# Using Human Knowledge Awareness to Adapt Collaborative Plan Generation, Explanation and Monitoring

Grégoire Milliez
CNRS, LAAS,
Univ de Toulouse, INP,
7 avenue du colonel Roche,
F 31400 Toulouse, France

Raphaël Lallement
CNRS, LAAS,
Univ de Toulouse, INSA,
7 avenue du colonel Roche,
F 31400 Toulouse, France

Michelangelo Fiore
CNRS, LAAS,
Univ de Toulouse, INSA,
7 avenue du colonel Roche,
F 31400 Toulouse, France

Rachid Alami
CNRS, LAAS,
Univ de Toulouse, LAAS,
7 avenue du colonel Roche,
F 31400 Toulouse, France

***Index Terms*—Human Knowledge Awareness; HTN; Joint Action; shared plan; HRI**

***Abstract*—One application of robotics is to assist humans in the achievement of tasks they face in both the workplace and domestic environments. In some situations, a task may require the robot and the human to act together in a collaborative way in order to reach a common goal. To achieve a collaborative plan, each agent (human, robot) needs to be aware of the tasks she/he must carry out and how to perform them. This paper addresses the issue of enhancing a robotic system with a dynamic model of its collaborator's knowledge concerning tasks of a shared plan. Using this model, the robot is able to adapt its collaborative plan generation, its abilities to give explanations and to monitor the overall plan execution. We present the algorithm we have elaborated to take advantage of the tree representation of our Hierarchical Task Network (HTN) planner to enhance the robot with appropriate explanation and execution monitoring abilities.**

**To evaluate how our adaptive system is perceived by users and how much it improves the quality of the Human-Robot interaction, the outcome of a comparative study is presented.**

## I. Introduction

One of the great challenges in robotics is to build a system able to collaborate with humans. Collaborating with human requires for the robot to plan its actions and anticipate its partner contribution to the overall plan, to monitor humans' inherently complex actions and to maintain a model of the world state. These interactions should also be performed in a socially acceptable manner in order to ensure human's comfort as well as legibility and acceptability of the robot behaviour.

We define a collaborative plan, or shared plan, as a set of linked actions involving several agents that cooperate towards the same goal. To generate a shared plan, the robot should take into account not only the environment configuration but also its human partner. A common way to do so consists in computing the affordances of the partner. On a social level, the robot should also be able to adapt the plan to the user's preference and knowledge on each task of the plan.

Once the robot has generated a plan to achieve the common goal, it needs to be shared with the human partner in order to ensure that they are aware of the tasks they have to carry out, and that they agree to perform them. When dealing with simple plans, infants can cooperate without using language if the plan and goal are simple enough. In situations requiring more complex ones, language is the preferred method [1], [2]. However explaining the whole plan at once would be inefficient and annoying for the human partner, especially if she/he is already aware of the way to achieve parts of it.

Research in psychology and philosophy have led to a better understanding of human behaviors during joint action and collaboration, to know how [3] and why [4] we collaborate and what is shared [5]. This research on joint actions was used in robotics to perform tasks involving human partners. While la substantial number of papers address the issue of plan generation for collaborative goals involving a human partner [6], only a few actually study how to efficiently adapt the plan generation and its execution to the user's knowledge. We also believe that representing and maintaining such information correctly during the interaction is a key issue. We argue that this user adaptation would significantly improve the social skills of the robot during the interaction.

We will first explain how we track human's knowledge on each HTN (Hierarchical Task Network) node, from high abstraction-level tasks to atomic actions. Then we present how we are able to take the human into account during collaborative plan generation. We then provide details on how our method uses the tree structure of the generated HTN to 1) present and negotiate the shared plan, and 2) explain and monitor tasks according to the user's knowledge level, in order to guide or teach the human when needed and to adapt the monitoring of their actions. Finally we will present an implementation of the system and a comparative study which involved two groups of users and discuss the results.

## II. Related Work

Previous studies, such as [7], have pointed out the relevance of using a joint plan. They describe experiments performed with naïve subjects and suggest that the joint plan should be fully communicated in order to sustain effective collaboration. In [8], dialog is used to teach new plans to the robot and to modify these plans. One way for the robot to learn is through "spoken language programming" where the human verbally explains the tasks to the robot. However this work does not address the situation where the robot has to explain the tasks. In [9], the system is able to learn a plan and to explain it to a new user. In this study the robot has two different modalities to adapt its behavior to users: a beginner and an expert mode. Our contribution aims at devising a more adaptive system by tracking the level of knowledge of each agent for each task and sub-task along with an online adaptive shared-plan generation.

Reasoning on others' mental state is called Theory of Mind [10]. An ability linked to this concept is perspective taking, which is widely studied in developmental literature [11], [12]. Perspective taking has been successfully used in several robotic applications to improve reasoning capabilities, leading to more appropriate and efficient task planning and interaction strategies. Previous research has shown how perspective-taking ability is a key feature for planning [13], understanding others' intentions [14] for coordination, efficient task learning by taking into account teacher's visual perspective [15], and improving dialog [16]. This research focused on the representation of other agents' visual perspective and belief state concerning the environment. In our work, we incorporate a model of the robot partner's (human) knowledge of the tasks contained in the collaborative plan to perform, in order to build a human-adaptive system for joint actions.

Research on Intelligent Tutoring Systems (ITS) [17] and on e-learning [18], has proven the necessity of keeping and updating a model of the learner's knowledge to efficiently teach a user. In our work, we maintain and update users' knowledge level for each task and use this information along with hierarchical plans to manage the interaction. The idea is not only to teach a plan to a user but to use her/his knowledge model to adapt the plan generation according to the policy of interaction (do we want efficiency, or to teach?), and during the execution, adapt the task explanation level and monitoring to the collaborator.

Some systems explicitly model shared plans during task execution, allowing the robot to adapt its plans to the users' actions, like Pike [19], [20], and Chaski [21]. In [22] dialog is used during collaborative plan execution in order to improve the performance of the team. Their approach is based on Markov Decision Problems and gives importance to the concept of agent's role in a task, which can be estimated and influenced using dialog.

Psychological studies show that humans form a shared representation of the tasks which includes the actions that every partner should perform [23]. Execution monitoring must then be linked to this shared representation of the tasks to better track the engagement level of each member in the joint action, and if there are errors that need to be dealt with.

## III. HUMAN-KNOWLEDGE TRACKING

### A. Situation Assessment and Mental States

To assess the knowledge state of its collaborator, the robot needs to understand the situation and extract information about agents. To do so, and to maintain a consistent world state, we use a situation assessment component to perform spatio-temporal reasoning based on data about humans, robots and objects [24]. It also computes affordances for each agent (reachability and visibility). This world state will be used by our plan generator to compute a plan adapted to the situation.

Using situation assessment, our previous work successfully managed a belief state for each agent, robot and human. Each belief-state model is independent and logically consistent. The robot belief regarding its counterparts' mental state about the environment is represented in these models.

In this paper we focus on the human's knowledge of tasks. This knowledge is represented as the vector <HUMAN, TASK, PARAMETERS, VALUE>. HUMAN is the human having this knowledge, TASK is the name of the task, PARAMETERS is the list of relevant parameters to describe the task knowledge (see below) and VALUE is the value (or level) of knowledge. As an example, the fact that the *human1* has an *expert* knowledge on assembling a furniture piece A with a piece B would be represented as: *<human1, assemble, [A,B], EXPERT>*. The possible knowledge values are *NEW*, *BEGINNER*, *INTERMEDIATE* and *EXPERT*.

Some tasks can be considered as common knowledge. For instance, putting ingredients in a bowl is considered simple enough to be a known action for any human. This kind of task will then be tagged as common knowledge and considered as known by the users no matter the parameters. Some other tasks knowledge might differ according to the parameters. For these tasks, the knowledge may be linked to a "type" of parameter instead of an instance of this class. As an example, we can consider that if a human knows how to paint the living-room, she/he will know how to paint any room. In this case we will put in their knowledge the type "room" for the task of painting instead of the instance living-room. To sum up, some tasks can be tagged as common knowledge while other tasks can be described by some of their parameters or parameter type. This formalism of task representation requires a domain expert to indicate how to represent the task knowledge.

### B. Knowledge Levels on Task

In this context, as the robot generates the shared plan, we assume that it knows all the tasks in the plan. Concerning the collaborator, we define four task-knowledge levels that will lead to different behaviors from the robot.

- *NEW*: this value will be used for tasks which have never been performed by the user. If the user observes the task being executed with explanation or if he performs it himself, the value will be changed to *BEGINNER*. However if the user observes the task being executed without any explanation, we keep the level as *NEW* since we consider that he has not been given enough information to link the observation with the task.
- *BEGINNER*: this value will be used for users who have already achieved the task but may still need explanation to perform it again. If the user successfully performs the task again, without asking for explanation, the value is changed to *INTERMEDIATE* and to *NEW* otherwise.
- *INTERMEDIATE*: this value will be used for users who are able to perform the task without guidance. If the user

successfully performs the task without guidance again, the value is changed to *EXPERT*. In case of failure, it is downgraded to *BEGINNER*.
- *EXPERT*: this knowledge level will be used for users who are able to perform the task without guidance and are experienced enough to explain it to a third party. If the user fails in performing the task, she/he is downgraded to *INTERMEDIATE*.

These task knowledge levels allow for adaptation of the collaborative plan generation, explanation and monitoring.

## IV. Human-Adaptive HTN Planner

We employ a hierarchical planning approach as it offers a better understanding of the context in which an agent is asked to carry out an action. This context is beneficial for plan explanation. Indeed, it implements an iterative context-sensitive refinement process. Hence, the system can guide users by telling them why they should perform an action and how it is linked to previous and following parts of the plan. We utilize an HTN, which provides domain representation and efficient planning.

To compute collaborative plans, we use a modified HTN planner specifically designed for robotics. It comes along with specific features concerning plan generation, such as:

- Agent based: it computes multi-agent plans with humans and robots acting.
- Cost driven: the best (or a "sufficiently" good) plan is found sooner (using plan pruning).
- Social rules: it 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…).

Each action in the domain provides a function that estimates its cost if added to the plan. So at any time it is possible to compute the cost of the partial plan while it is being built. Furthermore the current best plan score is stored; if at any moment the cost of the current partial plan exceeds this score, the plan is discarded and the search continues. This plan pruning helps speed up the search for the best plan. After each plan is computed a set of filtering rules are applied to sanction plans that do not exhibit certain social behaviors. Once the best plan (note that one can limit the search to a "sufficiently" good cost level) is retrieved, it is sent to the supervisor in the form of an HTN tree decomposition. In addition, a set of streams of actions is elaborated; each stream represents the actions an agent (human or robot) must carry out. To ensure proper action sequencing and synchronization between agents, causal links are embedded. Besides, the plan may include joint actions allocated simultaneously to two or more agents because they need tight collaboration (e.g. handover). Figure 1 depicts a part of the tree decomposition of a solution plan.

To take knowledge into account while planning, a new social rule was needed. The aim is to select the best suited plan for the given policy. We propose two policies: favor teaching, so the human can learn from the robot, or efficiency. With the

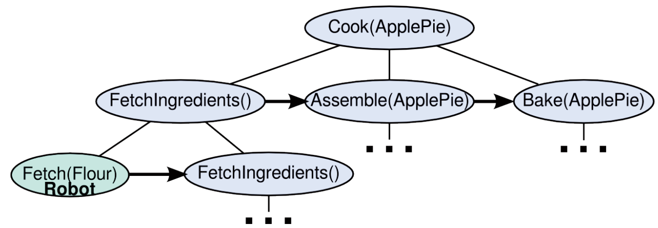


Fig. 1: Extract of a decomposition tree to cook an apple pie.

teaching policy, the planner tries to produce plans maximizing the number of human tasks where they have the opportunity to learn, while efficiency makes the planner select plans with the least amount of unknown tasks for the human in order to ensure that they can be more efficient. In case of a policy to favor efficiency, the rule is simply to apply a penalty every time the human has to perform an action they ignore. This penalty would be reversed for the teaching rule.

To illustrate our planner and the new social rule let us consider the toy example where a human and a robot have to cook an apple pie. A part of the solution plan is shown in Figure 1. In this context we can consider that the human knows how to carry out all the actions (pick, place, cut, and so on) but they may not know the exact order of steps (higher level task). If we favor teaching, the plan should contain a way to achieve the recipe with a minimal knowledge level on each task and, as much as possible, human will be in charge of those steps. On the other hand, if we favor efficiency the plan should contain the smallest amount of unknown tasks to be performed by the user. Using this rule, the robot is able to adapt its plan generation to the knowledge of the user concerning tasks contained in the shared plan. To properly compute the cost of a plan, the planner will also consider a task knowledge as upgraded once it is added to the plan. This allows the efficiency policy to prefer plans that reuse the same task many times and assign it to the same user to lower the cost, over some plans where different tasks are performed or the same task is performed by a different agent.

The planner is integrated in a dialog system that allows to negotiate plans (see below). More precisely it allows for asking about user preferences and abilities. If the user tells the system that she/he cannot perform a given task, it will not be added to the plan (invalidate the corresponding task precondition). Concerning user preferences, the negotiation step will update a database with the preferences expressed by the user. If the user specifies that she/he (does not) want to perform certain tasks, those tasks if added to the plan, will take an important reward (resp. penalty) cost. Hence plans which contain such tasks will be considered as unwanted, however if they are the only possible solutions (because of inability and so on) they will be kept and the planner will return the one with the least number of unwanted and maximum number of wanted tasks.

## V. Shared Plan Presentation and Negotiation

Once the robotic system has generated a collaborative plan adapted to the chosen policies (about learning, abilities and preferences), the plan must be shared with the human partner.

| | |
|---|---|
| agents(root) + have_to + root | "We have to cook an apple pie." |
| introduce_presentation | "I will tell you the steps." |
| agents(child[0]) + first + child[0] | "You will first fetch the ingredients," |
| then + agents(child[1]) + child[1] | "Then I will assemble the apple pie," |
| finally + agents(child[2]) + child[2] | "Finally, you will bake the apple pie in the oven." |

TABLE I: Presentation of a plan to cook an apple pie. Root is the root of the tree and child is a list with its children.

Speech is a "potent modality for the on-going maintenance of cooperative interaction" [7]. Indeed, Tomasello even suggests that the principal function of language is to establish and negotiate cooperative plans [3]. Considering this, we decided to use speech to present the plan to the collaborator.

### A. Plan Preprocessing

The HTN tree generated represents a solution to achieve the goal. However, it may not be suitable to present or explain it like it is to the collaborator, as it may contain refinement steps that would make the explanation unclear. To adapt the plan for explanation we use two rules. (1) We remove the recursive tasks. If a node $n$ of the HTN tree contains the same method (using $compare$ function) as its parent $parent(n)$, it will be replaced in the tree by its children $children(n)$. (2) We also replace nodes with a single child by their children.

1) **if** $(compare(n, parent(n)))$ **then** $n \leftarrow children(n)$
2) **if** $(children(n).size() = 1)$ **then** $n \leftarrow children(n)$

These rules build a lighter plan tree to process.

### B. Plan Presentation

Before executing the plan, the robot will present the goal and the proposed allocation of high-level tasks to give a global view on the plan. Standard NL generation is used as shown in Table I. To ensure the scalability of the system, when presenting the plan, the robot will verbalize only the $N$ first highest level tasks. For simplicity, we have chosen $N$=3 based on some runs carried out during the development process. We believe that this number would require further investigation depending on the domain or on the user and her/his confidence in the execution of the tasks. The robot will present the first steps of the plan, and then execute them. Once this execution is achieved, it will repeat the present/negotiate/execute process until the plan is completed or aborted.

### C. Plan Negotiation

Once the robot has presented the main tasks and repartition, it has to ensure that the human agrees with this shared plan. The robot will simply ask the human for approval and inquire what is wrong in case of disagreement. In the current version of our system, two kinds of human requests are handled. First the user can express her/his preferences, either the will to perform a task previously assigned to the robot or the denial to perform a task assigned to her/him. The other possibility is to inform the robot that the user cannot perform an action. This will be added to the user's model and stored in the database. The robot will then try to find a new plan that prevents the human from performing a task the user is not willing or able to perform. This plan will then be presented and the robot will ask again for the user's approval. In our system, the user's preferences have a higher cost than the teaching policy as we consider that the user should have the final decision.

## VI. Adaptive Plan Execution

### A. Plan Management Algorithm

Once the plan has been accepted by the collaborator, the execution can start. We give the algorithm for the adaptive plan execution, and then explain it.

**for** n:=nodes.start to n:=nodes.end **do**
  **if** $agents$(n) = {robot} **then**
    **if** $children(n) \neq \emptyset \wedge user\_kn(n)$ = NEW $\wedge$ $teachPolicy$ **then**
      $execute\_tree(children(n))$
      $user\_kn(n)$ := BEGINNER
    **else**
      $execute(n)$
    **end if**
  **else if** $user\_kn(n)$ = NEW **then**
    $explain(n)$
    **if** $children(n) \neq \emptyset$ **then**
      $execute\_tree(children(n))$
      $user\_kn(n)$ := BEGINNER
    **else**
      $monitor(n)$
    **end if**
  **else if** $user\_kn(n)$ = BEGINNER **then**
    **if** $propose\_explain(n)$ **then**
      $user\_kn(n)$ := NEW
      $(\dots)$ ▷ Same process as NEW
    **else**
      $monitor(n)$
    **end if**
  **else if** $user\_kn(n)$ = INTERMEDIATE $\vee$ $user\_kn(n)$ = EXPERT **then**
    $monitor(n)$
  **end if**
**end for**

- $execute\_tree(n)$ is the main function to manage the execution. This is called after the negotiation process. It has $nodes$, a list of nodes initially filled with the root's children, as an argument.
- $teachPolicy$ is a boolean to define if we are in teaching or efficient mode.
- $agents(n)$ returns the agents involved in the node $n$.
- $verbalize(n)$ will verbalize the current task, using the node context to present it (e.g. using sequential relations such as first, then or finally according to the node position in the list).
- $user\_kn(n)$ returns the knowledge level of the user concerning the task $n$.
- $propose\_explain(n)$ will lead the robot to propose an explanation for the current task. If the user accepts the explanation it will return true, and otherwise false.
- $explain(n)$ launches a procedure to explain the current task to the user. This procedure could be implemented as

a script to launch a video, an explanation speech or even to ask an expert to explain the task.

- $monitor(n)$ sends a request to the supervision system to monitor proper execution of the current node. If the request returns a success, the function will upgrade the user's knowledge and $execute\_tree$ function will continue. In case of failure, the function will downgrade the user's knowledge, exit the $execute\_tree$ function, and return a failure that will result in a replan request to the supervisor and a new execution if a plan is found.
- $execute(n)$ works in a similar way to the monitor but sends a request to execute the node by the robot.

### B. Explanation of the plan management algorithm

During the execution, we use the HTN processed tree to execute the plan, give explanation about tasks and monitor them according to the knowledge values in the user model. We use a depth first plan exploration process to proceed with the execution as it gives context to the task to perform. When reaching a node, several situations may occur.

*1) Only the robot is involved (lines 2- 8):* if the robot is the only agent in charge of the current node, if the collaborator has a knowledge level equal to *NEW* for the current task and the chosen policy for the interaction is teaching, the robot will execute the subtasks in "demonstration mode", meaning that it will verbalize each child task before performing it. Once it's done, the robot updates the human's knowledge on the current node to *BEGINNER*. The same process will be applied to the children, so the robot will verbalize each (and only) task that needs to be learned by the collaborator. If the robot is in charge, but the human collaborator already has knowledge on the task, or the current policy is efficiency, the robot will verbalize only the high-level task it performs.

Then, if the human is involved in the current node, the robot's behavior will depend on the human's knowledge level for the task, as it may have to explain it. The explanation could be done in several ways: showing a video, asking an expert to explain the task or simply verbally guiding the user, step by step. We will provide details about verbally guiding the user since it is the one actually involving the robot.

*2) The collaborator is NEW (lines 9- 16):* if the human has a level NEW for the current task, we explain it. When verbally guiding the user, if the current node has only one child node, we go deeper in the tree and apply again the corresponding behavior according to the knowledge level. If the current node is actually an operator (a leaf), the supervisor waits for the user to perform the current action. In case of success, the knowledge level for the task is upgraded to *BEGINNER* (in the above algorithm this is done in the *monitor* process).

*3) The collaborator is BEGINNER (lines 17- 23):* if the human has a level *BEGINNER* for the current task, we ask if he needs explanation. If so, we downgrade his knowledge level to NEW, on the current task, and apply the same process as the previous level. If the user refuses explanations, we simply monitor the execution of the current node. In case of success, the knowledge level for the current task is upgraded to *INTERMEDIATE*. This knowledge level will also be used as default. This way, when we don't know the knowledge level of an agent concerning a task, we just ask him if he needs an explanation and adapt the behavior accordingly.

*4) The collaborator is INTERMEDIATE (lines 24- 26):* if the human has a level *INTERMEDIATE* for the current task, we verbalize it without proposing explanations, since he has already succeeded with the plan at least once without help. Also, we do not go deeper in the tree and directly monitor the current task. If the user fails, we downgrade his knowledge to *BEGINNER*, otherwise we upgrade it to *EXPERT*.

*5) The collaborator is EXPERT (lines 24- 26):* in case of *EXPERT* knowledge level on the current task, we proceed as for the previous knowledge level, downgrading to *INTERMEDIATE* if the user makes a mistake and keeping the *EXPERT* level if he performs the task as expected.

### C. Execution and Monitoring

Once the current task to perform has been explained, the robot performs it if it is allocated to it, or it monitors its human partner's task performance. Consequently, the monitoring may be done at high-level tasks if the human has enough knowledge of it. We have chosen this adaptive way of monitoring since we believe the robot would be more efficient, sparing its resources, by focusing its attention more often on parts of the plan that have never been performed by the human partner, and less when he has some form of expertise, leaving more freedom to the human on the way to execute tasks he knows.

Monitoring human actions is complex, particularly with high-level tasks, where we are not monitoring a set of atomic actions. The system should have reasoning models for the robot to understand if the state of the world is coherent with the action that the human needs to perform. It should also be able to measure the level of engagement of the human to the task, in order to better assess if the human is executing or not his part of the shared plan, and to react accordingly.

### D. Failure and Replanning

The goal of monitoring human actions is to be able to recover from unexpected behaviors of the human. When this happens, the robot needs to inform the human of his potentially wrong behavior and downgrade his knowledge level. Consequently, the next time the human will perform this task the robot will guide and monitor his execution at a more refined level (the children tasks). The task planner is requested to compute a new plan to achieve the goal with the updated world state. One of the benefits of the dynamical update of human knowledge is that this new plan may have tasks that the robot has already explained or that the human has performed before the failure occurred. In this case, guiding the human through the new plan execution will be faster as the robot will not have to reexplain these tasks. This replanning behavior gives robustness to the robotic system and allows a socially acceptable recovery procedure where we inform the human with the error and reexplain the plan only at the needed level of detail.

## VII. Implementation

### A. Architecture

We have implemented the proposed mechanisms in our architecture[1]. Figure 2 illustrates the architecture and the interactions between the main modules. The *htn execution manager* module embeds the algorithm presented in Section VI-A. It is controlled by the supervisor and sends requests to the supervisor to monitor a task.

For our test, we use a PR2 by Willow Garage[2]. As perception is not the focus of the contribution; we simply use Motion Capture to track humans, and a tag-recognition software to track objects. At the start of a scenario the robot scans the environment, building a model of its world state. We have chosen for this first implementation a simple strategy for monitoring. The robot observes the environment, updating its world state accordingly, and monitors the expected outcome of actions. Actions performed are inferred using distance to a point of interest (e.g. container). We consider an action as "failed" if it has not been performed within a predefined amount of time.

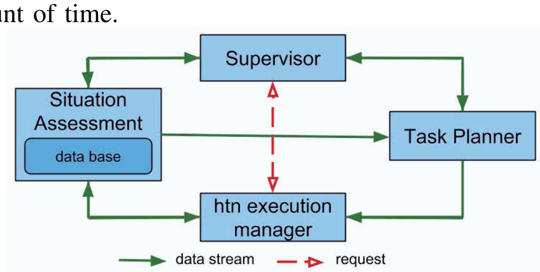


Fig. 2: System Architecture

### B. Experiment

To test our system, we have chosen a scenario where a human is coming back from work and has to prepare two desserts to bring to a diner party. She/he decides to cook an apple pie and a banana pie, but they have no previous knowledge on how to cook them. She/he asks their household robot for guidance and assistance as shown in Figure 3.

Fig. 3: Illustration of the cooking pies scenario

As in this scenario the user is in a hurry, we will use the efficiency policy. We have created a domain representing the necessary task knowledge to be used by the *task planner* and for *htn execution manager*. To cook the apple pie, five main tasks are needed. We imagine that, due to reachability constraints, the robot cannot make the dough or prepare the fruits. The task distribution will be as follows.

- The human will prepare the dough, meaning that she/he will make it and put it in the mould.
- The robot will prepare the mixture, meaning that it will make it and put it in the mould.
- The human will then prepare the fruits, by cutting them and putting them in the cooking mould.
- Then they will handle the dough for the top of the pie.
- Finally, the robot will take care of the baking by putting the pie in the oven and setting the timer.

After the first step, the human will have acquired knowledge on how to make the dough. Consequently, during the execution, when reaching for the second dough (for the top), the robot asks the user if he needs help. We imagine he answers "no". The robot doesn't explain it and the human has a level *INTERMEDIATE* on this task after succeeding with it. Concerning the second human task, it will be represented in the knowledge as *<human1, PrepareFruits, [fruit], VALUE>*. Indeed we consider that the process is the same for any fruit (cutting and putting in the mould) so we use the class fruit instead of the instance apple or banana. After cooking the first pie, the robot generates a plan to cook the banana pie. This time, we consider that both agents can perform all the tasks (everything is reachable for both). The banana pie is using the same tasks with different parameters. The mixture is different, so are the fruits but the method to prepare them is the same (cut and spread in a mould). Also the banana pie won't have dough on the top and will have a different baking time. The plan generated is presented in Figure 4. We can observe that the policy favors the efficiency by giving known tasks to the user (*PrepareDough* and *PrepareFruits*). During the execution, as the user has an *INTERMEDIATE* knowledge level on how to prepare dough, the robot will not explain it. Concerning the *prepareFruits* task, the robot will propose explanation to the human since she/he has performed it only once.

## VIII. User Study and Discussion

### A. User Study

We have conducted a comparative user study in order to have a first evaluation of the perception by users of our system's adaptability. Two groups of users were asked to follow the two-pies scenario. The first group interacted with a simulated robot equipped with a basic system (BS). BS has the same behavior as our system without the agent knowledge awareness mechanisms. The second group interacted with a second system, called knowledge system (KS), showing a behavior similar to what our system provides. All participants were then asked to evaluate the interaction with respect to several criteria. The task distribution presented in Section VII-B is used for both systems to achieve the apple pie cooking example. Once they have cooked the first dessert, the robot generates a plan to cook the banana pie. In KS, the robot has the same behavior as our system and favors a task distribution for the banana pie where the human performs tasks he has already performed for the apple pie (preparing dough and fruits). In BS, we imagine that the robot could ask the human to make the mixture instead of the dough.

[1] Open-Source modules: http://homepages.laas.fr/gmilliez/hri2016/

[2] https://www.willowgarage.com/pages/pr2/overview

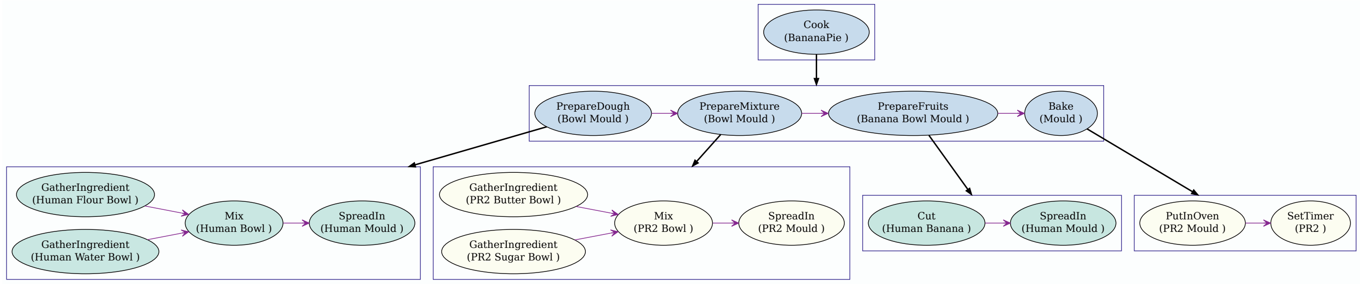


Fig. 4: Shared plan and associated HTN generated to collaboratively make a banana pie.

Two groups of 19 participants, from 18 to 60, interacted with each system. This was achieved thanks to an online user study[3], where we presented pictures of the task state and recordings of the robot's speech in French for each step of the interaction (as shown in Figure 5). On some steps, the user could choose the action to perform, making it possible to execute a wrong action leading to a replan from the robot. For simplicity, the replan was limited to cancel the wrong action before resuming the previous plan. Also, to focus on the adaptability of our solution from knowledge awareness in the different steps of the shared plan, we did not allow negotiation from the users. At the end of the simulated interaction, we have asked the same questions to both groups, concerning the adaptability of the system and the robot partner itself. The users gave marks along a Likert scale from one (disagree) to five (agree) to express their agreement with several statements (as shown in Figure 5). For instance, one question we asked: ”Did you feel that the robot adapts its explanation to the knowledge you acquired during the interaction?”.

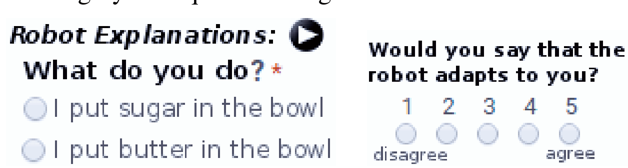


Fig. 5: *Left*: The user listens to recorded robot explanation and chooses the action to take. *Right*: At the end, the user evaluates the interaction using a Likert scale.

## B. Results

We collected the answers for each form, and computed the mean along with the standard deviation and p-value to evaluate the reliability. The p-value was computed using a t-distribution with 18 degrees of freedom and considering that the mean of KS would be higher than BS as stated in our hypothesis and identical as null hypothesis. Figure 6 summarizes the results. Comparing users' answers, we can see that the robot adaptation on users' knowledge concerning the explanation is well perceived with a mean of 3.74 for KS against 2.05 for BS. The users interacting with KS globally noticed that the task distribution took their knowledge into account by giving a mean rating of 3.42 for KS and 2.58 for BS. The last question concerned the freedom to choose the way to perform the task. The mean was 2.58 for KS compared to 1.89 for BS, and the p-value was lower than 0.05. So even on that aspect there was an improvement. With KS, the users attributed a mean of 3.11 for the global adaptability of the system against 1.89 for the basic one. We also asked how the robot partner was perceived. While in BS, the robot is not perceived as more verbose (2.53 for KS against 2.47 for BS), people found the interaction slightly more natural (2.74 against 2.42) and the robot appeared smarter (2.79 against 2.26). Even if these results strengthen our idea, as the p-value is higher than 0.1 for the naturalness, it doesn't prove that there is a noticeable difference between the two systems. We believe that other aspects might have been taken into account by the users such as the speech itself and lead users of both experiments to perceive the interaction as not so natural. For instance, improving the verbalization with a synonym dictionary could help to get more significant results.

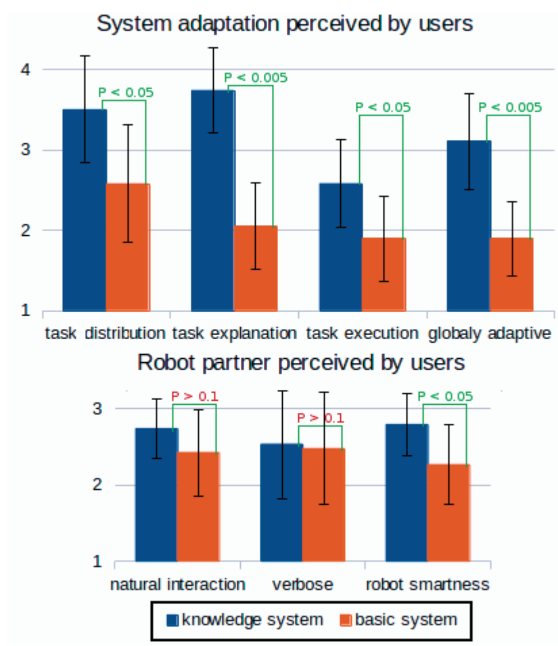


Fig. 6: Average users' rating of the interaction on several criteria. Blue for our system and red for basic system.

This study sheds light on how users were able to perceive the robot adaptation to their knowledge concerning task distribution, task explanation and monitoring. In addition, the robot partner was perceived as smarter and the interaction seemed a bit more natural to the users. However these first results need to be confirmed with study on a larger population. Also, as future work we envisage to conduct this users study on a real robot as we believe it would lead to more realistic opinions. In both studies, we asked the participants how the interaction

[3]User study for KS http://goo.gl/forms/qvbtu4vcFW, and BS http://goo.gl/forms/ZSvGcCi5le

could be improved. Several users suggested they would like to be able to choose which action to perform, putting forward the importance of the negotiation (we removed negotiation in the user study to focus on knowledge adaptation). A user suggested they would like to be informed about the progress from time to time. This can be easily added since at each moment the robot knows the number of nodes remaining in the shared plan to achieve the goal. Other comments concerned suggestions about the speech itself, the robot voice, intonation and chosen words, which were not the aim of the experiment but are indeed an important part of the interaction.

## IX. Conclusion

In this paper we have proposed a solution for a robot to manage a human-robot collaborative activity from plan generation to execution, guiding its human partner in an efficient and socially acceptable way. Our approach is based on a dynamic tracking and online updating of the partner's knowledge. It offers two policies, teaching or efficiency, which provide different levels of interactions. Our method incorporates the monitoring of high-level human actions, focusing on the completion of tasks instead of the details on how the task is completed. This provides some flexibility to the human when performing his/her assigned tasks.

We conducted a comparative online user study. Results were encouraging, users were able to perceive the adaptability of the robot on the three aspects of plan generation, task guidance and monitoring. Currently, this approach requires the robot to know all the methods to be performed (including each task corresponding decomposition). Adding the ability to teach the decomposition of tasks to the robot as in [25] would remove this restriction. For the negotiation part, we provide a first mechanism, assigning a different cost to a task when the user wants or does not want to perform it. However, this could be improved using previous work such as [26] which exposes a Propose/Evaluate/Modify framework to handle negotiation. Concerning monitoring, the current implementation uses a simple strategy based on assessment of task outcome after a given time. We plan to design more elaborate mechanisms to estimate the engagement of a human in a task and to better recognize when a monitored task has failed. The human could also inform the robot why he is not performing the task as expected. Finally, to improve the interactivity, it should be pertinent to allow the human to request explanations on the robot actions as in [27].

**Acknowledgments:** This work has been supported by l'Agence Nationale pour la Recherche under project reference ANR-12-CORD-0021 (MaRDi).

## References

[1] F. Warneken, F. Chen, and M. Tomasello, "Cooperative activities in young children and chimpanzees," *Child Development*, pp. 640–663, 2006.

[2] F. Warneken and M. Tomasello, "Helping and cooperation at 14 months of age," *Infancy*, vol. 11, no. 3, pp. 271–294, 2007.

[3] M. Tomasello, M. Carpenter, J. Call, T. Behne, and H. Moll, "Understanding and sharing intentions: The origins of cultural cognition," *Behavioral and Brain Sciences*, vol. 28, pp. 675–691, 10 2005.

[4] M. Tomasello, "Why we cooperate, michael tomasello. mit press, 2009. xviii + 206 pages.," *Economics and Philosophy*, vol. 27, pp. 183–190, 7 2011.

[5] S. A. Butterfill and N. Sebanz, "Editorial: Joint action: What is shared?," *Review of Philosophy and Psychology*, vol. 2, no. 2, pp. 137–146, 2011.

[6] R. Lallement, L. de Silva, and R. Alami, "Hatp: An htn planner for robotics," in *2nd ICAPS Workshop on Planning and Robotics*, 2014.

[7] S. Lallee *et al.*, "Cooperative human robot interaction systems: Iv. communication of shared plans with humans using gaze and speech," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 129–136, 2013.

[8] M. Petit *et al.*, "The coordinating role of language in real-time multimodal learning of cooperative tasks," *Autonomous Mental Development, IEEE Transactions on*, vol. 5, pp. 3–17, March 2013.

[9] M. Sorce *et al.*, "Proof of concept for a user-centered system for sharing cooperative plan knowledge over extended periods and crew changes in space-flight operations," in *IEEE RO-MAN*, 2015.

[10] D. Premack and G. Woodruff, "Does the chimpanzee have a theory of mind?," *Behavioral and brain sciences*, vol. 1, no. 04, pp. 515–526, 1978.

[11] B. Tversky and other, "Why do speakers mix perspectives?," *Spatial Cognition and Computation*, vol. 1, no. 4, pp. 399–412, 1999.

[12] F. U. Baron-Cohen S, Leslie AM, "Does the autistic child have a 'theory of mind'?," *Cognition*, vol. 21, no. 1, pp. 37 – 46, 1985.

[13] J. Guitton, M. Warnier, and R. Alami, "Belief management for hri planning," *BNC@ ECAI 2012*, p. 27, 2012.

[14] K. Talamadupula *et al.*, "Coordination in human-robot teams using mental modeling and plan recognition," in *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*, pp. 2957–2962, IEEE, 2014.

[15] C. Breazeal, M. Berlin, A. Brooks, J. Gray, and A. Thomaz, "Using Perspective Taking to Learn from Ambiguous Demonstrations," *Robotics and Autonomous Systems*, pp. 385–393, 2006.

[16] E. Ferreira, G. Milliez, F. Lefevre, and R. Alami, "Users' belief awareness in reinforcement learning-based situated human-robot dialogue management," in *IWSDS*, 2015.

[17] P. Brusilovskiy, "The construction and application of student models in intelligent tutoring systems," *Journal of computer and systems sciences international*, vol. 32, no. 1, pp. 70–89, 1994.

[18] P. Brusilovsky, S. Sosnovsky, and O. Shcherbinina, "User modeling in a distributed e-learning architecture," in *User Modeling 2005*, vol. 3538 of *Lecture Notes in Computer Science*, pp. 387–391, Springer Berlin Heidelberg, 2005.

[19] S. J. Levine and B. C. Williams, "Concurrent plan recognition and execution for human-robot teams," in *Twenty-Fourth International Conference on Automated Planning and Scheduling*, 2014.

[20] E. Karpas, S. J. Levine, P. Yu, and B. C. Williams, "Robust execution of plans for human-robot teams," in *Twenty-Fifth International Conference on Automated Planning and Scheduling*, 2015.

[21] J. Shah, J. Wiken, B. Williams, and C. Breazeal, "Improved human-robot team performance using chaski, a human-inspired plan execution system," in *international conference on Human-robot interaction*, 2011.

[22] A. S. Clair and M. Mataric, "How robot verbal feedback can improve team performance in human-robot task collaborations,"

[23] N. Sebanz, H. Bekkering, and G. Knoblich, "Joint action: bodies and minds moving together," *Trends in cognitive sciences*, vol. 10, no. 2, pp. 70–76, 2006.

[24] G. Milliez, M. Warnier, A. Clodic, and R. Alami, "A framework for endowing interactive robot with reasoning capabilities about perspective-taking and belief management.," in *Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication*, 2014.

[25] A. Mohseni-Kabir, C. Rich, S. Chernova, C. L. Sidner, and D. Miller, "Interactive hierarchical task learning from a single demonstration," in *Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, HRI '15, ACM, 2015.

[26] J. Chu-Carroll and S. Carberry, "Conflict resolution in collaborative planning dialogs," *International Journal of Human-Computer Studies*, vol. 53, no. 6, pp. 969–1015, 2000.

[27] M. Lomas *et al.*, "Explaining robot actions," in *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction*, HRI '12, 2012.