

Language Technology 2: Natural Language(-based) Interaction

www.coli.uni-saarland.de/courses/late2-10

Manfred Pinkal, Magdalena Wolska

dialogue modelling
(slides based on Ivana Kruijff-Korbayova's)

June 2010

late2.10.nli.dm

1

outline

- dialogue management
- dialogue models
 - state machines
 - frames
 - more flexibility?...
 - in grounding and verification
 - initiative and cooperation...
 - information state update
- current challenges

June 2010

late2.10.nli.dm

2

dialogue management

- dialogue flow control
- dialogue modeling → dialogue context, dialogue moves
- error handling
- initiative and cooperation
- adaptivity
- ...

June 2010

late2.10.nli.dm

3

dialogue flow control

- when to say something,
- when to stop
- ⇒ turn taking

June 2010

late2.10.nli.dm

4

turn taking

dialogue participants take turns (like in a game): A, B, A, B

dialogue turn: a continuous “contribution” to the dialogue by one speaker

turn passing at TRPs

turn taking in human-machine 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?

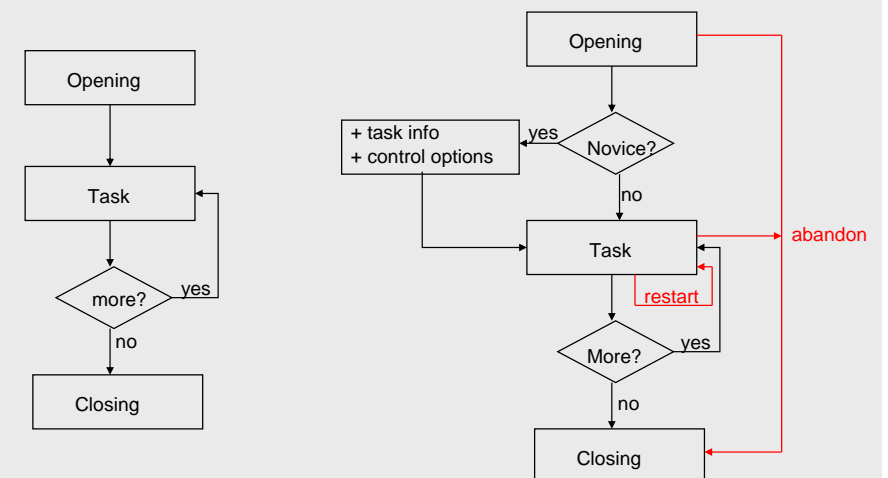
E.g., list with several options, complex info --> reference resolution

system barge-in: When appropriate at all? When is a TRP?

dialogue modeling

where we are
&
what to say next

global structure



dialogue models

script-based: finite automata

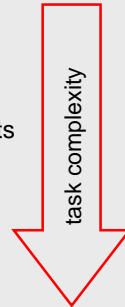
sequence of pre-defined steps (dialogue script)

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 models: state machines

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 automata

$\langle Q_{\text{states}}, \Sigma_{\text{alphabet}}, \delta_{\text{transitions}} (\text{mapping: } Q \times (\Sigma \cup \epsilon) \rightarrow P(Q)), Q_{\text{initial states}}, Q_{\text{final states}} \rangle$

variants: machines with

actions associated with states or transitions (Moore and Mealy machines; transducers)

transitions conditioned on no input symbol (a null)

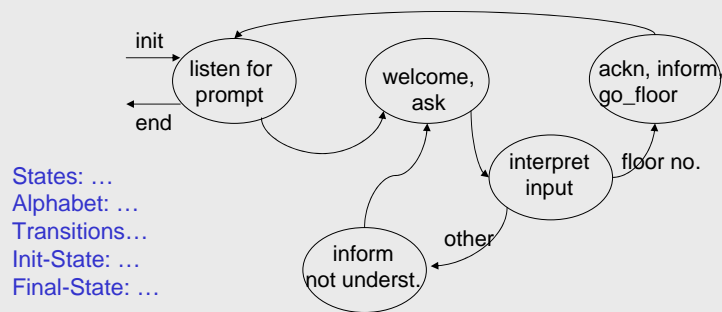
more than one transition for a given symbol and state (nondeterministic FSM)

states designated as accepting states (recognizer)

etc.

finite automata

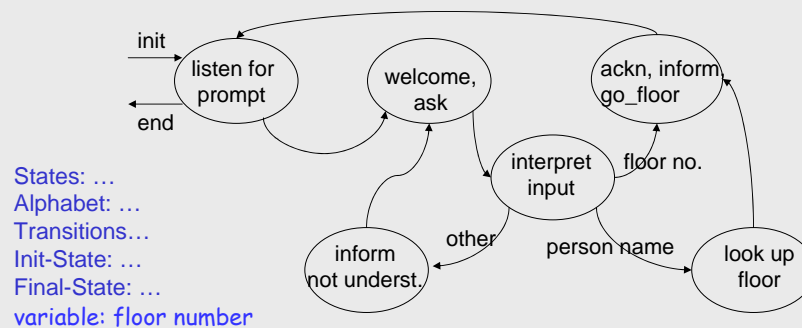
U: Elevator.
S: Hello. Which floor would you like to go to?
U: Third floor.
S: OK, I am taking you to the third floor.



States: ...
Alphabet: ...
Transitions...
Init-State: ...
Final-State: ...

finite automata

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.



States: ...
Alphabet: ...
Transitions...
Init-State: ...
Final-State: ...
variable: floor number

finite automata: summary

advantages

- fixed prompts can be pre-recorded
- speech recognition and input interpretation can be tuned for each state

disadvantages

- rigid dialogue flow
- inhibiting user initiative
- only suitable for simple tasks
- in principle, more flexible possible, but it quickly gets very complex

however: modular solutions are possible → e.g. DialogOS

frame-based models: 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	01/08/07
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": takes time, but facilitates more helpful system behavior)

frame-based models: delayed vs. immediate information retrieval

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: ...

frame-based models: VoiceXML

web-based markup language for representing spoken dialogs

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., error handling, help)

define behavior for predefined global conditions occur

control transfer conditions and subroutine constructs

(= special-purpose programming language)

<http://www.palowireless.com/voicexml/tutorials.asp>

http://www.vocomosoft.com/voicexml_tutorial.htm

<http://cafe.bevocal.com/docs/tutorial/index.html>

(btw: new, more expressive standard: State Chart XML)

frame-based models: summary

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

grounding

mis-communication or non-communication may arise are due to (among others):

- lack of perception or recognition
- ambiguity
- misunderstanding

decision: accept/reject/verify/clarify/repair/ignore ...

clarification and repair strategies, e.g., ask for repetition, rephrase, clarify

grounding (Clark and Schaefer 1989, Clark 1996, Traum 1994, 1998)

assumption: collaboration

humans collaborate in conversation to establish common ground

discourse is modelled in terms of **contributions**:

contribution : presentation + acceptance

grounding (Clark and Schaefer 1989, Clark 1996, Traum 1994, 1998)

levels of joint action in conversation → Clark's joint action ladder

Level	Speaker S's actions	Hearers's actions
4	S is proposing a joint project w to H	H is considering S's proposal of w
3	S is signalling that P for H	H is recognizing that P from S
2	S is presenting signal s to H	H is identifying signal s from S
1	S is executing behaviour b for H	H is attending to behaviour b from S

grounding (Clark and Schaefer 1989, Clark 1996, Traum 1994, 1998)

levels of joint action in conversation → Clark's joint action ladder

Level	Speaker S's actions	Hearers's actions
4	S is proposing a joint project w to H	H is considering S's proposal of w
3	S is signalling that P for H	H is recognizing that P from S
2	S is presenting signal s to H	H is identifying signal s from S
1	S is executing behaviour b for H	H is attending to behaviour b from S

coordinate the Hearer's attention with Speaker's communicative actions (e.g. S must make sure H is listening)

grounding (Clark and Schaefer 1989, Clark 1996, Traum 1994, 1998)

levels of joint action in conversation → Clark's joint action ladder

Level	Speaker S's actions	Hearers's actions
4	S is proposing a joint project w to H	H is considering S's proposal of w
3	S is signalling that P for H	H is recognizing that P from S
2	S is presenting signal s to H	H is identifying signal s from S
1	S is executing behaviour b for H	H is attending to behaviour b from S

coordinate the Hearer's attention with Speaker's communicative actions (e.g. S must make sure H is listening)

once a speech act successfully executed and attended to, Hearer must try to identify the presented signals (verbal and non-verbal)

grounding (Clark and Schaefer 1989, Clark 1996, Traum 1994, 1998)

levels of joint action in conversation → Clark's joint action ladder

Level	Speaker S's actions	Hearers's actions
4	S is proposing a joint project w to H	H is considering S's proposal of w
3	S is signalling that P for H	H is recognizing that P from S
2	S is presenting signal s to H	H is identifying signal s from S
1	S is executing behaviour b for H	H is attending to behaviour b from S

coordinate the Hearer's attention with Speaker's communicative actions (e.g. S must make sure H is listening)

once a speech act successfully executed and attended to, Hearer must try to identify the presented signals (verbal and non-verbal)

Hearer must understand what Speaker means

grounding (Clark and Schaefer 1989, Clark 1996, Traum 1994, 1998)

levels of joint action in conversation → Clark's joint action ladder

Level	Speaker S's actions	Hearers's actions
4	S is proposing a joint project w to H	H is considering S's proposal of w
3	S is signalling that P for H	H is recognizing that P from S
2	S is presenting signal s to H	H is identifying signal s from S
1	S is executing behaviour b for H	H is attending to behaviour b from S

coordinate the Hearer's attention with Speaker's communicative actions (e.g. S must make sure H is listening)

once a speech act successfully executed and attended to, Hearer must try to identify the presented signals (verbal and non-verbal)

Hearer must understand what Speaker means

once understood, decide whether to comply to the proposed project or action

grounding (Clark and Schaefer 1989, Clark 1996, Traum 1994, 1998)

levels of joint action in conversation → Clark's joint action ladder

Level	Speaker S's actions	Hearers's actions
4	S is proposing a joint project w to H	H is considering S's proposal of w
3	S is signalling that P for H	H is recognizing that P from S
2	S is presenting signal s to H	H is identifying signal s from S
1	S is executing behaviour b for H	H is attending to behaviour b from S

e.g. Can you pass the salt?

H must succeed (jointly with S) in construing the meaning of the utterance (Level 3) before H can consider S's *request* to pass the salt (Level 4).

grounding (Clark and Schaefer 1989, Clark 1996, Traum 1994, 1998)

levels of joint action in conversation → Clark's joint action ladder

Level	Speaker S's actions	Hearers's actions
4	S is proposing a joint project w to H	H is considering S's proposal of w
3	S is signalling that P for H	H is recognizing that P from S
2	S is presenting signal s to H	H is identifying signal s from S
1	S is executing behaviour b for H	H is attending to behaviour b from S

H can be in one of the following states:

H did not notice that S's uttered U

H noticed, but did not hear it correctly

H heard it correctly, but did not understand it

H understood

grounding (Clark and Schaefer 1989, Clark 1996, Traum 1994, 1998)

grounding theories

Clark & Schaefer's discourse contributions

Traum's conversational acts

the grounding criterion

S and H mutually believe that H understood what S said
to a criterion sufficient for the current purposes

grounding (Clark and Schaefer 1989, Clark 1996, Traum 1994, 1998)

grounding theories

Clark & Schaefer's discourse contributions

Traum's conversational acts

the grounding criterion

S and H mutually believe that H understood what S said
to a criterion sufficient for the current purposes

grounding acts (Clark and Schaefer 1989, Clark 1996, Traum 1994, 1998)

computational model of grounding in conversation

conversational acts (extension of speech acts theory)

turn-taking

grounding

core speech acts

argumentational acts

grounding acts (Clark and Schaefer 1989, Clark 1996, Traum 1994, 1998)

computational model of grounding in conversation

conversational acts (extension of speech acts theory)

turn-taking

grounding: Initiate, Continue, Cancel, ReqAck, Ack, ReqRepair, Repair

core speech acts

argumentational acts

grounding acts (Clark and Schaefer 1989, Clark (1996), Traum 1998)

Discourse Unit (DU): unit of content to be grounded
 goal: grounded state (mutually believed common ground)
 what is the function of utterance U?
 e.g. is H signalling how S's last utterance was interpreted?
 does U initiate, continue, complete a discourse unit DU?

grounding acts (Clark and Schaefer 1989, Clark (1996), Traum 1998)

Discourse Unit (DU): unit of content to be grounded
 goal: grounded state (mutually believed common ground)
 what is the function of utterance U?
 e.g. is H signalling how S's last utterance was interpreted?
 does U initiate, continue, complete a discourse unit DU?

Label	Description
initiate	Begin new DU, content separate from previous uncompleted DUs
continue	same agent adds related content to open DU
acknowledge	Demonstrate or claim understanding of previous material by other agent
repair	Correct (potential) misunderstanding of DU content
Request Repair	Signal lack of understanding
Request Ack	Signal for other to acknowledge
cancel	Stop work on DU, leaving it ungrounded and ungroundable

grounding acts (Clark and Schaefer 1989, Clark (1996), Traum 1998)

Discourse Unit (DU): unit of content to be grounded
 goal: grounded state (mutually believed common ground)
 what is the function of utterance U?
 e.g. is H signalling how S's last utterance was interpreted?
 does U initiate, continue, complete a discourse unit DU?

S: start DU
 F: final (DU grounded)
 D: DU abandoned
 I: initiator, R: responder

Label	Description
initiate	Begin new DU, content separate from previous uncompleted DUs
continue	same agent adds related content to open DU
acknowledge	Demonstrate or claim understanding of previous material by other agent
repair	Correct (potential) misunderstanding of DU content
Request Repair	Signal lack of understanding
Request Ack	Signal for other to acknowledge
cancel	Stop work on DU, leaving it ungrounded and ungroundable

Next Act	In State					
	S	1	2	3	4	F D
initiate^I	1					
continue^I		1			4	
continue^R			2	3		
repair^I		1	1	1	4	1
repair^R		3	2	3	3	3
ReqRepair^I		4	4	4	4	4
ReqRepair^R		2	2	2	2	2
ack^I				F	1	F
ack^R		F	F			F
ReqAck^I		1				1
ReqAck^R				3		3
cancel^I		D	D	D	D	D
cancel^R			1	1		D

grounding acts: example

(1) A: Move the boxcar to Corning
 A: and load it with pineapples
 B: OK
 A: I mean, oranges.
 B: OK.
 A: and now take the pinapples...
 initiate DU1
 continue
 ack
 Repair
 ack
 initiate DU2

(2) A: Move the boxcar to Corning
 A: and load it with pineapples
 B: OK.
 B: Pineapples?
 A: I mean, oranges.
 B: OK.
 initiate DU1
 continue
 ack
 ReqRepair
 Repair
 ack

grounding strategies in dialogue systems

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

grounding strategies in dialogue systems: examples

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?

grounding strategy in dialogue systems: factors

ASR confidence below/above threshold

pragmatic plausibility (Gabsdil & Lemon 2004)

combining ASR confidence with task interpretation confidence (plausible actions in context)

context-adaptive strategies

dialogue progress so far

→ reinforcement learning: learn optimal strategies from data based on rewards for „good“ 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

two models:

fixed initiative model → one participant in control

system-initiative (typical for script-based and form-based DM)

user initiative

mixed initiative model → either participant can assume initiative, depending
on knowledge, skills, situation, etc.

typical in human-human conversation

but: how to decide whether to take initiative?

cooperation

conversation (and communication in general) is a joint activity

has a purpose (agreed on by the participants)

involves collaboration/cooperation in achieving goals

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 uninterpretable,

when it has to be rejected (e.g., no database results)

or when too many database results

flexible dialogue modelling: Information State Update

information state

representation of the current state of dialogue

used by the system to:

(contextually) interpret user's turn

decide which external actions to take

decide what to say

store information (dialogue context representation)

utterances update the information state

(approaches to IS-base modelling differ in how IS is represented, what role it plays, what
it contains)

information state based theories of dialogue modelling: components

a description of the informational components of the IS

(aspects of common context, participants, common ground, linguistic and intensional structure, commitments, beliefs, intentions, user model...)

their formal representation

(e.g. lists, sets, typed feature structures, DRSs, propositions, modal operators, etc.)

set of dialogue moves (DMs) triggering the update of the IS

set of update rules governing the IS updates given various conditions of current IS and performed DMs

(e.g. set of selection rules that licence choosing a particular DM to perform given IS)

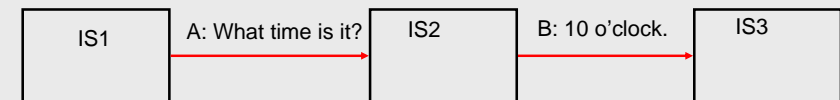
a control strategy to decide which update rule(s) to select at a given point in the dialogue

(e.g. „pick first that applies”, game theory, statistical methods)

information state: update rules

describe possible transitions from one information state to the next

```
if <conditions-on-IS-values>  
then <changes-to-IS-values>
```



conditions: when rule applicable

effects: how IS changes

information state: state machine model as IS update

IS: current-state + input

update rules:

```
if [state] & [input]  
then [output]; [next-state]
```

information state: frame-based model as IS update

IS: task-frame + user's move + system move

update rules:

e.g.,
if [user move = slot X value V] then [fill X with V]
if <conditions-on-frame-values> then <ask-slot-value Y>
(decision about next system move is also a rule!...)

information state: place of the task model

task- vs. dialogue structure

task → dialogue

but, dialogue does not have to follow task (execution) structure

„planning” dialogue → agenda

task model fills agenda with task-related goals

dialogue manager can add more goals, e.g., for grounding

update model:

Question Under Discussion (QUD-based): Godis

obligations-based: Edis

agent-based (collaborative problem solving): TALK

QUD-based information state update

information state in Godis:

```
PRIVATE : [ AGENDA : stack(Action)
            PLAN : stack(Action)
            BEL : set(Prop) ]
SHARED : [ COMMITMENTS : set(Prop)
           QUD : stack(Question)
           LU : [ SPEAKER: Speaker
                 MOVES: assocSet(Move) ] ]
```

+ module interface variables

```
INPUT : String
LATEST-MOVES: Set(Move)
LATEST-SPEAKER: Speaker
NEXT-MOVES: Set(Move)
OUTPUT: String
```

QUD-based information state update: example rules

U: "how much does a flight cost?"

if user asks Q, push respond(Q) on AGENDA

if respond(Q) on AGENDA and PLAN empty, find plan for Q and load to PLAN

if findout(Q) first on PLAN, ask Q

S: "where do you want to go?"

U: "Paris"

if LM=answer(A) and A **about** Q, then add P=Q[A] to SHARED.COM

if P in SHARED.COM and Q topmost on QUD and P **resolves** Q, then pop QUD

if P in SHARED.COM and P **fulfils goal** of findout(Q) and findout(Q) on PLAN, then pop PLAN

QUD-based information state update: example plan

```
findout(?x.transport(x))
findout(?x.dest-city(x))
findout(?x.depart-city(x))
findout(?x.dept-month(x))
findout(?x.dept-day(x))
findout({?class(economy), ?class(business)})
consultDB(?x.price(x))
respond(?x.price(x))
⇒ system's agenda
```

QUD-based information state update: answer integration

IS update rule for answer integration

```
integrateAnswer
pre: {
  in($SHARED.LU.MOVES, answer(A))
  fst($SHARED.QUD, Q)
  $DOMAIN:about(A, Q)
}
eff: {
  DOMAIN: combine(Q, A, P)
  add(SHARED.COM, P)
}
```

(before answer can be integrated, it must be matched to a question on QUD!...)

QUD-based information state update: answer integration example

...

S: "what class did you have in mind?"

U: "cheap"

if consultDB(Q) on PLAN, consult database for answer to Q and store result in PRIVATE.BEL

if Q on QUD and P in PRIVATE.BEL s.t. P resolves Q, answer(P)

S: "The price is £123"

QUD-based information state update

IS update rules take care of multiple issues, e.g.:

if user asks Q, push Q on QUD and load plan for dealing with Q

if users asks Q' while system is dealing with Q, **throw out plan for Q**, but Q remains on QUD

when Q' resolved, Q topmost on QUD will trigger reloading plan for dealing with Q

general rule: if SHARED.COM contains info resolving Q, don't ask Q so any resolved questions in plan will be thrown out

QUD-based information state update: QUD updates

U: I want price information [raise ?x.price(x)]

S: Where do you want to go?

U: London

S: When do you want to travel?

QUD=<?x.dept-month(x), ?x.price(x)>

U: Do I need a Visa? [raise ?visa]

QUD=<?visa, ?x.dept-month(x), ?x.price(x)>

S: Where are you travelling from?

U: Gothenburg

S: No, you don't need a Visa.

QUD=<?x.dept-month(x), ?x.price(x)>

PLAN empty;

QUD-based information state update: QUD and plan

(1)

U: OK.

```
QUD=<?x.dept-month(x), ?x.price(x)>
PLAN empty, so reload plan for dealing with ?x.price(x)
throw out all already resolved questions; raise the first unresolved question on plan
```

S: When do you want to travel? [= re-raise question]

U: I want to leave in April

S: What day do you want to leave?

(2)

U: OK, I want to leave in April [answers dept-month(april)]

```
QUD=<?x.price(x)>
PLAN empty, so reload plan for dealing with ?x.price(x)
throw out all already resolved questions; raise the first unresolved question on plan
```

S: What day do you want to leave?

...

June 2010

late2.10.nli.dm

61

information state update based systems

projects: Trindi, Siridus, D'Homme, BEETLE, WITAS, TALK, ...

software tools:

TrindiKit (Gothenburg U.) <http://www.ling.gu.se/trindi/trindikit/>

Dipper (U. of Edinburgh) <http://www.ltg.ed.ac.uk/dipper/>

MIDIKI (MITRE Corp.) <http://midiki.sourceforge.net/>

June 2010

late2.10.nli.dm

62

information state update based systems: summary

advantages:

- generic approach to dialogue modeling

- handling various dialogue phenomena

 - accommodation ("overanswering")

 - reraising of issues

 - task switching, sharing information across tasks

 - various dialogue genres (e.g., negotiation, tutoring...)

disadvantages

- static dialogue plans, not much work done on those

 - integrate with ideas in agent architecture where focus on task planning (current research)

June 2010

late2.10.nli.dm

63

Current Challenges

June 2010

late2.10.nli.dm

64

current challenges

adaptivity

systems need to be dynamically adaptive in a number of different ways:
to the environments in which they are used (modality), to their user's preferences and needs (personalisation), and to changes in task and context

ability to learn

systems need to be able to learn from interactions with users in order to provide an optimally usable interface that matches the current environment and user

standardisation

there is a need for a common set of standards to support re-usability for developers and to support usability for the users of spoken dialogue systems

pervasive systems

systems need to handle distributed dialogues (shifts to different dialogue situations / managers), concurrent dialogues (issues of co-ordination, synchronisation, redundancy);
interaction model needs to be predominantly event-based (external events, opportunistic)

references

- D. Jurafsky and J. Martin (2000): Speech and Language Processing, Chapter 19.
- McTear (2002): Spoken Dialogue technology. In ACM Surveys. pp. 1-80
- Traum and Larsson (2003): The Information State Approach to Dialogue Management. In Smith and Kuppevelt (2003), pp. 325-353.
- N. Blaylock, J. Allen, and G. Ferguson. (2003). Managing communicative intentions with collaborative problem solving. In Smith and Kuppevelt (2003), pages 63-84.
- J. van Kuppevelt and Ro.W. Smith, editors (2003). Current and New Directions in Discourse and Dialogue, volume 22 of Kluwer Series on Text, Speech and Language Technology. Kluwer Academic Publishers. Dordrecht, 2003
- D. Milward and M. Beveridge (2004) Ontologies and the Structure of Dialogue. In Proc. Of the Catalog workshop. pp.
- Malte Gabsdil and Oliver Lemon, " Combining Acoustic and Pragmatic Features to Predict Recognition Performance in Spoken Dialogue Systems" in Proceedings of ACL 2004.
- R. San-Segundo et al. (2001) Designing Confirmation Mechanisms and Error Recover Techniques in a Railway Information System for Spanish. SigDial Workshop.
- S. Ericsson and I. Lewin (2000). Dialogue Move Specifications for the Dialogue Move Engine. Siridus project deliverable D1.3.