Language Technology II
Dialogue Management

Ivana Kruijff-Korbayová
korbay@coli.uni-sb.de
www.coli.uni-sb.de/~korbay/ → Teaching

Outline

- Tasks of dialogue management
- Dialogue-flow control
- Finite State-Based DM
- Frame-Based DM
- ISU-Based DM
- Grounding and Verification
- Inititative and Cooperation
- Current challenges

Tasks of Dialogue Management

- Dialogue flow control
- Dialogue modeling
  - Dialogue context
  - Dialogue moves
- Error handling
- Initiative and cooperation
- Adaptivity
- ...

Dialogue Flow Control

when to say something,
when to stop
⇒ turn taking

Turn Taking

- Dialogue participants take turns (like in a game): A, B, A, B
- Dialogue turn = a continuous "contribution" to the dialogue from one speaker
- Though it is generally not obvious when a turn in natural dialog is finished, turn-taking appears fluid in normal conversation:
  - Minimal pauses between speakers (few hundred ms)
  - Less than 5% speech overlap
- How does it work?

Turn Taking Rules

- Conversational analysis
- When does turn-taking occur:
  - Transition-relevance places (TRPs) ---points where the dialog/utterance structure allows speaker shift to occur (typically at utterance boundaries, but also smaller units, e.g., phrases)
  - TRP signals include syntax (phrase boundaries), intonation, gaze, gesture; Also cultural conventions apply
- Who speaks next
  - At each TRP (current speaker A):
    - If A selected B as next speaker, B should speak
    - If A did not select the next speaker, then anyone may take a turn
    - If no-one else takes a turn, then A may (continue)
  - To get a turn if not selected, a speaker must "jump in" at a TRP
- When do we get pauses or lapses? When do we get overlaps?

Turn Taking in Human-Computer Dialogue

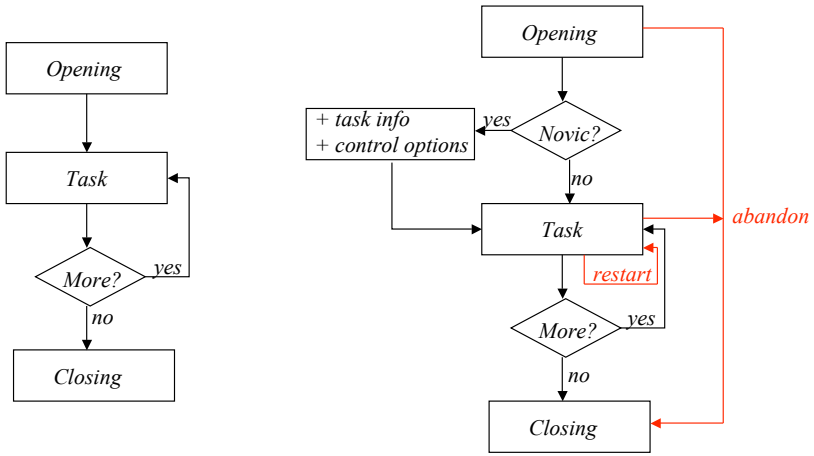
- Rigid: strict separation of system/user turns
  - How to determine the end of user's turn? (Is s/he finished?)
  - How long to wait for user's turn? (Is the user still engaged? Did s/he hear?)
  - Avoid user's speaking too early by explicit turn-taking signals
- Flexible, with barge-in:
  - User barge-in: system stops speaking when it detects input
    - Open-mic: system listening all-the-time
      - Problems: talk vs. noise; system's own talk is also "noise"
    - Push-to-talk: user pushes a button to open the mic (take a turn)

    Problem: What has actually been conveyed to the user? What is the resulting common ground between the system and the user? E.g., list with several options, complex info --> reference resolution
  - System barge-in: When appropriate at all? When is a TRP?

Dialogue Modeling

Where we are
&
What to say next

Global Dialogue Structure



Local Dialogue Structure

- Adjacency pairs or dialogue games:
  - Turns produced by different speakers
  - Ordered: First^Second (initiation - response)
  - Typed: particular First requires a particular Second
    - Greet-greet, ask-answer, request-grant, offer-accept, compliment-downplay, etc. ⇒ preferences, expectations
- Insertion sequences: APs can be embedded
  - E.g., "sub-dialogue", misapprehension-correction, clarification

Local Structure: Insertions

- "Sub-dialogue":
  A: Where are you going?
  B: Why do you want to know?
  A: I thought I'd come with you.
  B: I'm going to the supermarket.
- Clarification:
  A: I'd like three sausages.
  B: Which ones? Merquez or Lyoner?
  A: Merquez.
  B: Here you go.
- Misapprehension-Correction:
  A: When is the next train from SB to Hamburg?
  B: The next train to Homburg is at 1 p.m.
  A: Hamburg, not Homburg.
  B: Ah, Hamburg?
  A: Yes.
  B: The next connection to Hamburg Hauptbahnhof is at 3 p.m.

Methods of DM

- Script-based: Finite automata
  - Sequence of pre-defined steps (dialogue script)
- Frame-based (also: form-filling)
  - Set of slots to be filled (task template) and corresponding prompts
- Information-State Update
  - Declarative rules for updating dialogue context

Script-Based DM (Finite Automata)

DM Based on Finite Automata

- Automaton describes all possible dialogues
- Set of states and transitions
  - State determines system utterance
  - User utterance determines transition to next state (deterministic)
- No recursion! (= no nested subdialogues)
- Fixed dialogue script
- System-driven interaction

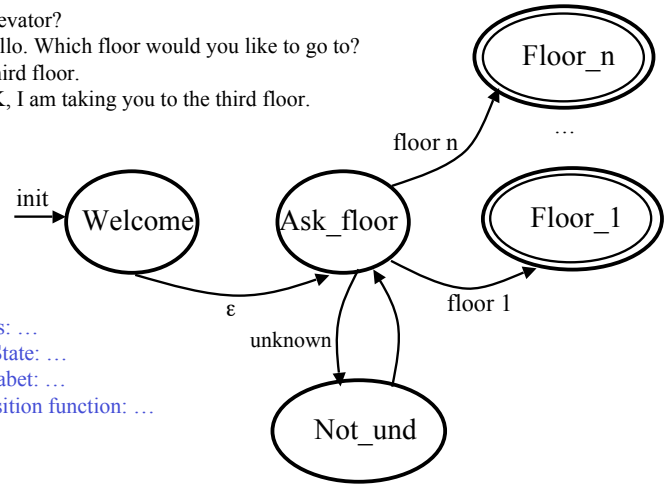
Finite Automaton (Finite State Machine)

- <States, Init-State, Alphabet, Transition-fction>
- Variants: machines having
  - actions associated with states (Moore machine)
  - actions associated with transitions (Mealy machine)
  - multiple start states
  - transitions conditioned on no input symbol (a null)
  - more than one transition for a given symbol and state (nondeterministic finite state machine)
  - states designated as accepting states (recognizer)
  - etc.

See, e.g., NIST http://www.nist.gov/dads/HTML/finiteStateMachine.html

FSM-Based Models

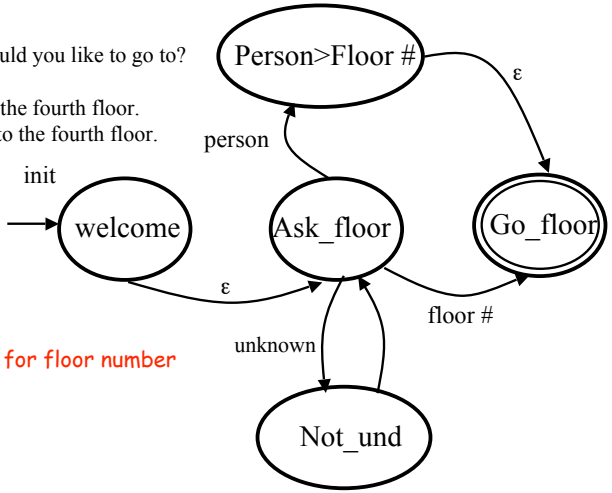
U: Elevator?
S: Hello. Which floor would you like to go to?
U: Third floor.
S: OK, I am taking you to the third floor.



States: …
Init-State: …
Alphabet: …
Transition function: …

FSM-Based Models

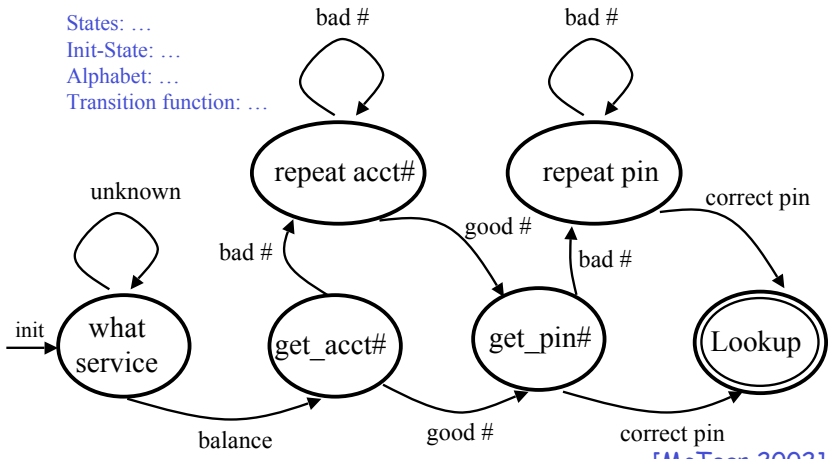
U: Elevator?
S: Hello. Where would you like to go to?
U: Prof. Barry.
S: Prof. Barry is on the fourth floor.
I am taking you to the fourth floor.



Extension: variable for floor number

FSM-Based Models

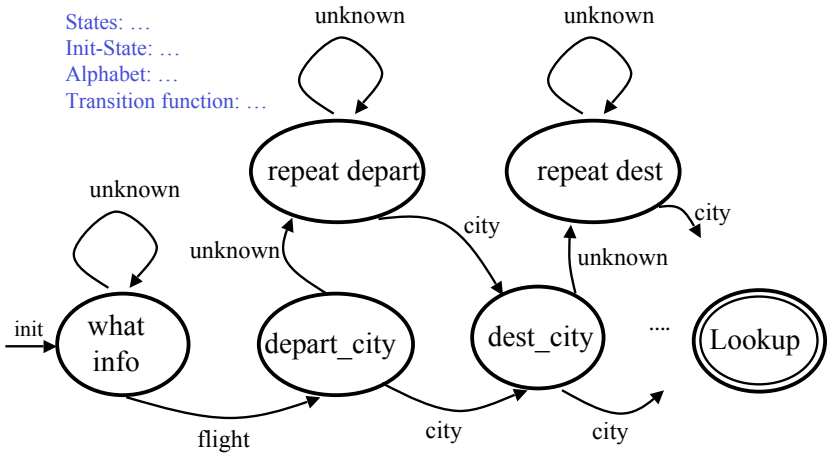
States: …
Init-State: …
Alphabet: …
Transition function: …



[McTear 2002]

FSM-Based Models

States: …
Init-State: …
Alphabet: …
Transition function: …



FSM-Based DM: Sum Up

- Advantages
  - Fixed prompts can be pre-recorded
  - Speech recognition and input interpretation can be tuned for each state
- Disadvantages
  - Very rigid dialogue flow
  - Inhibiting user initiative
  - Only suitable for simple tasks
  - In principle can make more flexible, but it quickly gets very complex

However: modular solutions are possible (--> DiaManT)

Frame-Based DM (Form Filling)

Frame-Based Models

- Frame (form): what info should be supplied by user

departure_city	?
departure_date	?
destination_city	?
return_date	?
…	

- Dialogue states: which slots are filled
- General routines for what system should do next (given which slots are filled)

Frame-Based Models

S: What can I do for you?
U: I want to fly to Paris

departure_city	?
departure_date	?
destination_city	Paris
return_date	?
...	

S: Where will you fly from?
U: From Berlin on August 1st.

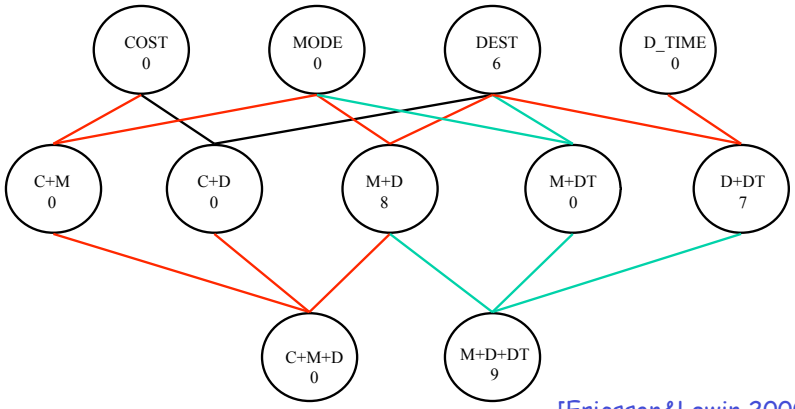
"Overanswering"

departure_city	Berlin
departure_date	1/8/05
destination_city	Paris
return_date	?
...	

Frame-Based Models

- Deciding what to do next
  - Next unfilled slot
  - Slot-combination weighting
  - Ontology-based coherence
- Database lookup
  - Delayed (typically; after certain slots filled)
  - Immediate (can be "expensive" = take time, but enables more helpful system behavior)

Slot-Combination Weighting



[Ericsson&Lewin 2000]

Ontology-based coherence

1. S: What is the patient's sex?
   U: Female with severe nipple discharge
   S: What is the patient's age?
   U: Fifty five
   S: Is the discharge bilateral?
   U: No

2. S: What is the patient's sex?
   U: Female with severe nipple discharge
   S: Is the discharge bilateral?
   U: No
   S: What is the patient's age?
   U: Fifty five

[Milward&Beveridge 2003]

Delayed vs. Immediate Lookup

S: What can I do for you?
U: I want to fly from Berlin to Paris.
S: When would you like to fly?
U: April 1st.
S: At what time would like to fly?
U: In the morning.
S: Sorry. There is no flight from Berlin to Paris on April 1st in the morning.
When would you like to fly?
U: Afternoon.
S: There is one flight. It leaves Berlin at 3 PM and arrives to Paris at 5 PM.

S: What would you like to hear?
U: Play Yesterday.
S: There are 5 songs called Yesterday. Which artist would you like?
U: The Beatles.
S: Sorry, I do not have Yesterday by the Beatles. Would you like another artist?
U: …

VoiceXML

- VoiceXML is a web-based markup language for representing spoken dialogs (analogic to HTML)
- VoiceXML application collects and processes info, and plays back info
- VoiceXML assumes a voice browser
  - Info conveyed to user by audio output (synthesized and/or recorded)
  - Info received from user as audio input (voice and/or telephone keypad tones)

VoiceXML

- Main elements of a VoiceXML document
  - Form: basic unit of functionality
  - Field: prompts for and accepts user input
  - Prompt: sequence of audio elements or TTS messages
  - Audio: audio file or TTS message to play
  - Filled: processes input, can pass control to other forms
- Form Interpretation Algorithm
  - Defines how fields in a form are filled in , and how the fill ordering can be modified
- Global event handlers (e.g., for error handling, help)
  - Define behavior when predefined global conditions occur
- Control transfer conditional and subroutine constructs (= special-purpose programming language)

VoiceXML Example

See VoiceXML tutorials
http://www.palowireless.com/voicexml/tutorials.asp

e.g.,
http://www.vocomosoft.com/voicexml_tutorial.htm
Or Chapters 1 and 2 of
http://cafe.bevocal.com/docs/tutorial/index.html

give good first steps

Frame-Based DM: Sum Up

- Advantages
  - More flexible dialogue
  - Enables some user initiative
- Disadvantages
  - Speech recognition more difficult, because user input less restricted
  - Not every task can be modeled by a frame

Grounding

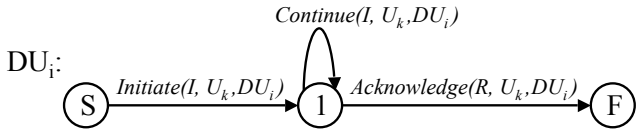
Establishing common ground
(Clark 1996)

Grounding

- Grounding problems are due to
  - Lack of perception or recognition
  - Ambiguity
  - Conflicts
  - Misunderstanding
- Decision: accept/reject/verify/clarify/repair/ignore …
- Clarification and repair strategies, e.g., ask for repetition, rephrase, clarify

Grounding Acts (Traum 1998, based on Clark 1996)

- What is the function of utterance U_k?
  - Does U_k initiate, continue of complete a discourse unit DU_i?

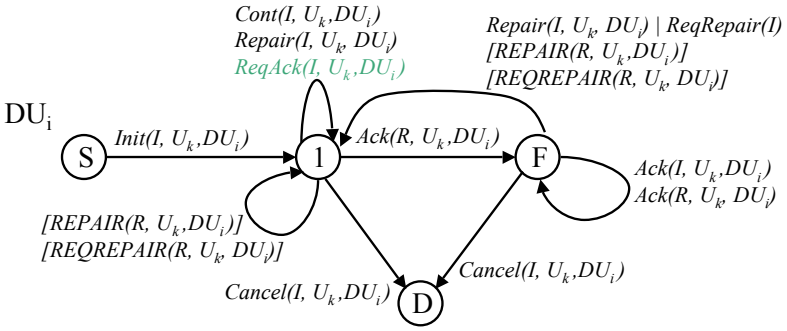


Discourse unit (DU_i): unit of (to be) grounded content

Grounding Acts Example

(1)	1:A: Move the boxcar to Corning	Init(A,1,DU1)
	2:A: and load it with pineapples	Cont(A,2,DU1)
	3:B: OK	Ack (B,3,DU1)
	4:A: I mean, oranges.	Repair(A,4,DU1)
	5:B: OK.	Ack(B,5,DU1)
(2)	1:A: Move the boxcar to Corning	Init(A,1,DU1)
	2:A: and load it with pineapples	Cont(A,2,DU1)
	3:B: OK.	Ack (B,3,DU1)
	4:B: Pineapples?	ReqRepr(B,4,DU1)
	5:A: I mean, oranges.	Repair(A,5,DU1)
	6:B: OK.	Ack(B,6,DU1)

Grounding Acts



Grounding Strategies

- Assuring correct understanding
  - Pessimistic strategy:
    - Immediate explicit verification
    - Terribly inefficient
  - Optimistic strategy
    - Delayed accumulated verification
    - Difficult to recover from errors
    - Error-chaining
  - Carefully optimistic strategy
    - "Implicit" verification by incorporating info to be grounded in next system turn

Verification Strategies in Systems

- Immediate explicit feedback (and verification request)
  S: Where do you want to go?
  U: Hamburg.
  S: Traveling to Hamburg. (OK?)
  U: Yes.
  S: When do you want to go?
- Delayed explicit feedback by summarizing at task end
  …
  S: So. Traveling from Saarbrücken to Hamburg on Monday June 6 …
- Immediate "implicit" feedback by incorporating material to be grounded in the next system turn (see if user accepts or protests)
  S: Where do you want to go?
  U: Hamburg.
  S: And when do you want to go to Hamburg?

Choice of Verification Strategy

- ASR confidence below/above threshold
- Pragmatic Plausibility (Gabsdil & Lemon 2004)
  - Combining ASR confidence with task interpretation confidence (plausible actions in context)
- Context-adapted strategies
  - Dialogue progress so far
  - Reinforcement learning: learn optimal strategies from annotated data, based on rewards for efficient dialogue and user satisfaction (Lemon et al. 2006)

Initiative & Cooperation

Initiative

- Who is in control of the dialogue progression?
- Being the one who's talking does not necessarily mean being in control, e.g., just answering a question
- Dialogue initiative vs. task initiative
- Basically, two models:
  - Fixed initiative model (one participant in control)
    - System-initiative (typical for script-based and form-based DM): system drives dialogue as wanted by prompting user, but this may be unnatural and inconvenient for user
    - User initiative: user can say what wants when wants, but difficult for system, because it doesn't know what is coming
  - Mixed initiative model (either participant can assume initiative, depending on knowledge, skills, situation, etc.)
    - Typical in human-human conversation
    - How to decide whether to take initiative?

Cooperation

- Conversation (and communication in general) is a joint activity
  - It has a purpose (agreed on by the participants)
  - It involves collaboration/cooperation
- Being cooperative: helping each other to accomplish goals by, e.g.,
  - Cooperative interpretation beyond literal meaning (inference), (indirect) dialogue act recognition
  - Cooperative answering
    - Complying with requests or directives when possible
    - Correcting false presuppositions or misconceptions
    - Intensional answers and generalizations
  - Taking initiative when this helps to accomplish the joint activity
    - Providing more information than requested (when it is relevant or useful), e.g., helpful responses (suggestions), when user's input uninterpretabl, when it has to be rejected (e.g., no database results) or when too many database results

References

- D. Jurafsky and J. Martin (2000): Speech and Language Processing, Chapter 19.
- McTear (2002): Spoken Dialogue technology. In ACM Surveys. pp. 1-80
- VoiceXML Forum: http://www.voicexml.org/
- H. Clark. Using Language. Chapters 4 and 8. Cambridge University Press. 1996.
- D. Traum (1998): A computational model of grounding. AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems.
- R. San-Segundo et al. (2001) Designing Confirmation Mechanisms and Error Recover Techniques in a Railway Information System for Spanish. SigDial Workshop.

References

- S. Ericsson and I. Lewin (2000). Dialogue Move Specifications for the Dialogue Move Engine. Siridus project deliverable D1.3. http://www.ling.gu.se/projekt/siridus/Publications/deliv1-3.pdf
- D. Milward and M. Beveridge (2004) Ontologies and the Structure of Dialogue. In Proc. Of the Catalog workshop. pp. 69-76. http://www.upf.edu/dtf/personal/enricvallduvi/catalog04/papers/10-milward-beveridge.pdf
- Malte Gabsdil and Oliver Lemon, " Combining Acoustic and Pragmatic Features to Predict Recognition Performance in Spoken Dialogue Systems" in proceedings of ACL 2004.
- Oliver Lemon, Roi Georgila, James Henderson, and Matthew Stuttle, "An ISU dialogue system exhibiting reinforcement learning of dialogue policies: generic slot-filling in the TALK in-car system", EACL 2006.