Planning and Plan Execution for Human-Robot Interaction Tasks

Luca Iocchi

Sapienza University of Rome, Italy
iocchi@diag.uniroma1.it

Ron Petrick

Heriot-Watt University, UK
R.Petrick@hw.ac.uk

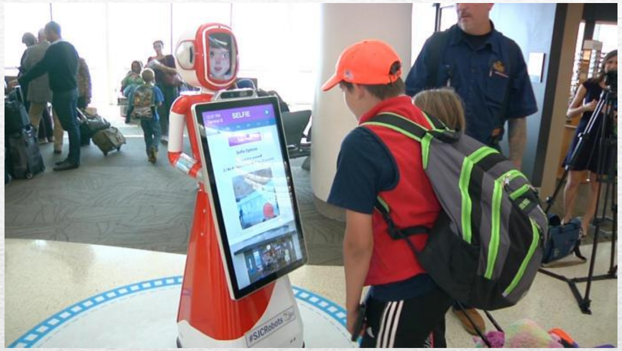
Outline

- Public space robots and human-robot interaction
- Cognitive social robots and planning
- Epistemic planning
  - PKS
  - $\mathcal{ALCK}_{\mathcal{NF}}$
- Plan execution
- Uncertainty in HRI planning

Robots in public spaces

https://youtu.be/vp0gkNOPryY

Robots in public spaces



Robots in public spaces

http://james-project.eu/

A task-based interaction

Two people, A and B, each individually approach a robot bartender.

Robot (to A): How can I help you?

Person A: A lemonade, please.

Person C approaches and attracts the attention of the robot by gesturing.

Robot (to C): Just a moment please.

Robot: (Serves A)

Robot (to B): What will you have?

Person B: A glass of water.

Robot: (Serves B)

Robot (to C): Thanks for waiting. How can I help?

Person C: I'd like a cola.

Robot: (Serves C)

What should the robot do next?

→

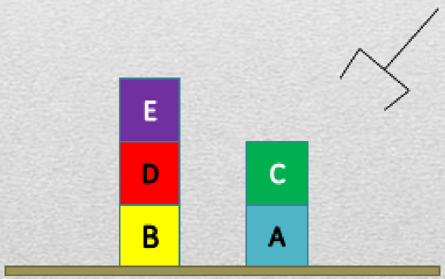
- Greet the customer
- Ask the customer for a drink
- Acknowledge the drink order
- Pickup the correct drink
- Serve the customer
- Close the transaction

Planning would be a good tool for this task….

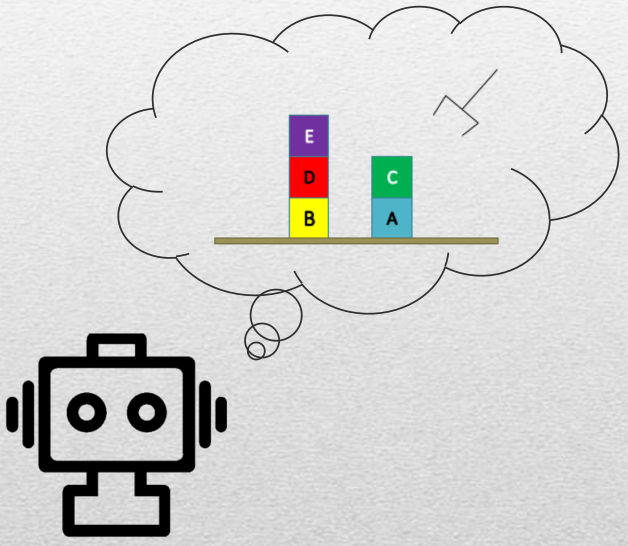
Classical Planning

Agent's knowledge about the problem is complete

- Full and perfect observations
- Full knowledge about transitions
- Deterministic effects
- Instantaneous actions (execution time does not matter)



World state



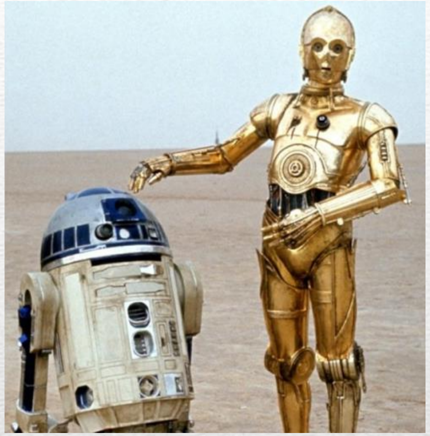
Agent state

Challenges for planning

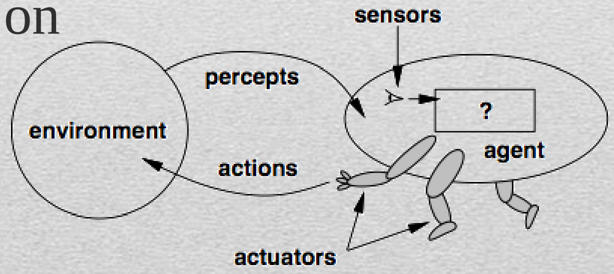
In human-robot interaction (HRI) tasks:

- Knowledge about the problem is **incomplete**
- Observations are **noisy**
- Action effects are **non-deterministic**
- Actions have **duration**
- Actions can **fail**
- Humans must be **explicitly modelled** but are often unpredictable (not just obstacles towards reaching the goal)

Cognitive social robots



- Interested in equipping **robots with high-level reasoning** capabilities about goals, actions, perception, collaboration
- With an **explicit representation of knowledge** about the environment, themselves, and other agents
- Where the **way in which a goal is achieved matters** and involves acting according to social norms, with a focus on acceptability



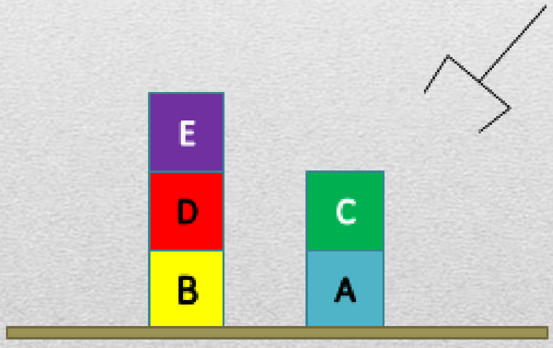
A task-based interaction

Two people, A and B, each individually approach a robot bartender.	
Robot (to A): How can I help you?	**Information gathering action**
Person A: A lemonade, please.	
Person C approaches and attracts the attention of the robot by gesturing.	
Robot (to C): Just a moment please.	**Social action**
Robot: (Serves A)	**Physical action**
Robot (to B): What will you have?	**Information gathering action**
Person B: A glass of water.	
Robot: (Serves B)	**Physical action**
Robot (to C): Thanks for waiting.	**Social action**
How can I help?	**Information gathering action**
Person C: I'd like a cola.	
Robot: (Serves C)	**Physical action**

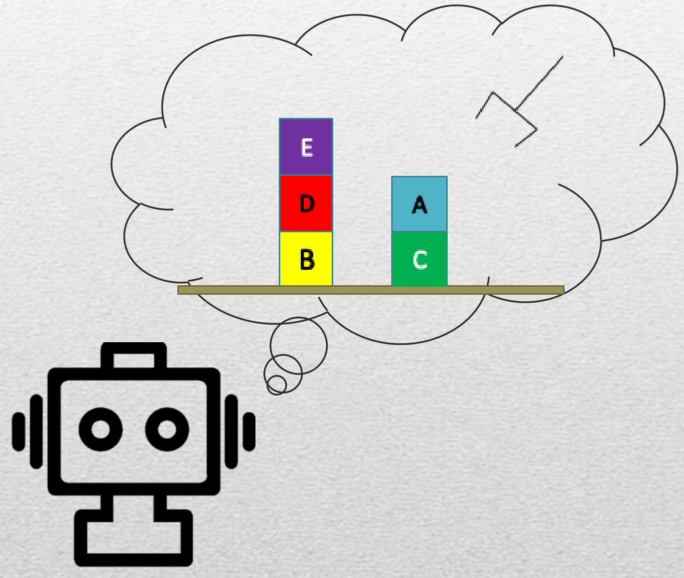
Epistemic state representation

Agent state representation potentially differs from actual world state

NOISE / ERRORS



World state

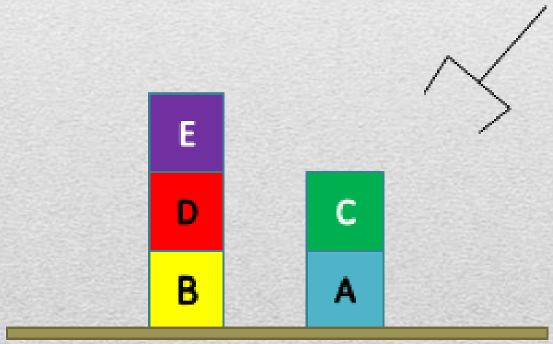


Agent state

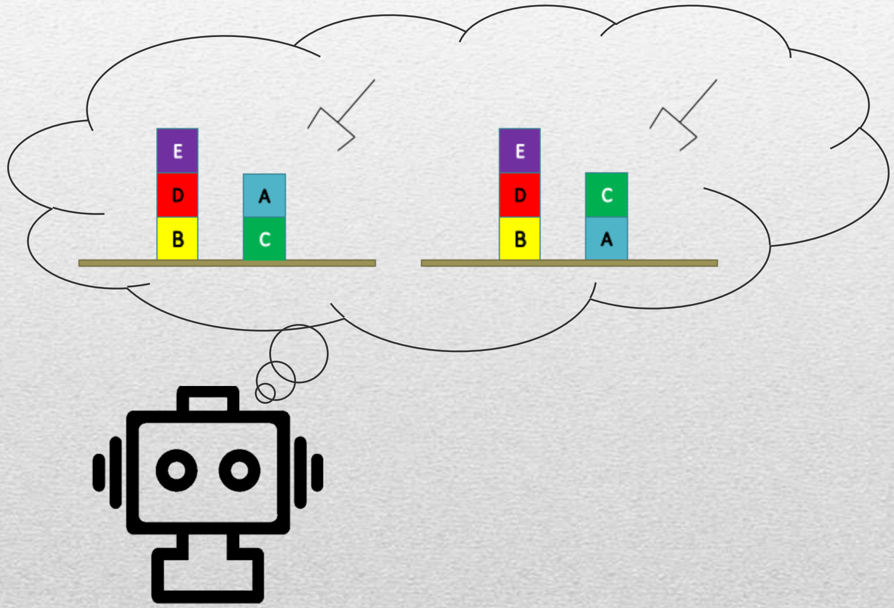
Epistemic state representation

Agent state representation potentially differs from actual world state

UNCERTAINTY



World state



Agent state

Epistemic Modal Logic

Logical frameworks to represent and reason about **knowledge** and **belief**

$$Ka$$

I know that a is true
a = false is impossible

$$K\neg a$$

I know that a is false
a = true is impossible

$$\neg Ka$$

I don't know that a is true
a=false is possible

$$\neg K\neg a$$

I don't know that a is false
a=true is possible

Epistemic Modal Logic

Logical frameworks to represent and reason about **knowledge** and **belief**

$$Ka \vee K\neg a$$

I know the value of a

$$\neg Ka \wedge \neg K\neg a$$

I don’t know the value of a

Epistemic Modal Logic

Model of a **SENSE** action (information gathering to know a)

Preconditions $\neg K a \wedge \neg K \neg a$

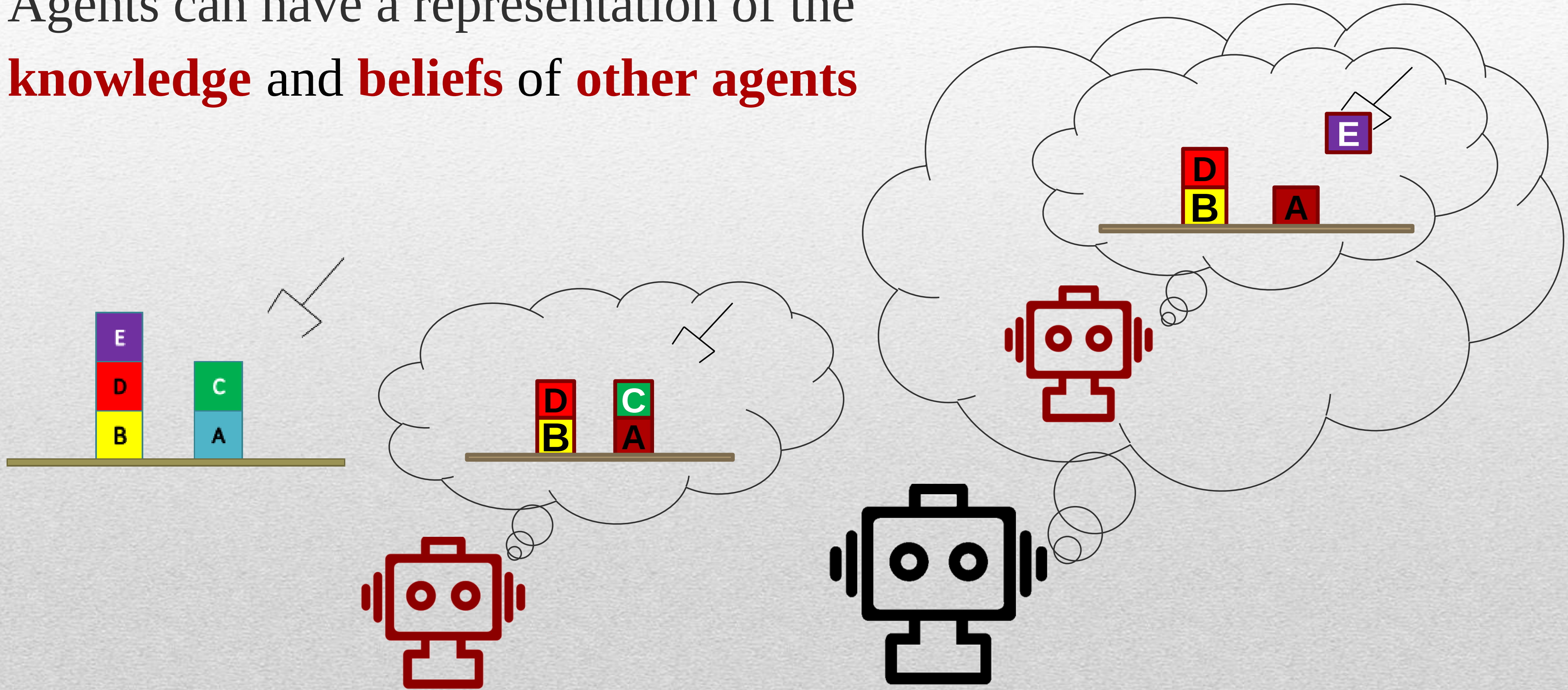
I don't know the value of a

Effects $K a \vee K \neg a$

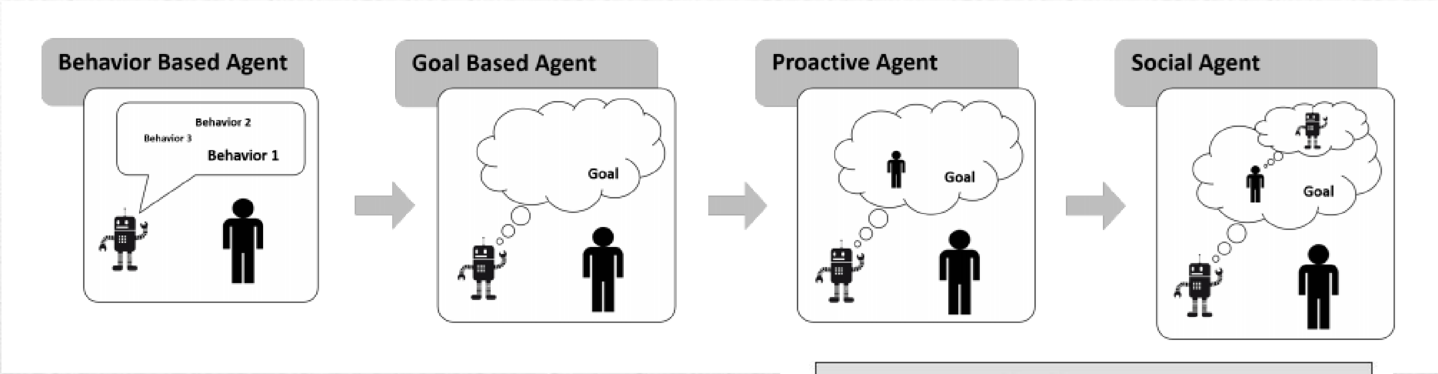
I know the value of a

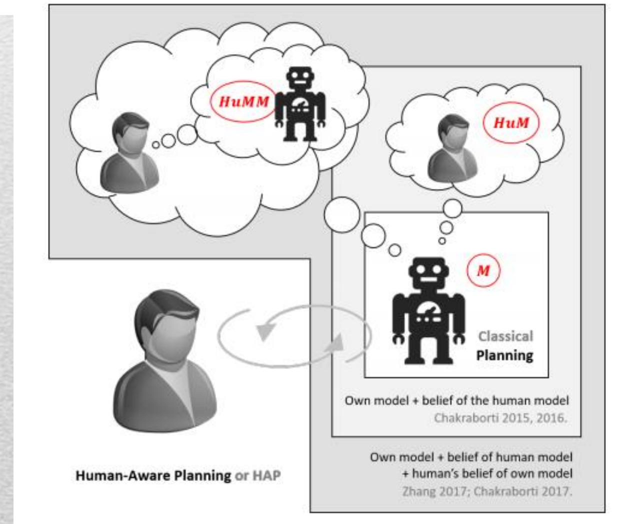
Epistemic state representation

Agents can have a representation of the **knowledge** and **beliefs** of **other agents**



Human models





- Explicit representation of Human Models and Human Mental Models

[Chakraborti et al. 2017]

Multi-Agent Epistemic Logic

Knowledge and **belief** shared among multiple agents

Example: two agents *H*, *R*

$$K_H a \vee K_H \neg a$$

H knows the value of a

$$\neg K_R a \wedge \neg K_R \neg a$$

R does not know the value of a

$$\neg K_H (K_R a \vee K_R \neg a)$$

H does not knows that R knows the value of a

$$K_H (\neg K_R a \wedge \neg K_R \neg a)$$

H knows that R does not know the value of a

Multi-Agent Epistemic Logic

Example model of **ASK** action (R asks H for a)

<u>Preconditions</u>

$$K_R(K_H a \vee K_H \neg a) \wedge \neg K_R a \wedge \neg K_R \neg a$$

R knows that H knows the value of a

R does not know the value of a

<u>Effects</u>

$$K_H(\neg K_R a \wedge \neg K_R \neg a)$$

H knows that R does not know the value of a

Epistemic Planning

- Planning at the level of knowledge
- Dynamic epistemic logic (DEL)
  (general framework, very high complexity)
- Many variations of DEL and related approaches that focus on making it efficient (e.g., see work of Bolander et al. at DTU, Nebel et al. at Freiburg, Geffner et al. at UPF, etc.)

KR 2020 Tutorial on Epistemic Planning
ICAPS 2020 Tutorial on Epistemic Planning

Epistemic Planning

This lecture:

- Practical planners for HRI
  - **PKS** [Petrick & Bacchus, 2002]
- Specific epistemic logics for planning
  - $\mathcal{ALCK}_{\mathcal{NF}}$ [Iocchi et al., 2000]

PKS

- **Planning with Knowledge and Sensing (PKS)**
  [Petrick & Bacchus 2002, 2004; Petrick & Foster 2013, 2020]

- Contingent planning with incomplete information and sensing

- **Knowledge-level representation of action**
  - Actions are modelled in terms of the changes they make to the planner's knowledge state rather than the world state
  - Based on manipulating a collection of databases with syntactic restrictions on the form of knowledge that can be represented

Representing knowledge in PKS

Kf: knowledge of positive and negative facts (not closed world!)

`onTable(blockA), !gripperEmpty, seeksAttention(customer1)`

Kw: know whether information (e.g., binary sensors)

`heavy(blockA)?, ordered(customer1)?`

Kv: know value information (e.g., multi-valued sensors)

`objectWeight(blockA)?, drink-order(customer1)?`

Kx: exclusive-or knowledge

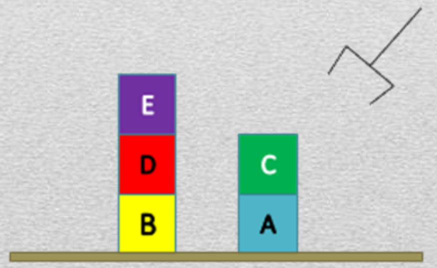
LCW: local closed world information [Etzioni et al., 1994]

Reasoning in PKS

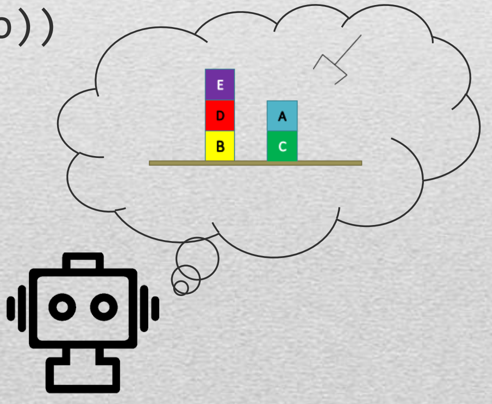
- A query language is used to ask simple questions about the planner's knowledge state:
  - **K(ϕ):** is ϕ known to be true?
  - **Kv(f):** is the value of f known?
  - **Kw(ϕ):** is ϕ known to be true or known to be false?
  - The negation of the above queries.
- Reasoning is restricted by querying the databases, but often involves more than just a single database lookup.

World-level vs knowledge-level action

```
action grasp(?o : object)
    preconds:
        gripperEmpty
        onTable(?o)
    effects:
        !gripperEmpty
        !onTable(?o)
        inGripper(?o)
```



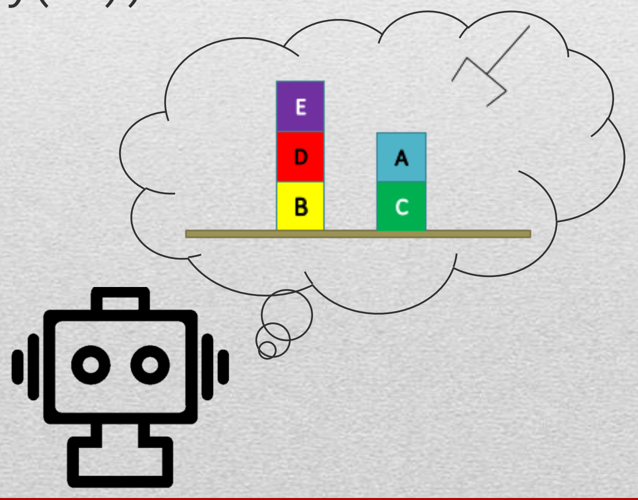

```
action grasp(?o : object)
    preconds:
        K(gripperEmpty)
        K(onTable(?o))
    effects:
        add(Kf,!gripperEmpty)
        add(Kf,!onTable(?o))
        add(Kf,inGripper(?o))
```



Sensing actions in PKS

```
action senseEmpty(?o : object)
   preconds:
      K(onTable(?o))
   effects:
      add(Kw,empty(?o))
```

```
action senseWeight(?o : object)
   preconds:
      K(inGripper(?o))
   effects:
      add(Kw,heavy(?o))
      add(Kv,weight(?o))
```



Epistemic planning in PKS

Planning in PKS = search through knowledge state transition system

```
Kf: gripperEmpty,onTable(bottle1)

Action: sense-empty(bottle1)

Kf: gripperEmpty,onTable(bottle1)
Kw: empty(bottle1)
```

Contingent plan branches:

```
Kf: empty(bottle1)
Kf: !empty(bottle1)
```

```
Action: grasp(bottle1)

Kf: !gripperEmpty,!onTable(bottle1),inGripper(bottle1)
Kw: empty(bottle1)

Action: sense-weight(bottle1)

Kf: !gripperEmpty,!onTable(bottle1),inGripper(bottle1)
Kw: empty(bottle1),heavy(bottle1)
Kv: weight(bottle1)
```

→ Run-time variable

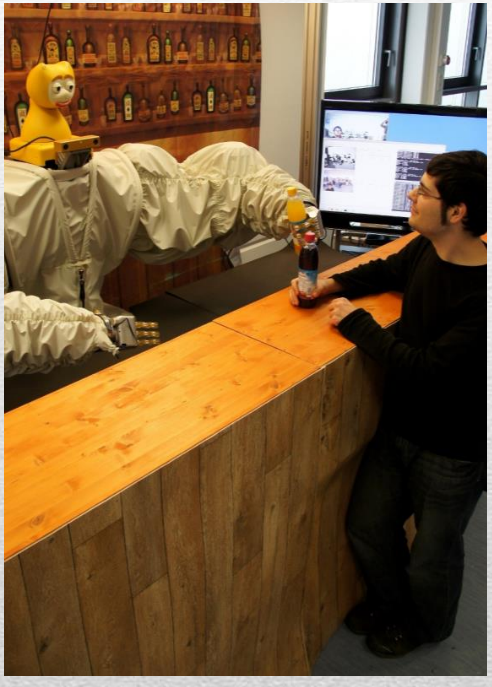
HRI actions

```
action greet(?a : agent)
    preconds:
        K(inTrans = nil)
        K(!ordered(?a))
    effects:
        add(Kf,inTrans = ?a)

action serve-drink(?a : agent, ?d)
    preconds:
        K(ordered(?a))
        Kv(request(?a))
        K(request(?a) = ?d)
    effects:
        add(Kf,served(?a))
```

```
action ask-drink(?a : agent)
    preconds:
        K(inTrans = ?a)
        K(!ordered(?a))
    effects:
        add(Kf,ordered(?a))
        add(Kv,request(?a))

action bye(?a : agent)
    preconds:
        K(inTrans = ?a)
        K(served(?a))
    effects:
        add(Kf,inTrans = nil)
```



Single agent interaction plan

`greet(a1)`	Greet agent a1
`ask-drink(a1)`	Ask a1 for drink order
`ack-order(a1)`	Acknowledge a1's order
`serve(a1,request(a1))`	Give the drink to a1
`bye(a1).`	End the transaction

Single agent interaction plan

http://james-project.eu/

Multi-agent interaction plan

`wait(a2)`	Tell a2 to wait
`greet(a1)`	Greet a1
`ask-drink(a1)`	Ask a1 for drink order
`ack-order(a1)`	Acknowledge a1's order
`serve(a1,request(a1))`	Give the drink to a1
`bye(a1)`	End a1's transaction
`ack-wait(a2)`	Thank a2 for waiting
`ask-drink(a2)`	Ask a2 for drink order
`ack-order(a2)`	Acknowledge a2's order
`serve(a2,request(a2))`	Give the drink to a2
`bye(a2).`	End a2's transaction

Another multi-agent interaction plan

`greet(a1)`	Greet a1
`ask-drink(a1)`	Ask a1 for drink order
`ack-order(a1)`	Acknowledge a1's order
`ask-drink-next(a2)`	Ask a2 for drink order
`ack-order(a2)`	Acknowledge a2's order
`serve(a1,request(a1))`	Give the drink to a1
`serve(a2,request(a2))`	Give the drink to a2
`bye(a2).`	End a1's transaction

Multi-agent interaction plan

http://james-project.eu/

Contingent interaction plan

`greet(a1)`	Greet agent a1
`ask-drink(a1)`	Ask a1 for drink order
`branch(request(a1))`	*Consider contingent plan branches*
`  K(request(a1)=juice):`	*If order is* juice
`      ...`	`[subplan]`
`      serve(a1,juice)`	Serve juice to a1
`  K(request(a1)=water):`	*If order is* water
`      ...`	`[subplan]`
`      serve(a1,water)`	Serve water to a1
`  K(request(a1)=cola):`	*If order is* cola
`      ...`	`[subplan]`
`serve(a1,beer)`	Serve cola to a1
`bye(a1).`	End the transaction

$\mathcal{ALCK}_{\mathcal{NF}}$

Description Logics with epistemic operators

- Concepts: properties of the environment (fluents)
- Individuals: states
- Roles: actions

[Iocchi et al. KR 2000]

Preconditions

$$\mathbf{K}C \sqsubseteq \exists \mathbf{K}R.\top$$

C: preconditions, R: action, **K**: epistemic operator

$\mathcal{ALCK}_{\mathcal{NF}}$

Effects

Ordinary actions

$$\mathbf{K}C \sqsubseteq \forall R.\, D$$

Sensing actions

$$\mathbf{K}C \sqsubseteq \mathbf{K}(\forall R_S.\, D) \sqcup \mathbf{K}(\forall R_S.\, \neg D)$$

$\mathcal{ALCK}_{\mathcal{NF}}$

Frame axioms (inertial properties)

$$\mathbf{K}C \sqsubseteq \forall \mathbf{K}R.\, \mathbf{A}\neg C \sqcup \mathbf{K}C$$

C: fluent, *R*: action

K: epistemic operator

A: default assumption operator

If *C* is true in the current state (before execution of R) and it is consistent to assume (i.e. we cannot prove the opposite) that *C* is true after the execution of the action, then *C* is true in any successor state.

$\mathcal{ALCK}_{\mathcal{NF}}$

Planning problem

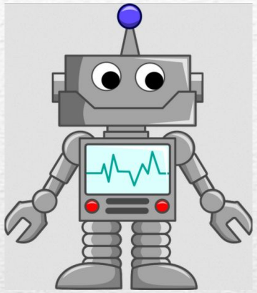
- Σ KB describing the planning domain
- $init$ individual representing the initial state
- C_G concept describing the goal

Plan generation: constructive proof of $\Sigma \models C_G(init)$

Complexity: plan existence is PSPACE-complete

Implemented planner (ask me / use PKS)

Short-term HRI with conditional plans

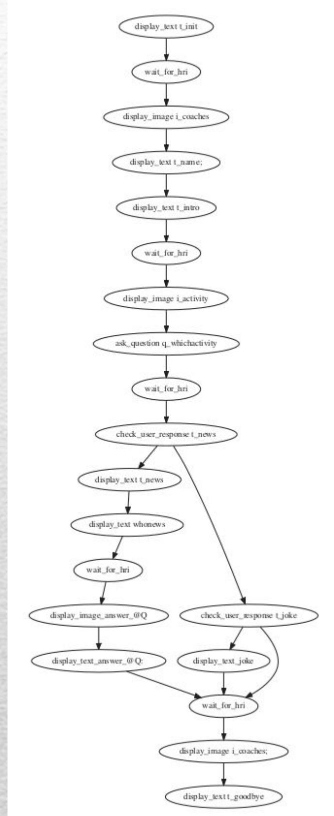


- KB modelling robot's knowledge about user needs and desires (e.g., thirsty / hungry)
- Ask/answers questions to gather information during the interaction
- Conditional plans

[Sanelli et al. ICAPS 2016]

Short-term HRI with conditional plans

- Contingent-FF
- ROSPlan
- Petri Net Plans



Short-term HRI with conditional plans

https://youtu.be/dH66_wXIrd4

Uncertainty

Sources of uncertainty in Human-Robot Interaction tasks

- Intentions
- Communication
- Perception (i.e., assessment of current state)
- Prediction of future states

HRI systems assuming perfect abilities of the robot have many limitations in actual deployment.

Unexpected situations

Situations not predicted/modelled/described in the plan

Unexpected situations can be described with variables <u>modelled</u> or <u>not modelled</u> in the (planning) domain

Unexpected situations **can be** described with **modelled** variables

- detect with plan monitoring
- solve with re-planning

Monitoring and replanning

Low-confidence speech recognition / timeouts

ask-drink(a1)	Ask customer a1 for a drink order
???	Customer a1 was not understood
[Replan]	Monitor detects expected info not available
not-understand(a1)	Alert a1 that they were not understood
ask-drink(a1)	Ask a1 again for drink order

Overanswering

greet(a1)	Greet customer a1
???	Customer a1 says "I'd like a beer"
[Replan]	Monitor detects drink order info available; don't execute ask-drink action
serve(a1,request(a1))	Serve a1 their drink

Unexpected situations in planning

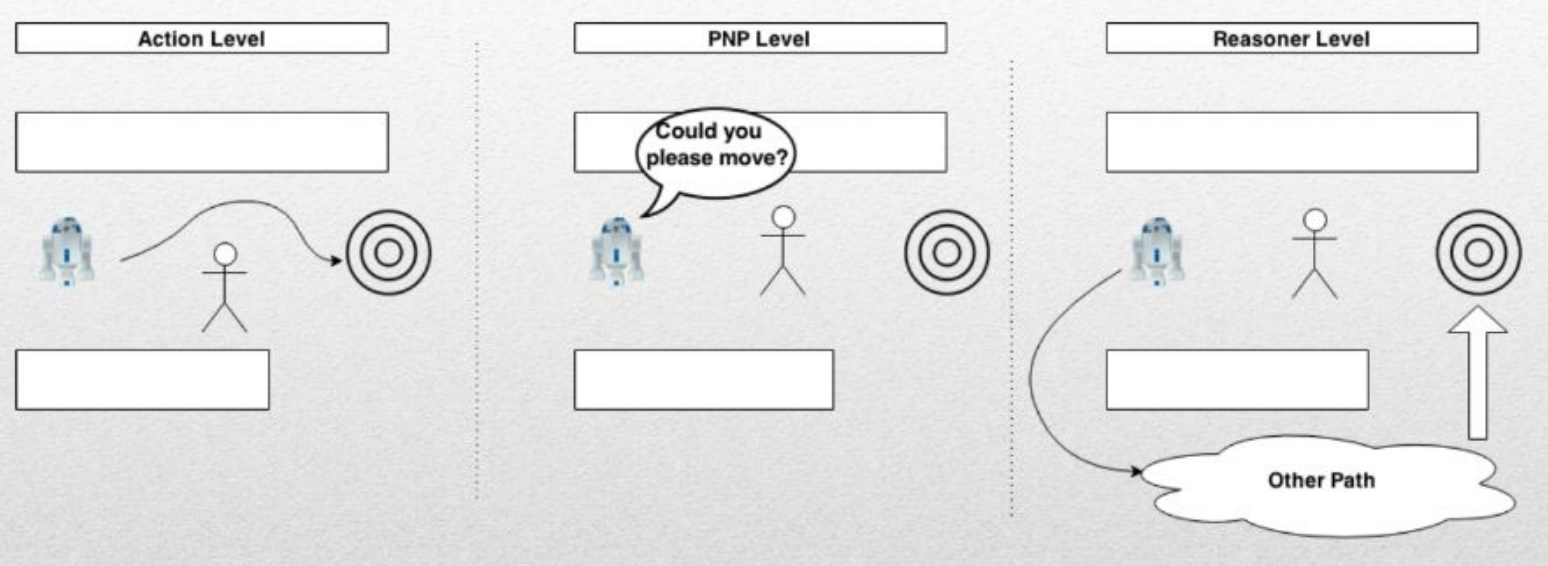
Unexpected situations **cannot be** described with **modelled** variables

- update the planning domain
- generate new planning problem
- generate new plan
- monitor
- replanning

Domain update (e.g. adding new state variables, changing action pre-conditions and post-conditions, etc.) is not trivial and brings performance issues.

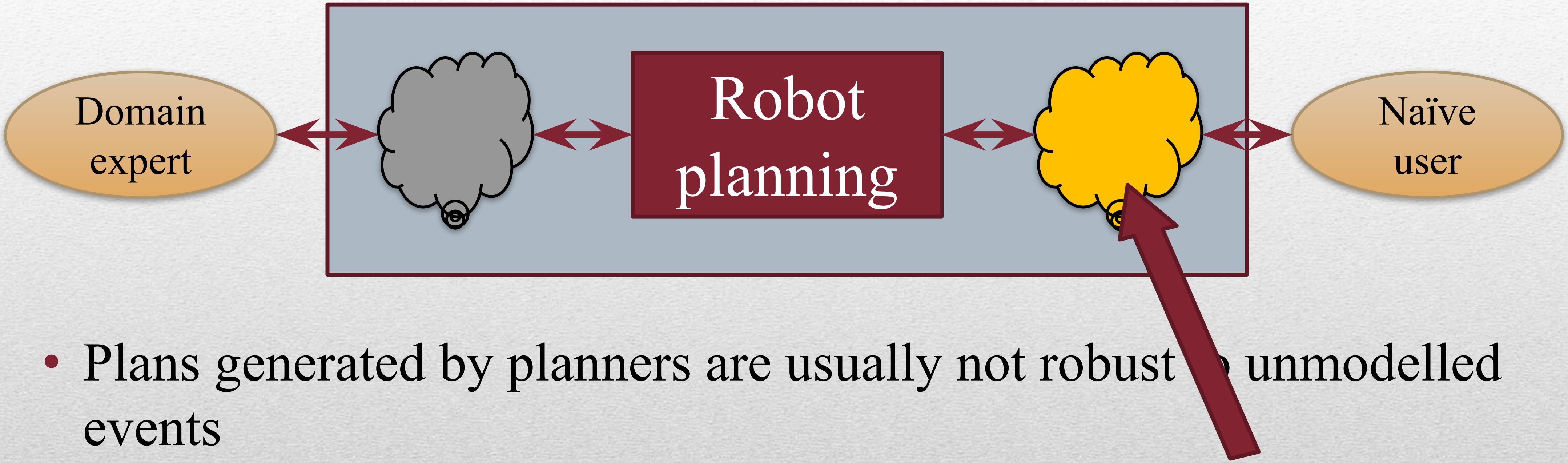
Example

A person is encountered in a corridor during a goto action



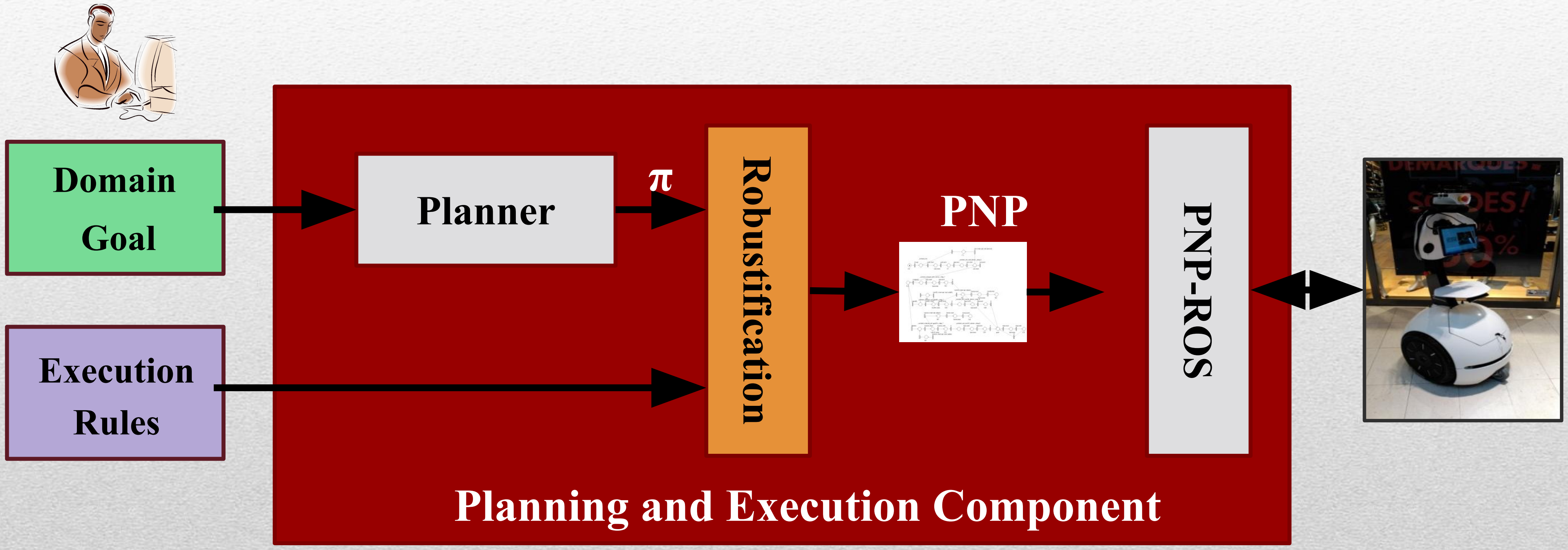
How can we express these requirements in a declarative way?

Robust plans



- Plans generated by planners are usually not robust to unmodelled events
- Interaction with naïve users requires increased **robustness** of plans

Execution rules and robustification



Execution rules

Declarative rules to monitor the execution of a plan and specify recovery procedures

if <something> **during** <some action> **do** <something>

Conditions are evaluated at execution time and they do not need to be variables of the planning domain (hence the approach is scalable).

[Iocchi et al. ICAPS 2016]

Petri Net Plans

High-level plan representation based on Petri Nets

[Ziparo et al. JAAMAS 2011]
pnp.diag.uniroma1.it

- Ordinary and sensing actions
- Conditions, loops, interrupts
- Parallel execution (fork and join operators)
- Multi-robot support

PNPGen generates PNP from several planners (MDP solver, ROSPlan, HATP, …)

PNP-ROS run plans including ROS actions

Execution rules in PNP

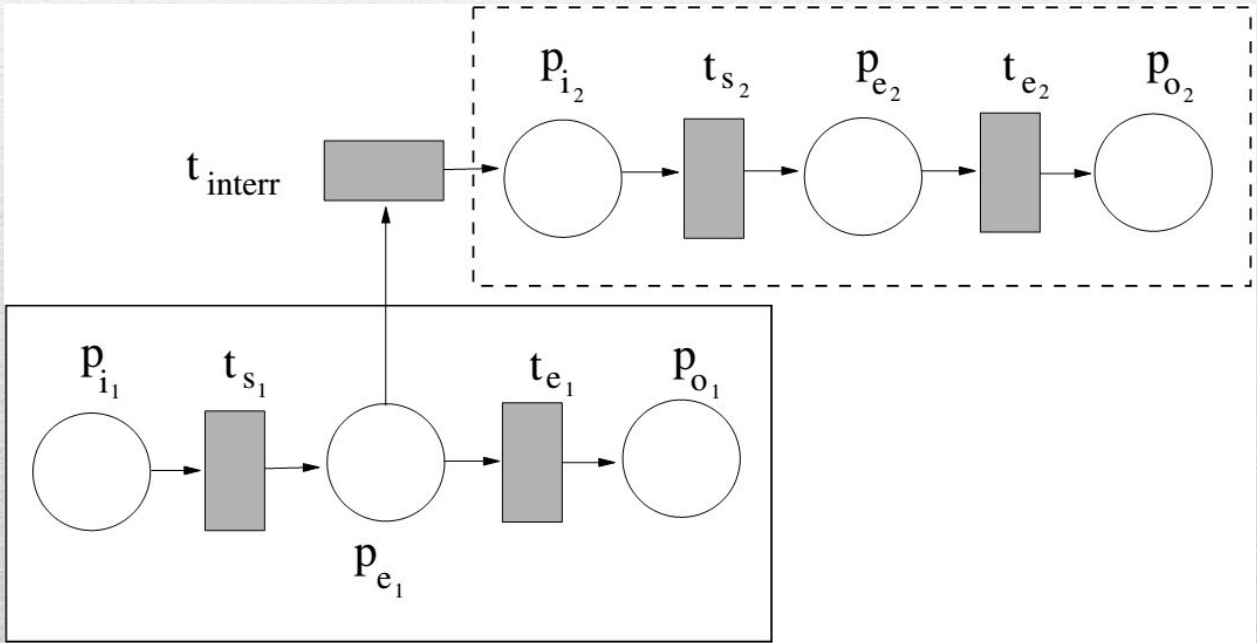
- High-level declarative rules
- Execution variables generally different from the ones in the planning domain

Adding to the conditional plan

- interrupt (special conditions that activate recovery paths)
- recovery paths (how to recovery from unexpected events)
- social norms
- parallel execution (multi-modalities)

Execution rules in PNP

Using the interrupts



Execution rules

Examples

if personhere and closetotarget **during** goto **do** skip_action

if personhere and not closetotarget **during** goto **do** say_hello; waitfor_not_personhere; restart_action

if lowbattery **during** * **do** recharge; fail_plan

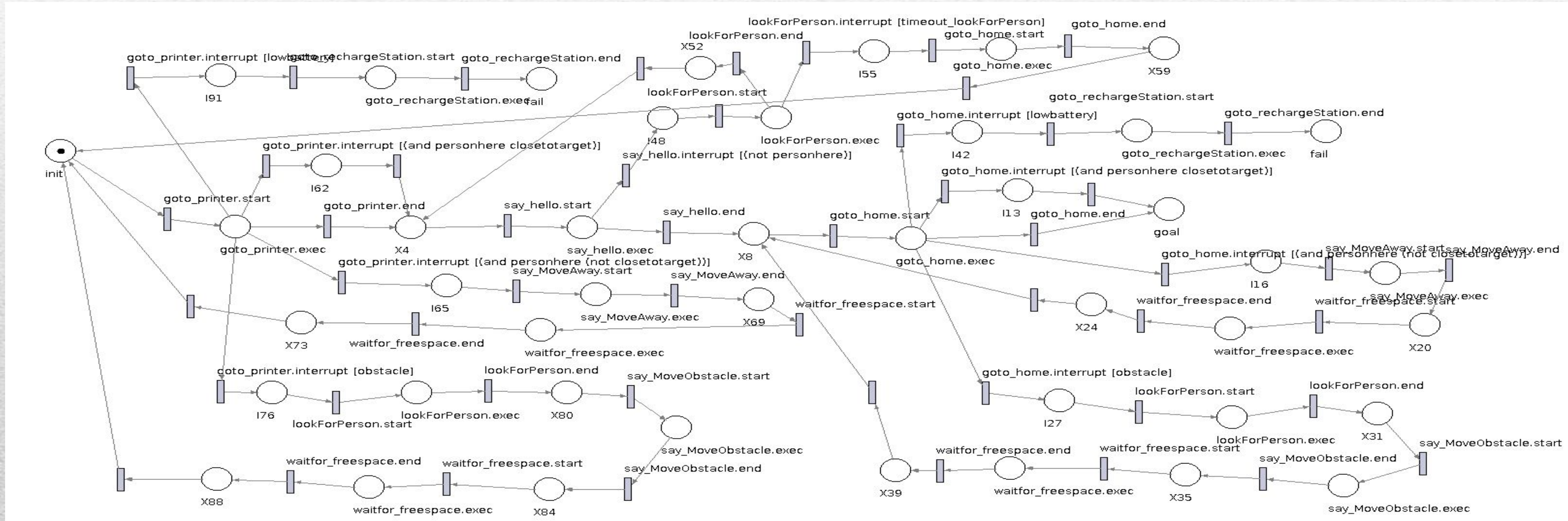
after receivedhelp **do** say_thanks

after endinteraction **do** say_goodbye

when say **do** display

Robot office assistant

Output: PNP with 17 actions, 45 places, 52 transitions, 104 edges



Robot office assistant

Conclusions and Opportunities

- Human-centered AI new perspective:
  "AI helps humans" vs. "AI defeats humans"
- Planning technology applied to HRI tasks
  - enables flexibility required to address complex problems
  - requires more theoretical/practical results
  - wide applicability to real-world systems

Acknowledgements

JAMES

james-project.eu

Stamina

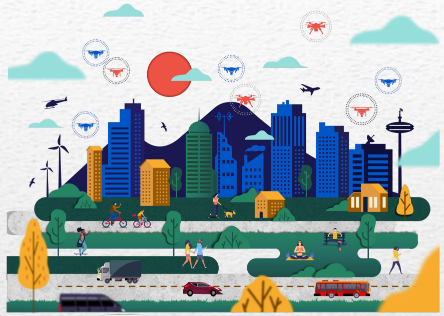
COACHES

AI4EU

Ongoing projects

AI4EU

SciRoc



BUBBLES™

ORCA HUB
Offshore Robotics for Certification of Assets
Remote Safety and Integrity

START MAKING SENSE
start-making-sense.org

Readings

[Iocchi et al., 2000] Luca Iocchi, Daniele Nardi, Riccardo Rosati: Planning with sensing, concurrency, and exogenous events: logical framework and implementation. In Proc. of the 7th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'2000), pages 678–689, 2000. https://www.diag.uniroma1.it/iocchi/publications/iocchi-kr00.pdf

[Petrick & Bacchus, 2002] Petrick, R.P.A., Bacchus, F.: A knowledge-based approach to planning with incomplete information and sensing. In Proceedings of AIPS, pp. 212–221 (2002) - https://www.aaai.org/Library/AIPS/2002/aips02-022.php

Readings

[Petrick & Foster 2013] Petrick, R. P. A. and Foster, M. E. Planning for social interaction in a robot bartender domain. Proceedings of the International Conference on Automated Planning and Scheduling, 389-397, 2013.
http://www.dcs.gla.ac.uk/~mefoster/papers/petrick-foster-icaps2013.pdf

[Bolander 2017] Thomas Bolander. A Gentle Introduction to Epistemic Planning: The DEL Approach. Proceedings of the 9th Workshop on Methods for Modalities, Electronic Proceedings in Theoretical Computer Science 243, 1-22, 2017.
https://arxiv.org/abs/1703.02192

Readings

[Petrick & Foster 2020] Petrick, R. P. A. and Foster, M. E. Knowledge Engineering and Planning for Social Human-Robot Interaction: A Case Study. Knowledge Engineeering Tools and Techniques for AI Planning. Springer, 261-277, 2020.

http://doi.org/10.1007/978-3-030-38561-3_14

[Engesser et al. 2017] Engesser, T., Bolander, T., Mattmuller, R., and Nebel, B. Cooperative Epistemic Multi-Agent Planning for Implicit Coordination. Proceedings of the 9th Workshop on Methods for Modalities, 2017.
http://eptcs.web.cse.unsw.edu.au/paper.cgi?M4M9.6.pdf

Readings

[Ziparo et al. 2011] Vittorio A. Ziparo, Luca Iocchi, Pedro U. Lima, Daniele Nardi, Pier Francesco Palamara: Petri Net Plans - A framework for collaboration and coordination in multi-robot systems. Autonomous Agents and Multi-Agent Systems, 23(3): 344-383 (2011) http://www.diag.uniroma1.it/iocchi/publications/iocchi-jaamas11-draft.pdf

[Chakraborti et al. 2017] Tathagata Chakraborti, Sarath Sreedharan, Yu Zhang, Subbarao Kambhampati: Plan Explanations as Model Reconciliation: Moving Beyond Explanation as Soliloquy. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI) 2017. - https://www.ijcai.org/Proceedings/2017/0023.pdf

Readings

[Iocchi et al. 2016] Luca Iocchi, Laurent Jeanpierre, Maria Teresa Lazaro, Abdel-Illah Mouaddib: A Practical Framework for Robust Decision-Theoretic Planning and Execution for Service Robots. In Proc. of International Conference on Automated Planning and Scheduling (ICAPS) 2016: 486-494, 2016.
https://www.diag.uniroma1.it/iocchi/publications/iocchi-icaps16.pdf

[Sanelli et al. 2017] V. Sanelli, M. Cashmore, D. Magazzeni, L. Iocchi. Short-Term Human Robot Interaction through Conditional Planning and Execution. In Proc. of International Conference on Automated Planning and Scheduling (ICAPS), 2017.
https://www.diag.uniroma1.it/iocchi/publications/sanelli-icaps2017.pdf