCXML: Dialogue handling

- Combination of VoiceXML and CCXML (hybrid applications)
 - Passing variables between VoiceXML and CCXML
 - Tags <dialogstart src="..."> and <dialogterminate .../>

```

<transition state="init" event="connection.CONNECTION_CONNECTED" name="evt">
  <log expr="Houston, we have liftoff."/>
  <dialogstart src="gimme.vxml" name="theDialog"/>
  <assign name="state0" expr="dialogActive"/>
  <send event="timeout" target="session.id" delay="20000"/>
</transition>

<transition state="dialogActive" event="dialog.exit" name="evt">
  <log expr="Houston, the dialog returned [" + evt.input + "]">
  <exit/>
</transition>

```

Call to VoiceXML

Variable returned from VoiceXML document

```

<?xml version="1.0"?>
<vxml version="2.0">
  <form id="Form">
    <field name="input" type="digits">
      <prompt>
        Please say some numbers ...
      </prompt>
      <filled>
        <exit namelist="input"/>
      </filled>
    </field>
  </form>
</vxml>

```

Doc: gimme.vxml

305 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://www.w3.org/TR/2002/WD-ccxml-20021011/ © W3C

CCXML

```

<?xml version="1.0" encoding="UTF-8">
<ccxml version="1.0">
  <var name="state" expr="init"/>
  <eventhandler statevariable="state">
    <transition state="init" event="connection.CONNECTION_ALERTING">
      <accept/>
    </transition>

    <transition state="init" event="connection.CONNECTION_CONNECTED" name="evt">
      <log expr="Houston, we have liftoff."/>
      <dialogstart src="gimme.vxml" name="theDialog"/>
      <assign name="state" expr="dialogActive"/>
      <send event="timeout" target="session.id" delay="20000"/>
    </transition>

    <transition state="dialogActive" event="dialog.exit" name="evt">
      <log expr="Houston, the dialog returned [" + evt.input + "]">
      <exit/>
    </transition>

    <transition state="dialogActive" event="user.timeout">
      <log expr="Houston, a timeout occurred"/>
      <dialogterminate sessionid="theDialog"/>
    </transition>

    <transition event="call.CALL_INVALID" name="evt">
      <exit/>
    </transition>

    <transition event="error." name="evt">
      <log expr="Houston, we have a problem: (" + evt.error + ")">
      <exit/>
    </transition>
  </eventhandler>
</ccxml>

```

Dialogue call

The dialogue is terminated

306 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://docs.voxeo.com/ccxml/1.0/home.htm

CCXML: Call connection

- CCXML allows connecting an inbound call to an outbound call
 - `<createcall>`
 - `<join>`

The ID of the inbound call is saved in `callid_in`

```
<?xml version="1.0" encoding="UTF-8"?>
<ccxml version="1.0">
  <var name="state" expr="init"/>
  <eventhandler statevariable="state">
    <transition state="init" event="connection.CONNECTION_ALERTING">
      <accept/>
    </transition>
    <transition state="init" event="connection.CONNECTION_CONNECTED" name="evt">
      <var name="callid_in" expr="evt.callid"/>
    </transition>
    <transition event="error." name="evt">
      <log expr="an error has occurred (' + evt.error + ')" />
      <exit/>
    </transition>
  </eventhandler>
</ccxml>
```

307 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://docs.voxeo.com/ccxml/1.0/home.htm

CCXML: Call connection (cont.)

Dialogue call

The dialogue returns the "pin" variable when it finishes

Checking PIN of inbound call

Music while the outbound call is placed

Placing the outbound call

```
<transition state="init" event="connection.CONNECTION_CONNECTED" name="evt">
  <var name="callid_in" expr="evt.callid"/>
  <assign name="state" expr="enterpin"/>
  <dialogstart src="enterpin.vxml"/>
</transition>
<exit name="pin"/>
... in enterpin.vxml document
<transition state="enterpin" event="dialog.exit" name="evt">
  <log expr="PIN = [' + evt.pin + ']" />
  <if cond="1234 != evt.pin">
    <exit/>
  </if>
  <var name="pin" expr="evt.pin"/>
  <assign name="state" expr="calling"/>
  <dialogstart src="holdmusic.vxml" name="holdMusicDlg"/>
  <createcall dest="8315551234"/>
</transition>
```

308 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://docs.voxeo.com/ccxml/1.0/home.htm

CCXML: Call connection (cont.)

- The outbound call may **not** be attended:
 - The "background music" dialogue is finished
 - A dialogue is initiated to inform the outbound call cannot be placed
 - After the informative dialogue, the "disconnect" event is thrown to finish the session

```
<transition state="calling" event="connection.CONNECTION_FAILED">
  <assign name="state" expr="callfailed"/>
  <dialogterminate sessionid="holdMusicDlg"/>
</transition>
<transition state="callfailed" event="dialog.exit">
  <assign name="state" expr="playingCallFailed"/>
  <dialogstart src="callfailed.vxml"/>
</transition>
<transition state="playingCallFailed" event="dialog.exit">
  <disconnect/>
</transition>
```

309 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://docs.voxeo.com/ccxml/1.0/home.htm

CCXML: Call connection (cont.)

- ... but it **may** be attended:
 - Call the dialogue that receives the "pin" parameter
 - The background music is finished
 - Inbound call is connected to outbound call
 - Check the outbound call is different to the inbound call

```
<transition state="calling" event="connection.CONNECTION_CONNECTED" name="evt">
  <var name="callid_out" expr="evt.callid"/>
  <assign name="state" expr="callAccepted"/>
  <dialogstart src="newcall.php" name="pin"/>
</transition>
<transition state="callAccepted" event="dialog.exit" name="evt">
  <if cond="evt.callid != callid_in">
    <exit/>
  </if>
  <assign name="state" expr="beforeBridging"/>
  <dialogterminate sessionid="holdMusicDlg"/>
</transition>
<transition state="beforeBridging" event="dialog.exit">
  <assign name="state" expr="bridged"/>
  <join sessionid1="callid_in" sessionid2="callid_out"/>
</transition>
```

310 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://docs.voxeo.com/ccxml/1.0/home.htm

CCXML: Multi-conference

- Connection of several calls to each other (several users can talk simultaneously)
 - `<createconference>` ... `<destroyconference>`

Creation of conference

The initial call is accepted in the multi-conference

Placing calls to the other users of the multi-conference

Event thrown when the conference is created

```
<transition state="init" event="connection.CONNECTION_ALERTING" name="evt">
  <log expr="... assigning incoming callid to [line_0] ..."/>
  <var name="line_0" expr="evt.callid"/>
  <log expr="... creating conference ..."/>
  <createconference id="smallConf"/>
</transition>
<transition state="init" event="ccxml.conference.created">
  <assign name="state" expr="docalis"/>
  <log expr="... conference created and its ID is [' + smallConf + ']' ..."/>
  <log expr="... accepting incoming call [line_0] ..."/>
  <accept callid="line_0"/>
  <log expr="... making outbound calls ..."/>
  <createcall dest="4078351111" name="line_1"/>
  <createcall dest="4078351112" name="line_2"/>
  <createcall dest="4078351113" name="line_3"/>
</transition>
```

311 R. López-Cózar, UGR ESSLII 2004 1. Introduction

http://docs.voxeo.com/ccxml/1.0/home.htm

CCXML: Multi-conference (cont.)

The call is joined to the multi-conference

If the join is successful the `ccxml.conference.joined` event is thrown

If anyone hangs up or the line is interrupted, the events `ccxml.conference.unjoined` and `call.CALL_INVALID` are thrown

When the initial call ends, the multi-conference is terminated

```
<transition event="connection.CONNECTION_CONNECTED" name="evt">
  <log expr="... call leg ' + evt.callid + ' has been answered ..."/>
  <join sessionid1="evt.callid" sessionid2="smallConf"/>
</transition>
<transition event="ccxml.conference.joined" name="evt">
  <log expr="... call leg ' + evt.callid + ' has joined the conference ..."/>
</transition>
<transition event="ccxml.conference.unjoined" name="evt">
  <log expr="... call leg ' + evt.callid + ' has unjoined the conference ..."/>
</transition>
<transition event="call.CALL_INVALID" name="evt">
  <if cond="evt.callid == line_0">
    <destroyconference id="smallConf"/>
    <assign name="state" expr="conferencedestroyed"/>
  </if>
</transition>
```

312 R. López-Cózar, UGR ESSLII 2004 1. Introduction

Outline

- Standards
 - VoiceXML
 - CCXML
 - CallXML

313

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

CallXML

- Similar to CCXML, supported by Voxeo Community
 - Interaction using DTMF
 - Robust error control
 - Multi-conference
 - Easy to learn
 - Audio message recording
 - Text-To-Speech
 - Audio prompt generation
- Three types of element (tags)
 - Action
 - Events
 - Block

<http://docs.voxeo.com/calxml/2.0/home.htm>
© Voxeo Community

314

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Element types

- **Action elements:** used to carry out actions (e.g. playback previously recorded file, file recording, telephone hang up, etc.)
 - <playaudio ...>
 - <text ...>
 - <goto ...>
- **Event elements:** decide what to do when an event is thrown (e.g. the user presses a telephone key)
 - Elements may have **attributes** (e.g. **termdigits**)
- **Block elements:** group action and event elements
 - <block ...>

315

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Fundamentals

- **Issues to learn**
 - Basic structure of a CallXML application
 - Generation of a pre-recorded message
 - Generation of message using TTS
 - Event capture
 - Enabling action interruption
- **Tags**
 - <playaudio>
 - <ontermdigit>
 - <text>
 - <goto>
- **Attribute**
 - <termdigits>

<http://docs.voxeo.com/calxml/2.0/home.htm>
© Voxeo Community

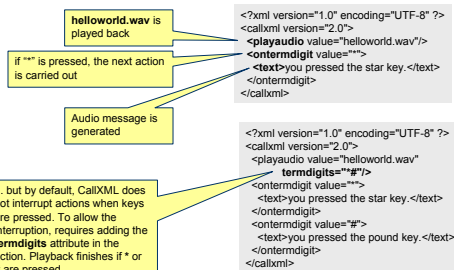
316

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

Fundamentals



317

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction

CallXML: Navigation inside a document

- **Issues to learn**
 - Grouping actions with <block>
 - Navigation using <goto>
- **Tags**
 - <block>
 - <goto>

<goto ..> tiene atributo que especifica donde continuar la ejecución

<block ..> tiene atributo que especifica el nombre del bloque

```
<?xml version="1.0" encoding="UTF-8" ?>
<calxml version="2.0">
  <block>
    <playaudio value="helloworld.wav"
      termdigits=""#"/>
    <goto value=""#one"/>
    <ontermdigit value=""#"/>
    <goto value=""#two"/>
    <ontermdigit value=""#"/>
  </block>
  <block label="one">
    <text>you pressed the star key.</text>
  </block>
  <block label="two">
    <text>you pressed the pound key.</text>
  </block>
</calxml>
```

318

R. López-Cózar, UGR

ESSLLI 2004

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

319 R. López-Cózar, UGR

http://docs.voxeo.com/calxml/2.0/home.htm
© Voxeo Community

CalXML: External navigation

- Issues to learn**
 - Link between documents using **<goto>**

<goto> specifies other document, that even can be in other server

<submit> specifies the variables that will be passed to the other document (* = all)

<method> specifies how the variables will be passed (**get** or **post**)

```
<?xml version="1.0" encoding="UTF-8" ?>
<calxml version="2.0">
  <block label="star">
    <text>you pressed the star key.</text>
  </block>
</calxml>
```

File helloworld-star.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<calxml version="2.0">
  <block>
    <playaudio value="helloworld.wav"
      termdigits="#" />
  </block>
  <ontermdigit value="#">
    <goto value="helloworld-star.xml"
      submit="#"
      method="get" />
  </ontermdigit>
  <ontermdigit value="#">
    <goto value="helloworld-pound.xml"
      submit="#"
      method="get" />
  </ontermdigit>
</calxml>
```

File helloworld-pound.xml

http://docs.voxeo.com/calxml/2.0/home.htm
© Voxeo Community

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

320 R. López-Cózar, UGR

http://docs.voxeo.com/calxml/2.0/home.htm
© Voxeo Community

CalXML: Answering machine

- Issues to learn**
 - Record a message using the action element **<recordaudio>**
 - Save an audio file as an e-mail attachment
- Tag attributes**
 - <format>**
 - <value>**
 - <termdigits>**
 - <beep>**
 - <maxtime>**
 - <maxsilence>**

Event throwers

ESSLI 2004

1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

321 R. López-Cózar, UGR

http://docs.voxeo.com/calxml/2.0/home.htm
© Voxeo Community

CalXML: Answering machine

Brief sound to indicate when the recording begins

Specify the format for the message to record

The file will be sent to this address

Maximum recording time before file is sent

Maximum silence time before file is sent

```
<?xml version="1.0" encoding="UTF-8" ?>
<calxml version="2.0">
  <block>
    <recordaudio format="audiowav"
      value="mailto:my-email@my-email-server.com"
      termdigits="#"
      maxtime="30s"
      maxsilence="10s"
      beep="true" />
  </block>
  <ontermdigit value="#">
    <text>Your message has been sent.</text>
  </ontermdigit>
  <onmaxtime>
    <text>You have reached maxtime. Message sent.</text>
  </onmaxtime>
  <onmaxsilence>
    <text>Maxsilence exceeded. Message sent.</text>
  </onmaxsilence>
</calxml>
```

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

322 R. López-Cózar, UGR

http://docs.voxeo.com/calxml/2.0/home.htm
© Voxeo Community

CalXML: caller ID and called ID

- Issues to learn**
 - Caller ID = number of telephone placing the call
 - Called ID = telephone number of the called application
 - Both are session, read-only variables of CalXML; they are automatically passed through the calls using **<goto>** and **<run>**
- Tag**
 - <playnumber>**

ESSLI 2004

1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

323 R. López-Cózar, UGR

http://docs.voxeo.com/calxml/2.0/home.htm
© Voxeo Community

CalXML: caller ID and called ID

<playnumber> is similar to **<text>** but is used for TTS of numbers instead of text; the **format** attribute allows to specify how to read the data (e.g. **format="digits"** or **format="number"**)

3 seconds pause

```
<?xml version="1.0" encoding="UTF-8" ?>
<calxml version="2.0">
  <block repeat="3">
    <text>Hello there. The caller id number is:</text>
    <playnumber value="$session.callerid;" format="digits">
      e.g. 1 2 3 4 5
    </playnumber>
    <text>The called id of this application is:</text>
    <playnumber value="$session.calledid;" format="number">
      e.g. 12.345
    </playnumber>
    <wait value="3s">
      3 seconds pause
    </wait>
  </block>
</calxml>
```

ESSLI 2004

1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

324 R. López-Cózar, UGR

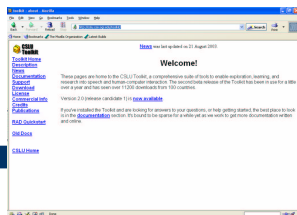
http://docs.voxeo.com/calxml/2.0/home.htm
© Voxeo Community

CalXML: Cache memory control

- Issues to learn**
 - Cache control from CalXML
 - Cache control from the server side
- Cache memory** (in our computer)
 - Accessed Internet resources are stored in our computer RAM memory
 - Advantage: unnecessary Internet accesses are avoided
 - Disadvantage: cache data may not be updated

ESSLI 2004

1. Introduction



CSLU Toolkit

- Free download for education, research and evaluation
- Features
 - Development of systems based on state-transition networks
 - Data capture and analysis
 - Dialogue corpus development
 - Research on ASR and NLP for several languages
 - Speaker recognition
 - Etc.

331 R. López-Cózar, UGR

ESSLI 2004

1. Introduction



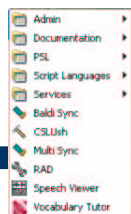
CSLU Toolkit

- ASR
 - Based on ANNs (Artificial Neural Networks) and HMMs (Hidden Markov Models)
 - Task-independent recogniser
 - Task-dependent recognisers (e.g. digits, letters, etc.)
 - Word-spotting
 - Barge-in
- User input
 - Speech, DTMF, clicks on screen
- System output
 - Speech (FESTIVAL), animated agents (*Gurmey, Lilly, Marge, Percy*, etc.), graphs, text

332 R. López-Cózar, UGR

ESSLI 2004

1. Introduction



CSLU Toolkit: Tools

- **Administration** (preferences, Web update, etc.)
- **Documentation** (RAD, vocabulary tutor examples)
- **Baldi Sync** → alignment of voice and lip movements of animated agent
- **CSLUsh** → command shell of the toolkit
- **Multi Sync** → similar to Baldi Sync but allows selecting several faces
- **RAD** → Rapid Application Developer

333 R. López-Cózar, UGR

ESSLI 2004

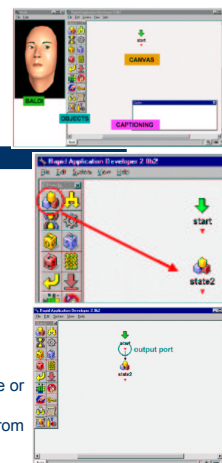
1. Introduction



CSLU Toolkit: RAD

- Components
 - Baldi (or other animate agent)
 - Objects
 - Canvas
 - Captioning
- System development
 - Object dragging to the canvas
 - Object connection
 - Each object has an input port and one or more output ports
 - Objects can be connected dragging from output ports to input ports

Video



334 R. López-Cózar, UGR

ESSLI 2004

1. Introduction



RAD (Rapid Application Developer)

- **Some objects**
 - Generic object (ASR or TTS)**
 - Communication media**
 - Subdialog**
 - List builder**
 - List recogniser**
 - Selection of user input**
 - Action**
 - Conditional**
 - Login**
 - Dialog end**
 - Digit recogniser**
 - Alphanumeric recogniser**
 - Flow transfer**

335 R. López-Cózar, UGR

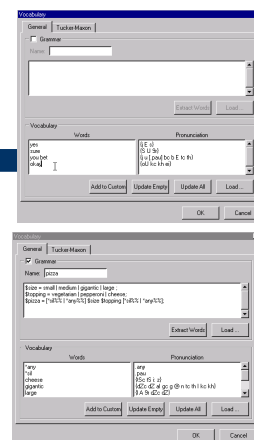
ESSLI 2004

1. Introduction



RAD

- **Generic object used for ASR**
 - Isolated words
 - Continuous speech (grammar)



336 R. López-Cózar, UGR

ESSLI 2004

1. Introduction



RAD

- Continuous speech recognition (needs a grammar)

Size = small | medium | gigantic;
 Stopping = vegetarian | pepperoni | cheese;
 Spizza = [*sil% | *any%] Size Stopping [*sil% | *any%]; Grammar: pizza.grammar

- Special characters in grammars

Characters	Use
[]	Optional parts of the grammar
< >	Parts of the grammar that can be repeated once or more times
{ }	Parts of the grammar that can be repeated zero or more times
	Boolean "or"
%	Following a word, indicates the word will not be in the recogniser output although it is recognised
%	Following a word, will substitute the next word into the recognition results if the first word is recognised
*sil	Used to recognise "silence"
*any	It represents "garbage", i.e. any sound that is not in the vocabulary, as for example out of vocabulary words (OOV), sneezes, background noise, etc.

337

R. López-Cózar, UGR

ESSLI 2004

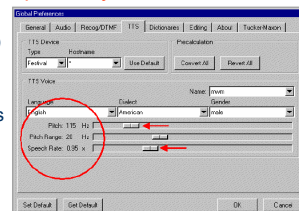
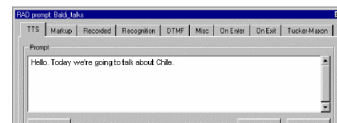
1. Introduction



RAD

- Generic object used for speech synthesis

- TTS
 - English (USA and UK)
 - Spanish
 - Portuguese
- Pre-recorded sentences



338

R. López-Cózar, UGR

ESSLI 2004

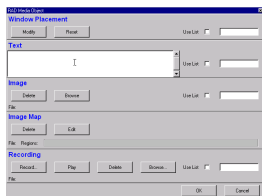
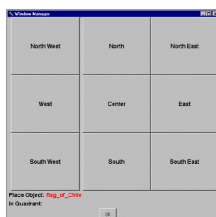
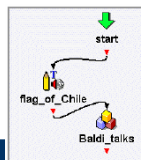
1. Introduction



RAD

- Communication media

- Show images on screen
- Generate sounds



339

R. López-Cózar, UGR

ESSLI 2004

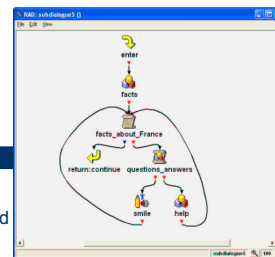
1. Introduction



RAD

- Subdialogue

- Used to avoid state overload in canvas



Subdialogue entrance



Subdialogue exit

340

R. López-Cózar, UGR

ESSLI 2004

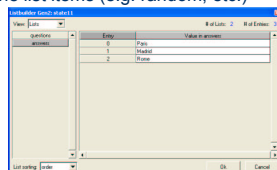
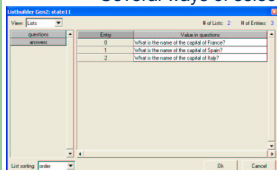
1. Introduction



RAD

- List builder

- Creation of associations between items of two lists A and B (e.g. A=questions, B=answers)
- Object output ports:
 - Blue: used to finish the interaction within the list
 - Red: used to continue the interaction within the list
- Several ways of selecting the list items (e.g. random, etc.)



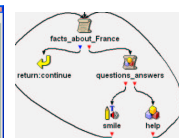
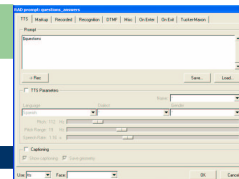
341



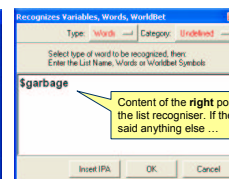
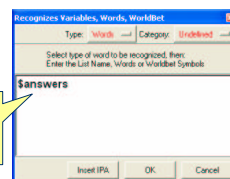
RAD

- List recogniser

- Receives as input the output of the list builder (item_A-item_B)
- Typically used for TTS one item and compare the recognition result with the other



Content of the left port of the list recogniser. If the user said the correct Capital ...



Content of the right port of the list recogniser. If the user said anything else ...

342

R. López-Cózar, UGR

ESSLI 2004

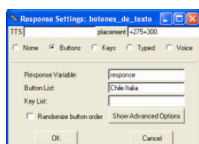
1. Introduction



RAD

User response selection

- **Button List** or **Key List** indicates the options for the user
- User selection can be made using buttons, text, etc.
- **Response Variable** stores the user selection



343

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



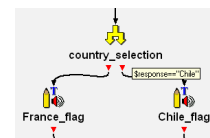
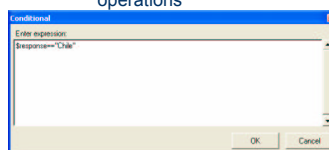
RAD

Action

- Allows using Tcl code (e.g. set background colour, remove objects from the screen, etc.) to enhance the applications

Conditional

- Decides the following state to visit as a result of Boolean operations



344

R. López-Cózar, UGR

ESSLLI 2004

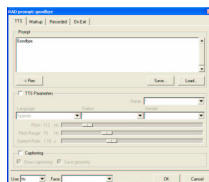
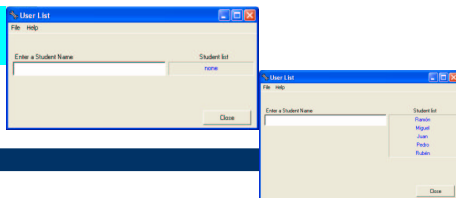
1. Introduction



RAD

Login

- Allows personalising the applications to the users, and also organise the capture of data provided by the user



345

R. López-Cózar, UGR

ESSLLI 2004

1. Introduction



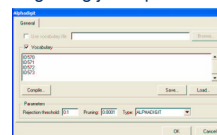
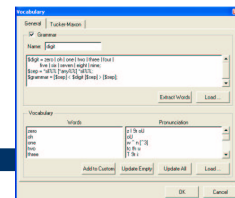
RAD

Digit recogniser

- Similar to the Generic Object for ASR but uses a specific grammar just for digit recognition

Alphanumeric recogniser

- Allows recognising just alphanumeric strings (letters & digits)



346

R. López-Cózar, UGR

ESSLLI 2004

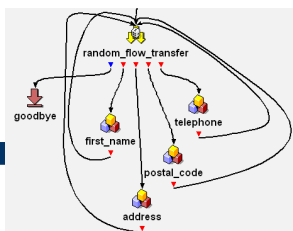
1. Introduction



RAD

Random flow transfer

- Random selection of all red output ports
- After each red port has been selected, the blue one is selected



347

R. López-Cózar, UGR

ESSLLI 2004

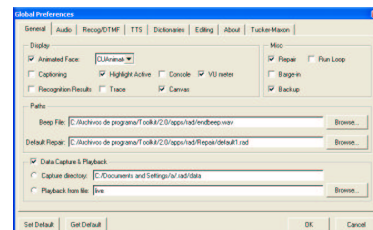
1. Introduction



RAD

Preferences

- Animated face
- Captioning
- Repair
- Barge-in
- Canvas
- Data capture



348

R. López-Cózar, UGR

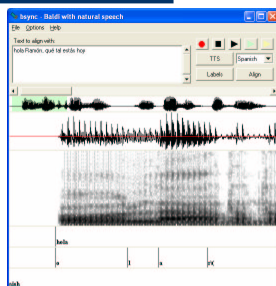
ESSLLI 2004

1. Introduction



CUAnimate and bsync

- CUAnimate
 - Change orientation of face and body
 - Change movement of eyebrows, head, etc.
 - Change facial expressions
- bsync
 - Synchronisation using TTS
 - Synchronisation using real-voice files
 - Human voice is used instead of TTS (files .sob)
 - Alignment of lip movements using the .sob file



349

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Outline

- CSLU Toolkit
- IBM Voice Toolkit for WebSphere Studio
- Voxeo Community
- Other tools

350

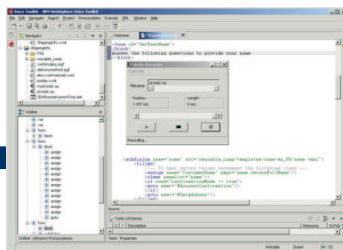
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

IBM Voice Toolkit

- Includes
 - Audio recorder
 - Application debugger
 - Tools for development and test of grammars
 - Pronunciation builder
 - Reusable components
 - Etc.



http://www-306.ibm.com/software/pervasive/voice_toolkit/
 © IBM

351

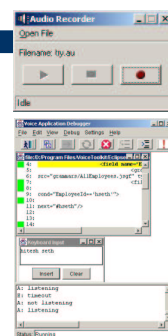
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

IBM Voice Toolkit

- Audio recorder
 - Pre-recorded prompts instead of TTS
 - .wav or .au files
- Debugger
 - Breakpoints, step-by-step execution, etc.



http://www-306.ibm.com/software/pervasive/voice_toolkit/
 © IBM

352

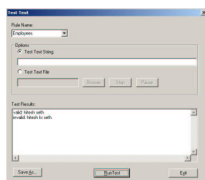
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

IBM Voice Toolkit

- Tools for grammar development and test
 - Several grammar types
 - JSGF (Java Speech Grammar Format)
 - BNF (Backus-Naur Form)
 - XML-based, etc.
 - Format interchange
 - Wizard to generate possible sentences
 - Checking whether a sentence (text/audio) is allowed by a grammar



353

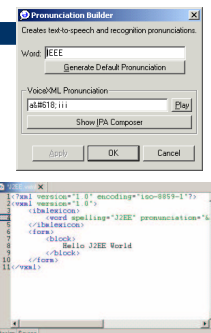
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

IBM Voice Toolkit

- Pronunciation builder
 - Allows creating pronunciations for unknown words using IPA (International Phonetic Alphabet)
 - E.g. change pronunciation of IEEE (i e e) for (i triple e)
 - The new pronunciation is automatically used
 - <ibmlexicon> tag



http://www-306.ibm.com/software/pervasive/voice_toolkit/
 © IBM

354

R. López-Cózar, UGR

ESSLI 2004

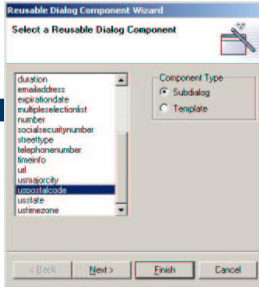
1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

IBM Voice Toolkit

http://www-306.ibm.com/software/pervasive/voice_toolkit/
© IBM

- Reusable components
 - Time / efforts saving
- Subdialogues
 - Confirmation
 - Telephone number
 - Currency, etc.
- Templates
 - Address (name + type + building number ...)
 - Credit card (number + expiration date), etc.
- Automatically generated messages
 - No input
 - Input example / spell
 - Discard data



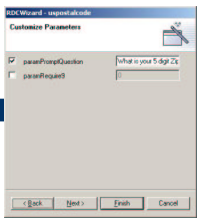
355 R. López-Cózar, UGR ESSLII 2004 1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

IBM Voice Toolkit

http://www-306.ibm.com/software/pervasive/voice_toolkit/
© IBM

- Reusable components
 - Wizard to configure parameters of the reusable components



```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="1.0">
  <script src="reusable_comp/subdialogs/ustpostalcode/en_US/ustpostalcode.es">
    var obj$ustpostalcode = new USPostalCode();
  </script>
  <form>
    <subdialog name="ustpostalcode"
      src="reusable_comp/subdialogs/ustpostalcode/en_US/ustpostalcode.vxml">
      <para name="paraSubdialog">
        <expr>obj$ustpostalcode</expr>
      </para>
      <para name="paraPromptQuestion">
        <expr>"What is your 5 digit Zip Code?"</expr>
      </para>
      <filled>
        <!-- To hear return values uncomment the following lines -->
        <!-- <value expr="ustpostalcode returnUSPostalCode"/> -->
        <!-- <value expr="ustpostalcode returnUSPostalCode"/> -->
      </filled>
    </subdialog>
  </form>
</vxml>
```

356 R. López-Cózar, UGR ESSLII 2004 1. Introduction

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

Outline

- CSLU Toolkit
- IBM Voice Toolkit for WebSphere Studio
- Voxeo Community
- Other tools

357 R. López-Cózar, UGR ESSLII 2004 1. Introduction

<http://community.voxeo.com>

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

Voxeo Community



- Web site that provides resources to build and test dialogue systems based on VoiceXML, CCXML and CallXML:
 - Tutorials
 - Pre-recorded audio files
 - VoiceXML grammars and samples
 - Graphic designer
 - Application debugger, etc.

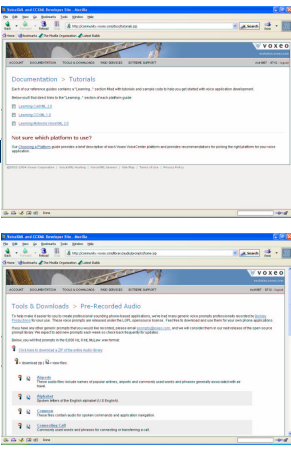
358 R. López-Cózar, UGR ESSLII 2004 1. Introduction

<http://community.voxeo.com>

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

Voxeo Community

- Tutorials
 - VoiceXML 2.0
 - CCXML 1.0
 - CallXML 2.0
- Pre-recorded audio files
 - Names of popular airlines, airports, etc.
 - English letters
 - Names of credit cards
 - Months of the year, days of the week
 - Etc.



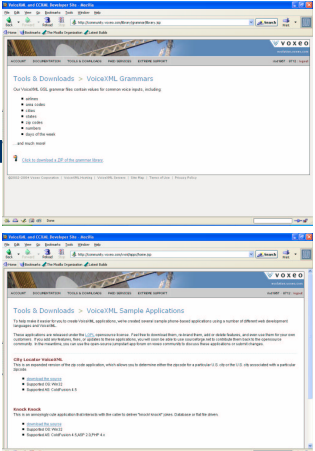
359 R. López-Cózar, UGR ESSLII 2004 1. Introduction

<http://community.voxeo.com>
© Voxeo Community

An Introduction to Spoken Dialogue Systems: Fundamentals, Standards and Tools

Voxeo Community

- VoiceXML grammars
 - Airlines
 - Area code
 - Cities
 - Zip codes
 - Numbers
 - Days of week
 - Etc.
- VoiceXML sample applications
 - City locator
 - VXML transfer
 - Etc.



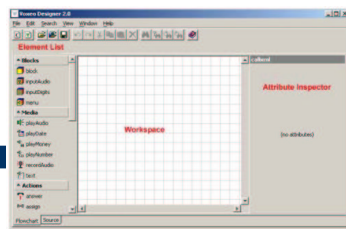
360 R. López-Cózar, UGR ESSLII 2004 1. Introduction



Voxeo Community

• Voxeo Designer

- Graphic tool to develop dialogue systems based on VoiceXML and CallXML
- Three parts:
 - Element list:** contains action, event and block elements
 - Workspace:** visioning graphically the applications
 - Attribute inspector:** see and edit attributes of the elements selected in the workspace



361

R. López-Cózar, UGR

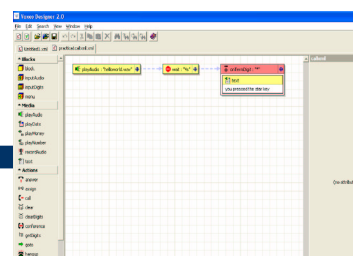
ESSLI 2004

1. Introduction



Voxeo Community

- Adding elements to the workspace
 - Click on the element
 - Some elements are inserted inside other elements



362

R. López-Cózar, UGR

ESSLI 2004

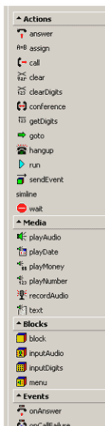
1. Introduction



Voxeo Community

• Element list

- Actions**
 - Main elements of the application
 - Used to respond to events and throw events
- Communication media** (only for CallXML)
 - Used to provide information and obtain user information (only audio and text)
- Blocks**
 - Several block types can be used to control the interaction flow
- Events** (only for CallXML)
 - Used to respond to events
 - Only applicable to some Communication Media and Actions



363

R. López-Cózar, UGR

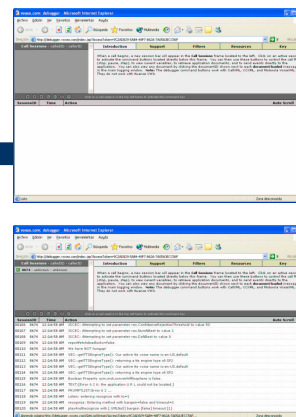
ESSLI 2004

1. Introduction



Voxeo Community

- Application debugger**
 - Allows tracing the execution of applications



364

R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Outline

- CSLU Toolkit
- IBM Voice Toolkit for WebSphere
- Voxeo Community
- Other tools



365

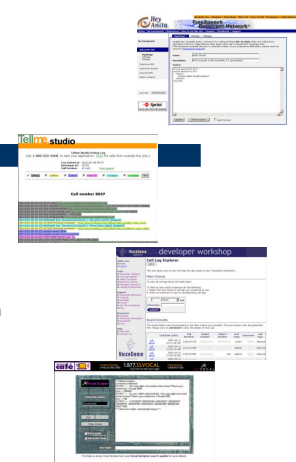
R. López-Cózar, UGR

ESSLI 2004

1. Introduction

Other tools

- HeyAnita RapidApp**
 - http://freespeech.heyanita.com
- Tellme Debug Log**
 - http://studio.tellme.com
- Voice Genie Call Log Explorer**
 - http://developer.voicegenie.com
- BeVocal Vocal Scripter**
 - http://cafe.bevocal.com



366

R. López-Cózar, UGR

ESSLI 2004

1. Introduction



Where to learn more

- **Apple** (<http://www.apple.com/macos/speech>)
- **AT&T** (<http://www.att.com/aspg>)
- **BBN** (<http://www.bbn.com/departments/dept-slp.html>)
- **IBM** (<http://ibm.com/software/speech>)
- **Lernout & Hauspie** (<http://www.lhs.com>)
- **Microsoft** (<http://www.microsoft.com/speech>)
- **Nuance** (<http://www.nuance.com>)
- **Philips** (<http://www.speech.be.philips.com>)
- **Speechworks** (<http://www.speechworks.com>)



An Introduction to Spoken Dialogue Systems. Fundamentals, Standards and Tools

16th ESSLLI
August 9-13, 2004
Nancy, France

Prof. Ramón López-Cózar Delgado
Dept. Languages and Computer Systems
Computer Science Faculty
University of Granada, Spain
E-mail: rlopezc@ugr.es
<http://www.ugr.es/~rlopezc>

Thanks for your attention !

