



**HAL**  
open science

## When the robot puts itself in your shoes. Managing and exploiting human and robot beliefs

Mathieu Warnier, Julien Guitton, Séverin Lemaignan, Rachid Alami

### ► To cite this version:

Mathieu Warnier, Julien Guitton, Séverin Lemaignan, Rachid Alami. When the robot puts itself in your shoes. Managing and exploiting human and robot beliefs. IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication, Sep 2012, Paris, France. 7p. hal-01977529

**HAL Id: hal-01977529**

**<https://laas.hal.science/hal-01977529>**

Submitted on 10 Jan 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# When the robot puts itself in your shoes. Managing and exploiting human and robot beliefs.

Mathieu Warnier, Julien Guitton, Séverin Lemaignan and Rachid Alami<sup>1,2</sup>

**Abstract**—We have designed and implemented new spatio-temporal reasoning skills for a cognitive robot, which explicitly reasons about human beliefs on object positions. It enables the robot to build symbolic models reflecting each agent’s perspective on the world. Using these models, the robot has a better understanding of what humans say and do, and is able to reason on what human should know to achieve a given goal. These new capabilities are also demonstrated experimentally.

## I. INTRODUCTION

We aim at building robots that share space and tasks with a human. Such robots could help disabled or elderly people in their home or could work among humans in factories much more flexibly than current industrial robots.

We focus here on tasks involving interactive object manipulation, where humans and robots can fetch, carry and place objects. Humans may enter or leave the interaction area at some points during the scenario. A typical example of such a task is the “clean the table” task in which a robot and a human must remove items from a table. This work builds upon an existing architecture, which already includes a complete set of abilities that allow the robot controller to effectively conduct such a collaborative task with a human partner in a flexible manner. Reasoning correctly about the human is a very important skill for the robot. It comes in many different flavors like visual perspective taking revealing what the human sees and affordances hinting at which basic actions the human can do.

In some more complicated scenarios a human may fail to notice changes in the world. The robot must then reason explicitly about this human’s belief on some object attributes that may differ from those currently held by the robot. It is necessary to understand what this human says and does, and to reason on what they should know to achieve a given goal.

The main contribution of this paper is to present an algorithm to explicitly detect, update and delete these position related beliefs in real time. The algorithm is demonstrated in real time, for several objects and humans with arbitrary geometric configurations. We discuss on the design and implementation of this algorithm. The robot can assess whether a human knows about one object’s position and whether this position is different from robot’s own beliefs based on spatial perspective taking on current state and human’s past beliefs. The robot builds an independent symbolic belief state for each agent participating in the task. We have carried out some experiments illustrating how these beliefs are generated and

how they lead to the production of different plans that yield better goal achievement results.

Section II reviews related work and analyses the context of our contribution. Section III introduces concepts on agent knowledge about position through a simple example. Section IV provides a quick overview of the robot control architecture. Section V describes our new position management features. Finally, Section VI presents the first experimental results.

## II. RELATED WORK

Perspective taking is a human ability allowing one to put himself/herself in another person’s point of view. Studied in psychology literature [7], [20], this ability is crucial when interacting with people by allowing one to reason on others’ understanding of the world in terms of visual perception, spatial descriptions and affordances. In the last years, it has been gradually employed in Human-Robot Interaction. [3] presents a learning algorithm that takes into account information about a teacher’s visual perspective in order to learn a task. [8] applies visual perspective taking for action recognition between two robots. [19] uses both visual and spatial perspective taking for finding out the referent indicated by a human partner. [15] rely on perspective taking to solve ambiguities in dialogue.

In psychology, theory of mind (ToM) is the ability to attribute mental states *e.g.* beliefs, intents, desires to oneself and others. Visual perspective taking is one of the most significant ToM precursors, and appears very early in human child development [11]. ToM encompasses a wide range of skills from instantaneous visual perspective abilities to complex interpretation of other agent’s intents, plans and feelings occurring over a long time period. Increased ToM skills directly lead to increased performance when interacting with other agents in a collaborative, as well as a competitive context. Being able to attribute false/distinct belief (to recognize that some others have different beliefs about the physical world) is considered as a milestone in ToM development.

Scassellati in [16] presents Leslie’s and Baron-Cohen’s ToM models and their potential use in robotics.

In the recent years there has been numerous and diverse contributions addressing ToM in a robotic context. Kim autonomously generates some ToM skills using evolutionary robotic techniques [9]. Butterfield uses Markov Random Fields as a sound mathematical model to make decisions that account for other agents beliefs [6]. Briggs updates robot beliefs about other agents and adapts robot speech based on the detected adverbial modifiers in human discourse [5].

<sup>1</sup> CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France <sup>2</sup> Univ de Toulouse, LAAS, F-31400 Toulouse, France  
Firstname.Name at laas.fr

Our focus is mostly on the belief modeling part of ToM. In psychology literature, the false belief task was first formulated in [21]. Breazal in [4] proposed one of the first human robot implementations, and showed some more advanced goal recognition skills relying on this false belief detection. In our scenario, the robot knows and helps the human updating his potentially wrong beliefs about the environment. The dialogue is used in a very simplified way. The robot speaks to the human who is assumed to understand the conveyed message immediately. It differs significantly from other contributions in which the humans are those who know and help the robot to improve its own modeling and perception of the environment through sophisticated dialogue.

### III. TWO ILLUSTRATIVE EXAMPLES

Fig. 1 and fig 2 show two illustrative examples. These are screen-shoots from the 3D model display of the spatial reasoning module, taken during the real experiments. It involves our cognitive robot and two humans, Patrick and Bob. Patrick is wearing a pink shirt and Bob has a blue shirt. The black box and the white box can be moved easily. The pink trashbin and the big grey box are larger objects. All these objects are on a table.

Fig. 1 illustrates the concept of *False/Distinct belief on object position*: in fig. 1(a), Patrick, Bob and the robot share the same belief state about all objects. In fig. 1(b) Patrick is away and Bob moves two boxes. In fig. 1(c) Patrick comes back. He notices the new black box position but cannot see neither the former white box position nor its new position. Consequently he still thinks the object is next to the grey box. He has a belief on the box position that is different from the current real position that is perceived by the robot. We represent it with a green sphere with full opacity localized where the human thinks the object is.

Fig. 2 illustrates the concept of *Lack of belief on object position*: in fig. 2(a), Patrick, Bob and the robot share the same belief state about all objects position. In fig. 2(b), Patrick is away and Bob moves one of the box. In fig. 2(c), Patrick cannot see the new box position but he can now notice that the object is not where he thought it was anymore. The robot concludes that Patrick does not have a belief on object position. This is represented by a partially transparent sphere placed where the human last thought the object was.

Without our algorithm, the robot would assume that the human holds the same beliefs as itself (i.e. the human knows the new tapes position).

### IV. A DECISIONAL FRAMEWORK

In this section we briefly introduce the decisional framework that is described more thoroughly in [1].

#### A. Global Overview

The proposed decisional framework consists of several entities, each having a specific role, as illustrated in fig. 3. We describe how the robot is controlled through an analysis of the three main activities performed by the robot controller:

- 1) Situation assessment and context management
- 2) Goals and plans management
- 3) Action refinement, execution and monitoring

The three robot controller activities presented above make use of a number of key components in the architecture:

- SPARK: Spatial Reasoning and Knowledge module [18]
- ORO: a knowledge management module [10]
- HATP: a Human-Aware Task Planner [2]
- A set of Human aware motion, placement and manipulation planners [17], [12], [13]

#### B. Geometric and Temporal Reasoning component

We assume that perception provides real-time identity and position of objects when they are in the field of view of the sensors. In our experiments, the robot is localized using a standard laser based localization system, the objects are identified and localized using ARToolkit, and the humans are tracked using a commercial motion capture system and a Kinect device from Microsoft. Consequently, the robot is aware of human presence, position and posture. We assume that objects are seen and recognized by a human if they are in his field of view.

SPARK, the geometric reasoning component, is responsible for geometric information gathering. It embeds a number of decisional activities linked to abstraction (symbolic facts production) and inference based on geometric and temporal reasoning. SPARK maintains all geometric positions and configurations of agents, objects, and furniture coming from perception and previous or *a priori* knowledge (object meshes).

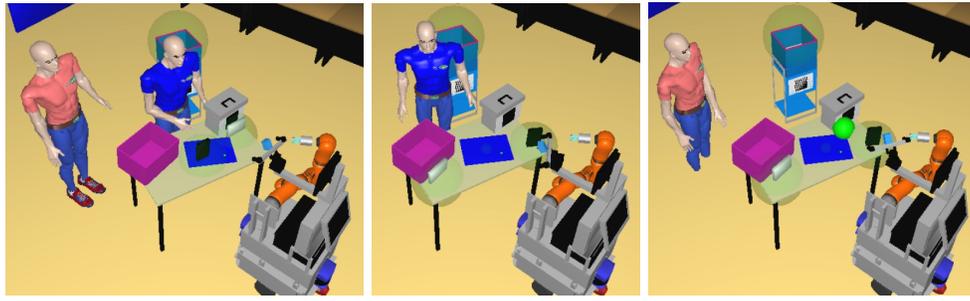
*Symbolic facts production*: Geometric state of the world is abstracted in symbolic facts that can be classified in three different categories.

- Relative positions of object and agents, e.g.  $\langle \text{BOX isOn TABLE} \rangle, \langle \text{BOX isNextTo TRASHBIN} \rangle$ .
- Perception and manipulation capacity and state of agents, e.g.  $\langle \text{ROBOT looksAt BOX} \rangle, \langle \text{BOX isVisibleBy HUMAN} \rangle$ .
- Motion status for object or agent parts, e.g.  $\langle \text{BOX isMoving true} \rangle, \langle \text{ROBOT.HEAD isTurning true} \rangle$ .

Reasoning about human perspective allows for computation of facts such as:  $\langle \text{BOX isLeftOf HUMAN} \rangle, \langle \text{BOX isVisibleBy HUMAN} \rangle$ .

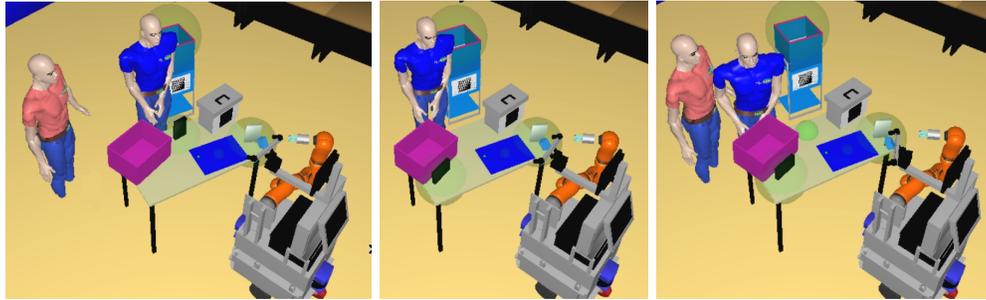
#### C. Symbolic facts and beliefs management

The facts produced by the geometric and temporal reasoning component are stored in a central symbolic knowledge base, called ORO. ORO stores independent knowledge models (in our implementation, as *ontologies*) for each agent (the robot and the humans it interacts with). The robot components (like the executive layer or the situation assessment component) can then store the agents' beliefs in specific models. Each of these models is independent and logically consistent, enabling reasoning on different perspectives of the world that would otherwise be considered as globally inconsistent (for instance, an object can be visible for the robot but not for the human).



(a) Patrick (Pink) and Bob (Blue) are here. They know each object position. (b) Patrick leaves and Bob moves two objects. (c) Patrick comes back and thinks the white box is still behind the big grey box.

Fig. 1. "False/Distinct belief" scenario.



(a) Patrick and Bob are here. They know each object position. (b) Patrick leaves and Bob moves one object. (c) Patrick comes back and does not know where the black box is.

Fig. 2. "Lack of belief" scenario.

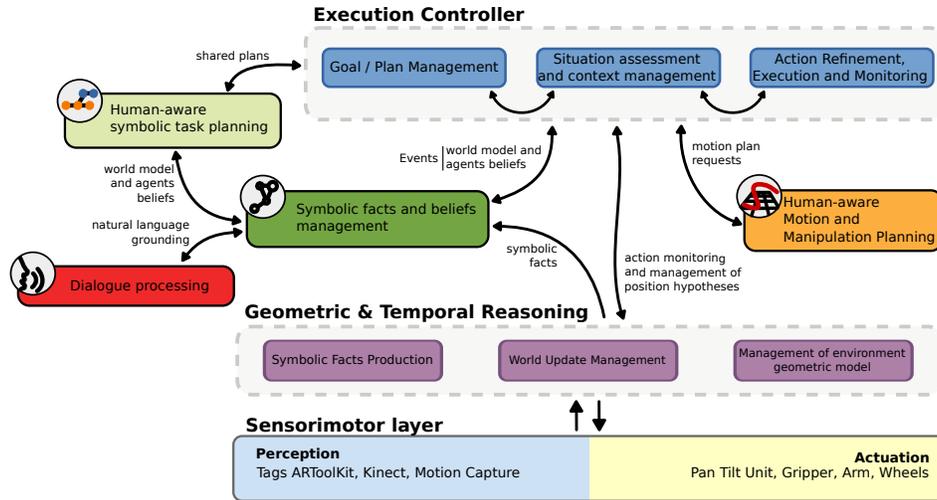


Fig. 3. Architecture of the robot control system

#### D. Symbolic Planning

In order to accomplish a desired goal, the system has to produce and execute a plan, *i.e.* a set of actions to be done by the robot and its human partners. HATP is a Hierarchical Task Network (HTN) planner. It is able to produce plans for the robot's actions as well as for the other agents (humans or robots). The resulting plan, called "shared plan" is a set of actions that forms a stream for each agent involved in the goal achievement. Depending on the context, some "shared

plans" contain causal relations between agents. For example if the second agent needs to wait for the success of the first agent's action to be able to start its own action. When the plan is performed, causal links induce some synchronizations between agents.

It can be tuned by setting up different costs from a set of constraints called social rules. This tuning aims at adapting the robot's behavior according to the desired level of cooperation of the robot.

HATP allows reasoning and planning for agents that have distinct or incomplete beliefs. It is based on a description of the agent’s beliefs, and a mechanism that produces and inserts communication actions into the current plan if and where necessary. To do this, the values of the attributes of each object linked to the task are verified if the task corresponds to an atomic action. If the agent involved is not the robot, and has an unknown or distinct value for an attribute, the planner will add respectively an ”inform” or ”contradict” action in the plan.

## V. REASONING ON DISTINCT BELIEFS

In this section we present the main contribution of this paper: the belief management algorithm.

### A. Facts computation without explicit distinct belief management.

Before managing belief for each agent, we were using algorithm 1 to construct a symbolic representation of the world. This algorithm is a simple loop, calling the function *agent.computeFacts(objects)* that computes the symbolic facts described in paragraph IV-B for each agent present in the scene.

---

#### Algorithm 1 computeFactsSimple (agents,objects)

---

```

for all agent in agents do
  if agent.isPresent then
    agent.computeFacts(objects)
  end if
end for

```

---

In this algorithm, new beliefs are not computed if the agent has left the scene. When he comes back, his beliefs are updated with the current state of the world, relying on the strong assumption that the agent becomes directly aware about all changes. In other words, all present agents share the exact same symbolic state of the world.

### B. Explicit distinct belief management.

The new algorithm for managing distinct beliefs for each agent is illustrated by algorithm 2. Before discussing the details of this algorithm, we present a set of new variables and functions used in the algorithm.

#### Variables used in the Algorithm:

In order to describe the beliefs about the object positions, we use three variables called *positionKnown*, *hasDistinctPosition* and *distinctPosition*.

The boolean variable *agent.positionKnown(object)* indicates whether an agent knows about the position of an object. *agent.hasDistinctPosition(object)* is a boolean variable used to express if the agent’s belief about the position is distinct from the belief of the robot, or not. *agent.distinctPosition(object)* contains the distinct belief about position if there is one, and is empty otherwise.

Robot belief about object position is stored in some other variable.

With these variables, it becomes easy to describe the different beliefs each agent has about the object positions. For example:

Patrick doesn’t know where the grey box is.

```

robot.positionKnown(greyBox) = true
patrick.positionKnown(greyBox) = false
patrick.hasDistinctPosition(greyBox) = false
patrick.distinctPosition(greyBox) = empty

```

Patrick and Robot both believe grey box is at a given position.

```

robot.positionKnown(greyBox) = true
patrick.positionKnown(greyBox) = true
patrick.hasDistinctPosition(greyBox) = false
patrick.distinctPosition(greyBox) = empty.

```

Patrick believes grey box is at a geometric position *someDistinctPosition1* in the 3D space known from robot but different from robot’s own belief about grey box position.

```

robot.positionKnown(greyBox) = true
patrick.positionKnown(greyBox) = true
patrick.hasDistinctPosition(greyBox) = true
patrick.distinctPosition(greyBox) = someDistinctPosition1

```

Patrick believes grey box is at some position unknown from robot.

```

robot.positionKnown(greyBox) = false
patrick.positionKnown(greyBox) = true
patrick.hasDistinctPosition(greyBox) = false
patrick.distinctPosition(greyBox) = empty

```

#### Functions used in Algorithm 2:

*agent.SaveCurrentSharedPositions(objects)*: This function is applied once when an *agent* leaves the scene and stores the current value of the robot position in the agent’s *distinctPosition* variable for all *objects* whose position is known by the agent and equal to the robot position.

*objects.previouslyUnseenAreUnknown(agent)* : this function is applied once to an appearing *agent*, and sets the *positionKnown* variable to false for all *objects* that were never seen by this agent before.

*object.isSeenAtRobotPosition(agent)* : this function answers whether an *agent* can see an *object* at the position in the robot model.

*agent.deleteDistinctPosition(object)* : this function deletes the *distinctPosition* variable of one *agent* for one *object*.

*objects.setDistinctPositionsInModel(agent)* : this function moves all *objects* in the 3D model to the position in *distinctPosition* in the *agent* model.

*agent.computeFacts(objects)* : computes the facts for all *objects* using positions defined currently in the 3D model that are the position beliefs hold by one specific *agent*.

*object.isSeenAtDistinctPosition(agent)* this function computes whether an *agent* can see an *object* at the position he thinks the *object* is.

*objects.resetRobotPositionsInModel()* : this function resets all *object* positions in the 3D model to the positions believed by the robot.

### C. Symbolic facts and beliefs management

The algorithm 2 for facts computation with belief management is presented below in pseudo-code using the variable and functions defined above.

When an agent is absent, some objects may be moved from one place to another. When the agent comes back, there are three different cases if an object has moved:

- The agent sees the current object position and consequently notices the transition, so he will not have distinct belief about this object position.
- The agent may not see the object, but notices that it is not at its previous position. In this case, the agent will know that he doesn't know where this object is.
- The human has no way of noticing the transition because the object is not visible from its current nor it would be if it was still at its previous position. In this case, the agent has a distinct belief from the robot, until these two conditions stay true or another agent communicates the object position.

---

**Algorithm 2** computeFactsWithDistinctBeliefs (*agents, objects*)

---

**Require:** *computeFactsSimple* (*robot, objects*)

**for all agent in agents except robot do**

**if** *agent.hasJustDisappeared* **then**

*agent.SaveCurrentSharedPositions(objects)*

**else if** *agent.isPresent* **then**

**if** *agent.hasJustAppeared* **then**

*objects.previouslyUnseenAreUnknown(agent)*

**end if**

**for all object in objects do**

**if** *agent.hasDistinctPosition(object)* or *Not(agent.positionKnown(object))* **then**

**if** *object.isSeenAtRobotPosition(agent)* **then**

*agent.deleteDistinctPosition(object)*

*agent.positionKnown(object) := true*

**end if**

**end if**

**end for**

*objects.setDistinctPositionsInModel(agent)*

**end if**

**if** *agent.isPresent* **then**

*agent.computeFacts(objects)*

**for all object in objects do**

**if** *agent.hasDistinctPosition(object)* **then**

**if** *object.isSeenAtDistinctPosition(agent)* **then**

*agent.positionKnown(object) := false*

*agent.deleteDistinctPosition(object)*

**end if**

**end if**

**end for**

*objects.resetRobotPositionsInModel()*

**end if**

**end for**

---

We rely on a simplifying assumption: agents present in the scene notice every action on all objects, and consequently know the new position of an object, even if these agents

cannot see the objects at their new position. Distinct position management is allowed only for some of the objects that are easily moved and hidden behind other bigger objects. This distinct position management can increase the computation required. Facts involving objects with distinct positions are recomputed for all agents that hold a distinct position belief for these objects. In the worst case, the number of re-computations would be multiplied by the number of agents. In our scenarios with less than three agents and less than five potentially moving objects, computation time is reasonably low.

## VI. EXPERIMENTS

Managing distinct beliefs for an agent is very helpful for understanding human speech, action, and focus of attention. In the context of symbolic task planning, it also allows the robot to plan whether to say something, what to say, and when to say it; to ensure proper plan realization. During the plan execution, the robot must decide whether to speak as planned, according to the evolution of the beliefs. The position belief management features presented above have been implemented in the spatio-temporal reasoning module in the architecture on our real robot. Distinct symbolic facts can now be produced and stored in each agent's models. The communication actions produced by the planner are managed by the robot controller. In these experiments we show some dialog examples and then present task planning applications more thoroughly. Some videos and 3D displays of these experiments can be seen at the following url <sup>1</sup>.

### A. Better dialogue management.

The robot can answer questions or propose some information proactively, based on its reasoning about other agent's beliefs. In section §III we present two scenarios to illustrate false/distinct belief and lack of belief. At the end of the second scenario, the state of the world for the robot (ROBOT) can be seen in fig. 2(c). Patrick (PATRICK) doesn't know where the black box is (B\_BOX) but the robot notices that he sees the white box (W\_BOX). We present an extract of the two symbolic models of the robot and Patrick below.

ROBOT	PATRICK
W_BOX isVisibleBy ROBOT	W_BOX isVisibleBy ROBOT
W_BOX notVisibleBy PATRICK	W_BOX notVisibleBy PATRICK
W_BOX isOn TABLE	W_BOX isOn TABLE
B_BOX isVisibleBy ROBOT	B_BOX hasKnownLocation false
B_BOX isVisibleBy PATRICK	
B_BOX isOn TABLE	
B_BOX isNextTo PINK_TRASHBIN	

Patrick asks the robot: "Where is the box?" The dialog disambiguation mechanism can easily understand from Patrick's model that he speaks about the box for which he has no localization *i.e.* the black box (B\_BOX). The robot can answer : "It is on the Table, next to the pink trashbin" It enriches the categorization algorithm proposed in [14]

At the end of the first scenario, the state of the world for the robot (ROBOT) can be seen in fig. 1(c). Patrick

<sup>1</sup><http://homepages.laas.fr/mwarnier/Roman2012.html>

(PATRICK) has a wrong belief about the white box (W\_BOX) position. Patrick then tries to look behind the grey box where the white box was. We present below an extract of the two symbolic models of the robot and Patrick in this new state.

ROBOT	PATRICK
W_BOX isNextTo PINK_TRASHBIN	W_BOX isNextTo BOX
W_BOX isOn TABLE	W_BOX isOn TABLE
	W_BOX isLookedAtBy PATRICK
B_BOX isOn TABLE	B_BOX isOn TABLE
B_BOX isNextTo BOX	B_BOX isNextTo BOX

Patrick looks at the white box (W\_BOX) position in his model. The robot can understand that the human currently looks for the white box. (W\_BOX). The robot can say proactively : "The object you are looking for is next to the pink trashbin"

### B. A planner coping with divergent beliefs.

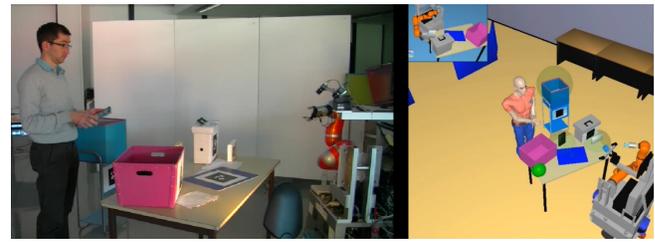
Again, we use the two scenarios presented in §III. The robot and Patrick must clean the table together. Fig. 4(a) is the plan produced with the state of the world in fig. 1(c). The plan's first action is a communication action. The robot contradicts Patrick about his belief about the white box position. This updates Patrick's belief about the white box position, so that Patrick can realize the second action, which consists of picking up the white box. Patrick would have tried to pick up the white box behind the big grey box otherwise. Fig. 4(b) is the plan produced with the state of the world in fig. 2(c). The plan's first action is a communication action. The robot informs Patrick about the black box position. This updates Patrick belief's about the black box position, so that Patrick can realize the second action, which consists of picking up the black box. Patrick would not have known where to reach for the black box otherwise.

The first two videos on the web page, whose links can be found above, show these two scenarios.

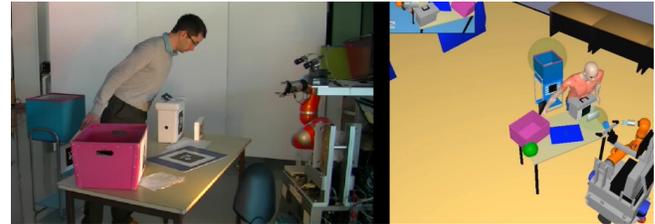
In this last experiment we demonstrate the robot controller's capacity to adapt the plan execution according to a new unexpected state of the world. In fig. 5(a), Patrick thinks that the white box is still close to the pink trashbin. Fig. 6(a) shows the plan produced. The communication action is inserted only when needed *i.e.* before the first human action that involves the white box. Fig. 5(b) shows that the human looks at the new white box position after his first action. This will update his belief about the white box position, so that the communication action is not needed any more. Fig. 6(b) shows the plan that is actually executed. The communication action is skipped, as its preconditions are not verified but its effects are.

### C. Results

These are preliminary experiments to show that the system is functional. We firmly believe the new algorithm to be useful. During the experiment we could sometime notices that when the robot failed to detect some distinct beliefs, it lead to some ambiguities in human actions. Yet we have not assessed the improvement brought by this new algorithm



(a) World state before planning



(b) Human look at the object after first action

Fig. 5. Skipping communication scenario world states

thoroughly. We plan to do some extensive user studies, that will compare the results from different scenarios with or without the new algorithm, to assess its benefit quantitatively.

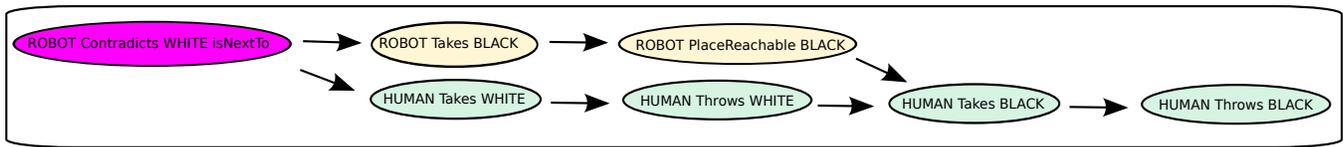
## VII. CONCLUSION AND FUTURE WORK

In this article we have presented a new feature of our cognitive robot spatio-temporal reasoning to manage human's potentially false/distinct or lacking belief on object position. It is based on visual perspective taking on current and previous object positions. Experiments were carried out, revealing our robot's capacity to assess human beliefs and use these in dialogue and task planning. It has yielded promising first results. This is a useful improvement of its ToM.

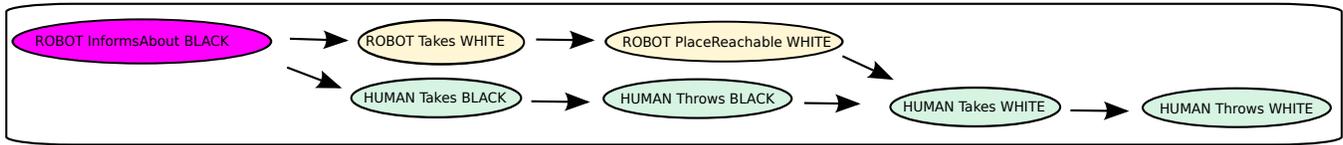
The algorithm could easily be extended to manage non position-related beliefs. The robot could reason on agent beliefs about temperature, content, weight, etc. The robot should be able to sense or be informed about these attributes and assess human perception or knowledge about them.

Regarding position beliefs, the next step could be to push the reasoning further when the robot detects that an agent knows that he doesn't know about some object's position. Can the agent then draw some conclusion on where the object could be among several possible alternatives based on occluded region and present agent's affordances? This kind of reasoning could first be used by the robot itself when it doesn't know where an object is anymore. It would then be used in conjunction with other agent's perspective to reason on another agent's belief about possible object positions.

Acknowledgments: This work has been conducted within the EU SAPHARI project (<http://www.saphari.eu/>) funded by the E.C. Division FP7-IST under Contract ICT-287513.



(a) False belief plan1

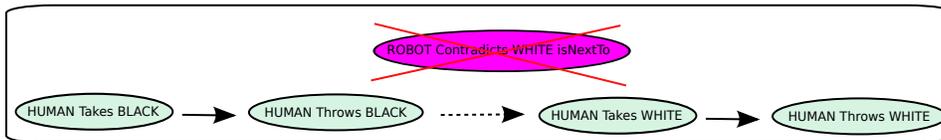


(b) Lack belief plan1

Fig. 4. Plans produced



(a) Plan produced



(b) Plan executed

Fig. 6. Skipping communication plans

## REFERENCES

- [1] R. Alami, M. Warnier, J. Guitton, S. Lemaignan, and E. A. Sisbot. When the robot considers the human... In *Proceedings of the 15th International Symposium on Robotics Research*, 2011.
- [2] S. Alili, V. Montreuil, and R. Alami. HATP task planner for social behavior control in autonomous robotic systems for hri. In *The 9th International Symposium on Distributed Autonomous Robotic Systems*, 2008.
- [3] C. Breazeal, M. Berlin, A. Brooks, J. Gray, and A. Thomaz. Using perspective taking to learn from ambiguous demonstrations. *Robotics and Autonomous Systems*, pages 385–393, 2006.
- [4] Cynthia Breazeal, Jesse Gray, and Matt Berlin. An embodied cognition approach to mindreading skills for socially intelligent robots. *I. J. Robotic Res.*, 28(5):656–680, 2009.
- [5] Gordon Briggs and Matthias Scheutz. Facilitating mental modeling in collaborative human-robot interaction through adverbial cues. In *Proceedings of the SIGDIAL 2011 Conference*, SIGDIAL '11, pages 239–247, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
- [6] Jesse Butterfield, Odest Jenkins, David Sobel, and Jonas Schwertfeger. Modeling aspects of theory of mind with markov random fields. *International Journal of Social Robotics*, 1:41–51, 2009. 10.1007/s12369-008-0003-1.
- [7] J. Flavell. *Perspectives on Perspective Taking*, pages 107–139. L. Erlbaum Associates, 1992.
- [8] M. Johnson and Y. Demiris. Perceptual perspective taking and action recognition. *Advanced Robotic Systems*, 2(4):301–308, 2005.
- [9] Kyung-Joong Kim and Hod Lipson. Towards a “theory of mind” in simulated robots. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, GECCO '09, pages 2071–2076, New York, NY, USA, 2009. ACM.
- [10] S. Lemaignan, R. Ros, L. Mosenlechner, R. Alami, and M. Beetz. Oro, a knowledge management module for cognitive architectures in robotics. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.
- [11] A.M. Leslie. Theory of mind as a mechanism of selective attention. *The new cognitive neurosciences*, pages 1235–1247, 2000.
- [12] J. Mainprice, E.A. Sisbot, L. Jaillet, J. Cortes, R. Alami, and T. Simeon. Planning human-aware motions using a sampling-based costmap planner. In *IEEE International Conference on Robotics and Automation*, 2011.
- [13] A.K. Pandey and R. Alami. Mightability maps: A perceptual level decisional framework for co-operative and competitive human-robot interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.
- [14] R. Ros, S. Lemaignan, E.A. Sisbot, R. Alami, J. Steinwender, K. Hamann, and F. Warneken. Which one? grounding the referent based on efficient human-robot interaction. In *19th IEEE International Symposium in Robot and Human Interactive Communication*, 2010.
- [15] Raquel Ros, E. Akin Sisbot, Rachid Alami, Jasmin Steinwender, Katharina Hamann, and Felix Warneken. Solving ambiguities with perspective taking. In *5th ACM/IEEE International Conference on Human-Robot Interaction*, 2010.
- [16] B. Scassellati. Theory of mind for a humanoid robot. *Autonomous Robots*, 12(1):13–24, 2002.
- [17] E. A. Sisbot, A. Clodic, R. Alami, and M. Ransan. Supervision and motion planning for a mobile manipulator interacting with humans. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI 2008)*, 2008.
- [18] E.A. Sisbot, R. Ros, and R. Alami. Situation assessment for human-robot interaction. In *20th IEEE International Symposium in Robot and Human Interactive Communication, (ROMAN)*, 2011.
- [19] J. Trafton, N. Cassimatis, M. Bugajska, D. Brock, F. Mintz, and A. Schultz. Enabling effective human-robot interaction using perspective-taking in robots. *IEEE Transactions on Systems, Man, and Cybernetics*, Part A:460–470, 2005.
- [20] B. Tversky, P. Lee, and S. Mainwaring. Why do speakers mix perspectives? *Spatial Cognition and Computation*, 1(4):399–412, 1999.
- [21] Heinz Wimmer and Josef Perner. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children’s understanding of deception. *Cognition*, 13(1):103 – 128, 1983.