Using Human Knowledge Awareness for Human Robot collaboration

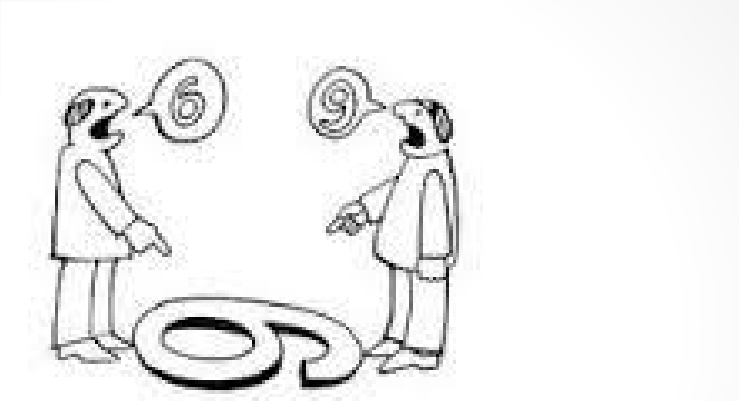
Grégoire Milliez,
Raphaël Lallement,
Michelangelo Fiore,
Rachid Alami,
LAAS-CNRS
Toulouse, France

Introduction

Statement: reasonning on **other's mental state** is a key feature for **Joint Action**!

Does the chimpanzee have a theory of mind? (Premack 1978)

Does the autistic child have a 'theory of mind'? (Baron Cohen 1985)



Issue: how to **model** and **properly use awareness** of the **human knowledge** on how to perform a task?

e.g.: We want to bake a pie. Does the human know how to prepare the dough?
How to represent this knowledge? How to use it?

Motivation: **efficient** and **socialy adapted** human-robot collaboration

Outline

1. Human Knowledge Representation
2. Human Adaptive HTN Planner
3. Shared Plan Presentation and Negotiation
4. Adaptive Plan Execution

Outline

1. Human Knowledge Representation
2. Human Adaptive HTN Planner
3. Shared Plan Presentation and Negotiation
4. Adaptive Plan Execution

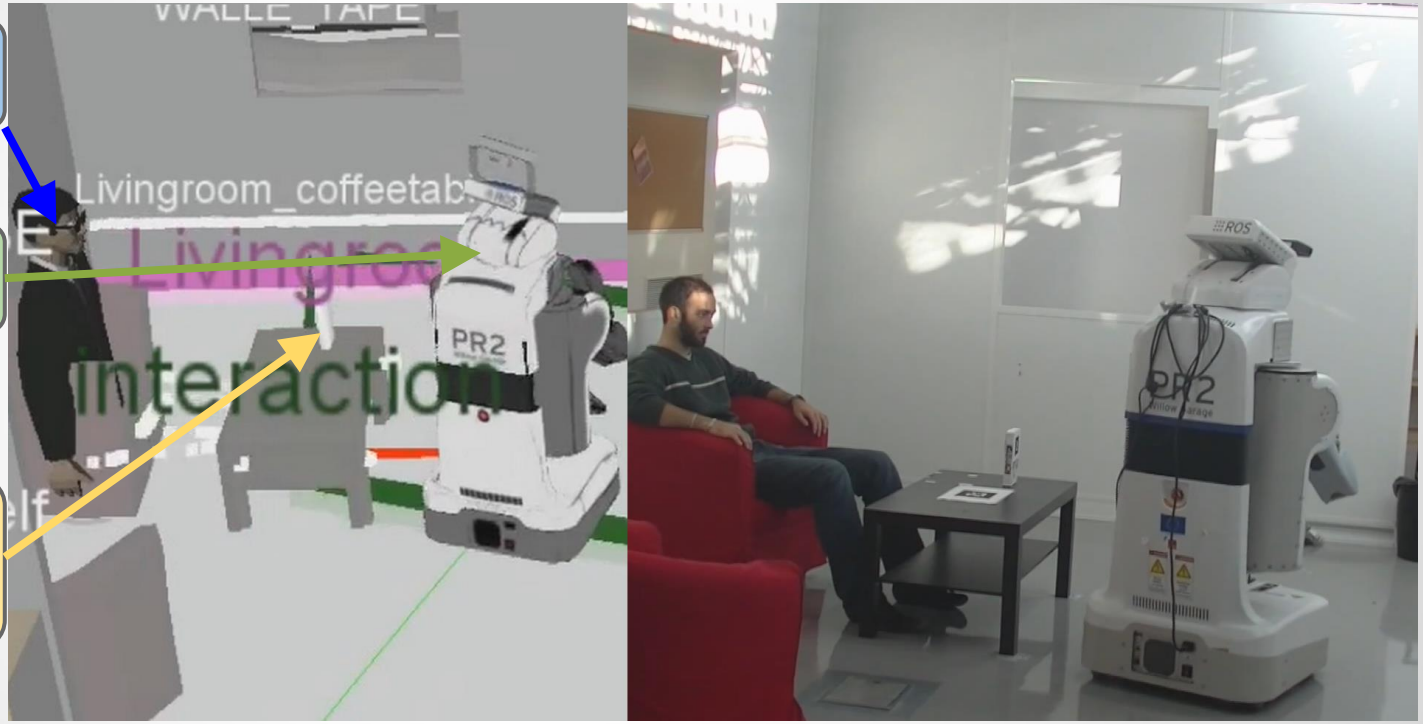
Human Knowledge Tracking

Knowledge tracking from situation assessment, using TOASTER

human tracker

proprioception

Object recognition



TOASTER: An Open-Source Situation Assessment Framework for HRI (RO-MAN 2016, *in review*)

https://github.com/Greg8978/toaster

Human Knowledge Tracking

Human Knowledge Modeling

In this work, we focus on the human's **knowledge on how to perform tasks**. This knowledge is represented as a vector

<HUMAN, TASK, PARAMETERS, VALUE>

e.g.: human1 has an expert knowledge on assembling a furniture piece A with a piece B:

<human1, assemble, [A,B], EXPERT>

HUMAN: human having this knowledge
TASK: name of the task
PARAMETERS: list of relevant parameters to describe the task knowledge
VALUE: value (or level) of knowledge

Human Knowledge Tracking

Human Knowledge Modeling

Knowledge Levels

1 NEW: human has no knowledge on how to perform the task

2 BEGINNER: human may know how to perform the task

3 INTERMEDIATE: human knows how to perform the task

4 EXPERT: human knows how to perform the task and is able to teach it

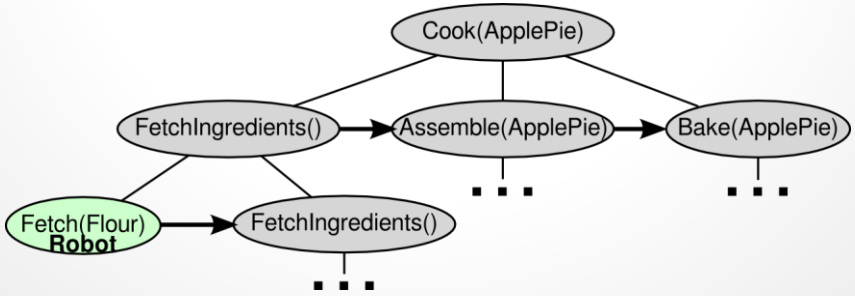
Outline

1. Human Knowledge Representation
2. Human Adaptive HTN Planner
3. Shared Plan Presentation and Negotiation
4. Adaptive Plan Execution

Human Adaptive HTN Planner

Hierarchical planner for better context

HATP: Hierarchical Agent-based Task Planner:
=> **Agents based:** computes multi-agent plans with humans and robots acting.
=> **Cost driven**: the best plan is found sooner (using plan pruning).
=> **Social rules**: refines the plans according to a set of rules designed to promote more socially acceptable plans (e.g. effort balancing depending on human preferences and context, social conventions).



Human Adaptive HTN Planner

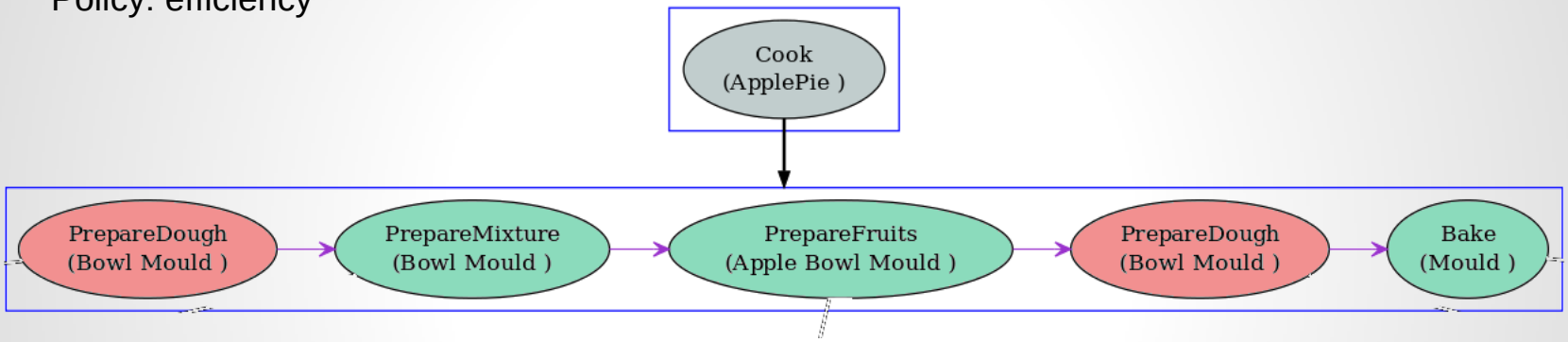
Human knowledge:

<*human1*, *PrepareDough*, *[ ]*, *EXPERT*>

Human request:

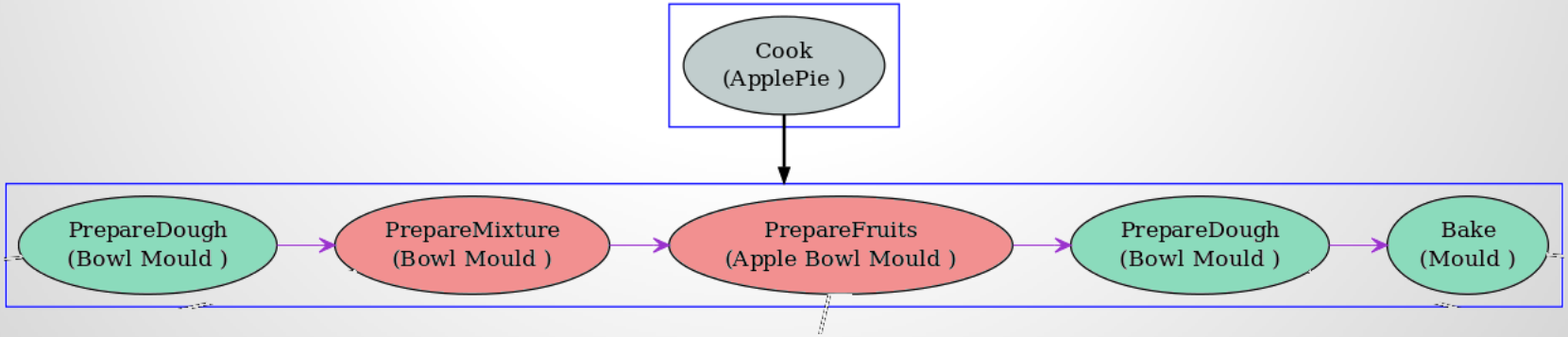
<*human1*, *Bake*, *[Mould]*, *Don' t want*>

Policy: efficiency



Robot Human

Policy: teaching



Outline

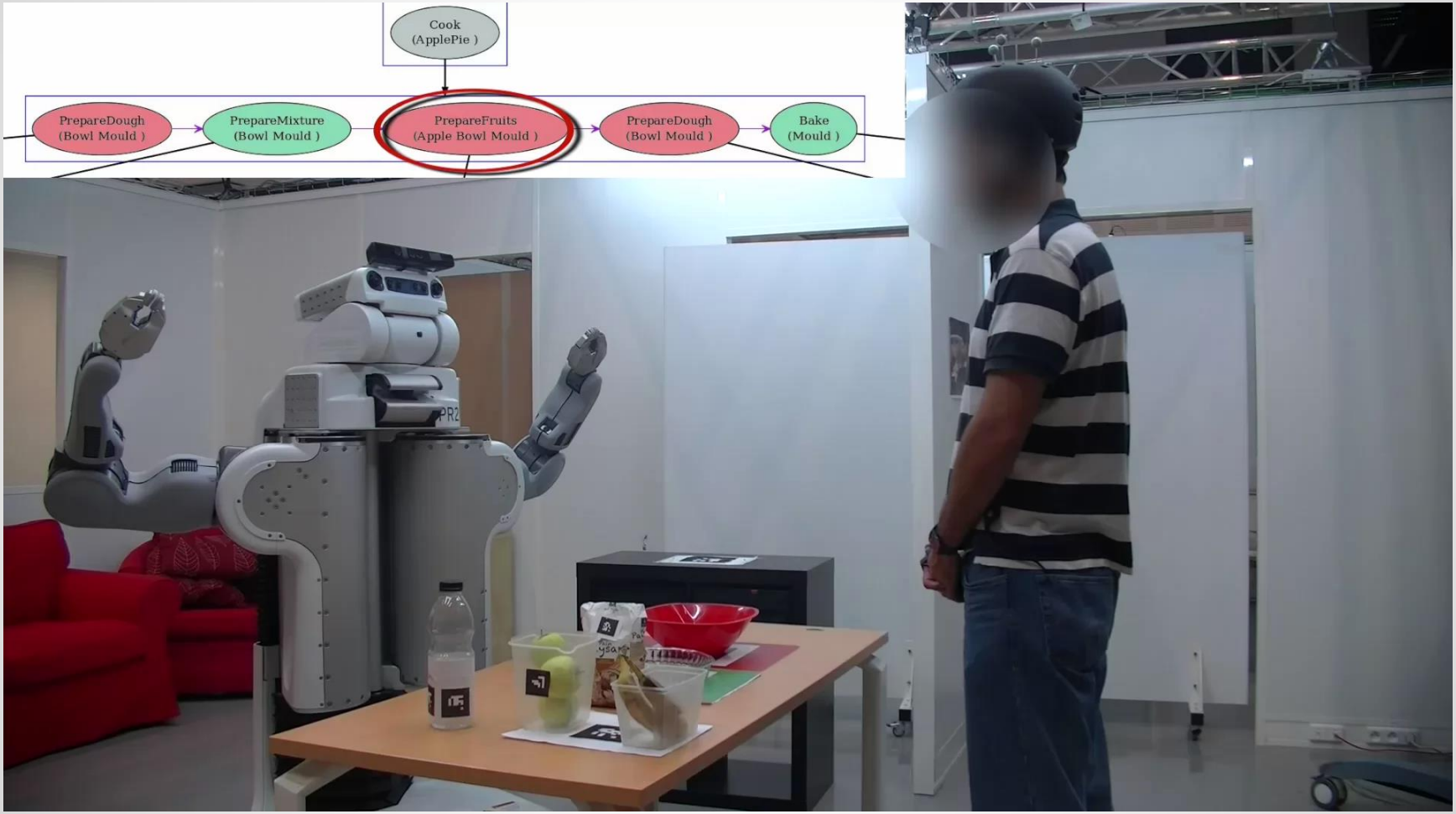
1. Human Knowledge Representation
2. Human Adaptive HTN Planner
3. Shared Plan Presentation and Negotiation
4. Adaptive Plan Execution

Shared Plan Presentation and Negotiation

Plan presentation

Shared Plan Presentation and Negotiation

Plan negotiation



Outline

1. Human Knowledge Representation
2. Human Adaptive HTN Planner
3. Shared Plan Presentation and Negotiation
4. Adaptive Plan Execution

Adaptive Plan Execution

beginner <*human1*, *PrepareDough*, *[ ]*, *BEGINNER*>

expert <*human1*, *PrepareDough*, *[ ]*, *EXPERT*>

```
1:  for n:=nodes.start to n:=nodes.end do
2:      if agents(n) = {robot} then
3:          if children(n) ≠ ∅ ∧ user_kn(n) = NEW
              ∧ teachPolicy then
4:              execute_tree(children(n))
5:              user_kn(n) := BEGINNER
6:          else
7:              execute(n)
8:          end if
9:      else if user_kn(n) = NEW then
10:         explain(n)
11:         if children(n) ≠ ∅ then
12:             execute_tree(children(n))
13:             user_kn(n) := BEGINNER
14:         else
15:             monitor(n)
16:         end if
17:     else if user_kn(n) = BEGINNER then
18:         if propose_explain(n) then
19:             user_kn(n) := NEW
20:             (...)                ▷ Same process as NEW
21:         else
22:             monitor(n)
23:         end if
24:     else if user_kn(n) = INTERMEDIATE
              ∨ user_kn(n) = EXPERT then
25:         monitor(n)
26:     end if
27: end for
```

Conclusion

⇒ How to assess and represent human knowledge on tasks
⇒ Use human knowledge awareness for plan generation, explanation and monitoring
⇒ 1st user study to show the improvement from our system

⇒ User study in real
⇒ Negotiating the decomposition? / teach the robot

Thank you!

Questions?