

dialog and dialog systems

elevator project seminar, ws06/07

today

what's in a dialog

- properties of dialogue

- turn-taking, dialog sequences, the structure of conversation

- speech acts

- joint activity, grounding

- implicature

dialog systems

- example tasks, modes of interaction

- sub-systems

- dialog management and dialog models

what's in a dialogue

spontaneous spoken dialog

linguistic properties: cohesive devices

structure manifested in the dialog partys' contributions

speech-related phenomena:

pauses and fillers („uh”, „um”, „...”, like, you know,...”)

prosody, articulation

disfluencies

overlapping speech

dialog specific phenomena:

dialog acts/speech acts, dialog sequences, grounding

spontaneous vs. practical dialogs

topic drifts vs. goal-orientedness → joint activity

interpreting monologue vs. dialogue

both (narrative) monologue and dialogue involve interpreting
information status
coherence/rhetorical relations
contextual references
intentions

dialogue additionally involves:

turn-taking
initiative and confirmation strategies
grounding
repairing misunderstandings

dialog: turn taking

dialog is made up of turns:

speaker A says something, then speaker B, then speaker A...

turn taking: who should talk and when

how do speakers know when it's time to contribute a turn?

btw, children learn turn taking within the first 2 years of life (Stern74)

conversation analysis

→ turn taking rules: determine who is expected to speak next

- if current speaker selects S, S must speak
- if current speaker does not specifically select, any speaker may speak
- if noone else takes the next turn, current speaker may take next turn

rules apply at Transition-Relevance Points (TRP):

where dialog/utterance structure allows for a speaker change
(typically at intonational phrase boundaries)

dialog: dialog sequences

some turns specifically select who the next speaker will be

→ adjacency pairs

regularly occurring, conventionalized sequences

conventions introduce *obligations* to respond (preferred responses)

greeting : greeting

question : answer

complement : downplayer

accusation : denial

offer : acceptance

request : grant

set up next speaker expectations

‘significant silence’ dispreferred

‘no’ without explanation dispreferred in response to request

abrupt topic changes dispreferred

dialog: joint activity

communication relies on collaboration

dialogue: collective act performed by the speaker and the hearer

cooperatively interpret and contribute

Gricean maxims of conversation + principles of rational behaviour

cooperative principles: quality, quantity, relevance, manner

certain stock of knowledge is taken for granted, assumed to be known both by the speaker and the hearer

BUT there may exist discrepancies between *private vs. mutual beliefs*

→ crucial: establish shared knowledge, *common ground*

dialog: grounding

establishing „common ground” (set of things mutually believed by both speaker and hearer)

the hearer must ground or acknowledge speaker's utterance
OR signal that there was a problem in reaching common ground

closure principle: agents performing an action require evidence, sufficient for current purposes, that they have succeeded in performing it (Clark96)

dialog: grounding

interpretation: multiple levels:

channel:	S executes, H attends
signal:	S presents, H identifies
proposition:	S signals that p, H recognizes that p
intention:	S proposes p, H considers p

grounding feedback possible at all levels

- continued attention
- relevant next contribution
- acknowledgement
- demonstration (e.g. paraphrase, completion)
- display (verbatim)

dialog: grounding

S: I can upgrade you to an SUV at that rate.

H gazes appreciatively at S (continued attention)

H: Do you have a RAV4 available? (relevant next contribution)

H: ok / mhmmm / Great! (acknowledgement/backchannel)

H: An SUV. (demonstration/paraphrase)

H: You can upgrade me to an SUV at the same rate? (display/repetition)

H: I beg your pardon? (request for repair)

dialog: grounding

problems:

- lack of perception

- lack of understanding

- ambiguity

- misunderstanding

→ clarification and repair strategies

dialog systems

dialog systems

goal-oriented conversational systems

challenges:

- need to understand

- interpretation context-dependent

- intention recognition

- anaphora resolution

- people don't talk in sentences...

- user's self-revisions

dialog systems

goal-oriented conversational systems

how:

- interactions in a limited domain

- prime users to adopt vocabulary the system knows

- partition interaction into manageable stages

- let the system take the initiative (predictability)

dialog systems

example tasks:

retrieve information → information-seeking dialogue

seek to satisfy constraints → negotiation dialogue

perform action → command-control dialog

collaborate on solving a problem → problem-solving dialog

instruct → tutorial/instructional dialogue

applications:

travel arrangements, telephone directory

customer service, call routing

tutoring

communicating with robots

voice-operated devices

dialog systems: travel arrangements (Communicator)

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

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

dialog systems: call routing (ATT HMIHY)

S: How may I help you?
U: *can you tell me how much it is to tokyo?*
S: You want to know the cost of a call?
U: *yes that's right*
S: Please hold on for rate information.

S: How may I help you?
U: *what's the area code for chicago?*
S: You want an area code.
U: *(silence)*
S: The area code for Chicago is 312.

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

dialog systems: tutorial dialog (ITSPOKE)

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

U: *uh the force of gravity*

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

U: *no*

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

U: *it will be moving*

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

U: *freefall*

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

dialog systems

modality: type of communication channel used to convey or acquire information

natural-language: spoken or textual keyboard-based or both

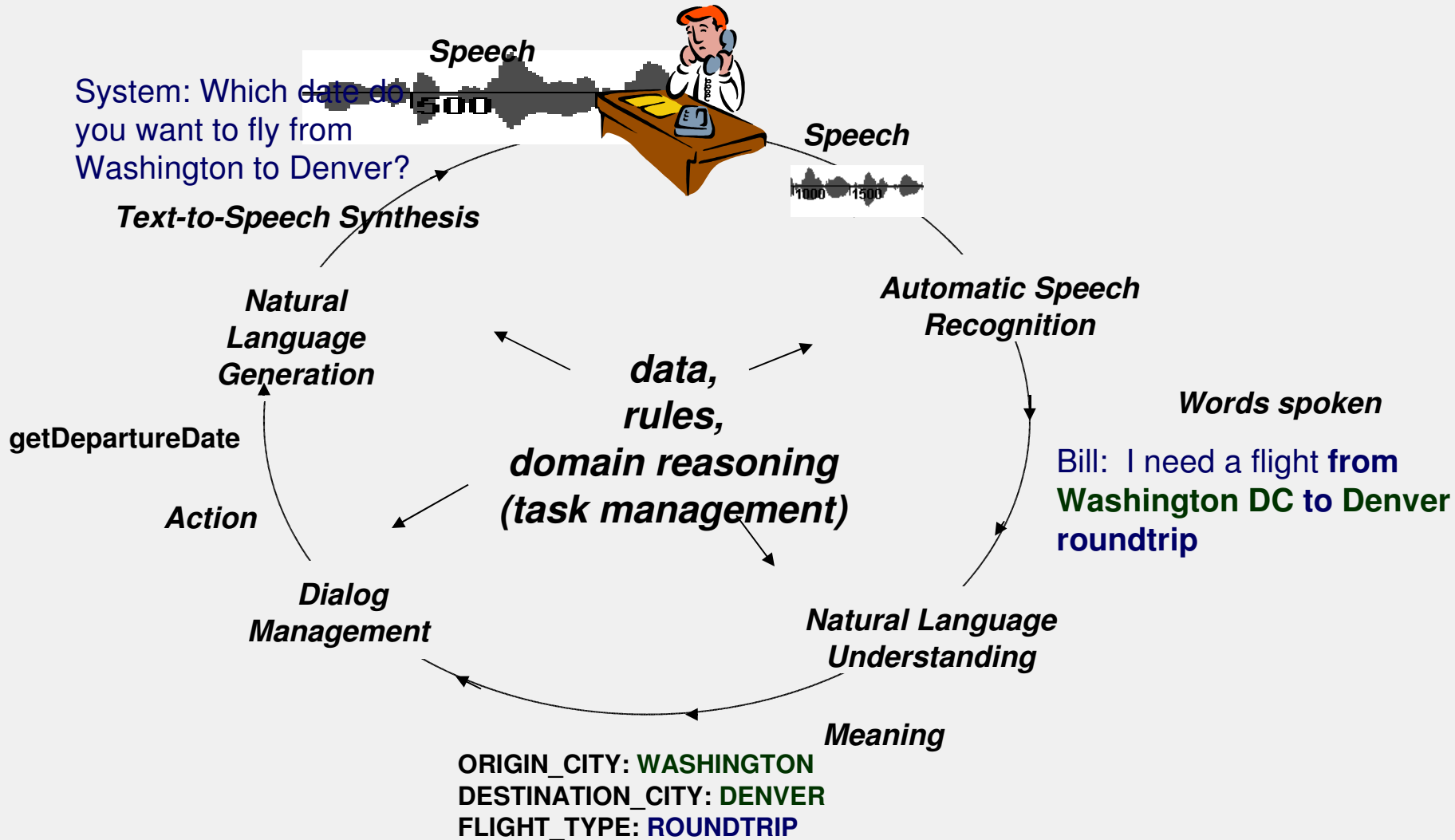
pointing devices

graphics, drawing

gesture

combination of one or more of above (multi-modal systems)

dialog systems



dialog systems

typical components:

ASR, NLU: tell system what was said

Dialog Manager: when to say, what to say

Task Manager: perform domain-relevant action

NLG: how to say

TTS: say

dialog systems

additional components:

speaker identification, verification; e.g. banking

system knows the speaker...

definitely: say „hi, Cindy”, go directly to appropriate account

probably: say “is that Cindy?”

possibly: say “have you used this service before?”

otherwise: say “hi, what’s your name”

user model

modality handlers (input fission, output fusion)

...

dialog systems: speech recognition

ASR: speech to words/meanings

language model + recognition grammar („semantic grammar”)

understanding user crucial → grammars typically hand-written context-free rather than statistical

```
REQUEST : tell me | I want | I'd like | ...
DEPARTURE_TIME : (after|around|before) HOUR | morning | evening
HOUR : one|two|three| ... |twelve (am|pm)
FLIGHTS : (a) DEPARTURE_TIME flight | DEPARTURE_TIME flights
ORIGIN : from CITY
DESTINATION : to CITY
CITY : London | Warsaw | New York | ...
```

dialog systems: speech recognition

(some) problems

grammar-writing time-consuming, expensive

limited coverage (grammar writer will probably miss many possible formulations because he/she just doesn't think about them)

(some) things to consider:

„restricted” language models: dependent on dialogue state

e.g. if asking for city name, model only with city names

could make use of the fact that the system know who the speaker is

adapt to speaker: acoustic, language model, pronunciation

will user be allowed to speak while the system is speaking?

need to correctly detect speech (esp. in noisy environments)

using recognition confidence values

overall utterance, individual words, combined

dialog systems: generation and speech synthesis

NLG: based on content (meaning) to be expressed:
plans sentences
chooses how to express concepts with words; syntactic structures and
lexemes → surface realization
 simplest method: „canned” utterances (with variable slots)
 → „template-based” generation
if possible, assigns prosody (according to context)

Text-to-Speech component
takes NLG output
synthesizes a waveform

dialog systems: generation and speech synthesis

(some) NLG considerations:

system prompts influence dialog coherence and „naturalness” →
variation:

S1: Please say the departure time

S2: Please say the departure city.

S3: Please say the destination city.

S1: First, tell me when you would like to travel

S2: And from which city would you like to leave?

S3: Thanks. Now, what is your destination?

dialog systems: generation and speech synthesis

(some) NLG considerations:

system prompts influence dialog coherence and „naturalness” → tapered prompts (gradually shorter if same sub-dialog):

S: Now, what's the first company to add to your watch list?

U: Cisco

S: What's the next company name? Or, you can say, „Done”

U: IBM

S: Tell me the next company name, or say, „Done.”

U: Intel

S: Next one?

U: America Online.

S: Next?

U: ...

dialog systems: dialog management

dialog engine's tasks:

when to say? → control the flow of dialog

what to say? → dialog modeling

takes input from ASR/NLU

maintains some sort of „dialog state”

communicates with Task Manager

passes output to NLG/TTS

dialog systems: dialog management

control the flow of dialog

when to say something and when to listen (turn-taking), when to stop

update dialog context with current user's input and output the next action in the dialog

deal with barge-in, hang-ups

dialog modeling

what is the context

what to say next

goal: achieve an application goal in an efficient way through a series of interaction with the user

dialog systems: turn-taking strategies

rigid turn taking

system speaks till it completes turn, stops, and only then listens to user
system waits till user stops speaking and responds again

problems: users must wait for system to finish turn
 users often speak too early,
 make too long pause while speaking (interpreted as end of turn)

flexible turn taking

user barge-in; as in natural conversation → more efficient

problems: backchannel or noise misinterpreted as user turn
 system interprets own output as input



„push-to-talk” instead of „open microphone” if possible
system sound when microphone open

dialog systems: initiative strategies

directive prompt

explicit instruction on what information user should supply at given point

open prompt

no/few constraints on what user can say

restrictive grammar

constrains the ASR/NLU system based on dialogue state

non-restrictive grammar

open language model, not restricted to a particular dialogue state

grammar prompt	open	directive
restrictive	—	system initiative
non-restrictive	user initiative	mixed initiative

dialog systems: initiative strategies

system initiative

S: Please give me your arrival city name.

U: Baltimore.

S: Please give me your departure city name....

user initiative

S: How may I help you?

U: I want to go from Boston to Baltimore on November 8.

mixed initiative

S: How may I help you?

U: I want to go to Boston.

S: What day do you want to go to Boston?



dynamically adjust strategy e.g. change from mixed initiative to system initiative if, e.g. ASR problems detected, many user corrections (dialogue history), based on user model and/or context model

dialog systems: dialog models

why need dialog models?

system and user work on a task
dialog structure reflects the task structure

BUT:

dialog need not follow the task-steps
need for grounding

dialog systems: dialog models

examples of dialog models

FSA

frame-based

Information State (aka ISU)

the choice depends on the complexity and nature of the task

dialog systems: dialog models

FSA-based dialog models

dialog modelled as a directed graph: set of states + transitions

system utterance determined by state

(interpretation of) user utterance determines next state (deterministic transition)

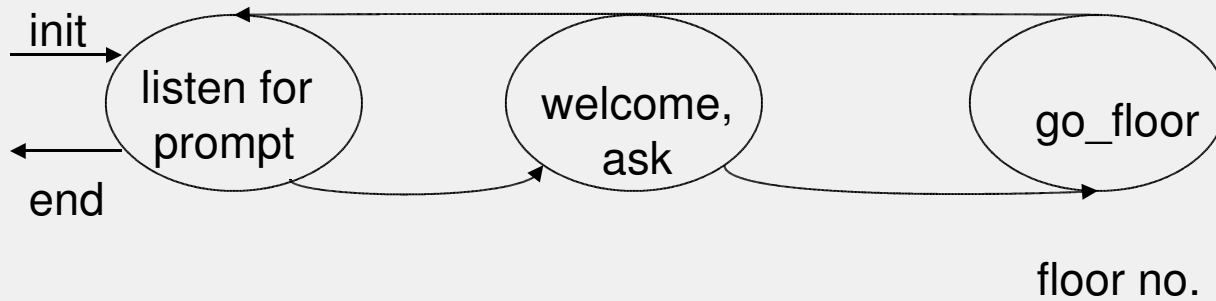
dialog systems: dialog models

FSA-based dialog models

```
start 01 getName
02 getTransactionType
03 if type == balance goto 10
03 if type == deposit goto 20
...
50 ask(„another transation?“)
if „yes“ goto 02
else stop
```

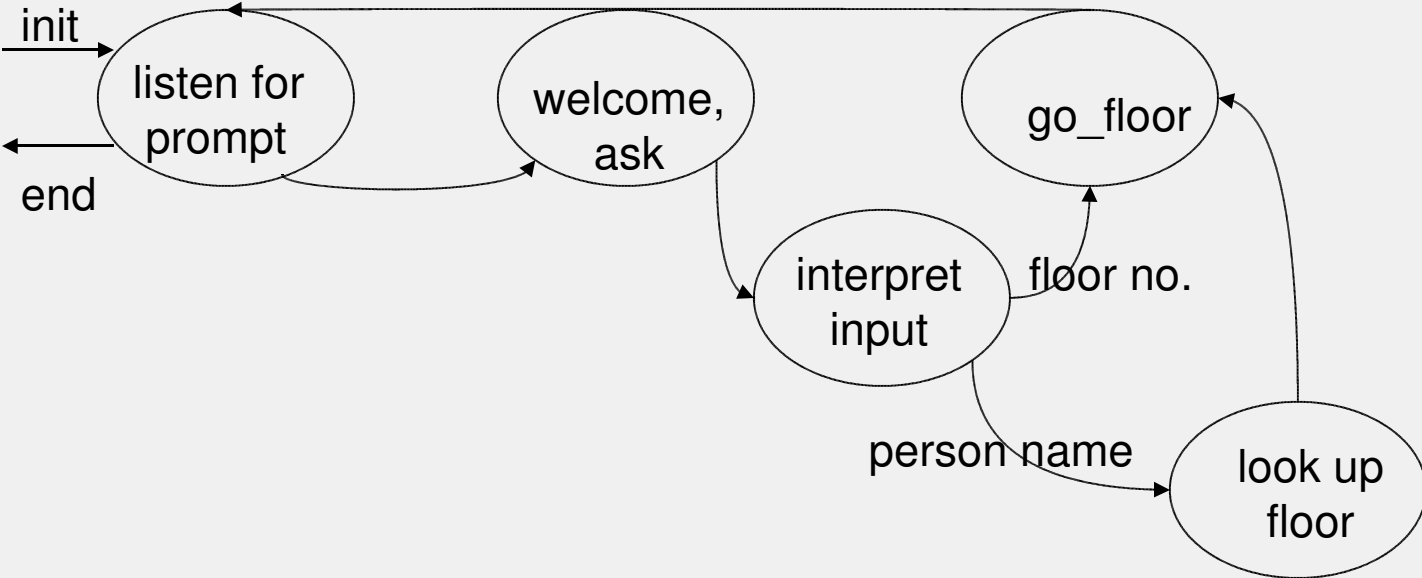
dialog systems: dialog models

FSA-based dialog models



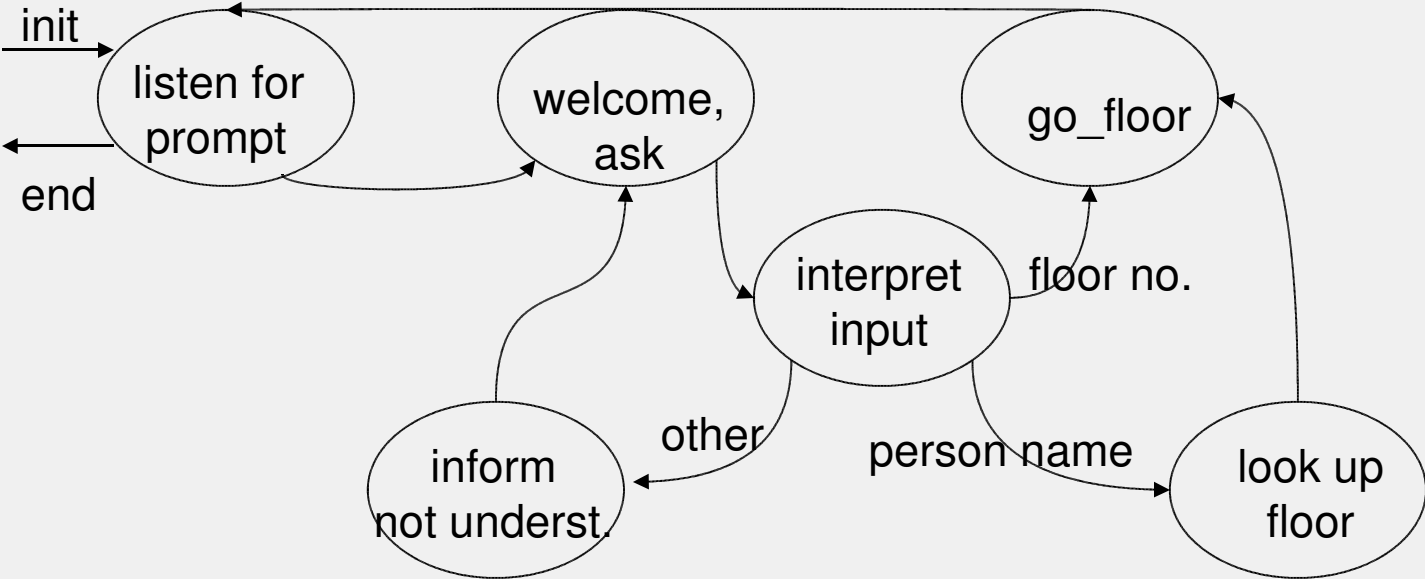
dialog systems: dialog models

FSA-based dialog models



dialog systems: dialog models

FSA-based dialog models



dialog systems: dialog models

FSA-based dialog models

fixed dialog script, system driven interaction

pros: fixed prompts (can pre-record)

ARS and interpretation can be tuned for each state

cons: rigid dialogue flow
user initiative?

in principle, more flexibility possible, but graphs grow complex quickly

suitable for simple fixed tasks

dialog systems: dialog models

frame-based dialog models

sets of precompiled templates for each data item needed in the dialog

system's agenda → fill the slots in the template

system maintains initiative → directed-questions (prompts)

slots need not be filled in a particular sequence → over-answering,
actions triggered on other slots

dialog systems: dialog models

frame-based dialog models

```
[SHOW:  
  FLIGHTS:  
    (getOrigin CITY)  
    (getDate DATE) (getTime TIME)  
  DEST:  
    (getDestination CITY)]
```

U1: Show me flights to SF.

U2: Show me morning flights from Boston to SF on Tuesday.

dialog systems: dialog models

frame-based dialog models

pros: enables some user initiative
more flexible than FSA

cons: user input less restricted → ASR more difficult
not every task can be modeled by frames

not suited to dynamic complex dialogs
doesn't handle multiple topics/conversation threads

dialog systems: dialog models

Information State-based models

Information State (IS) is a representation of current dialog state

dialog contributions viewed as dialog moves (DMs)

dialog move types similar to speech acts, e.g. command, wh-question, revision, etc.

IS is used to:

interpret user's utterances → *update* the dialog state

decide which external actions to take

decide when to say what

store information (dialogue context representation)

dialog systems: dialog models

Information State-based models

pros: allows for contextual interpretation
 rich representation (includes dialog context, obligations, etc.)
 dialog is not scripted
 dialog history stored → multi-threaded conversations
 allows for mixed-initiative

cons: complex apparatus

both FSA and frame-based models can be represented as ISU-models

dialog systems: grounding

ASR and input interpretation are error prone

grounding helps to make sure system interpreted correctly

users of speech-based interfaces are confused when system doesn't give them an explicit acknowledgement signal (Stifelman et al.93, Yankelovich et al.95)

→ in fact, *crucial* in design of dialog systems

grounding strategies

pessimistic: immediate explicit verification (awfully inefficient)

optimistic: delayed accumulated verification (errors accumulate, recovery difficult)



carefully optimistic: implicit verification → incorporate data to be verified in next system turn

choosing strategy: ASR confidence (e.g. combined with system's belief about task state)

dialog systems: grounding

grounding examples:

carefully optimistic strategy: acknowledgement + relevant next contribution

U: I need to travel in May.

S1: What day in May did you want to travel?

S2: And you're flying into what city?

S3: And what time would you like to leave?

grounding negative responses:

S: Did you want to review some more of your personal profile?

U: No.

S1: What's next? **(bad!)**

S2: Okay, what's next?

dialog systems: grounding

(some) considerations:

if system rejects user's utterance or displays misunderstanding/non-understanding, then user will probably make a correction:

reply „no” to confirmation question, repeat, rephrase, etc.

BUT corrections are typically **harder** to recognize than normal sentences (misrecognized twice as often as non-corrections...)

why? hyperarticulation, users tend to speak louder, user's rephrasing may aggravate because out-of-grammar

→ need for efficient error recovery strategies

e.g. dynamicaly change prompt strategy, switch to restrictive grammar

if dialog does nowhere, should the system be allowed to hang-up on a user?...

dialog systems: grounding, clarification

- A: Welcome to Busman, the Tampere bus information service! How may I help you?
- U: When does the next bus leave from Central Square to Her ehhh correction from Central Square to Hervanta?
- A: Sorry, I didn't recognize that. Please speak clearly.
- U: What now?
- A: Busman understands questions such as "When does the next bus to Hervanta depart?", ... [help continues]
- U: I'm leaving from Central Square, and I'm going to Hervanta.
- A: Sorry, I still didn't recognize that.
- A: Please tell me where you are leaving from.
- U: From Central Square.
- A: What is your destination?
- U: Hervanta.
- A: From Central Square to Hervanta use lines 13, 20, 23 and 30.
- A: Do you want to hear timetables?
- U: Yes.
- A: From Central Square to Hervanta on weekdays at 11:15 bus in line 23...

dialog systems: actually...

many practical implemented systems:

simple dialog model (e.g. frame)

system initiative + universals

„universals”: commands you can say anywhere

e.g. in FSA, at every state user can say things like „help”,
„start over”, „correct”

next: dialog authoring with Diamant

