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

Dialogue Modeling
as
Information State Update

Information State (IS)

- Representation of the current state of dialogue
- Used by system to
  - Interpret user's turn
  - Decide which external actions to take
  - Decide what to say
  - Store information (dialogue context representation)
- Utterances update information state
- Approaches to DM differ in how IS is represented, what role it plays, what it contains

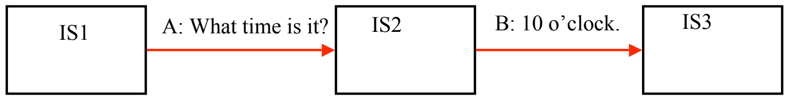
ISU-Based Theories

- Any theory following the ISU aproach should define:
  - A descripition of the informational components of the IS and their formal representation
  - A set of dialogue moves triggering the update of the IS
  - A set of update rules governing the IS updating
  - A control strategy to decide which update rule(s) to select at a given point in the dialogue

IS Update Rules

- Describe possible transitions from one information state to the next

  If <conditions-on-IS-values>
  then <changes-to-IS-values>



- When applicable
- What IS change

FS as ISU

- IS: current-state; input
- Update rules:
  - If [state] & [input]
    then [output]; [next-state]
- Example for elevator or account-balance

Frame-Based as ISU

- 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
- Example for travel dialogue

ISU-Based Dialogue Modeling

- Task- vs. Dialogue-Structure
  - Task --> dialogue
  - But, dialogue does not have to follow task (execution) structure
- "Dialogue planning" (agenda)
  - Task model fills agenda with task-related goals
  - Dialogue manager can add more goals, e.g., for grounding
- Some approaches:
  - QUD-based (Godis)
  - Obligation-based (Edis)
  - Agent-based: collaborative problem solving (TALK)

QUD-Based ISU Modeling

- 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 ISU Modeling

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 ISU Modeling

- Sample dialogue 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 ISU Modeling

- 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 an answer can be integrated by the system, it must be matched to a question on QUD

QUD-Based ISU Modeling

…

S: ”what class did you have in mind?”

U: ”cheap”

- if consultDB(Q) on PLAN, consult database for answer to Q; 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 ISU Modeling

- Dealing with multiple issues:
  - 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 ISU Modeling

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-date(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 ISU Modeling

(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 questions which have already been resolved; raise the first unresolved question on plan
S: When do you want to travel? [= question re-raising]
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 questions which have already been resolved; raise the first unresolved question on plan
S: What day do you want to leave?
...

ISU-Based DM

- A range of systems developed in various 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.sourgeforge.net/
  - Rubin (CLT, Saarbruecken)

QUD-Based ISU Modeling

- 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-based, where focus on task planning = current research

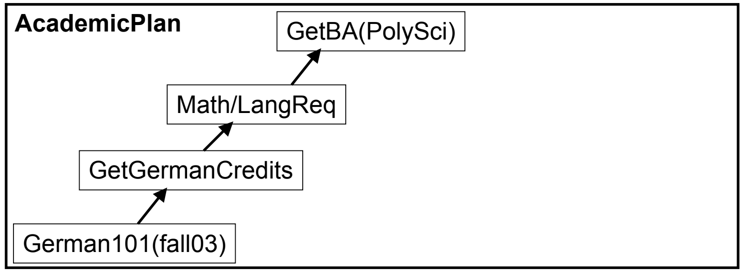
Agent-Based DM

- Communication is a joint activity: Agents communicate to establish common ground
- Collaborative problem solving by (rational) agents
  - Neither agent can accomplish the task alone
  - Need joint goals and mutual understanding
  - Agents collaborate to establish and achieve their goals
- Agents have knowledge about solving tasks
  - deciding on goals (objectives): adopt, select, defer, abandon, release
  - forming plans to achieve goals (recipes)
  - executing those plans (acting)
  - revising decisions (replanning, abandoning goals, etc.)
- Agents reason about beliefs and actions
- Intention recognition

Intention Recognition

Given: plan for getting a BA

U: I'll take German 101 fall semester.



Collaborative Planning&Acting

User: Send ambulance one to Parma right away.
(initiate (c-adopt (action (send amb1 Parma))))
(initiate (c-select (action (send amb1 Parma))))
System: OK. [sends ambulance]
(complete (c-adopt (action (send amb1 Parma))))
(complete (c-select (action (send amb1 Parma))))
System: Where should we take the victim once we pick them up?
(initiate (c-adopt (resource (hospital ?x))))
User: Rochester General Hospital.
(continue (c-adopt (resource (hospital RocGen))))
System: OK.
(complete (c-adopt (resource (hospital RocGen))))

[Blaylock et al. 2003]

Agent-based

- Advantages
  - Flexibility and adaptivity
  - Any task can be modeled
- Disadvantages
  - Intention recognition
  - Lots of reasoning

  --> see QUD-based for "shortcuts"

Current Challenges

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. Pre-print version: http://people.ict.usc.edu/~traum/Papers/traumlarsson.pdf
- N. Blaylock, J. Allen, and G. Ferguson. (2003). Managing communicative intentions with collaborative problem solving. In Smith and Kuppevelt (2003), pages 63-84. http://www.coli.uni-saarland.de/~blaylock/Pubs/Files/sigdialbook03.pdf
- 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.