UNIVERSITÄT
DES
SAARLANDES

Multimodal Dialogue Systems

Volha (Olga) Petukhova

Spoken Language Systems Group
Saarland University

Summer Semester 2020

Outline

Introduction

Dialogue Systems | Examples

Dialogue System Architecture

Components | Tasks

Dialogue Management

Script-based | Frame-based | Plan-based | Information State Update | Agent-based | Statistical DS | End2End DS | ChatBots

Development toolkits

CSLU | LUIS | Virtual Human | OpenDial | ….

Introduction

Multimodal natural-language based dialogue as human-machine interface

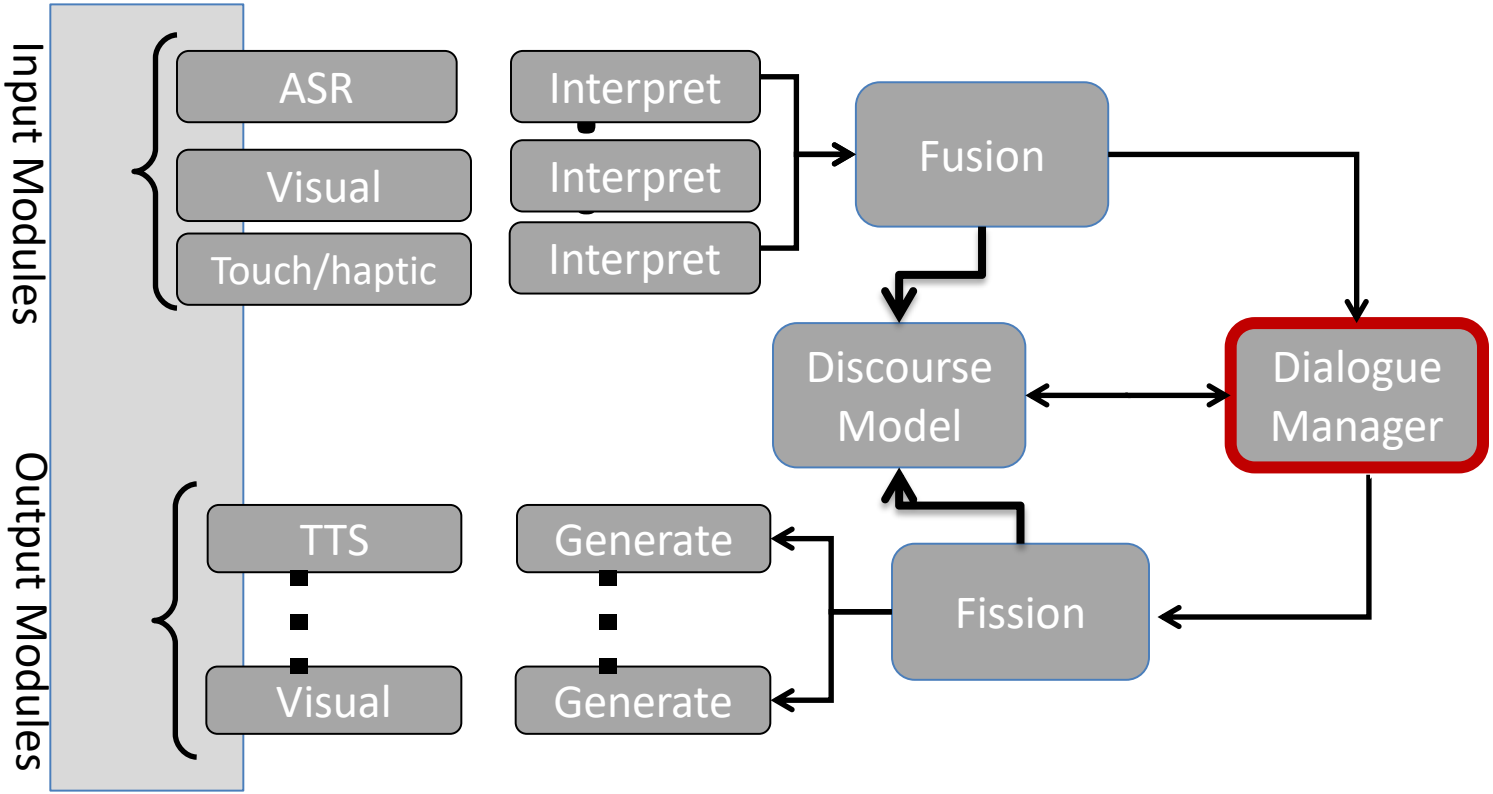
Examples ☺

- https://www.youtube.com/watch?v=zIFMq5IWVjI
- https://www.youtube.com/watch?v=t7Krn-DH3tw

Non-verbal behaviour

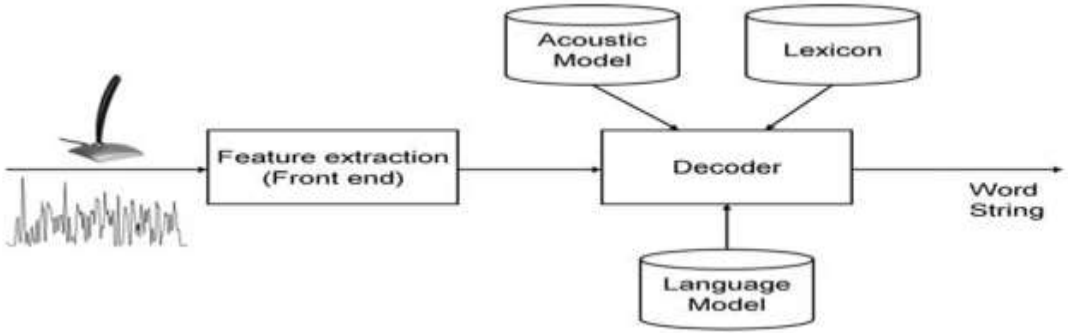
- https://www.youtube.com/watch?v=YZizCoOctPo
- https://www.youtube.com/watch?v=1X1vNlIf0xY

Dialogue Systems: general architecture



Dialogue Systems: Automatic speech recognition

- Nuance
- Sphinx (http://cmusphinx.sourceforge.net/)
- Kaldi (http://kaldi.sourceforge.net)
- Google API (http



Dialogue Systems: modern sensors

Kinect tracking

SMI eye-tracking glasses

Google glass
MS HoloLens

Biometrical sensors: MYO,
Nexus EXG

Intel RealSense technology

Dialogue Systems: multimodal interactive behaviour interpretation

- Verbal input: natural language understanding (NLU)
- Non-verbal input
- Fusion

Dialogue Systems: multimodal dialogue acts

- **Feedback acts** (68.5%): positive (65.3%), negative (3.2%)
- **Time Management** (24.8%)
- **Turn Management** (4.7%)
- **Discourse Structuring** (2%)

Dialogue Systems: roles of non-verbal signals

Articulating semantic content (about 39%):

They are relating to the propositional or referential meaning of an utterance

For example deictic gestures:

wording: *Press this little presentation*

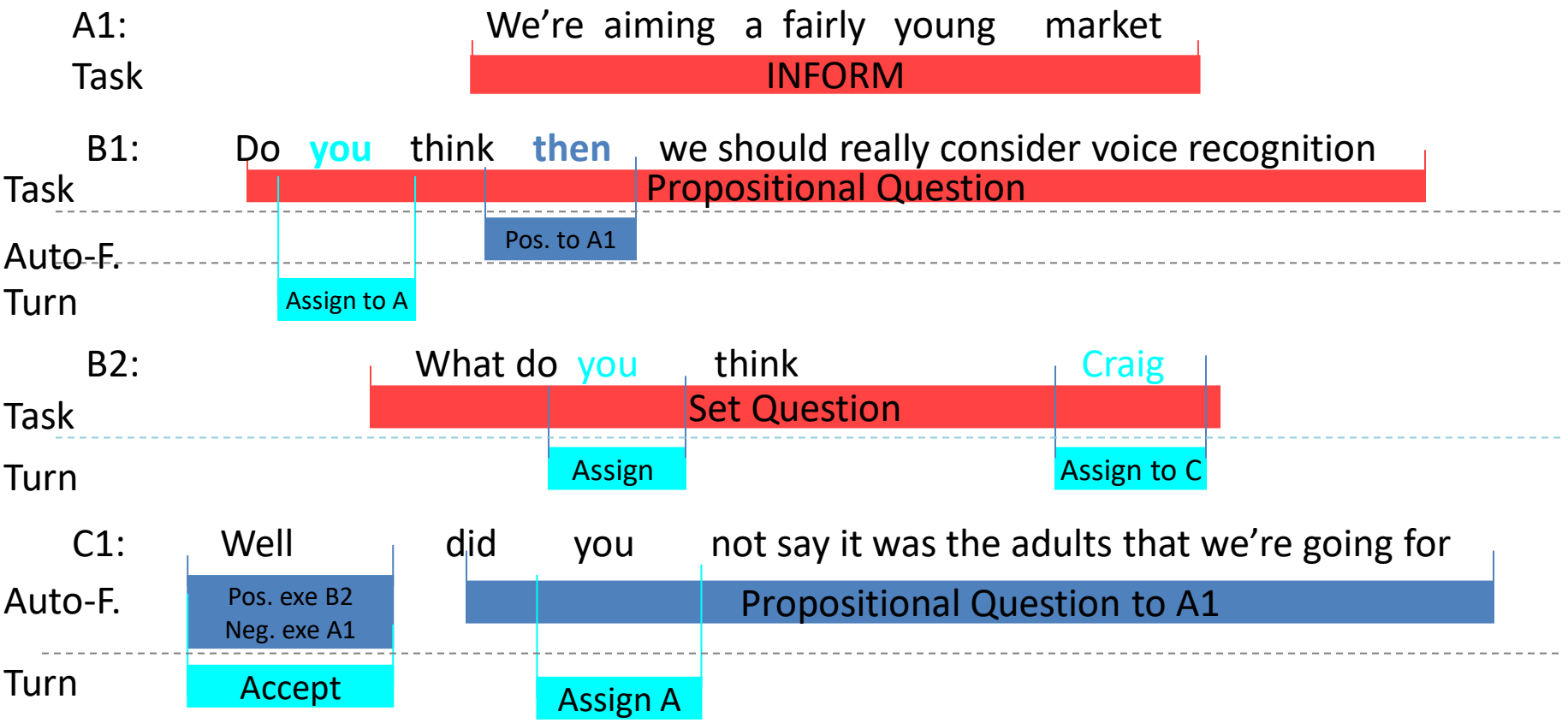
hand: .....*point*.....

pure semantic acts, as a rule do not have a communicative function on their own

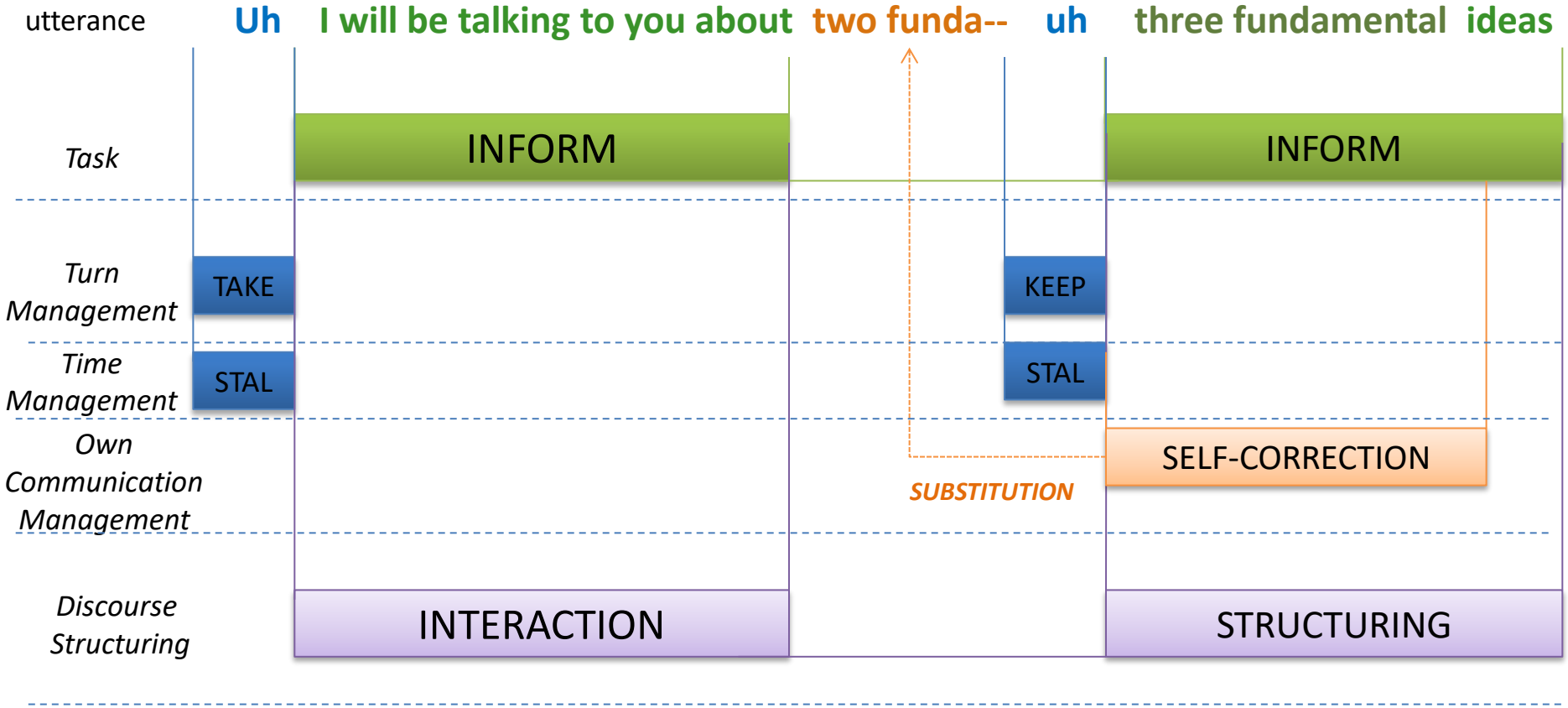
Dialogue Systems: roles of non-verbal signals

- adjustment of the level of feedback (understanding vs agreement)
- express degree of certainty about the validity of the proposition
- reveal speaker's attitude towards the addressee(-s), towards the content of what he is saying, or towards the actions he is considering to perform
- signal speaker's emotional or cognitive state (Pavelin (2002): *modalizers*)

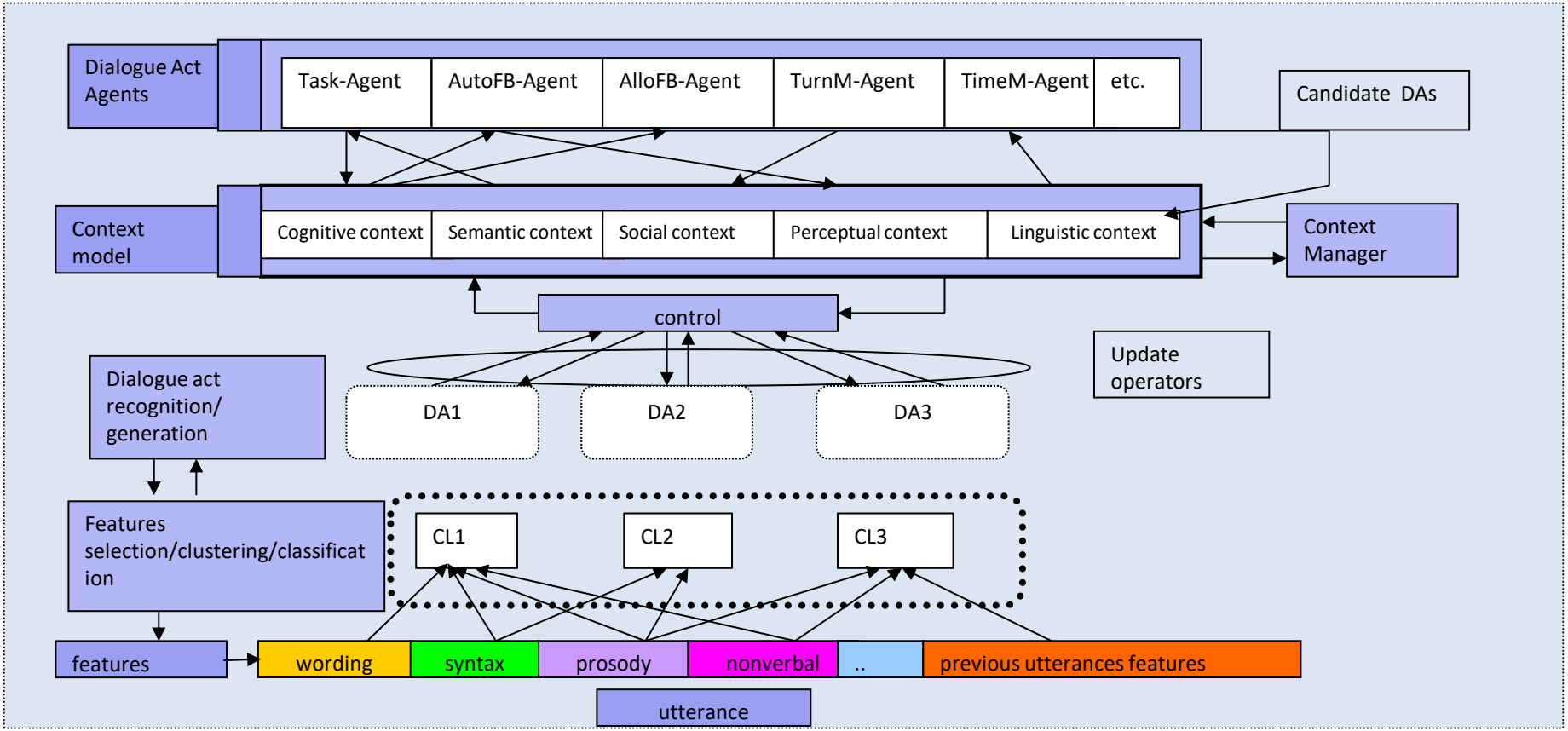
Dialogue Systems: segmentation



Dialogue Systems: disfluences



Dialogue Systems: processing flow



Tasks of Dialogue Management

Dialogue flow control

Dialogue modeling

➔Dialogue context

➔Dialogue acts

Dialogue act decision making

Dialogue phenomena:

- Error handling
- Initiative and cooperation
- Adaptivity
- – …

Dialogue Modelling: approaches

Script-based (state machines)
Sequence of pre-defined steps (dialogue script)

Frame-based (also: form-filling)
Set of slots to be filled (task template) and corresponding prompts

Plan-based
Collaborative problem solving

Information-State Update
Declarative rules for updating dialogue context

Statistical (PO)MDP-based models
Probability distribution of the events or user states observed/learned from the observed past

End-to-End models
sequence2sequence models learned from large amount of data

Script-based DM

- Script describes all possible dialogues
- Typically finite state machine
- Set of states and transitions
  - State determines system utterance
  - User utterance determines transition to next state (deterministic)
- No recursion! (= no nested sub-dialogues)
- Fixed dialogue script
- OK for system-driven interaction

Finite State Machine

<States, Init-State, Alphabet, Transition-function>

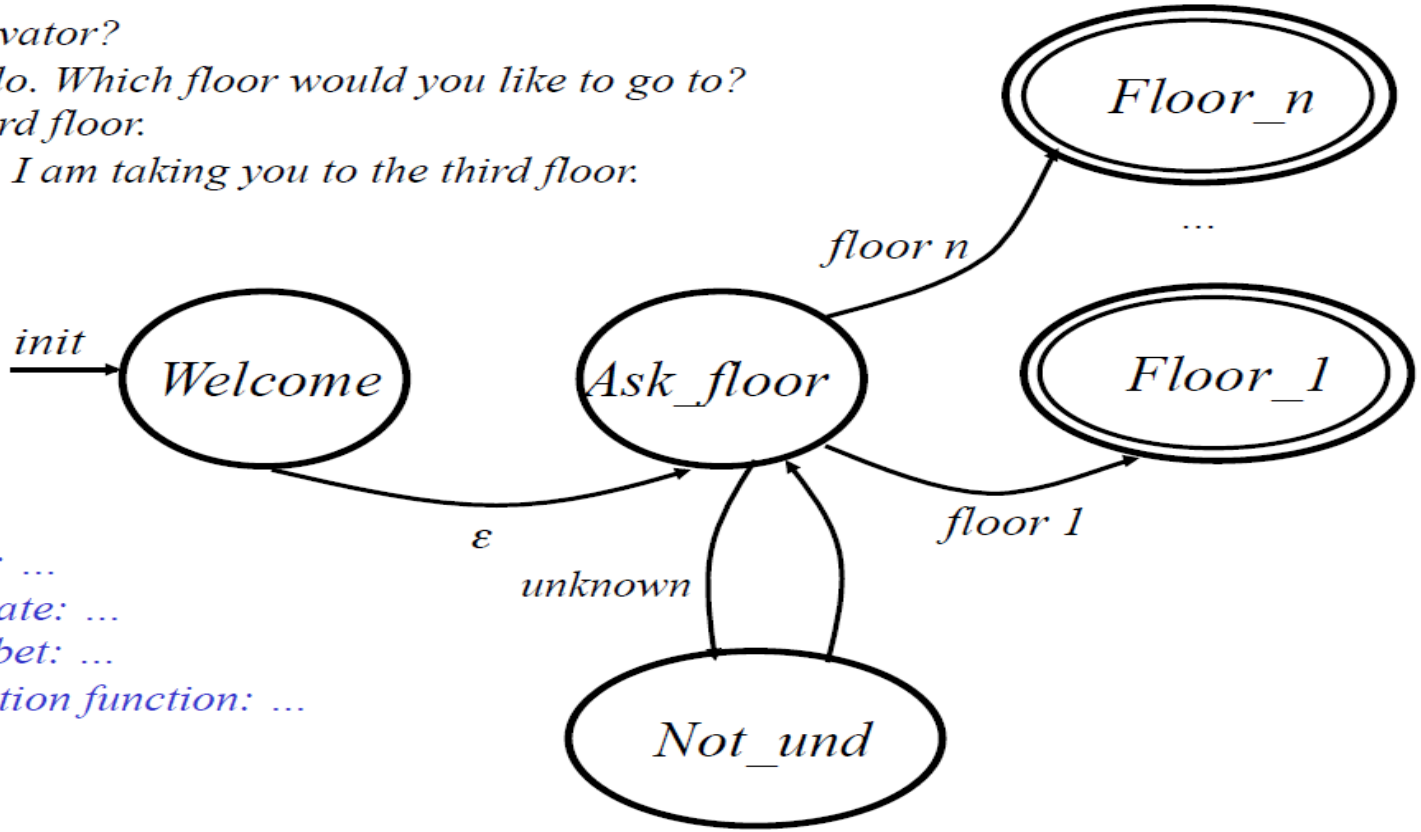
Variants: machines having

- 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

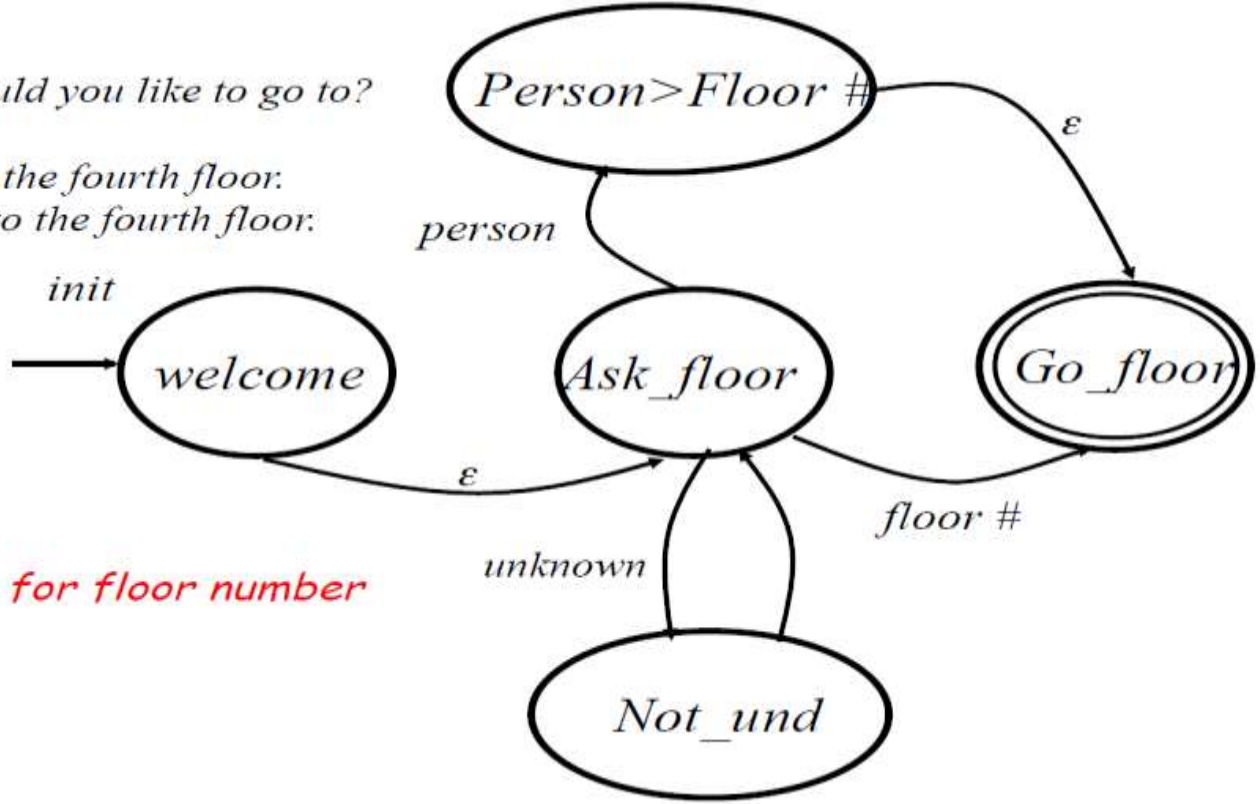
U: Elevator?
S: Hello. Which floor would you like to go to?
U: Third floor.
S: OK, I am taking you to the third floor.



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

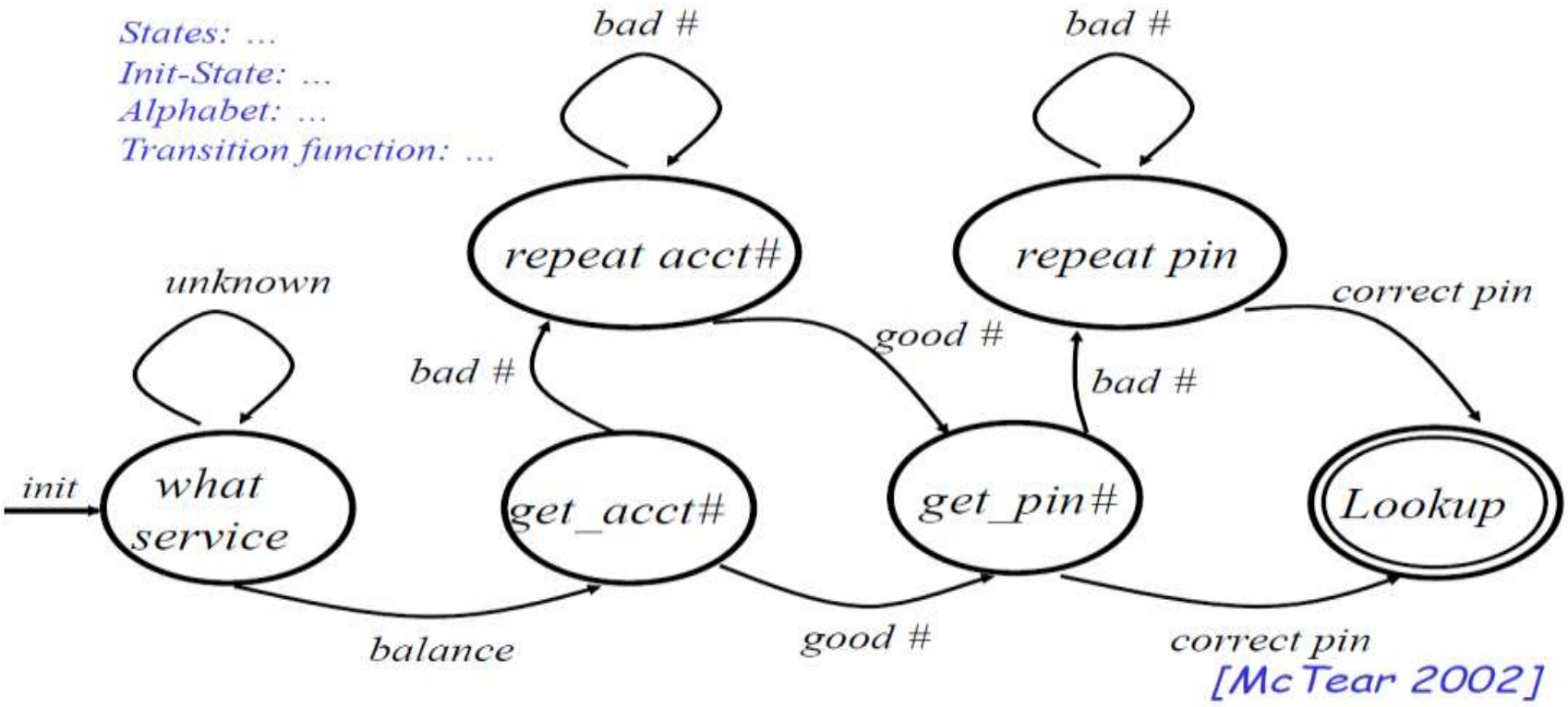
FSM-based Models

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.

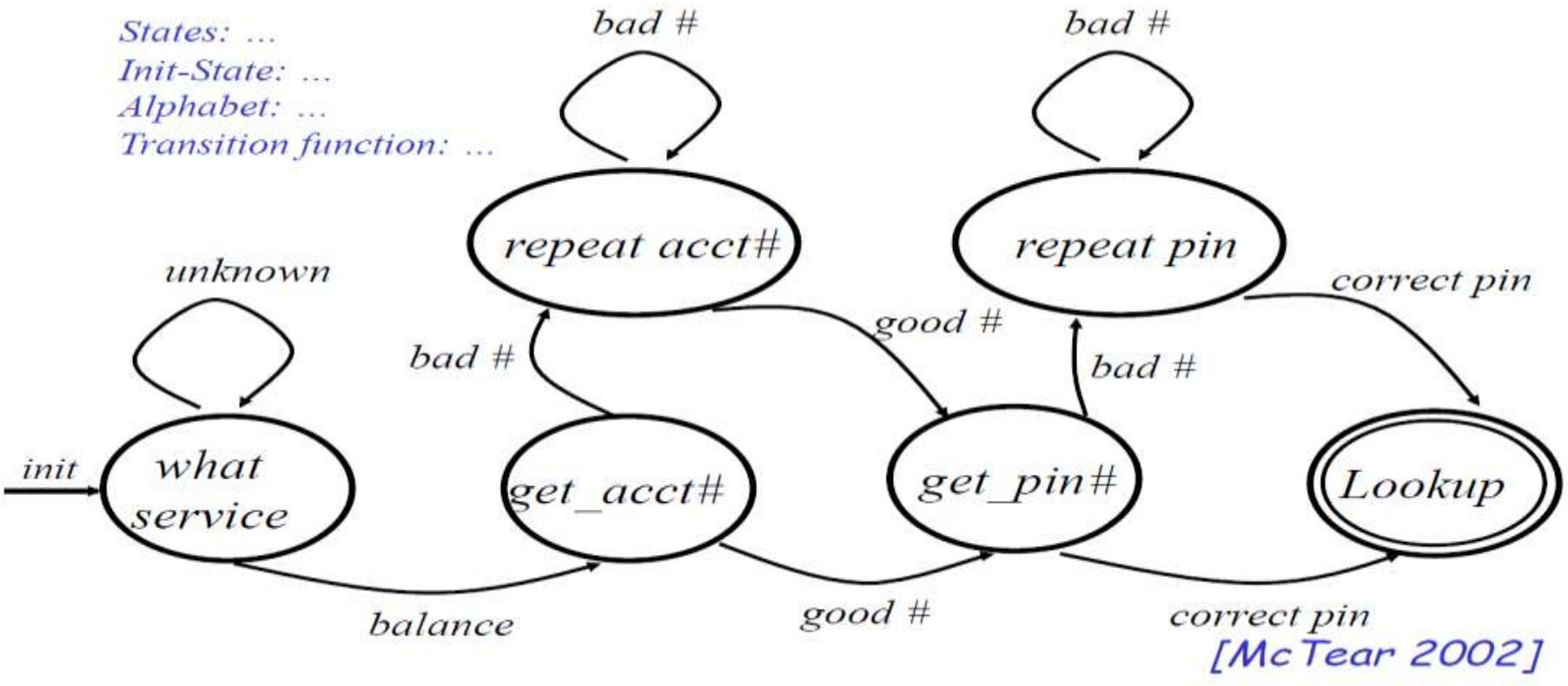


Extension: variable for floor number

FSM-based Models



FSM-based Models



FSM-based Models: sum up

- 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 can make more flexible, but it quickly gets very complex; modular solutions are possible

Frame-Based DM (form filling)

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: Where do you want to go?
U: Paris

S: Where will you travel from?
U: From Berlin.

S: When will you travel?
U: August 1st.

departure_city	*?*
departure_date	*?*
destination_city	*Paris*
return_date	*?*
...	

...

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

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"

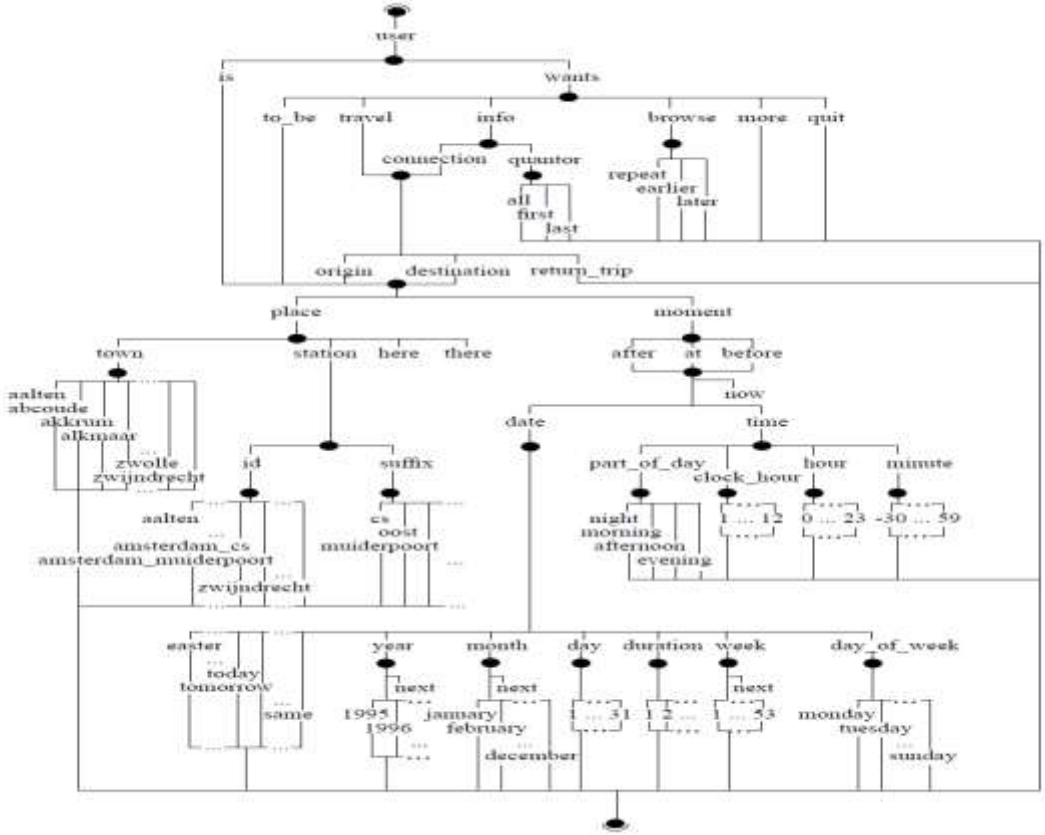
departure_city	*Berlin*
departure_date	*1/8/05*
destination_city	*Paris*
return_date	*?*
...	

Frame-Based Models

- Strategies for deciding what to do next
  - Next unfilled slot
  - Slot-combination weighting
  - Ontology-based coherence

- Options for database lookup
  - Delayed (typically; after certain slots filled)
  - Immediate (can be “expensive” = take time, but enables more helpful system behavior)

Frame-Based system: example (OVIS system, Aust et al., 1994)



Frame-Based Models: 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

Plan-based Models

- Communication is a joint activity: participants communicate to establish common ground, participants collaborate to accomplish a task
- 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
  - Decide on goals (objectives): adopt, select, defer, abandon, release
  - Form plans to achieve goals (recipes)

Plan-based Models

Automated planning: STRIPS; planning operators: actions, reconditions, post-conditions

- Executing plans (acting)
- Revising decisions (re-planning, abandoning goals, etc.)

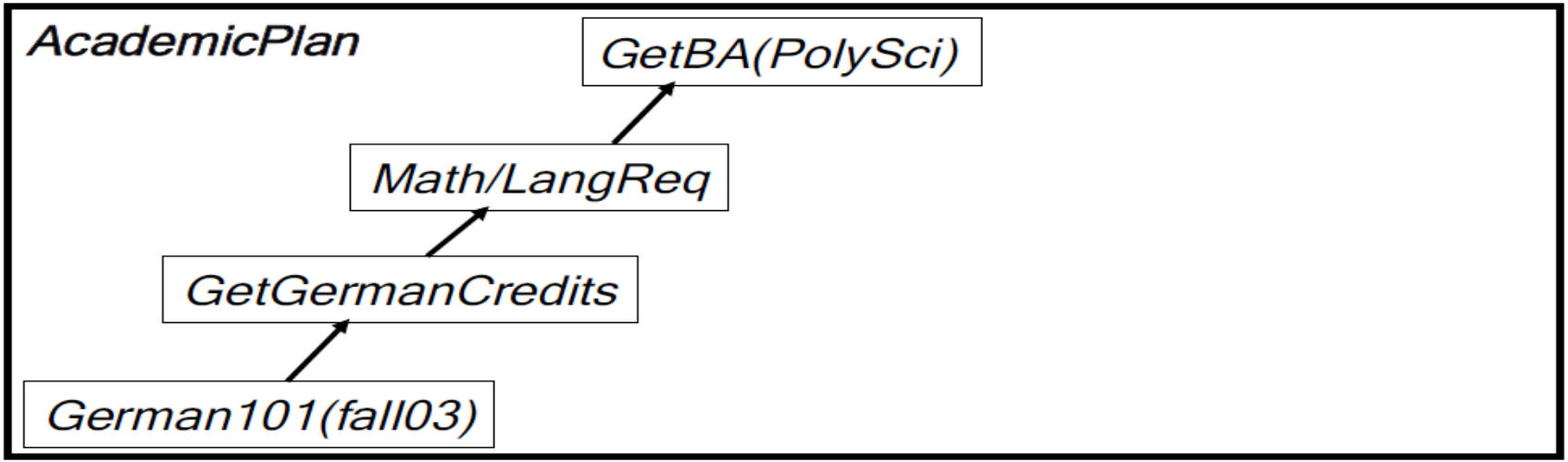
Agents reason about beliefs and actions

Intention recognition

Plan 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))))

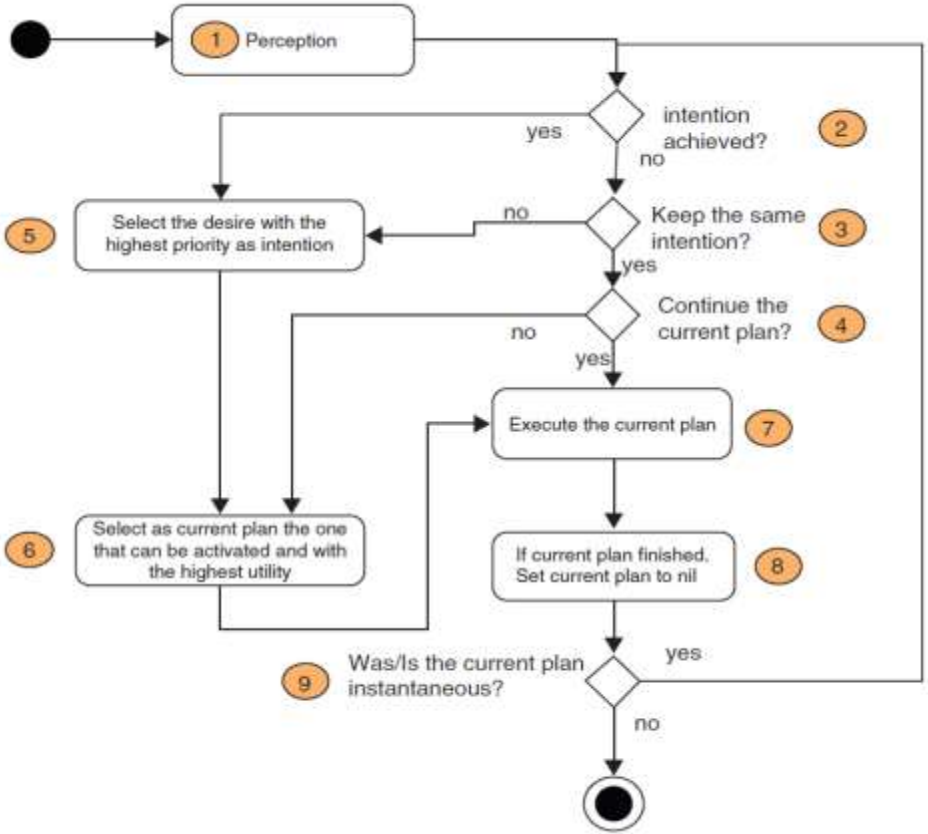
System: Where should we take the victim once we pick them up?
(initiate (c-adopt (resource (hospital ?x))))

User: Rochester General Hospital
(continue (c-adopt (resource (hospital RocGen))))

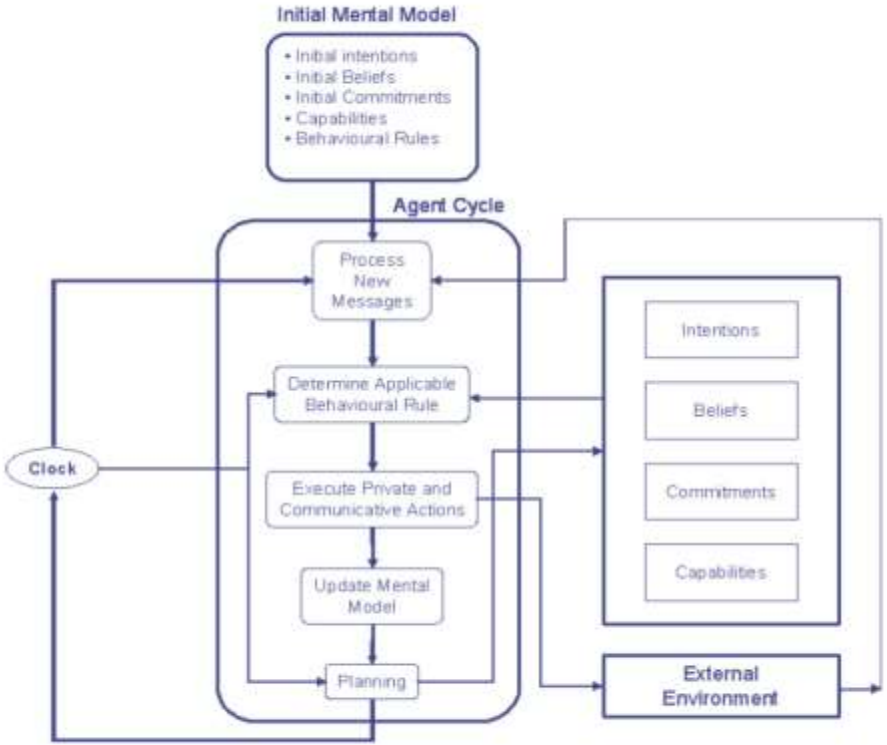
System: OK
(complete (c-adopt (resource (hospital RocGen))))

[Blaylock et al. 2003]

Activity diagram (Caillou et al., 2015)



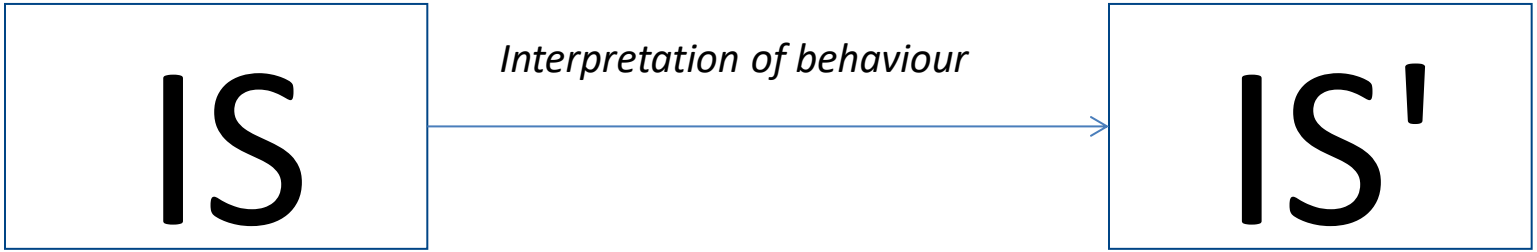
The BDI agent model (Allen and Perrault, 1980)



Plan-Based DM: sum up

- Advantages
  - Flexibility and adaptivity
  - Any task can be modeled
  - … the ultimate solution
- Disadvantages
  - Specifying planning operators is as hard as writing dialogue scripts
  - Plan recognition is a hard problem
  - Lots of reasoning needed

Information State Update



Information State

- Representation of the current state of dialogue
- Used by system to
  - Interpret user's contribution
  - Decide which 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 Dialogue Modelling

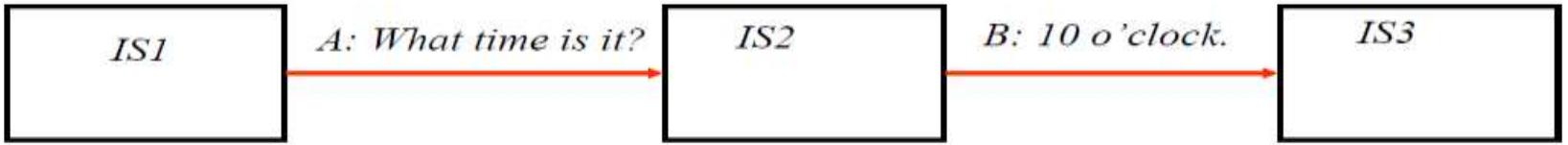
Components:

- a description of the information state 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 acts (DAs) triggering the update of the IS
- set of update rules governing the IS updates given various conditions of current IS and performed DAs (e.g. set of selection rules that license 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)

IS Update Rules

Describe possible transitions from one information state to the next

If <conditions-on-IS-values>



IS Update Rules: example (Traum and Larsson,2003)

(1) U-Rule: **SelectAskCity**
PRE: fst (private.Agenda, raise(destCity(X))
EFF: set (NEXT-MOVE, ask(destCity(X))

(2) U-Rule: **IntegrateSysAskCity**
PRE: { val (shared.LM, ask(destCity(X))), fst (private.Agenda, raise(destCity(X)))
EFF: { pop (private.Agenda), push (shared.QUD, destCity(X))

(3) U-Rule: **IntegrateUsrAnswerCity**
PRE: { val (shared.LM, answer(destCity(X))), fst (shared.QUD, destCity(X))
EFF: add (shared.BEL, destCity(X))

(4) U-Rule: **DowndateQUDCity**
PRE: { fst (shared.QUD, destCity(X)), in (shared.BEL, destCity(X))
EFF: pop (shared.QUD)

(a) The update rules.

IS: [private: [BEL: { } Agenda: <raise(destCity(X)), ... >] shared: [BEL: { } QUD: < > LM: . . .]]

U-Rule: **SelectAskCity**
EFF: set (NEXT-MOVE, ask(destCity(X))

System: Where do you want to go?

U-Rule: **IntegrateSysAskCity**
EFF: { pop (private.Agenda) push (shared.QUD, destCity(X))

IS: [private: [BEL: { } Agenda: < ... >] shared: [BEL: { } QUD: < destCity(X) > LM: ask(destCity(X))]]

User: Berlin

U-Rule: **IntegrateUsrAnswerCity**
EFF: add (shared.BEL, destCity(Berlin))

U-Rule: **DowndateQUDCity**
EFF: pop (shared.QUD)

IS: [private: [BEL: { } Agenda: < ... >] shared: [BEL: {destCity(Berlin)} QUD: < > LM: answer(destCity(Berlin))]]

(b) The example dialogue.

ISU: belief transfer

S: It's raining outside

preconditions:

Bel(S; p)
Want(S; Bel(A; p))

expected understanding:

Bel(S,MBel({S,A},WBel(S,Bel(A;Bel(S, p)))))
Bel(S,MBel({S,A},WBel(S;Bel(A;Want(S;Bel(A; p))))))

expected adoption:

Bel(S,MBel({S,A},WBel(S,Bel(A, p))))

ISU: belief transfer

A: no, it isn't

understanding:

Bel(A, MBel({S,A}, Bel(S,p)))
Bel(A, MBel({S,A}, Want(S, Bel(A, p))))

adoption:
~~Bel(A, MBel({S,A}, Bel(A,p)))~~

preconditions:

Bel(A, ¬p)
Want(A, Bel(S, Bel(A, ¬p)))

expected understanding:
…
expected adoption:
…

ISU: belief transfer

A: yes, it is

understanding:

Bel(A, MBel({S,A}, Bel(S,p)))
Bel(A, MBel({S,A}, Want(S, Bel(A, p))))

adoption:
Bel(A, MBel({S,A}, Bel(A,p)))

preconditions:

Bel(A, p)
Want(A, Bel(S, Bel(A, p)))

expected understanding:
…
expected adoption:
…

State Machine Model as ISU

- IS: current-state; input
- Update rules:

  If [state] & [input]

  then [output]; [next-state]

Frame-Based Model 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>

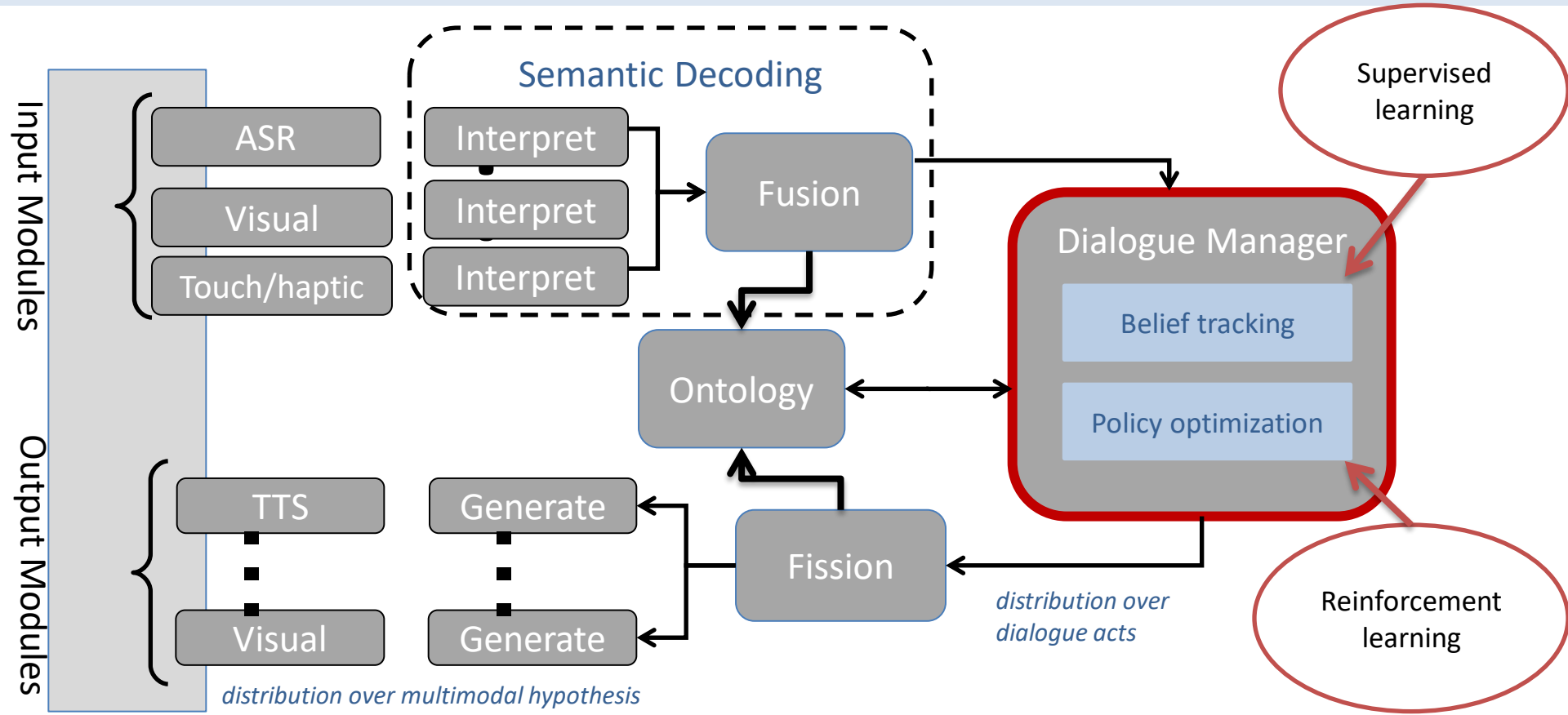
  then <ask-slot-value Y>

Decision about next system move is also a rule

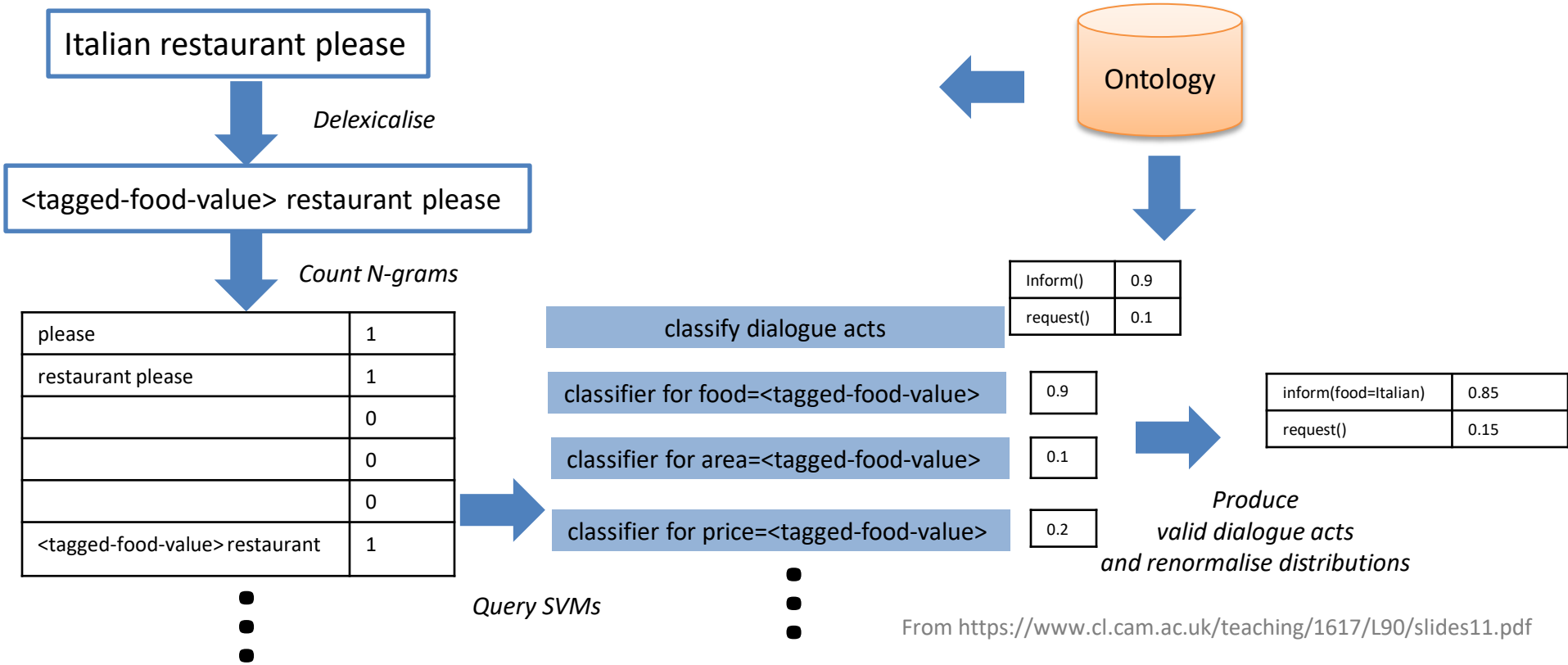
ISU-based Dialogue Modelling

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

Dialogue Systems: statistical DM



Statistical semantic encoding



From https://www.cl.cam.ac.uk/teaching/1617/L90/slides11.pdf

Statistical models: state change

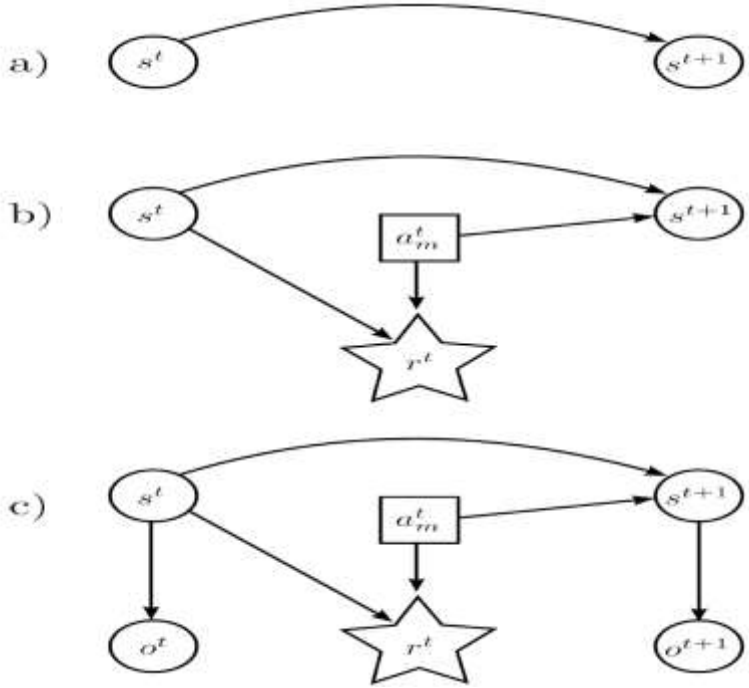
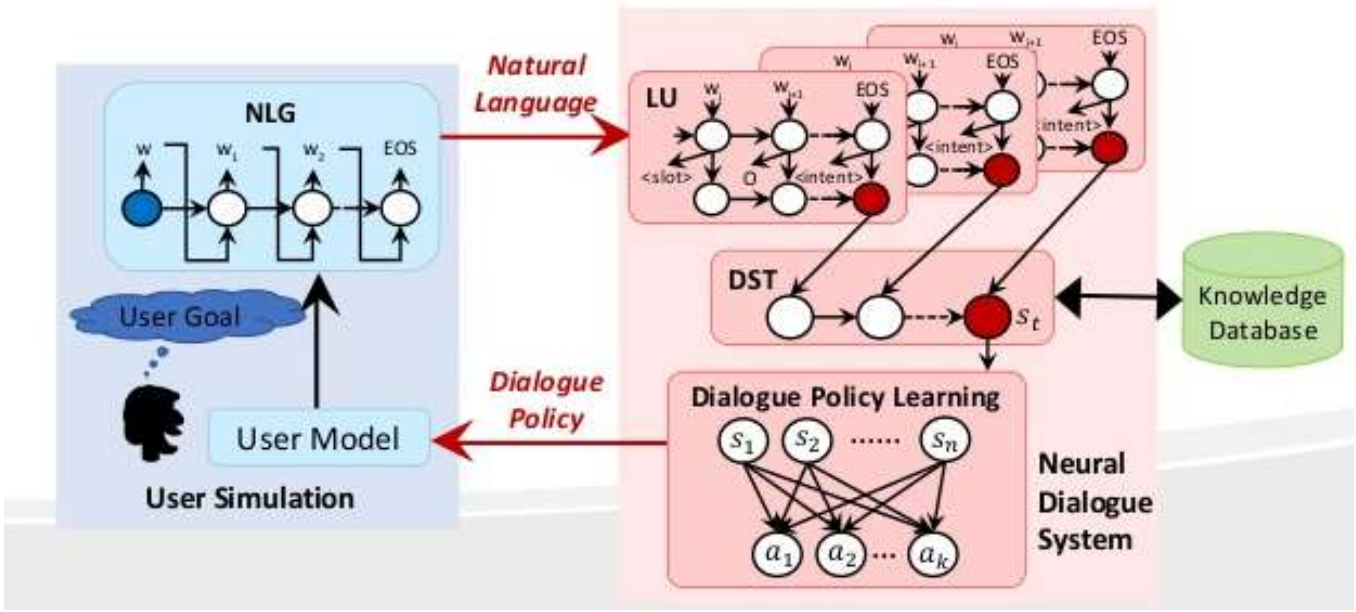


Diagram of state changes for different models. a) is a Markov chain, where *st* denotes the state at time ***t***; b) is a **MDP**, where at ***m*** is the system action and ***rt*** the reward at time ***t***; c) is a **POMDP**, where ***ot*** denotes the observation at time ***t***.

End-to-end dialogue systems



From https://www.slideshare.net/YunNungVivianChen/endtoend-taskcompletion-neural-dialogue-systems

End-to-end multimodal dialogue systems

Enhance end-to-end DS architectures with image input

- no video input
- no avatars, facial expressions, gestures etc.
- not much graphics output either

Using off-the-shelf components

- especially for image recognition –ready-made convolutional architectures
- textual parts based on known architectures (HRED, MemNN etc.)

Mostly just end-to-end prediction

- pretrained image recognition parts are kept fixed, no end-to-end training

Caption: A man and woman on bicycles are looking at a map
Person A (1): where are they located
Person B (1): in city
Person A (2): are they on road
Person B (2): sidewalk next to 1
Person A (3): any vehicles
Person B (3): 1 in background
Person A (4): any other people
Person B (4): no
Person A (5): what color bikes
Person B (5): 1 silver and 1 yellow
Person A (6): do they look old or new
Person B (6): new bikes
Person A (7): any buildings
Person B (7): yes
Person A (8): what color
Person B (8): brick
Person A (9): are they tal
Person B (9): i can't see
Person A (10): do they lo
Person B (10): they are

Caption:
A sink and toilet in a small room.

Q3: can you see anything else ?
A3: there is a shelf with items on it
Q4: is anyone in the room ?
A4: nobody is in the room
Q5: can you see on the outside ?
A5: no, it is only inside
Q6: what color is the sink ?
A6: the sink is white
Q7: is the room clean ?
A7: it is very clean
Q8: is the toilet facing the sink ?
A8: yes the toilet is facing the sink
Q9: can you see a door ?
A9: yes, I can see the door
Q10 what color is the door ?
A10 the door is tan colored

Visual dialogue & Visual Question Answering

VQA

Q: How many people on wheelchairs ?

A: Two

Q: How many wheelchairs ?

A: One

Captioning

Two people are in a wheelchair and one is holding a racket.

Visual Dialog

Q: How many people are on wheelchairs ?

A: Two

Q: What are their genders ?

A: One male and one female

Q: Which one is holding a racket ?

A: The woman

Visual Dialog

Q: What is the gender of the one in the white shirt ?

A: She is a woman

Q: What is she doing ?

A: Playing a Wii game

Q: Is that a man to her right

A: No, it's a woman

From http://www.cs.toronto.edu/~fidler/slides/2017/CSC2539/Sayyed_slides.pdf

ChatBots

A ChatBot is a conversational agent that interacts with users using natural language

- First chatbot - ELIZA (Weizenbaum 1966), which emulated a psychotherapist: http://nlp-addiction.com/eliza/
- ALICE is a chatbot (**Artificial Intelligence Mark up Language)**: **ALICE System** http://www.alicebot.org/about.html
- Machine Learning based Chatbots, typically Sequence2Sequence learning
  - Requieres lots of data; Best datasets to train Chatbots: https://lionbridge.ai/datasets/15-best-chatbot-datasets-for-machine-learning/

DS authoring tools and development environments

DM approach	Example task	Toolkit/ Authoring environment
Finite state machines	Long-distance calling	CSLU toolkit [1]
Statecharts	Virtual receptionist	SCXML [2]; IrisTK [3]
Frame-based	Getting travel information	CMU Communicator [4]; VoiceXML [5]
Information State Update	Human-robot interaction	TRINDI [6]; Dipper [7]
Plan-based	Medical diagnosis	RavenClaw [8]
Agent-based	Collaborative planning and acting	ViewGen [9]
Probabilistic approaches	Car driving assistant	OpenDial [10]
	Various information-seeking tasks	Alex DSF [11]; PyDial [12]
Neural approaches	Negotiations	PyOpenDial [13]
Chat-oriented;	Retail `chat commerce'	AIML [14]
interactive pattern-matching	Personal assistant	Facebook: Botsify, Chatfuel, Chatsuite, etc.
information-retrieval techniques	Question-answering	NPCEditor [15]

DS authoring toolkits and development environments:

references

[1] Stephen Sutton and Ronald Cole. The CSLU toolkit: rapid prototyping of spoken language systems. In Proceedings of the 10th annual ACM symposium on User interface software and technology, pages 85–86. ACM, 1997.

[2] Jenny Brusk and Torbj¨orn Lager. Developing natural language enabled games in (Extended) SCXML. In Proceedings from the International Symposium on Intelligence Techniques in Computer Games and Simulations (Pre-GAMEON-ASIA and Pre-ASTEC), Shiga, Japan, March, pages 1–3, 2007.

[3] Gabriel Skantze and Samer Al Moubayed. IrisTKfs: a statechart-based toolkit for multi-party face-to-face interaction. In Proceedings of the 14th ACM international conference on Multimodal interaction, pages 69–76. ACM, 2012.

[4] Alexander I Rudnicky, Christina Bennett, Alan W Black, Ananlada Chotomongcol, Kevin Lenzo, Alice Oh, and Rita Singh. Task and domain specific modelling in the Carnegie Mellon Communicator system. Technical report, CMU, Pittsburgh PA School of Computer Science, 2000.

[5] VoicaXML. VoiceXML specification, 2007.

[6] Staffan Larsson and David Traum. Information state and dialogue management in the Trindi dialogue move engine toolkit. Natural Language Engineering, 6(3-4):323–340, 2000.

[7] Johan Bos, E. Klein, O. Lemon, and T. Oka. DIPPER: description and formalisation of an information-state update dialogue system architecture. In Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue, pages 115–124, 2003.

[8] Dan Bohus and Alexander I Rudnicky. The RavenClaw dialog management framework: Architecture and systems. Computer Speech & Language, 23(3):332–361, 2009.

DS authoring toolkits and development environments:

References (cont.)

[9] Afzal Ballim and Yorick Wilks. Beliefs, stereotypes and dynamic agent modeling. User Modeling and User-Adapted Interaction, 1(1):33–65, 1991.

[10] Pierre Lison and Casey Kennington. OpenDial: A toolkit for developing spoken dialogue systems with probabilistic rules. In Proceedings of ACL-2016 System Demonstrations, pages 67–72, 2016.

[11] Filip Jurcıcek, Ondrej Dusek, Ondrej Platek, and Lukas Zilka. Alex: A statistical dialogue systems framework. In International Conference on Text, Speech, and Dialogue, pages 587– 594. Springer, 2014.

[12] Stefan Ultes, Lina M Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Inigo Casanueva, Paweł Budzianowski, Nikola Mrksic, Tsung-Hsien Wen, Milica Gasic, et al. Pydial: A multi-domain statistical dialogue system toolkit. In Proceedings of ACL 2017, System Demonstrations, pages 73–78, 2017.

[13] Youngsoo Jang, Jongmin Lee, Jaeyoung Park, Kyeng-Hun Lee, Pierre Lison, and Kee-Eung Kim. PyOpenDial: A python-based domain-independent toolkit for developing spoken dialogue systems with probabilistic rules. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, pages 187–192, 2019.

[14] Richard Wallace. The elements of AIML style. Alice AI Foundation, 139, 2003.

[15] Anton Leuski and David Traum. NPCEditor: Creating virtual human dialogue using information retrieval techniques. Ai Magazine, 32(2):42–56, 2011.