A Task Planner for an Autonomous Social Robot

Samir Alili, Rachid Alami, Vincent Montreuil

A Task Planner for an Autonomous Social Robot

Samir Alili, Rachid Alami, and Vincent Montreuil

Abstract This paper describes a high-level task planner called HATP (for Human Aware Task Planner). which is designed to synthesize plans for human robot teamwork while respecting social conventions and adopting acceptable collaborative behaviors. We provide an overall description of HATP and discuss the features of its plans evaluation mechanisms. We also illustrate its performance on a real scenario achieved by a robot in interaction with a human partner in a realistic setup.

1 Introduction

One challenge in robotics research is to develop socially interactive and cooperative robots. *Fong et al*[8] define "Socially interactive robots" as entities which "operate as partners, peers or assistants, which means that they need to exhibit a certain degree of adaptability and flexibility to drive the interaction with a wide range of humans". This definition was made explicit by *Klein et al*[11] with what they called "ten challenges for human robot teamwork". We believe that several of these challenges can be handled at the task planning level. Indeed, the robot should be able (1) to signal in what tasks it can/wants participate, (2) to act in a predictable way to ensure human understanding of what it is doing, (3) to exhibit its status and its intentions, (4) to negotiate about tasks with its human partner in order to determine roles and select how to perform the tasks and (5) to deal with social conventions, as well as its human partner abilities and preferences.

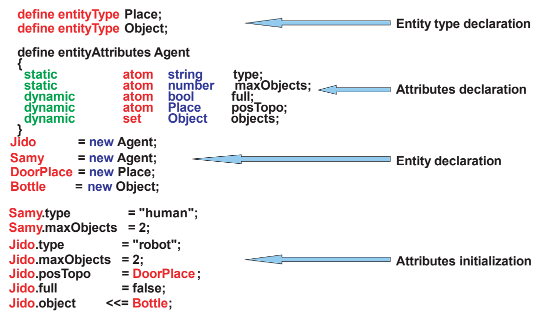
The work presented here consists of an approach, a model and an implemented planner called HATP (for Human Aware Task Planner) which has been specially designed for the interactive action between heterogeneous agents, in our case humans and robots. It is based on hierarchical task planning[10] which gives to the planner some ability to answer to certain issues defined above. It also integrates "social

LAAS-CNRS; University of Toulouse, 7 Avenue Colonel Roche, 31077 Toulouse Cedex 4, France. e-mail: `firstname.lastname@laas.fr`

behavior rules”, which orient the robot decision toward the synthesis of socially acceptable plans.

Section §2 presents HATP domain representation: world, methods and actions, and explains the rules that we have chosen to favor robot behaviours that are more suitable to Human-Robot Interaction (HRI) context. Section §3 gives details of plan evaluation to obtain social behaviors. Section §4 describes the implemented system and illustrates its use through several examples involving a mobile manipulator acting in interaction with a person. Finally, section §5 draws the main conclusions and discusses future work.

2 HATP planner description



```
define entityType Place;
define entityType Object;

define entityAttributes Agent
{
  static      atom   string    type;
  static      atom   number    maxObjects;
  dynamic     atom   bool      full;
  dynamic     atom   Place     posTopo;
  dynamic     set    Object    objects;
}
Jido      = new Agent;
Samy      = new Agent;
DoorPlace = new Place;
Bottle    = new Object;

Samy.type        = "human";
Samy.maxObjects  = 2;
Jido.type        = "robot";
Jido.maxObjects  = 2;
Jido.posTopo     = DoorPlace;
Jido.full        = false;
Jido.object     <<= Bottle;
```

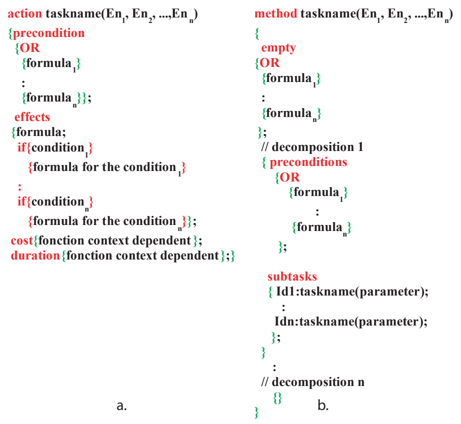
Fig. 1 World Data Base Representation in HATP: in this example, we define the entities *Place* and *Object*. The entity *Agent* is predefined. A set of atributes is associated to it: (type, maxObjects, posTop, full, objects) which represent respectively a type of Agent (human, robot...), its capacity, its current position, a boolean which indicates if the Agent has reached his max capacity, the list of objects that the agent is holding. We can see that *objects* and *Place* are of type *Entity*. The last lines illustrate the initialisation of the attributes of a robot called “Jido” and human called “Samy”.

HATP is a hierarchical task planner a problem is represented with a tuple $P =< W_0, D, TTL >$, where W_0 is initial world state, D is a domain and TTL represents a set or tree of tasks. The domain is represented by a pair $D =< T, R >$, where T is a set of tasks and R is a set of rules. We can distinguish two types of tasks in $T = Op \cup M$. Op correspond to basic primitives and M represent non-primitive tasks. R is a set of “social rules”. Each $r_i \in R$ is represented by tuple $r_i =< B_i, Pe_i^{Ag,ct}, pr_i >$ where B_i is the description of the rule, (i.e. description of patterns to recognize in the plan structure). $Pe_i^{Ag,ct}$ is a penalty to be added to the plan score if the rule r_i is violated in the plan solution P. The value depends on the agent who violates the rule and the context in which the rule has been violated. A priority $pr_i >$ is associated to each rule.

Let us give more details about the elements of HATP planning domain. First, we give the description of HATP syntax and a formal description of actions (also called operators) and methods. Then, we define the different forms of social rules available in HATP.

World database: In HATP the world description is represented by a set of entities $Wb =< En_1, En_2, En_3\ , ..., En_n >$, each entity is unique and defined by set attributes that are similar to predicates or state variables in classical planning. They can be of different types, each type is defined as tuple $Att =< Ct_1, Ct_2, Ct_3 >$, where Ct_1, Ct_2 and Ct_3 range respectively over the set $\{Static, Dynamic\}$[1], $\{Atom, Set\}$[2] and $\{Numeric, Boolean, String, Entity\}$. Figure 1 illustrates an example of a HATP domain description.[3]

Operators represent the actions i.e. tasks that can be directly executed. They are defined as a tuple $Op =< Pecd, E, C, Du >$ where $Pecd$ is a precondition formula, E is the effect on the world after action execution. $C : (Ct, agent) \rightarrow \Re^+$ is the cost function of action execution; it depends on the agent and on the context, $D \in \Re^+$ is the duration of action execution (see Fig. 2.a).



```
action taskname(En1, En2, ...,Enn)
{precondition
 {OR
  {formula1}
  :
  {formulan}};
 effects
 {formula;
  if{condition1}
    {formula for the condition1}
  :
  if{conditionn}
    {formula for the conditionn}};
 cost{fonction context dependent};
 duration{fonction context dependent};}
```

a.

```
method taskname(En1, En2, ...,Enn)
{
  empty
{OR
  {formula1}
  :
  {formulan}
 };
 // decomposition 1
 { preconditions
    {OR
       {formula1}
            :
        {formulan}
     };

   subtasks
   { Id1:taskname(parameter);
      :
    Idn:taskname(parameter);
   };
 }
   :
 // decomposition n
   {}
}
```

b.

Fig. 2 HATP method and action representation: we can note disjunctive set of preconditions and also conditional effects.

Methods represent the tasks that must be further decomposed into subtasks. They are defined as pairs $M =< goal, D >$. If the $goal$ formula is true then the goal is already reached and the method is reduced to `no-op`[4]. $D = \{D_1, \ldots, D_n\}$ is a set of alternative decompositions. Each decomposition D_i is a tuple $D_i =< Pecd_i, TL_i, C_i >$

[1] $Dynamic$ attributes can be modified during the planning process whereas $Static$ attributes can not.

[2] Set attributes can store several values whereas $Atom$ can have only one value at a time.

[3] The entity $Agent$ is predefined by the system.

[4] `no-op` is an Operator whose precondition is always true, whose effect is empty and which has a null duration and cost.

where $Pecd_i$ is a precondition, $TL_i \in T$ is a set of subtasks which decompose the method M according to the constraints C_i which are generally precedence constraints (see Fig. 2.b).

Social Rules: We define a social rule as a convention which associates penalties to a behaviour of an agent in a given context. In human environments most of these rules are implicit between the members of a community and they vary from a society or a culture to another. In HATP we have chosen six rule types that seem to be general rules. A given domain definition may include or not any number of rules:

Undesirable states correspond to world states which are dangerous, unpleasant or inadequate. Example: the robot lets the fridge door open, the robot puts an object on the floor which may impede the movement of other agents or, more dangerous, leaves the gas switched on.

Undesirable sequences some action combinations can conduct to an uncomfortable feeling for robot partners. For example, a plan in which the robot puts down an object and its partner picks it up immediately after might be considered as an awkward sequence if a the robot can hand over directly the object.

Bad decompositions this rule is used to select better choices at a high level of decomposition. The idea is that specific decompositions must remain possible but they must be used only if necessary. For example, when the robot has to leave an object for someone, it is better to place it on a "visible" place.

Effort balancing The idea is to establish a balance of effort among partners. If we have a team of two robot agents, the amount of effort could be the same for all the staff. If the team is composed of a human and a robot, then the robot must provide more effort than the human.

Timeouts the idea is to prevent long waiting time between two actions done by a same agent, because it is prejudicial for the plans quality.

Intricate links between plan streams Crossed links represent precedence links between actions of two different agents. Such links make plans fragile and introduce a lot of dependency between agents.

3 Social plan evaluation

In HATP, the planning process is composed of two threads; a refinement thread and an evaluation thread.

The refinement thread algorithm is responsible for the plan searching it is largely inspired from SHOP2 procedure [13]. The main differences between them are that HATP can have as input a set of tasks but also partial trees. HATP can reduces time exploration for solution by making the assumption that if two tasks are ”parallel” (i.e. they do not have causal link between them), it will try to obtain a final plan where they are as independent as possible (for more details, refer to[12]).

The evaluation thread is responsible of complete evaluation of plans; it is based on the Decision by Objectives theory, more precisely The Analytic Hierarchy Process (AHP)[9]. It is a generic method[15, 14], designed to facilitate decision making based on several heterogeneous criteria in complex environments. It decomposes the decision problem into a hierarchy of more easily comprehended sub-problems, each of which can be independently analyzed.

HATP plan evaluation: The criteria considered in the plan evaluation are social rules described in the HATP domain and the cost of all actions involved in a given plan. The evaluation in HATP in performed in two phases. First, construction of the criterion dominance vector and computation of the best score solution.

The first phase is done off-line. It consists in making pairwise comparisons between all criteria and in the synthesis of the dominance degree of each criterion. In order to do this, HATP uses priorities pr_i associated to each social rule. A priority value is between -8 and 8. 0 represents the reference point and is the default attached to the criteria cost. Then we can compute a comparison matrix A and also $W = [w_1, w_2, \dots, w_n]$ (n represent number of criterion) the eigenvector associated to it, while respecting the technique described in [9], Each w_i represents the weight of criteria i in plan evaluation (figure3).

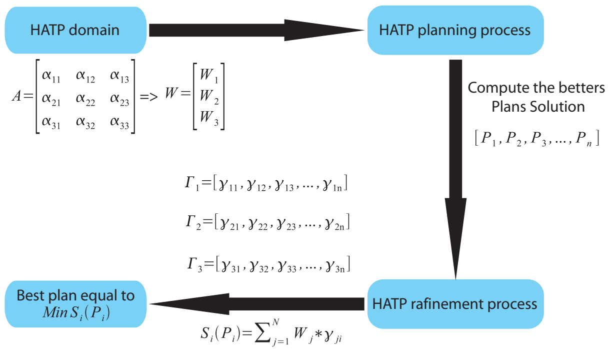

HATP domain
HATP planning process
$A=\begin{bmatrix}\alpha_{11} & \alpha_{12} & \alpha_{13}\\ \alpha_{21} & \alpha_{22} & \alpha_{23}\\ \alpha_{31} & \alpha_{32} & \alpha_{33}\end{bmatrix} \Rightarrow W=\begin{bmatrix}W_1\\ W_2\\ W_3\end{bmatrix}$
Compute the betters Plans Solution
$[P_1, P_2, P_3, \dots, P_n]$
$\Gamma_1=[\gamma_{11}, \gamma_{12}, \gamma_{13}, \dots, \gamma_{1n}]$
$\Gamma_2=[\gamma_{21}, \gamma_{22}, \gamma_{23}, \dots, \gamma_{2n}]$
$\Gamma_3=[\gamma_{31}, \gamma_{32}, \gamma_{33}, \dots, \gamma_{3n}]$
Best plan equal to $Min\, S_i(P_i)$
HATP rafinement process
$S_i(P_i)=\sum_{j=1}^{N} W_j * \gamma_{ji}$

Fig. 3 HATP plans evaluation: Let us assume a problem with three criteria α_1,α_2,α_3. We can construct a matrix A of pairwise comparison that provides eigenvector $W = [w_1, w_2, w_3]$ where w_i corresponds to α_i. The planning process produces a reference plan solution P_{ref} and a set P of all other solutions that have a null improvement. Compute the second local weight that gives three eigenvector Γ_1, Γ_2 and Γ_1. In each γ_{in} i refers to a criterion and n to the solution. Then we can calculate a plan sore/quality $S_i(P_i) = \sum_{j=1}^{3} w_j \times \gamma_{ji}$

The second phase is performed during the planning process. The first solution produced by the planner is taken as a reference solution. Now we compare all the new generated solutions with it, and choose the solution that improves the plan. The improvement is calculated in the following way: let P_{new} be a new solution plan and P_{ref} be reference plan, then the improvement of P_{new} is $\frac{\Sigma_{i=0}^{N} w_i * \frac{Val_i^{ref} - Val_i^{new}}{Val_i^{ref}}}{N}$. If the value of the average is negative P_{new} is immediately rejected and the process continues. Otherwise P_{new} becomes the new reference. In the case where we have an null improvement, we make pairwise comparison between all solutions with respect to all criteria. We obtain several matrices and eigenvectors, the number of eigenvectors is equal to the number of criteria; the dimension depends on the number of solutions (see Fig. 3). Then, a simple linear combination between different weighting gives the score that represents the plan quality. To determine the best solution we use the formula $S_i(P_i) = \sum_{j=1}^{N} w_j \times \gamma_{ji}$, which represents the plan evaluation. The best solution is the one that minimizes the score.

4 Experimental results

We illustrate below HATP performance through two scenarios. The first one is ”Fetch-And-Carry” and shows the influence of social rules on plan quality as well the capacity of HATP to handle contingency. The second scenario, named ”Serve-And-Clean” illustrates the performance of HATP implemented on “Jido”, a mobile manipulator that dedicated to human robot interaction and cooperation [5][5] Both

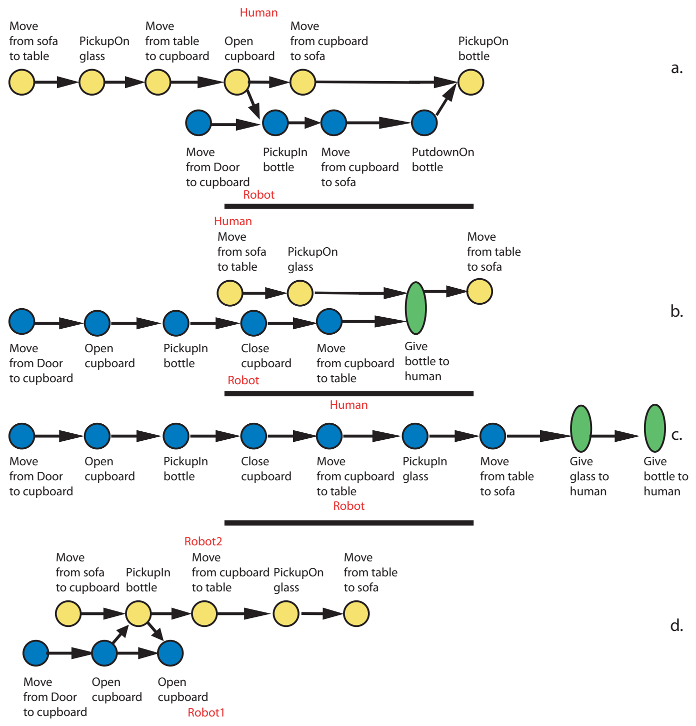


Fig. 4 Effect of social rules on the plans: the yellow circles represent human actions, the blue circles represent robot actions and the green ellipses represent joint actions.

[5] Some videos and pictures are available at (http://www.laas.fr/~salili/media.php).

scenarios run in a home. We have a "living-room" with 3 types entities: Agent (robot, human), furniture (cupboard, table, sofa. . .), and object (glass1, glass2, bottle). The robot has a symbolic as well as a geometric description of its environment. It is able to navigate robustly in its environment[7, 16], to detect and track persons[2, 3, 17, 18], and to recognize, localize and manipulate a (limited) set of objects. The overall robot architecture is described in [6, 5]. HATP interacts essentially with a robot supervisor, called **SHARY**[4, 6, 1].

"Fetch-And-Carry" scenario: In the initial state, the robot is at the door, the human siting is at the sofa and wants to have drink. To achieve this goal we choose to begin withe a partial plan, composed of three parts as : (1) the human must have a glass, (2) the human must have a bottle and (3) the human must reach the sofa. Besides, a specific decomposition is imposed for task (1): Jido must get the glass and transmit it to the human. The glasses are on the table and the bottle is in a closed cupboard.

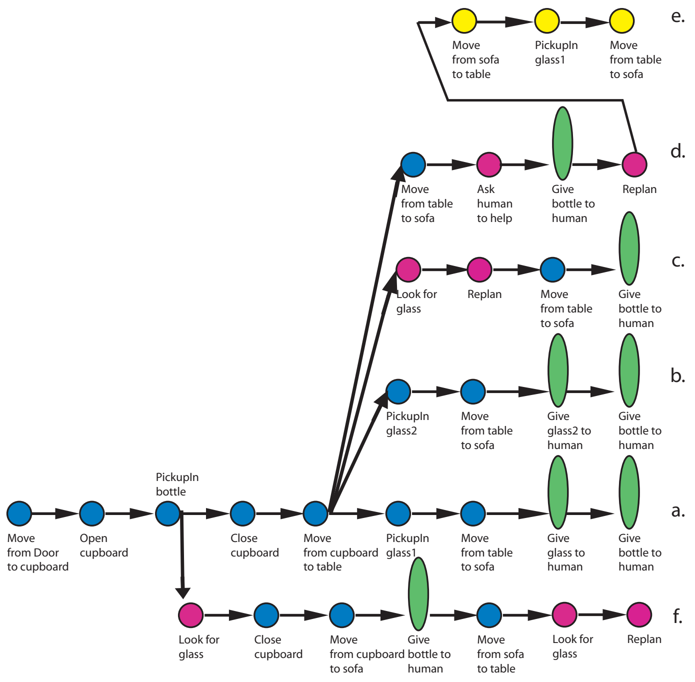


Fig. 5 HATP failure and contingency handling: the yellow circles represent human actions, the blue circles represent robot actions and the green ellipses represent joint actions. The purple action are a non-deterministic actions except for action "re-plan" which is a deferred planning request.

Figure 4.a illustrates the plan produced by HATP when social rules are inhibited. We can note that this plan is correct but it has a number of defects: the human opens the cupboard for the robot, the robot puts the bottle on the sofa and then the human picks it up. Now, if we introduce the following set of rules (avoid to have a bottle or glass put on the sofa, avoid to leave the cupboard open, avoid awkward sequences such that robot puts down an object and human picks it up just after, avoid that open/close sequences performed by different agents), we can see in Figure 4.b, that most defects have been corrected. However, this is not sufficient because we can observe that the human moves from the sofa to the table and comes back only for getting the glass, and there is a risk to have him wait for the robot to achieve its own part of the plan. If we introduce rules that try to avoid intricate synchronisations between the robot and the human we observe (Figure 4.c) that all defects have been corrected. Now if we replace the human by a robot and change the equilibrium point for the “Effort balancing” rule , we can see the effect of these modifications in Figure 4.d. We note that the system tries to make a good balance between the agents with respect to their abilities.

Now we will focus on HATP ability to handle contingency. In Figure 5.a the plan represents HATP answer for the same goal as above. If we consider the case where the object $Glass1$ is invisible or unreachable for Jido or Jido doesn’t know its current position, HATP will produce a plan where $Glass1$ is substituted by an other object of the same type $Glass2$ (Figure 5.b), if the two glasses are invisible to Jido, HATP produces the plan in Figure 5.c where the robot goes to last known position of the object and tries to look for it. The purple action ”Look for” represents a non-deterministic action. If the action succeeds (robot can see object) the supervisor SHARY will ask HATP for a new plan, else the supervisor will carry on in the same plan. Concerning the action ”re-plan” it is a fictive action for the supervisor. It is used in order to induce a new invocation of HATP after the execution of a non-deterministic action. In the case where the two glasses are unreachable, the robot is in a situation where it is unable to solve the problem himself. HATP produces a plan (Figure 5.d) where Jido asks the human for help. If the human agrees, the supervisor will ask HATP for a new plan (Figure 5.e) involving the human. If not, the supervisor abandons the task. Finally, if the position of the two glasses is unknown, then HATP produces a (Figure 5.f) where the robot will search for the objects in all possible positions.

“Serve-And-Clean” scenario: The human asks Jido to clean a bottle that is on the living-room table. We can see on Figure 6 the ”clean-up” task execution. At the left part of the figure we see the interaction between HATP and SHARY while the right part exhibits the interaction and plans execution of the two agents: Jido and the human called H1. At the beginning H1 asks Jido to clean the yellow bottle. SHARY replies by updating HATP world database and sending a planning request (CleanUp(YELLOWBOTTLE)), HATP answers by producing a that Jido begins to achieve under SHARY supervision. When Jido tries to take the bottle on the table, SHARY realizes that the bottle is unreachable. In this situation SHARY sends a new planning request. HATP produces a plan where Jido needs to interact with human to solve the situation. If H1 agrees, SHARY sends again the same planning request to HATP, which produces a new plan where the H1 is asked to get and give the bottle to Jido which will then will be able to navigate throw it in the trash bin.

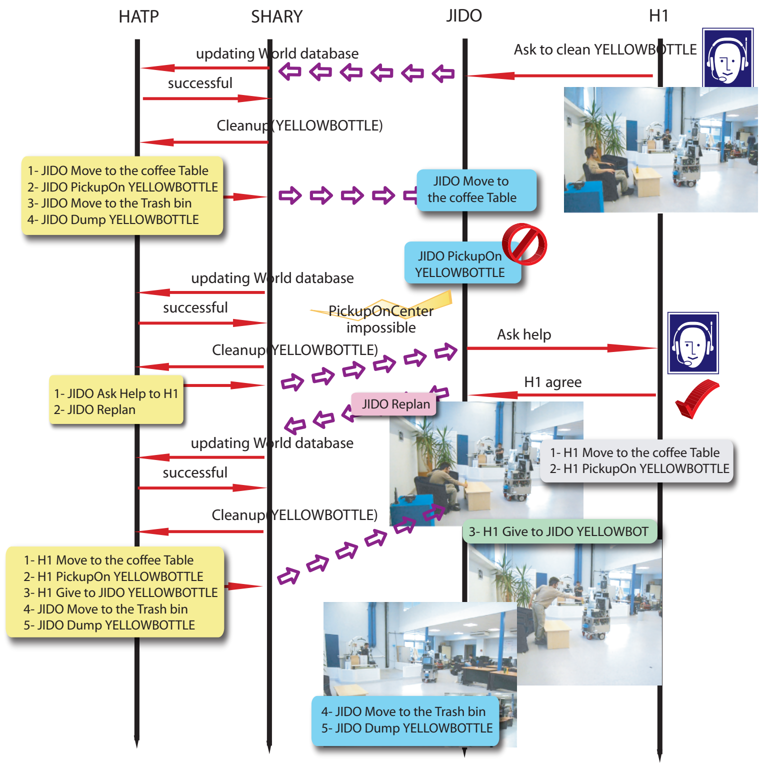

HATP
SHARY
JIDO
H1
Ask to clean YELLOWBOTTLE
updating World database
successful
Cleanup(YELLOWBOTTLE)
1- JIDO Move to the coffee Table
2- JIDO PickupOn YELLOWBOTTLE
3- JIDO Move to the Trash bin
4- JIDO Dump YELLOWBOTTLE
JIDO Move to the coffee Table
JIDO PickupOn YELLOWBOTTLE
updating World database
successful
PickupOnCenter impossible
Cleanup(YELLOWBOTTLE)
Ask help
H1 agree
1- JIDO Ask Help to H1
2- JIDO Replan
JIDO Replan
updating World database
successful
1- H1 Move to the coffee Table
2- H1 PickupOn YELLOWBOTTLE
Cleanup(YELLOWBOTTLE)
3- H1 Give to JIDO YELLOWBOT
1- H1 Move to the coffee Table
2- H1 PickupOn YELLOWBOTTLE
3- H1 Give to JIDO YELLOWBOTTLE
4- JIDO Move to the Trash bin
5- JIDO Dump YELLOWBOTTLE
4- JIDO Move to the Trash bin
5- JIDO Dump YELLOWBOTTLE

Fig. 6 `CLEAN-UP` task execution: At the top left of the figure, we see a simple version of the first HATP plan computed to achieve SHARY requests. In the middle and at the right side of the figure, we see the execution stream corresponding to this plan execution. This first plan failed due to robot inability to catch the bottle (even when it has perceived it). Then SHARY asks for a new feasible plan. HATP finds a plan with a higher cost and two streams and where the person is requested to participate by giving the bottle to Jido. The robot can then proceed and move to throw the bottle in the trash bin.

5 Conclusion and Future work

We have described in a robot task planner called HATP that has been designed for interactive and collaborative robotic applications. We have illustrated its ability to produce so-called "socially acceptable plans" for several agents while taking into account the agents abilities, preferences and desires and respecting social aspects related to the interaction of a robot with humans. We have seen its ability to handle contingency and to adopt, when necessary, proactive behaviors. We have also seen that, it is able to deal with partially specified plans, allowing the robot to comply with human desires and to re-plan after a failure.

Concerning future work, several aspects will be investigated. The first one is the development of heuristics in the refinement process in order to explore the most promising parts of the solution space and to allow the use of HATP in more complex situations with a more elaborate model of its activities. Another aspect is about the improvement of temporal constraints management as well as the elaboration of a task representation that deals explicitly with uncertainty.

References

1. A.Clodic, R.Alami, V.Montreuil, , and et al. A study of interaction between dialog and decision for human-robot collaborative task achievement. In *16th IEEE RO-MAN*, 2007.
2. C. Breazeal, A. Edsinger, P Fitzpatrick, and et al. Active vision for sociable robots. *IEEE Transactions on Systems, Man and Cybernetics,*, 2001.
3. B. Burger, I. Ferrane, and F. Lerasle. Multimodal interaction abilities for a robot companion. In *ICVS*, 2008.
4. A. Clodic. *Supervision pour un robot interactif : Action et Interaction pour un robot autonome en environnement humain*. PhD thesis, University of Toulouse, 2007.
5. A. Clodic, H. Cao, S. Alili, V. Montreuil, and et al. Shary: a supervision system adapted to human-robot interaction. *11th ISER*, 2008.
6. A. Clodic, M. Ransan, R. Alami, and V. Montreuil. A management of mutual belief for human-robot interaction. In *IEEE SMC*, 2007.
7. R.Alami E.A.Sisbot, T.Simeon and et al. A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 2007.
8. Terrence W Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous Systems*, 2003.
9. E. Forman and M. A. Selly. *Decision By Objectives*. World Scientific, 2001.
10. M. Ghallab, D. Nau, and P. Traverso. *Automated Planning - theory and practice*. Morgan Kaufmann Publishers, 2004.
11. G. Klein J. M. Hoffman, D. D. Woods, Bradshaw, and et al. Ten challenges for making automation a ”team player” in joint human-agent activity. *IEEE Intelligent Systems*, 2004.
12. V. Montreuil, A. Clodic, and R. Alami. Planning human centered robot activities. In *IEEE Int. SMC*, 2007.
13. D. Nau, T. C. Au, O. Ilghami, U. Kuter, and et al. Shop2: An htn planning system. *Journal of Artificial Intelligence Research*, 2003.
14. T. L Saaty. *The Analytic Hierarchy Process for Decisions in a Complex World*. RWS Publications, 1999.
15. T. L Saaty. *Fundamentals of the analytic hierarchy process*. RWS Publications, 2000.
16. E.A. Sisbot, L.F. Marin-Urias, and R. Alami. Spatial reasoning for human robot interaction. In *IEEE/RSJ IROS*, 2007.
17. F. Lerasle T. Simon T. Germa, L. Brthes. Data fusion and eigenface based tracking dedicated to a tour-guide robot. *ICVS*, 2007.
18. P.Danes L.Brethes T. Germa, F.Lerasle. Human/robot visual interaction for a tour-guide robot. *IEEE/RSJ IROS*, 2007.