Εθνικόν και Καποδιστριακόν Πανεπιστήμιον Αθηνών
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Φωνητικές Διεπαφές Χρήστη-Τεχνολογίες Φωνής

Ενότητα 6.1: **Ανάπτυξη Διαλογικών Φωνητικών Εφαρμογών**

Γεώργιος Κουρουπέτρογλου
koupe@di.uoa.gr

Περιεχόμενα ενότητας

Μεθοδολογίες και εργαλεία ανάπτυξης φωνητικών διαλογικών εφαρμογών

Διαχείριση Διαλόγου (1 από 7)

- the central component of a dialogue system:
  - accepts spoken input from the user,
  - produces messages to be communicated to the user,
  - interacts with external knowledge sources
  - controls the dialogue flow.
- Various ways in which dialogue management functions can be performed:
  - From a fairly simple interaction (users retrieve information and perform routine transactions)
  - To complex tasks involving negotiation and problem solving (require a more advanced type of interaction).

Διαχείριση Διαλόγου (2 από 7)

- The complexity of the task will be reflected in the level of spoken language that is to be used:
  - from a minimal subset of natural language (consisting perhaps of only a small set of words such as the digits 0-9 and the words "yes" and "no“),
  - to large vocabulary systems supporting relatively free-form input.

Διαχείριση Διαλόγου (3 από 7)

- The input may be:
  - spoken
  - typed
  - combined with other input modes (e.g Dual Tone Multiple Frequency –DTMF- touch-tone),
- the output may be:
  - spoken
  - displayed as text on a screen,
  - accompanied by visual output in the form of tables or images.
- Multimodal I/O will be discussed in next chapters

Διαχείριση Διαλόγου (4 από 7)

- who controls the initiative in the dialogue:
  - system-directed (the system asks a sequence of questions to elicit the required parameters of the task from the user)
  - user-directed (the user controls the dialogue and asks the system questions in order to obtain information)
  - mixed-initiative (dialogue control is shared: The user can ask questions at any time, but the system can also take control to elicit required information or to clarify unclear information)

Διαχείριση Διαλόγου (5 από 7)

- control the dialogue flow involves:
  - determining what questions the system should ask,
  - in what order,
  - when.

Διαχείριση Διαλόγου (6 από 7)

- methods for controlling the dialogue flow:
  - From scripts (σενάριο) as a sequence of choices in a dialogue network (or dialogue grammar).

    Generally used for system-directed dialogues.
- To more open-ended systems: in which the choice of the next action is determined dynamically, based on the current state of the dialogue

  "agent-based" approach.

Διαχείριση Διαλόγου (7 από 7)

Grounding (θεμελίωση): the process whereby participants in dialogue try to ensure that what has been said in the dialogue has been mutually understood.

- Grounding involves:
  - verifying that utterances have been correctly understood
  - asking for clarification of utterances that are unclear or ambiguous.
- Ways to achieve grounding:
  - from simple confirmations
  - to more complex grounding procedures.

The current presentation focuses mainly on dialogues:

- scripted and
- system-directed

implementation issues in text weeks

Dialogue Initiative (1 από 2)

The process of dialogue between two humans can be viewed as an exchange of information in which the initiative may shift between the two participants.

For example:

- at one time A might be:
- asking all the questions,
- narrating some event, or
- commenting on some issue,
- while B might be in the role of listening and responding minimally.
- At a later point the roles might be reversed.

Dialogue Initiative (2 από 2)

In casual conversation: initiative may be fairly evenly distributed and each person may initiate new topics as they see fit.

- Generally speaking, in spoken dialogue systems the initiative does not change within a dialogue in the way it does in dialogues between humans and
- Tend to fall into one of the following categories:

1. System-directed (the system has the initiative).
2. User-directed (the user has the initiative).
3. Mixed-initiative (the initiative is shared).

System-directed Dialogue (1 από 3)

- The system asks one or more questions to elicit some information from the user so that it can submit an appropriate query to the external knowledge source.
- Most current spoken dialogue systems are system-directed.
- Typical example: a travel application: the system has to find out the user's requirements: values for parameters such as: destination, date and time of travel:

  System: Where are you traveling to?

  User: London.

  System: What day do you wish to travel?

  User: Friday.

  System: At what time?

  User: 9 a.m.

System-directed Dialogue (2 από 3)

- Generally constructed in such a way that the user's input is restricted to single words or phrases that provide responses to carefully designed system prompts.
- Major advantage: the required vocabulary and grammar for each response can be specified in advance.

Speech recognition and language understanding are constrained and are likely to be more accurate.

For simple dialogues of this type there are well-understood methods for design and implementation.

System-directed Dialogue (3 από 3)

Disadvantages:

- restrict the user's input to predetermined words and phrases,
- correction of misrecognised items difficult
- no opportunity for the user to take the initiative and ask questions or introduce new topics.

User-directed Dialogues (1 από 3)

- The user asks one or more questions that the system interprets and answers.

The user determines the questions to be asked and the role of the system is to attempt to answer the questions.

- The system may ask clarification questions, if some aspect of the user's question is unclear.
- Like a natural language interface to a database, in which the user queries the database using natural language input.

User-directed Dialogues (2 από 3)

example:

User: How many employees living in the London area earn more than 50,000€?

System: Fifty four

User: How many are female?

System: Eleven.

User: And managers?

System: Nine.

User-directed Dialogues (3 από 3)

Disadvantages:

- the system needs to have comprehensive speech and language processing capabilities in order to process and interpret a potentially wide range of input.
- the user needs to be aware of the words and phrases that the system can interpret.
- the system may also need the ability to keep track of the entities discussed in previous queries.

Mixed-initiative Dialogue (1 από 2)

- Either participant can take the initiative to ask questions, initiate topics, request clarifications, and so on.

example:

System: Where are you traveling to?

User: I want to fly to London on Friday.

System: At what time do you want to fly to London?

User: Are there any cheap flights?

Mixed-initiative Dialogue (2 από 2)

- The term "mixed-initiative dialogue" has been used in a more restricted sense, particularly within the VoiceXML framework, to describe dialogues in which the system has overall control of the dialogue but the user has some flexibility to volunteer more information than is requested in the system's questions.

example:

System: Where are you traveling to?

User: I want to fly to London on Friday "over-informative"

System: At what time do you want to fly to London?

User: 9 a.m.

Dialogue Control

- methods for representing and implementing the flow of the dialogue:
- **"finite state-based"** systems: the dialogue flow can be determined in advance and represented in terms of a dialogue network or grammar.
- **"agent-based"** systems: the dialogue flow is determined dynamically through a process in which the dialogue manager performs some reasoning in order to determine its next actions.
- **"frame-based"** systems: dialogue control is not completely predetermined, yet not as open-ended, as in agent-based systems.

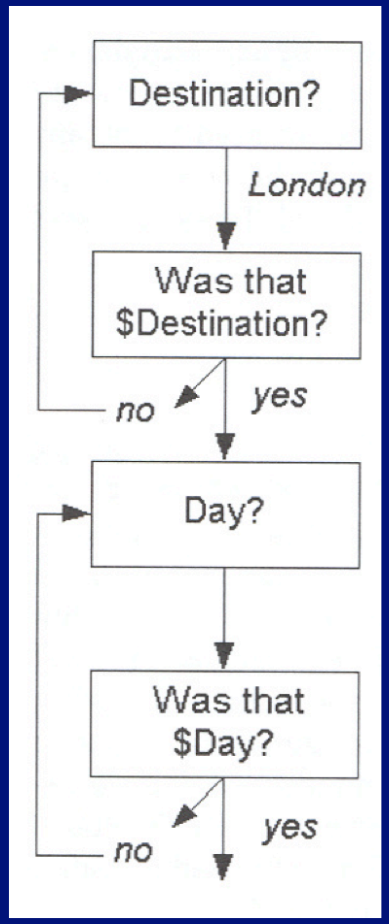
a way of implementing the limited mixed-initiative dialogues

Finite State-based Dialogue Control (1 από 6)

- The dialogue structure can be represented in the form of a state transition network (or graph) in which:
  - the nodes represent the system's questions
  - the transitions between the nodes represent all the possible paths through the network.
- The graph specifies all legal dialogues
  - each state represents a stage in the dialogue in which some information:
    - is elicited from the user
    - is confirmed with the user,
    - or some action is performed by the system.

Finite State-based Dialogue Control (2 από 6)

example of a partial network for a travel system



Finite State-based Dialogue Control (3 από 6)

- There are many ways in which dialogue networks can be extended to permit more complex dialogue sequencing.

For example,

- sub-dialogues may be used to represent sub asks in the dialogue, or tasks such as eliciting a date that occur frequently.

In the Center for Spoken Language Understanding (CSLU) toolkit there is a default repair sub-dialogue to handle situations:

  - where the system cannot detect any input from the user, or
  - when recognition accuracy is beneath an acceptable threshold.

Finite State-based Dialogue Control (4 από 6)

- Advantage: simplicity:

State transition networks are a natural way of modelling dialogues that are system-directed and that involve well-structured tasks and information that is to be exchanged in a predetermined sequence.

- Toolkits (e.g. CSLU toolkit) that allow the dialogue flow to be represented graphically, thus providing an intuitive interface for the developer.
- But, inflexible: as the dialogue paths are specified in advance, there is no way of managing deviations from these paths.

Finite State-based Dialogue Control (5 από 6)

- Problems arise if the user needs to correct an item or introduce some information that was not foreseen at the time the dialogue flow was designed.

Tasks involving some form of negotiation between system and user cannot be modelled, as the course of the dialogue cannot be determined in advance.

For example, planning a journey may require the discussion of constraints that are unknown by either the system or the user at the outset.

Finite State-based Dialogue Control (6 από 6)

- The most widely used form of dialogue control in commercial systems at the present time

Fewer technological demands are put on the system components, particularly the speech recogniser, as the user's responses can be constrained and therefore predicted.

Given this restricted input, there is no requirement for a sophisticated natural language understanding component.

The lack of flexibility and naturalness may be justified as a trade-off against the technological demands.

Frame-based Dialogue Control (1 από 9)

- Similar to finite state-based systems: suitable for form-filling tasks in which the system asks the user a series of questions to gather information, and then consults the external knowledge source,
- But in frame-based systems the questions do not have to be asked in a predetermined sequence.

Example 1:

System: Where are you traveling to?

User: I want to fly to London on Friday.

Assuming that there are three items to information to be elicited in this task - destination, date and departure time - two of these have been supplied by the user in one utterance.

Frame-based Dialogue Control (2 από 9)

In a finite state-based system only one item of information can be supplied at a time:

given that the system would have a recognition grammar to recognise destinations following the system's prompt:

- the speech recogniser might return "London" and ignore the remainder of the utterance. In this case the system would then proceed to the next question and ask for the travel date - which would obviously be confusing to the user at this point.
- Alternatively the recogniser might fail to return anything because the user's input did not sufficiently match anything in the recognition grammar. In this case the system would ask the user to repeat, leading potentially to the same problem all over again.

Frame-based Dialogue Control (3 από 9)

Requires three components:

- A frame (or template) that keeps track of the items of information that the system has to elicit from the user.
- A more extensive recognition grammar.
- A dialogue control algorithm that can determine the system's next actions based on the contents of the frame.
- The frame can be a simple data structure consisting of a series of slots to be filled with appropriate values.

example of a frame:

before the system's initial prompt:

Destination: unknown
Date: unknown
Departure time: unknown

Frame-based Dialogue Control (4 από 9)

The frame can be more complex:

e.g.: three possible values:

- "unknown"
- "filled"
- One indicating whether the value has been verified or not.

- Within the 3rd value there could be further distinctions,

e.g.:

- the level of confidence in the filled value,
- the way in which it was verified.

Frame-based Dialogue Control (5 από 9)

These additional elements would have implications for the dialogue control algorithm in terms of the sorts of questions that the system should ask next.

e.g.

- a value with low confidence would require a more explicit form of verification (ρητή επιβεβαίωση),
- a value with high confidence might not require any further verification and the item could be considered to be implicitly grounded.

Frame-based Dialogue Control (6 από 9)

Recognition grammar: more complex than the grammars used in finite state-based systems

e.g. . in response to the system's prompt in Example 1, the user's response might include various permutations of the three required elements, such as:

Destination

Destination + Date

Destination + Time

Destination + Date + Time

Destination + Time + Date

The grammar would require rules to match any of these combinations + verifications and corrections.

Frame-based Dialogue Control (7 από 9)

Advantages for the user (compared to finite-state):

- greater flexibility

It can be difficult to constrain users to the responses required by the system, even when the system prompts have been carefully designed to do just that

- the system can process the user's over-informative answers and corrections :
  - use of natural language
  - use of multiple slot-filling
- the transaction time for the dialogue can be reduced,
- more efficient and more natural dialogue flow.

Frame-based Dialogue Control (8 από 9)

Advantages for the developer (compared to finite-state):

- Implementing higher degree of flexibility in a finite state-based system becomes cumbersome, if not impossible: A large number of states and transitions are required to deal with all the different paths through the dialogue network.
- A frame-based system can be specified declaratively and the dialogue control algorithm will find a path dynamically through the implicit dialogue network.

Frame-based Dialogue Control (9 από 9)

Dialogues involving more complex interactions may require more sophisticated dialogue control due to factors such as the following:

- Different users may vary in the level of knowledge they bring to the task: a wide range of responses is required by the system.
- The state of the world may change dynamically during the course of the dialogue:

  it is not possible to specify all possible dialogue states in advance.

- The aim of the dialogue is not just to obtain sufficient information from the user to execute a database query or carry out some action

  instead, the dialogue involves the negotiation of some task to be achieved, involving planning and other types of collaborative interaction.

Agent-based Dialogue Control (1 από 4)

- Based on techniques from Artificial Intelligence (AI)
- Focus on the modelling of dialogue as collaboration between intelligent agents to solve some problem or task.
- Appropriate for more complex tasks, such as negotiation and problem solving
- There are many variants, depending on what particular aspects of intelligent behavior are included in the system.

Agent-based Dialogue Control (2 από 4)

Example

User: I'm looking for a job in the Calais area. Are there any servers?

System: No, there aren't any employment servers for Calais. However, there is an employment server for Pas-de-Calais and an employment server for Lille. Are you interested in one of these?

Agent-based Dialogue Control (3 από 4)

- Communication is viewed as interaction between two agents, each of which is capable of reasoning about its own actions and beliefs, and sometimes also about the actions and beliefs of the other agent.
- The dialogue model takes the preceding context into account with the result that the dialogue evolves dynamically as a sequence of related steps that build on each other.
- There are mechanisms for error detection and correction, and the system may use expectations to predict and interpret the user's next utterances.

Agent-based Dialogue Control (4 από 4)

- Tend to be mixed initiative, which means that the user can take control of the dialogue, introduce new topics, or make contributions that are not constrained by the previous system prompts.
- The form of the user's input cannot be determined in advance as consisting of a set number of words, phrases or concepts and, in the most complex systems, a sophisticated natural language understanding component is required to process the user's utterances.

Θεμελίωση (Grounding)

- The Dialogue Manager of a spoken dialogue system is confronted with a number of problems that do not normally arise in systems using a graphical user interface (GUI):

1. The system cannot guarantee that the representation of the user's input, that it receives from the speech recognition and language understanding components, is accurate.
2. There may be discrepancies between the information that the user requests and what is available in the external knowledge source.

Grounding: Processing the User's Input (1 από 17)

The input from the user may be unclear or incomplete for a variety of reasons:

1. The speech recogniser may have detected silence even though the user had spoken.

In this case no words will be returned.

2. Only a part of the user's utterance has been recognised and returned.

The beginning of the user's input may have been cut off because the user began to speak before the speech recognition engine had started - typically, where the user spoke before a system beep. Alternatively, the end of the user's input could have been lost because the engine stopped listening too early - usually because a pause was detected that was interpreted as the end of the input.

Grounding: Processing the User's Input (2 από 17)

3. All of the input has been captured but some or all of the words were incorrectly recognised.
4. Even though all the words were correctly recognised, the language understanding component was either unable to assign the correct meaning or there were a number of possible meanings due to ambiguity.

Grounding: Processing the User's Input (3 από 17)

Clarification of sub-dialogues:

- The simplest approach to handling ill-formed or incomplete input is to simply report the problem back to the user and to request a reformulation of the input.

It fails to distinguish the different ways in which the input may be ill-formed or incomplete, and it relies on the user knowing how to reformulate the input.

- A more useful approach: to address the problem explicitly (ρητά)

For example, if silence has been detected, the system can output a message such as: Sorry I did not hear anything. Would you please speak a little louder.

Grounding: Processing the User's Input (4 από 17)

- If the system is unable to assign a meaning to the words returned by the recogniser, a message could be output such as:

Sorry I didn't quite understand that. Would you please rephrase?

- Facilities exist within most toolkits for handling such commonly occurring problems:

e.g. in VoiceXML:

- detection of silence is handled as a (no input) event for which a default message is output.
- In the case where the input cannot be interpreted, a (no match) event is thrown with an appropriate message.
- In both cases the developer can change the default message and even devise a sequence of messages that provide increasing help in case the event is repeated without being resolved.

Grounding: Processing the User's Input (5 από 17)

Verification

- Even where no error requiring clarification has been detected, it is important that the system should verify that it has correctly understood the user.
- Verification is common in human-human dialogues where there is a potential for error, *for example, human agents in call centres will verify important information such as credit card numbers, addresses and dates before proceeding with a transaction.*
- Establishing common ground is even more necessary in spoken dialogues with computers given the possibility of recognition and understanding errors.
- There are two commonly used methods for verification:
  - explicit verification and
  - implicit verification.

Grounding: Processing the User's Input (6 από 17)

Explicit Verification (ρητή επιβεβαίωση)

- Takes the form of a question that asks explicitly for confirmation of the input.
- This may be accompanied by a request to answer with "yes" or "no”:

System: So you want to fly to London? (Please answer yes or no.)

- Disadvantage: it can become tedious for the user to explicitly verify each value and the number of turns required to complete the dialogue is increased.

Grounding: Processing the User's Input (7 από 17)

- Verification can be deferred until all the values have been collected.

However, this may lead to difficulties when one or more values are incorrect.

example:

System: Where are you travelling to?

User: London.

System: What day?

User: Friday.

System: So you want to travel to London on Sunday?

Grounding: Processing the User's Input (8 από 17)

Explicit Verification (ρητή επιβεβαίωση)

How is the user to correct "Sunday" to "Friday"?

- obvious way: say something like "No, not Sunday, I said Friday".

However, such a response is likely to be beyond the system's capabilities.

(Recall that any user input must conform to the speech recognition grammar.)

Grounding: Processing the User's Input (9 από 17)

Explicitly ask the user which values are incorrect.

example:

System: Where are you travelling to?

User: London.

System: What day?

User: Friday.

System: So you want to travel to London on Sunday?

User: No.

System: Which value do you wish to correct: the destination or the day?

User: Day.

System: What day?

Grounding: Processing the User's Input (10 από 17)

- For this strategy to work, the user must know that only a "yes" or "no" answer is permitted in response to a verification question.
- Following a "no" answer the system can then ask for the value to be corrected.
- This strategy is fairly robust;

However it becomes more complex if more than one value is to be corrected. In this case, can the user say all the values that are to be corrected, which would require the system to have appropriate grammars to represent all possible permutations?

Grounding: Processing the User's Input (11 από 17)

Implicit Verification (εν δυνάμει ή έμμεση επιβεβαίωση)

- The system embeds in its next question a repetition of its understanding of what the user said in the response to the previous question.
- The user can still correct the repeated value, but if the user answers the next question without correcting the value, then that value has been implicitly confirmed.

example:

User: I want to fly from Belfast to London.

System: At what time do you want to fly from Belfast to London?

User: Seven in the evening.

Grounding: Processing the User's Input (12 από 17)

In utterance 2 the system includes the departure and destination cities that it has recognised in its question about the departure time.

In utterance 3 the user gives the departure time but does not mention the departure and destination cities, so that these values are taken to be confirmed implicitly, as otherwise the user should have corrected them.

- Implicit verification can decrease the transaction time for a dialogue, as fewer questions need to be asked.
- The resulting dialogue also has a more natural flow.
- However, requests for verification can give rise to a wider range of possible responses, which may subsequently put greater demands on the recognition and understanding processes.

Grounding: Processing the User's Input (13 από 17)

Implicit Verification

Example:

I want to fly from Belfast to Luton.

but the system had misrecognised "Luton" as "London" and produced the question at utterance 2 ("At what time do you want to fly from Belfast to London?").

There are a number of ways that the user could correct "London" to "Luton", including the following:

No, not London, Luton.

Luton, not London.

I said Luton.

Belfast to Luton.

Belfast Luton.

Grounding: Processing the User's Input (14 από 17)

Implicit Verification

- A further problem: verification requests involve combining the values elicited from the user (either correctly or incorrectly) with the next value to be queried (e.g., in this case, the departure time).
- With even a relatively small number of values the combinatorial possibilities expand rapidly, giving rise to a large number of possible verification questions.
- The implicit verification strategy relies on the convention that the user will correct the system if some value in the implicit verification question is incorrect.

However, users may not realise that a correction is required, or they may think that they can correct the value later.

Grounding: Processing the User's Input (15 από 17)

- A further possibility: the user was more intent on answering the next question and did not attend to the items returned by the system in the implicit verification question, for example, that the system had incorrectly recognised "Luton" as "London".

Grounding: Processing the User's Input (16 από 17)

- Verification is one of the most challenging issues in spoken dialogue systems.
- A failure to verify correctly may lead to miscommunication, while an explicit verification strategy may result in an unreasonably lengthy dialogue, which has an adverse effect on user satisfaction.
- Considerable research is being directed toward the development of effective and efficient verification strategies that allow the system to degrade gracefully when faced with recognition or interpretation problems.

For example, when grounding a value the system could first try implicit verification, then move to explicit verification, and finally to a more constrained mode such as an explicit "yes" or "no" or even spelling of the problematic value.

Grounding: Processing the User's Input (17 από 17)

example:

User: I want to fly from Belfast to Luton.

System: At what time do you want to fly from Belfast to London?

User: No, not London, Luton.

System: Do you want to fly to London? Please answer yes or no.

User: No.

System: Please spell your destination.

User: L-U-T-O-N.

System: So you want to fly to Luton?

User: Yes.

System: At what time do you want to fly from Belfast to Luton?

Accessing Information from an External Knowledge Source (1 από 8)

- How to connect a spoken dialogue system to an external knowledge source (such as a database)

using the CSLU toolkit and VoiceXML in next Chapters

- Problems arise when there are discrepancies between the information that the user requests and what is available in the external knowledge source:
  - The vocabulary of the dialogue does not map directly on to the vocabulary of the application
  - The data that is retrieved is ambiguous (αμφιλεγόμενα) or indeterminate (ασαφή).

Accessing Information from an External Knowledge Source (2 από 8)

Problems with Vocabulary

- If the terms extracted from the user's utterances do not correspond to the vocabulary of the knowledge source, then will be returned: a) no information, b) partial information, or c) incorrect information.
- Problems due to: a) misspellings, b) different spellings of items (such as names, abbreviations) or c) different ways of referring to the same item.
- Some may be handled in an ad hoc way *(by providing alternative representations of the items).*

Accessing Information from an External Knowledge Source (3 από 8)

- A more general approach: use an **Information Manager** (that deals with complex information processing involving the application knowledge source) includes a data model which contains a number of vocabulary models that are each associated with one vocabulary within the application, (so that a distinction can be made between how items are represented in the database and how they may be referenced within a spoken dialogue).

Accessing Information from an External Knowledge Source (4 από 8)

Ambiguous (αμφιλεγόμενα) **and Indeterminate** (ασαφή) **Data**

- Various methods have been developed to handle these problems
- Usually involve some mechanism that has been specifically (ad hoc) devised to handle problems that have been predicted in advance for a specific domain.
- mechanism for handling under-specified or ambiguous values.

e.g disambiguate train stations with the same name ("Frankfurt am Main" and "Frankfurt an der Oder“) , which might both be referred to in a dialogue using the shorter name "Frankfurt".

Accessing Information from an External Knowledge Source (5 από 8)

- mechanisms for combining values,

e.g if a user called in the afternoon with the utterance "today at 8", the two values were combined into the single value 20:00 hours given that the value 08:00 hours was no longer valid.

- Database access may be unsuccessful because a value did not find an exact match in the database.

For example, a query concerning a flight to London at 8 p.m. might be unsuccessful, although there may be flights to London just before or just after this time.

- One approach is to relax (εκτονώνω) some of the parameters of the query until a suitable result can be found in the database.

Accessing Information from an External Knowledge Source (6 από 8)

- In other cases, it may not be clear which item should be relaxed.

example,

Is there a train from Birmingham to London arriving around 10 in the morning?

Relaxing the time parameter might return trains arriving at 9 a.m. and 11 a.m.

Relaxing the transport parameter might return a flight or a bus that arrives around 10 a.m.

- In other cases, the user might even be happy with a change in the departure or destination cities, as would be the case with alternative airports in the same city.
- Making judgments about which parameters to relax requires detailed analysis of the domain, so that there may not be any general solutions to this problem.

Accessing Information from an External Knowledge Source (7 από 8)

- There are problems concerning how the output is to be presented to the user.

If a number of database solutions have been found, it is necessary to decide how many to present.

- While a large number of records can be presented to the user in tabular form in a graphical user interface (GUI), people are generally unwilling to listen to long lists of items and are likely to forget the details by the time the end of the list has been reached.
- One approach is to divide the records to be listed into small groups of less than five and to read these out in turn.

  However, this is still unsatisfactory in the case of large numbers of records,

  for example, 25 flights matching the user's query.

Accessing Information from an External Knowledge Source (8 από 8)

- An alternative method is to constrain the query so that fewer records are returned.

For example, if the user requested flights from London to Boston, the system could ask for more specific parameters to constrain the query, such as flights with particular airlines, specific departure and arrival times, and so on.

- Again most of these methods are likely to be fairly domain--specific and it is difficult to devise more generally applicable solutions.

Knowledge Sources for Dialogue Management (1 από 4)

The dialogue manager may draw on a number of knowledge sources, which are sometimes referred to collectively as the **dialogue model**.

A dialogue model might include the following 6 types of knowledge

1. **A dialogue history.** A record of the dialogue so far in terms of the propositions that have been discussed and the entities that have been mentioned. This representation provides a basis for conceptual coherence and for the resolution of anaphora and ellipsis.

Knowledge Sources for Dialogue Management (2 από 4)

2. **A task record.** A representation of the information to be gathered in the dialogue. This record, (often referred to as a form, template or status graph), is used to determine what information has not yet been acquired. This record can also be used as a task memory for cases where a user wishes to change the values of some parameters, such as an earlier departure time, but does not need to repeat the whole dialogue to provide the other values that remain unchanged.
3. **A world knowledge model.** This model contains general background information that supports any commonsense reasoning required by the system.

   for example, that Christmas day is December 25

Knowledge Sources for Dialogue Management (3 από 4)

4. **A domain model.** A model with specific information about the domain in question,

   for example, flight information.

5. **A generic model of conversational competence.** This includes knowledge of the principles of conversational turn-taking and discourse obligations,

   for example, that an appropriate response to a request for information is to supply the information or provide a reason for not supplying it.

Knowledge Sources for Dialogue Management (4 από 4)

4. **A domain model.** A model with specific information about the domain in question,

   for example, flight information.

5. **A generic model of conversational competence.** This includes knowledge of the principles of conversational turn-taking and discourse obligations,

   for example, that an appropriate response to a request for information is to supply the information or provide a reason for not supplying it.

6. **A user model.** This model may contain:

- relatively stable information about the user that may be relevant to the dialogue - such as: the user's age, gender and preferences –
- information that changes over the course of the dialogue, such as the user's goals, beliefs and intentions.

Dialogue control strategies

Feature/Dialogue Control Strategy	State-based	Frame-based	Agent-based
Input	Single words or phrases.	Natural language with concept spotting.	Unrestricted natural language.
Verification	Explicit confirmation - either of each input or at end of transaction.	Explicit and implicit confirmation.	Grounding
Dialogue model	Information state represented implicitly in dialogue states. Dialogue control represented explicitly with control diagram.	Explicit representation of information states. Dialogue control represented with control algorithm.	Dialogue history Context Model of system's intentions, goals, beliefs.
User model	Simple model of user characteristics or preferences.	Simple model of user characteristics or preferences.	Model of user's intentions, goals, beliefs.

Issues for Developers (1 από 3)

Dialogue management is the component where most development work is done.

The developer has to make a number of choices regarding dialogue management and has to design and implement the dialogue manager according to these choices.

The following are the tasks involved:

- **Choice of dialogue initiative** (system-directed, user-directed or mixed-initiative) determined:
  - partly on the basis of an analysis of the type of interaction that is required to accomplish the selected task,
  - partly in the light of what resources are available.

    e.g: user-directed and mixed-initiative will require robust and extensive speech recognition and natural language understanding components.

Issues for Developers (2 από 3)

- **Choice of dialogue control strategy**: (finite state-based, frame-based or agent-based) determined:
  - Partly by the choice of dialogue initiative,
  - Partly on the basis of available resources.
- **Design of system prompts**: can constrain the range of possible user responses and so assist the speech recognition and natural language understanding components, as well as minimising the occurrence of errors.
- **Choice of verification strategy**: (required to ensure grounding of information between the system and the user) often involves a compromise between ensuring accuracy at all costs and avoiding very lengthy transactions.
- **Choice of error recovery strategy**: (involves detecting errors in the first place) methods can range from domain-specific checks to more general methods of error detection.

Issues for Developers (3 από 3)

- **Designing and implementing links to external knowledge sources.**
  - There can be problems in mapping between the vocabulary entries of a dialogue system and the vocabulary of an external knowledge source.
  - The data that is retrieved may often be ambiguous or indeterminate.
  - While it is often maintained that a spoken dialogue system can be easily linked to existing back-end systems, such as application web servers, there are often issues that are unique to the mode of spoken interaction with these back-end systems that require careful consideration.

Άσκηση 6.1

Επιλέξτε ένα από τα διαλογικά συστήματα ομιλίας που έχετε χρησιμοποιήσει στις προηγούμενες ασκήσεις το οποίο χρησιμοποιεί διάλογο ελεγχόμενο από το σύστημα (system-controlled). Σκεφτείτε πως το σύστημα θα δούλευε σε μορφή mixed-initiative. Κατασκευάστε 2-3 παραδείγματα του διαλόγου που περιγράφει το σύστημα να λειτουργεί σε μορφή mixed-initiative.

Τέλος Ενότητας

Ανάπτυξη Διαλογικών Φωνητικών Εφαρμογών

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.

Σημείωμα Αναφοράς

Σημείωμα Αδειοδότησης

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

Σημείωμα Χρήσης Έργων Τρίτων (1 από 2)

Οι φωτογραφίες που περιέχονται στην παρουσίαση αποτελούν πνευματική ιδιοκτησία τρίτων. Απαγορεύεται η αναπαραγωγή, αναδημοσίευση και διάθεσή τους στο κοινό με οποιονδήποτε τρόπο χωρίς τη λήψη άδειας από τους δικαιούχους.

Σημείωμα Χρήσης Έργων Τρίτων (2 από 2)