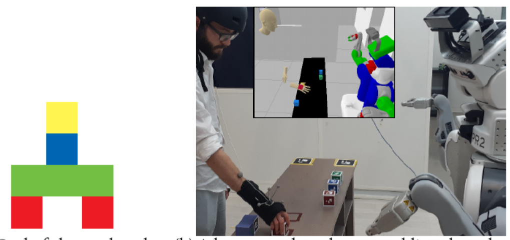
HATP/EHDA: A Robot Task Planner Anticipating and Eliciting Human Decisions and Actions

Guilhem Buisan, Anthony Favier, Amandine Mayima, Rachid Alami

HATP/EHDA: A Robot Task Planner Anticipating and Eliciting Human Decisions and Actions

Guilhem Buisan[1], Anthony Favier[1,2], Amandine Mayima[1] and Rachid Alami[1,2]*

Abstract— **The variety and complexity of tasks autonomous robots can tackle is constantly increasing, yet we seldom see robots collaborating with humans. Indeed, humans are either requested for punctual help or are given the lead on the whole task. We propose a human-aware task planning approach allowing the robot to plan for a task while also considering and emulating the human decision, action, and reaction processes. Our approach, named Human-Aware Task Planner with Emulation of Human Decisions and Actions (HATP/EHDA), is based on the exploration of multiple hierarchical tasks networks albeit differently whether the agent is considered to be controllable (the robot) or uncontrollable (the human). We present the rationale of our approach along with a formalization and show its potential on an illustrative example.**



(a) Goal of the stack task (b) A human and a robot assembling the cube stack

Fig. 1. (a) A robot and a human assembling a (b) cube stack. The robot environment estimation is depicted on top. For the best efficiency, the robot must plan for and adapt to the human possible decision and action processes.

I. MOTIVATION

Imagine that you are working in a factory and order your robotic assistant to assemble a small piece of furniture (represented by a stack, Fig. 1(a)) including several components (represented by cubes, Fig. 1(b)). Some components are only accessible by one, others are reachable by both of you and still others are reachable by a another human, carrying her own task. Many ways of realizing this task are available for the robot: depending on how you formulated your request, it can assume that you will help it or not; it can try to perform it on its own, if you are busy doing something else, even if it takes longer; it can request punctual help from the other human, to more efficiently perform the task if verbal communications are feasible; finally, it can also share the entire task with this third agent, fully involving her in the assembly. All these solutions may or may not be feasible, and can be more or less efficient, depending on the world state, the goal, and more importantly the estimated decision and action processes of the surrounding humans.

To tackle these issues, we present a novel task planning approach dedicated to HRI which, by planning for both the human and the robot, tries to satisfy multiple objectives:

1) **Plan with or without assuming a prior shared goal**. The robot and the human are not always sharing a goal. Our planner can balance between integrating the sharing of a goal with a human (assumed to be collaborative) in the plan and making the robot do the task alone, or integrate the possibility to ask for punctual human help.

2) **Model the human decision processes**. When taking part in a task, a human (assumed willing to collaborate) will also plan to reach their (potentially shared) goal. Our robot must be able to account for this to provide plans that are expected by the human partner and explainable to them.

3) **Help the human decision, but not compel it**. By modeling the human decision processes and their potential effects, the planner will need to consider various courses of action, leaving more latitude to the human and avoiding replanning

4) **Act and decide on the different agents' beliefs**. It is important to be able to represent actions as having different effects on the beliefs of the robot or the human. Indeed, some robot actions are partially or not observable by the human. Therefore, when performing them, the human has no way of knowing the complete new world state. Moreover, these effects and their observability often depend on the current world state, which representation must be supported by the planner. Finally, decisions made while planning may require to reason on both the robot and the human beliefs (*e.g.* knowledge alignment communication actions, robot asking questions).

To do so, we took inspiration from previous work on human task modelling and human-aware task planning to build a task planner focusing on human-robot collaboration. Our planner is able to explore in a distinct manner the deliberation and plan elaboration processes of the robot and the human, in order to build robot plans and anticipate the decisions, actions and reactions of the human. Moreover, it maintains one belief base per agent (human or robot).Actions preconditions and effects can be expressed in any of these belief bases. It allows to represent situational or inherently non observable actions from agents, knowledge transfer actions, and to detect beliefs divergences and plan accordingly. Our scheme is designed to provide a suitable framework enabling the anticipation of the beliefs, potential decisions, reactions and contributions to a shared goal or to an interaction

[1] LAAS-CNRS, Universite de Toulouse, CNRS, Toulouse, France

[2] This work was partially supported by the Artificial and Natural Intelligence Toulouse Institute (ANITI) under grant agreement No: ANR-19-PI3A-0004.

* Contacts: {guilhem.buisan, anthony.favier, amandine.mayima, rachid.alami}@laas.fr

situation of both agents. This overall anticipation process takes place within the robot decisional activity.

In the sequel, we first discuss related work before presenting our approach, its intents and rationale. Then, we provide a formalization of our planner scheme. Finally, some examples are presented before concluding.

II. BACKGROUND

To plan for the human decisions and actions, we need to model the humans decision and action processes. Thus, we first present common ways of modelling human activity excerpted from human-computer interaction literature. Then, we review several contributions to elaborate shared human-robot plans.

A. *Human Task Modeling*

Multiple approaches try to model the human activity. They are mainly used to design a system that integrates gracefully in human tasks, in order to improve their performance. These tasks can be seen as hierarchical, where abstract ones decompose into smaller, more concrete ones.

A common way of representing human activity and interaction with computer at high abstraction level is by using *task models*. The hierarchical structure of human activity was first exploited by Annett and Duncan [1]. They state that tasks can be described at several levels of abstraction until a certain criterion is met. Each task can thus be refined into subtasks detailing the procedure followed by the human to achieve the higher level task.

Task modeling has then evolved to introduce interaction with systems, produced and needed information, potential errors and a wide variety of operators specifying how tasks interacts with each other during their execution. Task models are now commonly used in user-centered and user interface designing processes. Most advanced notations include *ConcurTaskTrees* [2] and *HAMSTERS* [3].

These models are used to design or to evaluate interactive systems. They allow the designer to better understand the user task or to study the user workflow using their system. However, these models contain too little information for a system to be able to reason and take decision on them (either in planning or acting).

We drew inspiration from these models, sharing commonalities with Hierarchical Task Networks (HTNs), to represent human decision processes in HATP/EHDA.

B. *Planning a Task for Both the Human and The Robot*

While human activity modeling and autonomous planning have been studied separately for decades, there are still only few systems proposing to incorporate human activity into planning for intricate interactive tasks, leading to *shared plans* [4]. Planning for both a robotic agent and a human differs largely from multi-robot planning. Indeed, while the robotic agents are planning for itself and will surely execute the plan, the human is not directly controllable (making them follow the plan may require the robot to at least communicate and perhaps negotiate) and can also have their own plan they are trying to execute.

Most approaches to the so-called *human-aware task planning* problem assume a fully controllable human, willing to participate in the accomplishment of a common goal [5], [6]. Then, information and plan sharing are done in a post-processing step or at execution time [7], [8].

The Hierarchical Agent-based Task Planner (HATP) proposes a hierarchical approach to multi agents task planning [9], [10]. This HTN-based planner is able to elaborate a multi-agents plan based on a single HTN tree. Moreover, it maintains one beliefs base per agent allowing to write task decomposition rules and actions preconditions and effects in any agent beliefs base. Finally, HATP also computes costs for the plans found based on action costs and predefined social rules. However, HATP assumes that a shared goal has been established between the human and the robot prior to the planning process and that the generated plan will be shared with the human before the execution [11]. Indeed, HATP does not represent the human as an agent having a separate decision process that may lead to diverging plans without robot communication.

Other approaches are explicitly considering an external human model, which can be used to predict future human actions, and plan accordingly. Hoffman and Breazeal present an approach where the robot anticipates human actions to provide efficient support [12]. The environment is represented as a first-order Markov process (implying that the human can be represented as an hidden Markov model acting upon the, observable, environment). Buckingham *et al.* propose a planning scheme questioning humans mental models returning the effects of expected future humans actions [13]. The planner is then able to determine a robot policy influencing the humans actions. In this work, they show how this framework is able to cope with interactive tasks even without assuming that the human is collaborating. Similarly, Unhelkar *et al.* proposed a POMDP-based approach called CommPlan [14]. The POMDP is built using a user defined MDP (Markov Decision Process) representing the collaborative task and an AMM (Agent Markov Model) representing the human decision-making process. This POMDP is then solved to produce a robot policy which, inter alia, decides when the robot has to communicate about its beliefs, to question the human about theirs and to ask the human to perform an action. Besides, the human AMM is not only specified by the programmer but also refined during the interaction via learning. However, these approaches consider the human model as an oracle on which reasoning is hardly possible during the planning process. Moreover communication models and human decision processes are coarse and do not offer any insight for helping the robot planning process.

Additionally, to cope with the uncertainty of the human knowledge, Petrick and Foster propose to use conditional planning allowing to plan for incomplete information [15]. By doing so, the planner elaborates a plan for the robot accounting for multiple possible human choices, and depending on the knowledge received the execution component can execute the right branch of the plan.

Likewise, Sanelli *et al.* ([16]) present an approach not only elaborating conditional plans for the robot depending on the possible human choices (*e.g.* the choice of activity the human wants to perform), but they are able to transform this conditional plan into a Petri net plan to handle its execution. This contribution is inspired by a previous work of Nardi and Iocchi ([17]) in which they present a method for transforming (linear) joint plan into a Petri net plan managing its execution. Interestingly, the human actions from the plan are changed into a part of the Petri net where the robot elicits the action (*e.g.* via a verbal communication) if the human does not perform it by themselves. However, these approaches only request the human to make single actions, instead of sharing a high level goal, which can become unpleasant if done repeatedly.

Finally, Chakraborti *et al.* use both the robot model and the estimation of the model the human has of the robot to improve plan explicability [18]. Indeed, they propose a novel approach called *model reconciliation* which they present as a classical planning problem. In this problem, the goal is to make identical both the optimal plans generated via the robot model and the human estimation of the robot model. To do so, they define a list of operators on the models in order to modify them until the plans match. However, it only has been applied to robot plans and not to joint plans. Indeed, the generated plans contain only robot actions, and the robot and the human do not directly collaborate in the presented tasks.

III. DESCRIPTION

A. Rationale

We separate the agents who may take part in a given task into two categories: the controllable agent (*i.e.* the robot) for which the planner needs to select the best course of actions to generate a plan; and the uncontrollable agent (*i.e.* the human) on whom the planner has no direct control but, still, has a representation of their decision, action and reaction models.

The two agent types are fundamentally different:

1) the robot is controllable since the process is run by the robot itself,

2) the human agent is not controllable since the process can only "speculate" on their decisions and actions, but can model that the robot actions can still influence them and that some of them are observable by the robot,

3) the two agents are not equivalent, the robot agent role is to help, assist and facilitate human and to synthesize pertinent, legible and acceptable behavior.

We want to devise a planner allowing the controllable agent to plan for its actions while anticipating the decisions, actions and reactions of the uncontrollable agent. Moreover, we want the planner to be able to generate plans where the robot actions elicit situations calling for human decision, action and reaction, thus creating and anticipating collaboration and interaction.

This problem may be seen as a classical non deterministic planning problem, but enriched with the ability of the robot to model the actions, beliefs and decision process of the human. Thus, we have to consider distinct action models, beliefs and execution streams for each of the agents involved. HTN approaches have already been shown to be suitable for HRI as they allow to communicate about the plan more easily [10]. Therefore, we chose to use HTN planning for both the controllable and uncontrollable agents. HTN planning aims at decomposing abstract tasks into atomic, primitive tasks by choosing from a list of available context-dependent refinements for each abstract task, ensuring that preconditions and effects of refined primitives tasks are satisfied throughout the created plan. Similarly to HATP [6], our planner elaborates a plan with several streams of actions each assigned to an agent involved in the task. But while in HATP, all the streams are built starting from an initial root node corresponding to a shared goal between all agents, our planner starts from multiple initial root nodes corresponding to the first task of the initial agenda of each different agents.

B. Definitions

The main structure manipulated by our planner is the **agent**, more precisely two will be represented, the *human* and the *robot*. Each agent has their own **beliefs**, **action model**, **agenda**, **plan** and **triggers**. The planner has to use their action models and beliefs to decompose the tasks in their agenda into primitive tasks (actions) that are inserted in their plan. By doing so, it also has to update the beliefs of each agent and to model their reaction by executing the triggers.

a) **Agents:** First, we define an agent state as a tuple $\sigma_\alpha = \langle d_\alpha, \pi_\alpha, s_\alpha \rangle$, with d_α the agenda, π_α the partial plan and s_α the beliefs of the agent α (more details are presented in what follows). Then, we define an agent as being $\alpha = \langle \text{name}_\alpha, \sigma_\alpha, \Lambda_\alpha, Tr_\alpha \rangle$, with name_α the agent name, σ_α the agent state, Λ_α the action model and Tr_α the triggers of the agent α (detailed in what follows). Then we define two agents: the controllable one — the *robot* — ; and the uncontrollable one — the *human* —. We have $\sigma = \langle \sigma_{robot}, \sigma_{human} \rangle$ representing an agents state, being the state of all the agents at a certain plan step. Let Σ be the set of all the possible agents states.

b) **Beliefs:** Let S be the set of all possible world states, we call beliefs of an agent α the state $s_\alpha \in S$ in which this agent thinks the world is in. It is important to note that the state of the controllable agent (robot) is assumed to be the real world state estimation for the planner, as we consider the planner as being part of the controllable agent.

c) **Action models:** We represent the action model of an agent α as $\Lambda_\alpha = \langle Op_\alpha, Ab_\alpha, Me_\alpha \rangle$ where Op_α are the primitive tasks (*i.e.* operators, actions) that the agent α can perform, Ab_α the set of abstract tasks and Me_α are the methods (*i.e.* decompositions) describing how an agent α can perform an abstract task through a refinement process. It is important to note that while this representation makes a clear distinction between the robot and the human tasks, it does not prevent representing joint abstract tasks or tasks that can be either done by one or the other agent. Indeed, as we show later, complementary abstract tasks can be represented and some tasks can have the same operational model even if they are not in the same agent action model.

More precisely, the primitive tasks (operators) are defined as functions: $Op \ni o : \Sigma \rightarrow \Sigma \cup \perp$ which produce new agents state, being the effect of the application of the primitive task, or *false* if the task is not applicable. We represent operators as being instantaneous (or all having the same duration) in their realization. With this definition of operators, we are able to represent action effecting the beliefs of any agents (*e.g.* depending on the *observability* of a robot action, the operator will or will not update the beliefs of the human). Moreover, we can represent actions whose only effects are knowledge sharing (*e.g.* verbal communication for belief alignment).

Then, methods are defined as tuple, containing an abstract task and a decomposition function: $Me \ni m = \langle \tau, \delta \rangle$ with $\tau \in Ab$ and $\delta : \Sigma \rightarrow (Op \cup Ab)^n \cup () \cup \perp$ with $n \in \mathbb{N}^*$, which, depending on agents state, decompose the abstract task returning a sequence of tasks (primitive or abstract), an empty sequence if the abstract task does not need to be decomposed, or *false* if the task cannot be decomposed in the current state. Multiple methods can address the same abstract task, the goal of the HTN planner is then to choose the right one to create a plan.

d) **Agents agendas and plans:** An agenda d_α and a plan π_α (this agent only stream of actions) are defined for each agent α. The agenda d_α is a sequence of tasks (abstract or primitive) having to be performed by the agent. The plan π_α is a sequence of primitive tasks, built from the agenda, which the agent has to perform. The links of actions order between the two streams of actions (plans) are kept in each plan, allowing for coordination.

e) **Agent triggers:** Finally, we define for each agent α a set of so-called *trigger functions* Tr_α. These trigger functions aim at representing reactions of agents to certain situations (subsets of world states). They are useful to model event-driven behavior, as in PRS [19], when a specific world state *triggers* a reaction from an agent. Besides, these triggers can be used to represent social norms as defined in [20], where the user can specify literals which, if true in the world state during the planning process, add some specific robot actions to the plan.

Trigger functions are defined as: $Tr \ni t : \Sigma \rightarrow (Op \cup Ab)^n \cup ()$ with $n \in \mathbb{N}^*$, returning a sequence of tasks to be inserted in an agent agenda as a reaction to specific agents state. For now, the tasks returned by a trigger function are added on top of the agenda, thus preempting any task that may have started to be decomposed. A considered solution is to support the flagging of some abstract tasks in the domain as being *atomic*. We can then prevent the tasks returned by a trigger to be inserted between any tasks resulting from the decomposition of an atomic task.

IV. THE PLANNING PROCESS

To start planning, HATP/EHDA must be given the two action models (the robot and the human HTNs), the initial beliefs of both agents (which can differ) and the initial agenda of both agents. The initial agenda of the robot represents the task to decompose, while the agenda of the human represents any task the human is estimated to be commited to. If a shared goal has been established prior to planning between the robot and the human (*e.g.* the human asking to perform a task with the robot), the agenda of both agent will be filled with the same task.

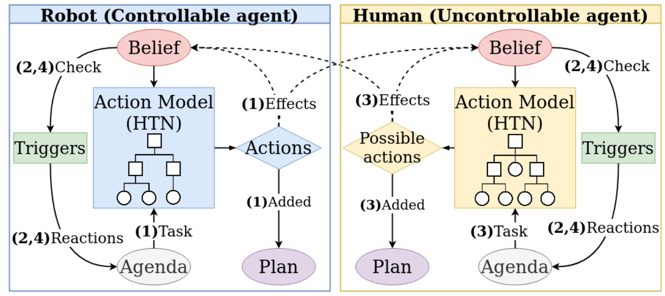


Fig. 2. The HTNs exploration, as explained in IV-A, consists in iterative loops of four steps : (1) Get possible robot actions from the robot HTN, add them in the plan and apply their specific effects on the H & R beliefs, (2) Check Triggers and add the reactions in the corresponding agendas, (3) Get possible human actions based on his/her updated (estimated) beliefs, add them in the plan and apply their effects on the H & R beliefs, (4) Check Triggers again and add the reactions in the corresponding agendas.

The planning process is done in three parts: (1) First, both HTNs are explored in a turn taking fashion, resulting in a valid joint plans tree. (2) Then, based on this tree, robot actions are selected according to action, plan-wide and social costs, resulting in a conditional plan, where at each step multiple human actions can be performed but only one robot action is set. (3) Finally, causal and threat links are added between actions of the conditional plan to ease its execution.

A. HTNs Exploration

The robot HTN exploration is a pretty standard depth-first algorithm. The first task λ from its agenda d_{robot} is popped, then if it is an abstract task $\lambda \in Ab$, all the applicable methods are applied, and their results are prepended to the agenda, thus giving new agents state (with the same beliefs as the previous ones but with the robot agenda updated) and branching our search space. We recursively iterate with the new task popped from the new robot agenda. Eventually, the popped task will be a primitive one $\lambda \in Op$, its function will then be applied to the currently explored agent states. If it returns *false* ($\perp$), the action is not applicable, and the exploration backtracks to another decomposition of an abstract task. However, if the action is applicable (returns a new agents state), the action is added to the robot plan and the triggers are run for each agent, updating their agenda if necessary. Then, the human HTN is queried to get their possible next actions from this new agents state. The possible actions found are added to the human plan, and, for each possible new agents state, we apply the triggers of each agent then we continue the robot HTN exploration. This exploration continues until the robot agenda is empty, or all the branches return *false*. The HTNs exploration process is summarized on Fig.2.

The human HTN exploration differs from classical HTN planner as the goal is not to produce a complete plan, but rather to list all the actions the human is likely to perform in a given agents state. We recursively decompose the first task of the human agenda d_{human} with every applicable method, until we reach an applicable operator. All the operators from all the applicable decompositions are returned to the robot HTN exploration and applied.

Two special cases are handled during the exploration. If the human agenda is empty whereas the robot one is not, the exploration returns a default action *IDLE* for the human (which does not modify agents beliefs nor agendas). This action represents the non-involvement of the human in a task. Besides, if no applicable action is found for the human a default action *WAIT* is returned (which does not modify agents nor agendas). This action represents the impossibility of the human to act in the current situation, making them wait for the robot to proceed. This default action can also be used in a domain to represent the human decision to wait for the robot to act.

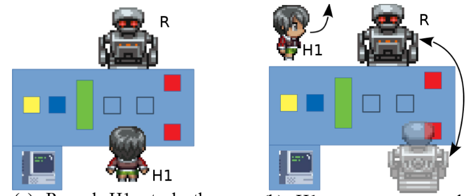
Once the robot agenda is emptied, the agents state is set as a success, the plan is added to the valid plans tree and the search can be continued until no decomposition is left for any task.

B. Conditional Plan Selection

Once this exhaustive search has been done, the result is a valid plans tree of alternating robot and human feasible actions along with their current beliefs leading to a task completion. The goal of this second planning step is to select robot actions such as each human action in the plan has only one robot action as a child. To do so, we define a cost function $cost : \sigma \times Op \mapsto \mathbb{R}^+$ representing the cost of an action in a specific state. The data structure is now similar to a two players game tree. However, *MinMax* approaches are not suitable here, as we are not in an adversarial setup but more into a collaborative one. Indeed, trying to minimize the maximal possible cost is assuming that the human will always do the actions leading to the worst plan. This defensive behavior could lead to non optimal plans. We thus propose to explore this tree differently.

Moreover, like in HATP we allow to define *social costs* functions. These functions take a complete human and robot sequence of actions (π_r and π_h) and return a cost ($\mathbb{R}^+$) which is added to the cost of the plan previously determined. By doing so, we can penalize non acceptable sequence of robot actions (*e.g.* serving a meal just after taking out the trash) or non satisfactory human required contribution (*e.g.* the robot requesting the human to perform small tasks multiple times instead of giving the big picture of the real task to perform).

The approach we propose for plan selection is to minimize the average cost. It represents the human potentially selecting any course of actions in their stream (while still respecting the action model defined in their HTN). The algorithm is given the root action of the task network previously generated and returns the cost of the conditional plan selected while having selected the robot actions in the task network minimizing the average cost of plans.



(a) R and H1 stack the cubes together for a shared goal requested by H1.
(b) H1 requests to stack the cubes and R acts alone.
(c) H1 requests to stack, R decides to punctually involve H2
(d) H1 request to stack, R decides to involve H2 with a shared goal.

Fig. 3. Cube stacking scene: A different plan is selected for each scenario, involving nearby humans in the less disturbing way possible.

(a) R and H1 build the stack together as a shared goal requested by H1.	**(b) H1 requests to stack the cubes and R acts alone.**
R-PickAndPlace(red, base) H1-PickAndPlace(red, base) R-PickAndPlace(green, bridge) H1-PickAndPlace(blue, top) R-PickAndPlace(yellow, top)	R-PickAndPlace(red, base) R-moveTo(red) R-PickAndPlace(red, base) R-moveTo(init) R-PickAndPlace(green, bridge) R-PickAndPlace(blue, top) R-PickAndPlace(yellow, top)
(c) H1 requests R to build the stack, R decides to punctually involve H2.	**(d) H1 requests R to build the stack, R decides to invite H2 to a shared goal.**
R-PickAndPlace(red, base) H2-IDLE R-AskPunctualHelp(red) H2-PickAndPlace(red, base) R-PickAndPlace(green, bridge) H2-IDLE R-PickAndPlace(blue, top) H2-IDLE R-PickAndPlace(yellow, top)	R-PickAndPlace(red, base) H2-IDLE R-AskSharedGoal() H2-PickAndPlace(red, base) R-PickAndPlace(green, bridge) H2-PickAndPlace(blue, top) R-PickAndPlace(yellow, top)

TABLE I

EXECUTION TRACE OF A SELECTED PLAN FOR EACH SCENARIO.

Finally, causal and threats links are computed on the conditional plan, to help the supervision in its execution.

V. EXAMPLES

To highlight the potential of our approach we present as example a cube stacking scene. The scene is depicted in four different scenarios on Fig. 3. The goal consists in stacking the colored cubes on the empty marks to match the colors of Fig. 1(a). All cubes placed in the middle of the table are reachable from anywhere. However, when close to one side a cube is only reachable from this specific side.

The partial robot and human actions models, as well as their exploration are presented in Fig. 4.

Each scenario is commented below with their corresponding selected plan shown in TABLE I.

a) H1 and R act together: First, the human sets a shared goal by asking the robot to stack cubes with him.

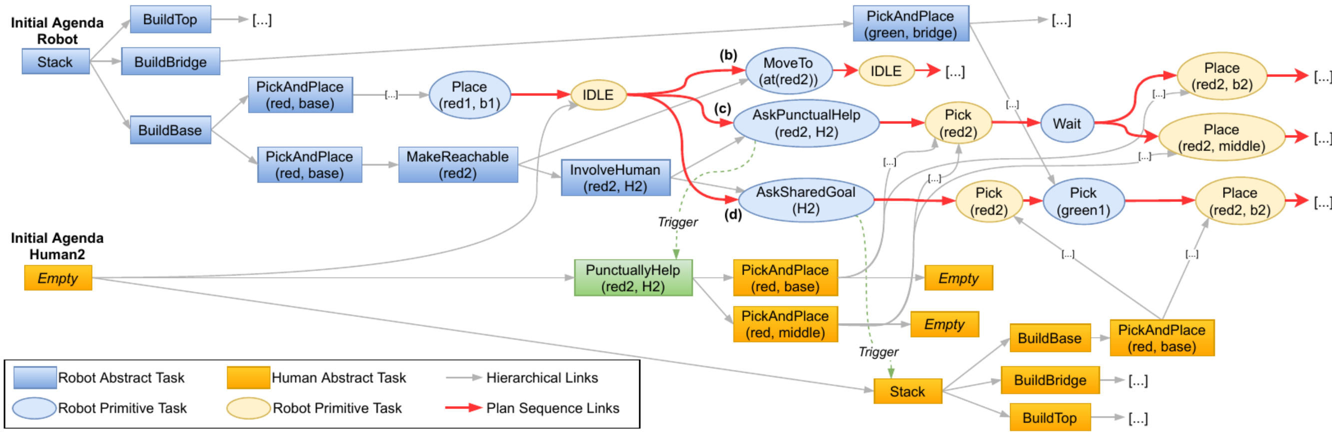


Fig. 4. Illustration of the incremental exploration of various courses of actions corresponding to scenarios depicted in Fig. 3(b), (c) and (d). Since H1 requests the robot to achieve the goal, it is not a shared goal, the robot agenda is filled with the task to achieve and the agenda of H2 starts empty.

Since it is a shared goal, Both human and robot agendas are initialized with the "Stack" task. Thus, the robot anticipates that the human will pick the unreachable second red cube by querying the human action model. The selected plan to collaboratively stack can be seen in TABLE I(a).

b) Robot acts alone: This time the human asks the robot to stack the cubes but then leaves the scene, the robot must act alone. Hence, the only applicable method to make the second red cube reachable is to move to the other side even though the movement action is expensive (we can imagine a table way longer than shown on the figure).

c) Robot asks punctual help: The first human acts the same way but another one is present, not involved in the task. The robot starts exploring its HTN and, thanks to the presence of the other human, a new method is applicable allowing the robot to ask for help. It can either ask for punctual help or involve completely the other human in the task. Of course, just asking help for one cube is less costly than asking to build the whole stack together. However, asking help to someone not already involved in a common task is still expensive since they have to put themselves in the context of the task. Yet, this punctual help is actually less costly for the robot than moving to the other side so the robot chooses to ask for punctual help. Note that we model the fact that after being asked to punctually help, the human can either stack the cube herself or just make it reachable to the robot by placing it in the middle. Only the first branch is shown in TABLE I(c) but the selected plan is in fact conditional with two branches as depicted in Fig. 4.

d) Robot invites Human2 to share a goal: Same initial setup but now two cubes are out of reach. Asking for punctual help is still less costly than moving around the table. Nevertheless, this disturbs the human so each new punctual request becomes more expensive than before. Thus, this time setting a shared goal becomes less costly for the robot than asking twice for punctual help.

Using HATP/EHDA on a real collaborative setting

HATP/EHDA has been integrated into a complete robotic architecture dedicated to human-robot collaboration [21], inspired by [22]. Besides HATP/EHDA, the architecture embeds a situation assessment component [23] together with a semantic knowledge management [24] to assess the state of the environment and estimate human beliefs. When the supervision decides of a task to perform and if this task corresponds to a human-robot shared goal or an individual robot goal, it invokes HATP/EHDA. Then, it controls the execution of the obtained plans. Their structure allows the controller to determine in which condition it has to simply wait for the human to choose or to act, or if it has to ask them to make a decision. An attached video illustrates several situations where it is shown that the human can make a choice or another in the case of a shared goal (Fig. 1). It also shows the case where the robot decides how and when to involve a human in the realization of a task.

VI. CONCLUSION

HATP/EHDA is a task planning approach allowing a robot to plan its actions while accounting for human decision, action and reaction, resulting in a conditional joint plan. To do so, a robot HTN is explored and human HTN models are queried (corresponding to human task modeling) to retrieve the possible human decisions, and actions. In fact, both human and robot action effects and preconditions can be expressed in either agent beliefs, allowing to predict, maintain or realign any belief divergence during the task planning process. We claim that HATP/EHDA planning framework is well suited and can be tuned to cover most of the human-robot planning problems and schemes previously envisaged in the literature: planning for human-robot while assuming a shared goal ([10], [14], [17]), planning in conditions where the robot can help and facilitate known human activities ([12]), planning for the robot alone and deciding when and in which conditions it can request help from human ([25]), integration of communication acts in the plans ([26], [7], [27]), incremental human-robot plan negotiation ([11], [18]) and determining which decision should be better postponed to leave more latitude to the human partner ([28]).

REFERENCES

[1] J. Annett and K. D. Duncan, "Task analysis and training design." *Journal of Occupational Psychology*, 1967.

[2] F. Paternò, "Concurtasktrees: an engineered notation for task models," *The handbook of task analysis for human-computer interaction*, pp. 483–503, 2004.

[3] C. Martinie, P. Palanque, E. Bouzekri, A. Cockburn, A. Canny, and E. Barboni, "Analysing and demonstrating tool-supported customizable task notations," *Proceedings of the ACM on Human-Computer Interaction*, vol. 3, no. EICS, pp. 1–26, 2019.

[4] B. J. Grosz and S. Kraus, "Collaborative plans for complex group action," *Artificial Intelligence*, vol. 86, no. 2, pp. 269–357, 1996.

[5] A. Torreño, E. Onaindia, A. Komenda, and M. Štolba, "Cooperative multi-agent planning: A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–32, 2017.

[6] E. Sebastiani, R. Lallement, R. Alami, and L. Iocchi, "Dealing with on-line human-robot negotiations in hierarchical agent-based task planner," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 27, no. 1, 2017.

[7] S. Devin and R. Alami, "An implemented theory of mind to improve human-robot shared plans execution," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2016, pp. 319–326.

[8] S. Nikolaidis, M. Kwon, J. Forlizzi, and S. Srinivasa, "Planning with verbal communication for human-robot collaboration," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 7, no. 3, pp. 1–21, 2018.

[9] S. Alili, R. Alami, and V. Montreuil, "A task planner for an autonomous social robot," in *Distributed autonomous robotic systems 8*. Springer, 2009, pp. 335–344.

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

[11] G. Milliez, R. Lallement, M. Fiore, and R. Alami, "Using human knowledge awareness to adapt collaborative plan generation, explanation and monitoring," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2016, pp. 43–50.

[12] G. Hoffman and C. Breazeal, "Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, 2007, pp. 1–8.

[13] D. Buckingham, M. Chita-Tegmark, and M. Scheutz, "Robot planning with mental models of co-present humans," in *International Conference on Social Robotics*. Springer, 2020, pp. 566–577.

[14] V. V. Unhelkar, S. Li, and J. A. Shah, "Decision-making for bidirectional communication in sequential human-robot collaborative tasks," in *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 2020, pp. 329–341.

[15] R. Petrick and M. E. Foster, "Planning for social interaction in a robot bartender domain," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 23, no. 1, 2013.

[16] V. Sanelli, M. Cashmore, D. Magazzeni, and L. Iocchi, "Short-term human-robot interaction through conditional planning and execution," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 27, no. 1, 2017.

[17] L. Nardi and L. Iocchi, "Representation and execution of social plans through human-robot collaboration," in *International Conference on Social Robotics*. Springer, 2014, pp. 266–275.

[18] T. Chakraborti, S. Sreedharan, Y. Zhang, and S. Kambhampati, "Plan explanations as model reconciliation: Moving beyond explanation as soliloquy," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 156–163. [Online]. Available: https://doi.org/10.24963/ijcai.2017/23

[19] F. F. Ingrand, R. Chatila, R. Alami, and F. Robert, "PRS: A high level supervision and control language for autonomous mobile robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 1996, pp. 43–49.

[20] F. M. Carlucci, L. Nardi, L. Iocchi, and D. Nardi, "Explicit representation of social norms for social robots," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 4191–4196.

[21] G. Sarthou, M. Amandine, B. Guilhem, B. Kathleen, and A. Clodic, "The director task: a psychology-inspired task to assess cognitive and interactive robot architectures," in *IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2021, pp. 1–8.

[22] S. Lemaignan, M. Warnier, E. A. Sisbot, A. Clodic, and R. Alami, "Artificial cognition for social human–robot interaction: An implementation," *Artificial Intelligence*, vol. 247, pp. 45–69, 2017.

[23] Y. Sallami, S. Lemaignan, A. Clodic, and R. Alami, "Simulation-based physics reasoning for consistent scene estimation in an hri context," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7834–7841.

[24] G. Sarthou, A. Clodic, and R. Alami, "Ontologenius: A long-term semantic memory for robotic agents," in *28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2019, pp. 1–8.

[25] S. Tellex, R. Knepper, A. Li, D. Rus, and N. Roy, "Asking for help using inverse semantics," in *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[26] G. Buisan, G. Sarthou, and R. Alami, "Human aware task planning using verbal communication feasibility and costs," in *International Conference on Social Robotics*. Springer, 2020, pp. 554–565.

[27] K. Belhassein, V. Fernández Castro, and A. Mayima, "A horizontal approach to communication for human-robot joint action: Towards situated and sustainable robotics," in *Culturally Sustainable Social Robotics*. IOS Press, 2020, pp. 204–214.

[28] S. Devin, A. Clodic, and R. Alami, "About decisions during human-robot shared plan achievement: Who should act and how?" in *International Conference on Social Robotics*. Springer, 2017, pp. 453–463.