Human-Aware Task Planning for Mobile Robots

Marcello Cirillo and Lars Karlsson and Alessandro Saffiotti
AASS Research Center, Örebro University, 701 82 Örebro, Sweden
{ mco,lkn,asaffio }@aass.oru.se

***Abstract*— Robots that share their workspace with people, like household or service robots, need to take into account the presence of humans when planning their actions. In this paper, we present a framework for human-aware planning that would make the robots capable of performing their tasks without interfering with the user in his every day life. We focus in particular on the core module of the framework, a human-aware planner that generates a sequence of actions for a robot, taking into account the state of the environment and the goals of the robot, together with a set of forecasted possible plans of the human. We describe the planner and its relations to other system components like a plan recognizer, and present a series of experiments performed with a household robot in a small apartment.**

I. INTRODUCTION

In the past decades, robots have been confined into special working cells under controlled conditions. Recently, the interest of the public for home robots has increased, and people are looking at robots as a new mean to improve the quality of their everyday life. The aging of the population, for instance, could open a wide space for new robotic applications [28]. The robots could then become silent workers, precious butlers and, eventually, friendly helpers in our houses.

As many researchers have already pointed out, the presence of humans introduces other challenges besides how the robots should interact with them [8], [6]. It also has a profound influence on how the robots perform high level reasoning and especially how they plan their actions. Classical AI planning systems, in which the state of the world is only affected by the actions of the robot [24], are no longer applicable. Humans are agents that act independently of the robot, thus, when planning, the robot needs to consider two different processes affecting the state, both its own actions and the actions of the human. In this paper, we present the design and first implementation of a framework for human-aware robot task planning and execution, in which the human and the robot both have their own goals, but the robot should prepare its own plan taking into account the presence of the human user. In a nutshell, our approach to human-aware planning can be described as follows:

- A plan recognition system generates a set of forecasted possible human plans.
- A human-aware planner takes as input these human plans, a set of goals (i.e. the tasks to perform), and a set of constraints regarding how the robot can interact with the human.
- The human-aware planner then attempts to find a plan that for all the forecasted human plans is guaranteed to (a) achieve the goals and (b) not violate the constraints.
- When changes in the plan forecasts are discovered during the execution, replanning can be triggered.

We have implemented a proof-of-concept system according to the points above, partly using existing techniques (in particular for the plan recognition part) and partly developing novel techniques (the planner itself). Our goal is to demonstrate the validity of our approach, albeit under some initial restrictive assumptions — only one human and one robot, limited plan recognition capabilities, and simple sequential plans. The full system, from data acquisition to plan execution, has been tested in a simple scenario involving a robot in a small apartment performing cleaning tasks.

II. RELATED WORK

Many researchers have studied planning with external events, as human actions can be considered from the robot perspective. An early example is the work by Blythe [5], which used Bayesian nets to compute the probability of success in the presence of external events. Our approach is also reminiscent of early work on collaborative planning [13].

In the robotic field, some works have considered human-robot co-habitation. These works take a viewpoint which is different from the one adopted here, by focusing on aspects such as safety (e.g., within MORPHA [11]), acceptable motion (e.g., within COGNIRON [1]) or human-aware manipulation [27]. The research on coordination of Multi Robot Systems [9], although inspiring for us, is not immediately applicable in our case, because we do not consider the human as a part of the system that can be controlled.

As noted by Hoffman and Breazeal [17], [18], cooperation between robots and humans can be greatly improved if the robots can predict the actions of the humans. Our approach, however, differs from their work in two points: first, our aim is not to forecast one human action at a time, but a full plan for a wider timespan. Second, we use planning techniques to select the appropriate actions that the robot should perform to achieve its goals without interfering with the human.

The problem of task planning in the presence of humans is currently open, although some researchers have started exploring the issue [2], [3], [6]. Montreuil et al. [22] and Galindo et al. [10] have addressed the complementary problem of how a robot could generate collaborative plans involving a human. In our work, by contrast, the robot does not plan actions for the human, but instead tries to forecast the future actions of the human and adapt the robot plan to these. Our approach also diverges from the techniques developed for plan merging [12] in two respects: first, in our case the human is not controllable and therefore his actions cannot be re-scheduled in time; second, we consider that human actions can prompt new goals for the robot, hence the plan of the robot cannot be created *a priori* while not knowing what the human will do.

An important aspect of our planning problem is that actions can be executed simultaneously and have durations, and this aspect has been addressed before in the literature. For instance, Mausam and Weld [21] present an MDP model (fully observable) based on the concept of interwoven epoch search space, adapted from Haslum and Geffner [14].

III. A Framework for Human-Aware Planning

The general framework is represented in Figure 1. In the following, we briefly describe the modules that compose it, except for the planner, which will be analyzed in more detail separately.

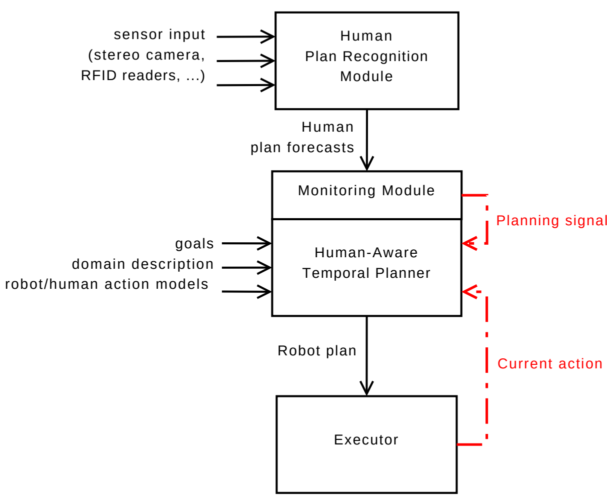

Fig. 1. The framework for human-aware planning.

A. Plan Recognition Module

In our framework, we assume that the system has access to an estimate of the plan that the human is currently executing — or, more generally, a set of possible plans that the human may be executing, with an associated probability distribution. The task of the plan recognition module is to produce this estimate, by taking as input sensor readings and a set of pre-defined plans that the human can execute.

In our experimental system, the plan recognition module has a multi-layered structure, as it can be seen from Figure 2, and it refines and abstracts the data received as input from the sensors step by step. At the base layer, sensor data is collected and coupled. Using a rule based system, simple *instantaneous actions* are detected. For instance, the presence of the user in the kitchen and the fact that the fridge door is open let us infer that the human is using the fridge. In the experiments reported below, we use this layer to recognize 8 simple actions: using_fridge, taking_medication, watching_tv, cooking, eating, using_notebook, preparing_suitcase and going_near_entrance.

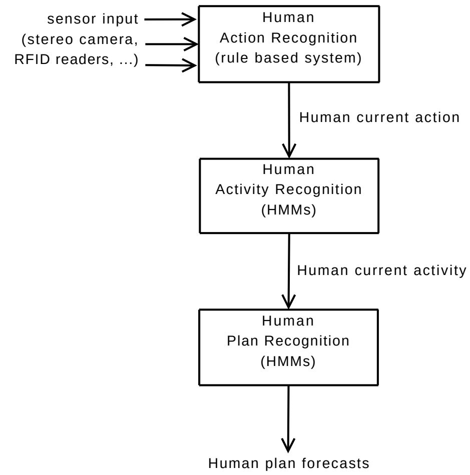

Fig. 2. A graphical representation of the layers that compose the Human Plan Recognition Module used in our implementation.

The simple actions are then passed to the next layer, where we use Hidden Markov Models to identify more complex activities (we define an *activity* as a short human plan, composed by multiple instantaneous actions, e.g.: a meal). Similar approaches have already been successfully used in literature [4], [15], although in our case the HMMs are defined beforehand and not learned from training data, and we rely on a rule based system to identify low level actions and not on unsupervised learning algorithms. In our system, for instance, a sequence of actions in which the user is in the kitchen, uses the fridge and finally eats sitting at the kitchen table would trigger the recognition of the *meal* activity. In our experiments, we used this module to identify four activities: meal, relax, work and going_out.

The third and final layer takes sequences of activities as input to identify predefined longer plans, again using HMMs. We use the Viterbi algorithm [20] to calculate the probabilities associated to each model for the sequence of activities detected. The output of this layer is one or more sequences of human actions that constitute the most probable human plans identified using a dynamic threshold. A new forecast is provided every 5 seconds based on the activities observed in the last 2 hours.

It should be emphasized that our current plan recognition module is not intended to be a state-of-the-art component, and could be easily replaced by another one in future experiments. Our module, however, proved to be sufficiently robust to noise in the sensor data for the purposes of our experiments.

B. Monitoring Module

The monitoring module is a plug-in of the planner. When the plan recognition module updates the human plans identified, then a replanning signal is raised for the planner and a new sequence of robot actions is calculated, according to the specifications detailed in Section IV.

C. Executor

The planner passes to the executor the sequence of actions that the robot should perform. In this first implementation of the full framework, the executor guides the movement of the robot in the experimental environment.

IV. Human-Aware Planner

The focus of our research is the human-aware planner, its design and implementation.

A. Plan format

We assume that both the human plans and the robot plans are composed of atomic actions, like `move-to-kitchen`, with an associated starting time and a fixed temporal duration. The time scale is discrete and in our experiments we used a granularity of one minute. We assume that human plans are linear sequences, while the robot plans generated by the temporal planner may include conditional branches. The fixed duration of the actions should not be seen as a strong constraint: Mausam and Weld [21] showed that by first planning with only the expected duration and then improve or replan the policy with other possible/actual durations, one can still obtain policies that are quite close to the optimum.

In the implementation of the planner that we present here, we assume the presence of only one human and a single robot in the environment. We also assume during planning that the human will follow one of the forecasted plans exactly: if the human deviates from the forecasted plans, that will be handled by replanning.

B. Planning algorithm: input

Our planner is an extension of PTLPlan [19], a probabilistic conditional planner that we extended to cope with human actions and external constraints. The extended planner works as follows. It takes four inputs from different sources. The first input is a planning domain description, that specifies the actions the robot can perform. Each action is detailed in terms of name, preconditions, effects (or results) and time. The effects may be context dependent and/or stochastic, and may involve sensing (for the robot). In other words, the actions are of the type commonly found in POMDPs. An example of a robot action is the following:

```
name: robot-clean(r)
precond: room(r) and room-cleanable(r)
results: dirt(r):=(dirt(r)- 1) and cost:=2
time: 5
```

The formal description of this action is decomposed into four parts.

name: the logic identifier of the action and its parameters; in this case, the room that would be cleaned.

precond: the preconditions that must hold for this action to be applicable; here, r must be a room and it must be cleanable.

results: at the end of the action, the room will be cleaner. We quantized the estimate of the dirt on the floor to allow the robot to refine its task with multiple sweeps of the same room. There is also a cost for the action specified here (in terms of battery consumption, for instance).

time: the time expressed in minutes that the action takes to be completed.

The domain description and the specification of the robot actions also include the human-robot interaction policies that the planner must consider during the search for an acceptable robot plan. These policies can be considered to all effects as maintenance goals, that is, they define conditions that must remain satisfied in every state [16]. For instance, a room can be cleaned only if the human is not present during the duration of the robot action (`room-cleanable(r)` $\equiv$ `not human-in(r)`).

The second input that the planner receives comes from the plan recognition module: a set of forecasted plans, where each plan is a sequence of human actions. The effects and durations of these actions are specified in almost the same manner as for the robot actions, but in the human case preconditions are not provided, as in our system the human is considered to act independently.

The third input for the planner is an initial belief state, representing the robot's incomplete knowledge about the state of itself and the environment, and of what plan the human is performing. A belief state b consists of a set of states $\{s_1, ..., s_n\}$ with associated probabilities $P_b(s_i)$. Each state in turn contains (a) a logic description of the environment as it is at a specific moment in time, and (b) a possible human plan together with an indication of how much of that plan has been executed.

The fourth input for the planner is a set of goals, that is, a number of predicates that must be true after a sequence of robot actions. An example of goal specification is:

```
dirt(bedroom)=0 and robot-in()=robotdocking
```

This goal specifies that at the end of the plan, the bedroom must be clean and the robot back at its charging station.

C. Planning algorithm: action application

An action a with duration t is applied to a belief state b as follows. Before the action can be applied, the preconditions must be verified in each state s_i of b. Then, for each state s_i, time is progressed by t time units. If there are actions in the human plan that will be terminated within the time t, their results are applied to s_i, and the human plan of s_i is updated accordingly (recall that each s_i might have a different human plan). For each human action, the planner also checks if no violations of the human-robot interaction constraints arise. Finally, the results of a are applied to s_i. The states produced in this way will consitute the new belief state b'. If a involves sensing and can result in different observations $o_1, ..., o_n$ (which are specified in the result part, e.g. `obs(door-open)`), then that action will result in one belief state b'_i for each observation o_i. However, in the experiments detailed in this paper we do not use observations. Therefore, the application of an action a in a belief state b will lead to a single new belief state b'.

D. Planning algorithm: finding a plan

Once the planner has received all the four inputs, is starts to search for a plan. The planner begins from the initial belief state and tries all the applicable robot actions. If a constraint violation is detected, that particular robot action is discarded.

An example of a discarded action is depicted in Figure 3. In this case, the plan recognition module forecasts that the user will be first in the livingroom and then he will move to the kitchen. While the algorithm can safely try to apply the first robot action (that is, the robot stays in the kitchen while the human is away), the second action is not acceptable, because our policies state that the robot should not attempt to clean a room where the human might spend some time.

Note that different human-robot interaction policies can be defined for different domains. If, for instance, instead of a cleaning robot we would plan for an autonomous walking aid for an elderly person, then we might want to have it in the same room as the user when the user is expected to move, while planning to go to the charging station when needed. When incorporated in the planner, different policies would result in different types of robot behavior.

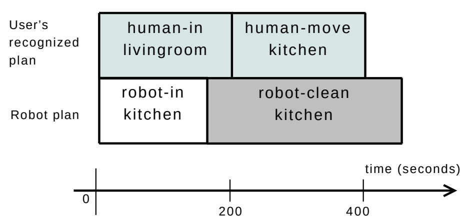

Fig. 3. An example of the search algorithm of the planner. The upper part of the figure represents the forecasted human plan, while the lower part summarizes the actions planned for the robot. The second robot action is discarded, because during the cleaning of the kitchen the user is in the same room, thus violating the human-robot interaction policy.

Each successful application of a robot action yields a new belief state (or possibly several). The planner continues its search from the newly generated belief states until a conformant robot plan is found for which a belief state where the goal holds is reached with sufficient probability. This means that the robot plan should reach the goal for all, or at least sufficiently many (if the sufficient probability is given as less than 1), of the human plans.

The planning algorithm takes advantage of user-specified control rules (specified in temporal logic) to prune certain unpromising sequences of belief states [19]. In our experimental setup, the planner was able to complete each computation and to provide the executor with a new sequence of robot actions in a matter of seconds, which is more than acceptable in our scenarios, considering that the actions of both human and robot have a minimum granularity of one minute.

E. Execution and monitoring

Once a conformant plan is found, it is passed to the executor, which guides the robot in its tasks action by action. The successful execution of every action triggers an update of the internal state of the robot, that keeps track of the status of the environment.

In case a new forecast arrives for the user, then the planner stops the execution of the previous plan, updates the current belief state using the information received from the executor and starts replanning using it as initial belief state. This means that the new starting belief state accounts for all the actions successfully completed by the robot until the replanning signal and also for all the results of the actions possibly performed by the user, according to the previous forecast.

V. Experiments

To demonstrate the feasibility of our approach, we have implemented a simple version of the full framework described above. Our main purpose is to show that our approach can work in a closed loop: from real sensor data, to plan recognition, robot action planning, full physical execution and monitoring. Closing the loop with the physical world is an ambitious goal, and therefore we use in this paper a few simplifying assumptions (for instance, only one human and one robot in the environment and a limited set of human plans) that will be relaxed in our future work.

We test the system in a household environment where one human is performing daily activities in the morning, from 8 am to 1 pm. During the same time span, a robotic vacuum cleaner has the task to clean the floor in all the rooms that need it, while avoiding interference with the user. It must, therefore, both operate and wait in rooms that the system has predicted as not occupied by the human.

The parameters that we use to evaluate the success of each experimental run are the following:

- At the end of the run, the robot has achieved its goals.
- The plan recognition system has correctly forecasted the plans that the user was executing, and refined its predictions over time.
- The sequences of robot actions generated by the planner respect the human-robot interaction policies that we defined.

A. Experimental Setup

In our tests, the system can recognize four pre-defined plans of the human. The user plans are described as sequences of actions that span five hours in the morning. Two of the plans describe a working day, while the other two depict possible holiday mornings.

In the following, we describe schematically the four mornings, detailing the timing of the actions. The location where the user will mostly spend his time during the action is specified between brackets.

```
holiday1
08:00 beginning of the day (bedroom)
08:00-09:00 breakfast (kitchen)
09:00-12:00 relax (living room)
12:00-13:00 lunch (kitchen)

holiday2
08:00 beginning of the day (bedroom)
08:00-09:00 breakfast (kitchen)
09:00-10:00 relax (living room)
```

```
10:00-12:00 promenade (outside)
12:00-13:00 lunch (kitchen)
```

```
workhome
08:00 beginning of the day (bedroom)
08:00-08:30 breakfast (kitchen)
08:30-12:00 work (kitchen table, kitchen)
12:00-13:00 lunch (kitchen)
```

```
normalwork
08:00 beginning of the day (bedroom)
08:00-08:05 getting ready (kitchen)
08:05-13:00 out for work (outside)
```

In two of the plans (*holiday1* and *holiday2*), the environment is affected by the actions of the human: the long breakfast induces the system to mark the kitchen as dirty. Therefore, the robot must take into account the modifications to fulfill its goals.

We tested our system in the PEIS-Ecology [25], [26], a real environment that provided a reliable communication infrastructure among our modules and allowed us to acquire the sensor data needed (Figure 4).

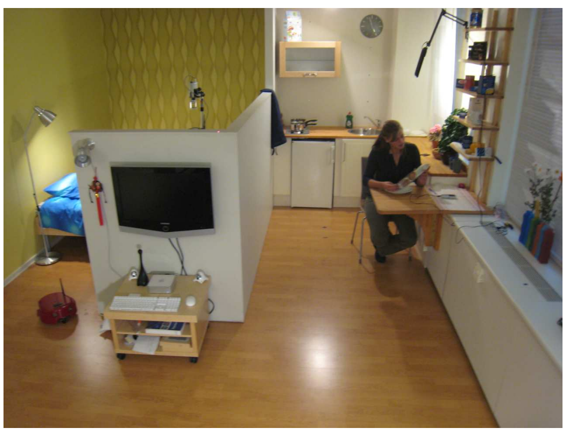

Fig. 4. The PEIS ecology is a real environment composed by a bedroom, a livingroom and a small kitchen.

As a testbed for the system, we used the four possible user's plans above, and we run four experiments in three different scenarios. Considering that each run would take 5 hours in real time (8 am to 1 pm, as we said), we decided to run only the first one in real time. The second run was made with the same setup, but at an accelerated pace, with the times of both human and robot actions accelerated by 6 times with the goal to verify that running an experiment at accelerated time would not compromise the results. The results we obtained were almost identical to the ones in the first run and small differences were originated by the natural fluctuations in the sensors' readings. Therefore, we run all the remaining experiments at accelerated time.

We used different sensors distributed into the PEIS environment as input for our plan recognition module: a stereo camera, with a people stereo tracking system [23]; two pressure sensors to detect the status of the fridge door; and two RFID readers located under the kitchen table and inside the fridge. A number of objects had RFID tags attached to be recognized by the system, e.g., dishes, books, and a laptop. No sensor was used to detect the dirt on the floor; this information was inferred by the actions that the human was perceived to perform.

To simulate an autonomous vacuum cleaner, we used an Amigobot from ActiveMedia Robotics, controlled using Player [7] with the *vfh* and *wavefront* drivers for obstacle avoidance and path planning. Localization relied on simple odometry.

In all our scenarios, the goal of the robot is to have the apartment clean and to be back at its docking station by the end of the morning, while avoiding the interference with the user and minimizing the cost of its actions.

The goal is specified in a simple formula:

```
dirt(bedroom) = 0 and
dirt(livingroom) = 0 and
dirt(kitchen) = 0 and
robot-in = robotdocking
```

The initial state was the same for all scenarios: the robot was at its docking station, the human was in the bedroom, and the only room that must be cleaned was the bedroom. A part of the symbolic representation of the initial state follows:

```
robot-in := robotdocking and
human-in := bedroom and
dirt(bedroom) := 3 and
dirt(livingroom) := 0 and
dirt(kitchen) := 0
```

Subsequent human actions can change the status of the environment and introduce new constraints for the actions of the robot.

B. First Scenario

The goal of the first scenario is to test the basic capabilities of our system in a linear situation: the user executed all and only the actions related to the human plan *normalwork*.

This scenario was repeated in two runs, the first at normal speed (5 hours in total) and the second at accelerated time (50 minutes). In both cases, the execution ended with success: the user's plan was correctly identified, the robot didn't interfere with the user and at the end of the day the bedroom was clean and the robot was back in its original position. The generated robot plan, coupled with the identified user's actions is graphically represented in Figure 5.

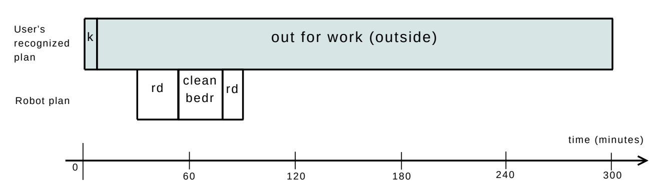

Fig. 5. First scenario, first run. In the robot plan, **rd** stands for **robot docking** and it means that the vacuum cleaner is on its docking station. In the human plan, **k** stands for **kitchen**.

C. Second Scenario

The goal of the second scenario was to test the system's capability to deal with predicted changes in the environment introduced by user's actions. In this scenario, the user executed all the actions in the plan *holiday1*. As we explained, in this case the kitchen is marked as dirty at the end of the breakfast and the robot must also attend to that room. This run was performed at accelerated time. This scenario also introduced another complication for the system: three out of four defined human plans start with a breakfast, thus complicating the recognition of the correct one.

At the beginning of the run, the system identified *holiday1*, *holiday2* and *workhome* as possible human plans. Those three plans have been created to be incompatible: as can be seen in Figure 6 (top), in the first two plans the kitchen has to be cleaned for the robot to achieve its goal, while in the third one the user remains in the kitchen all morning thus making it impossible for the robot to clean it. As a consequence, no conformant plan could be found at the beginning (time $t1$), and no robot action was performed. At time 600 (accelerated time) the human left the kitchen. After some time, the plan recognition module refined its forecast, identifying *holiday1* and *holiday2* as the only two possible human plans. The planner found a conformant plan that was then executed by the robot. Figure 6 (bottom) shows a schematic representation of the situation at this point (time $t2$): the figure shows the two recognized human plans, and the generated robot plan. The recognition of the human plan may be later refined to single out *holiday1* as the most likely one. This however, will not require any change in the robot plan.

At the end of the run, the robot had executed all the actions needed to clean both the kitchen and the bedroom, going back to its docking station at the end.

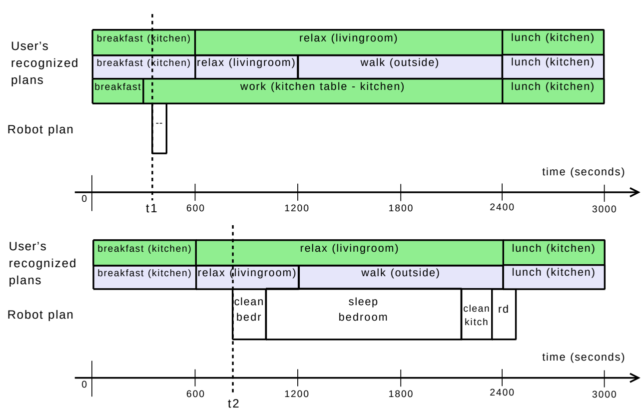

Fig. 6. The situation at instants **t1** (top) and **t2** (bottom) in the second scenario. At time **t1**, *workhome*, *holiday1* and *holiday2* are considered as possible human plans. Since in such situation is not possible for the system to find a conformant plan, no actions were planned for the robot. At time **t2**, *workhome* is not considered as possible anymore and the planner has generated a sequence of actions that will clean the house without interfering with the human. Note that these include cleaning the kitchen, since it was used by the human for a long breakfast.

D. Third Scenario

The third scenario was intended to test the capability of the system to recognize a change in the plan of the human and to re-plan accordingly. The user performed for the first part of the run all the actions related to the *workhome* plan. Then, after about 20 minutes in accelerated time (1200 sec), he switched plan to *holiday1*.

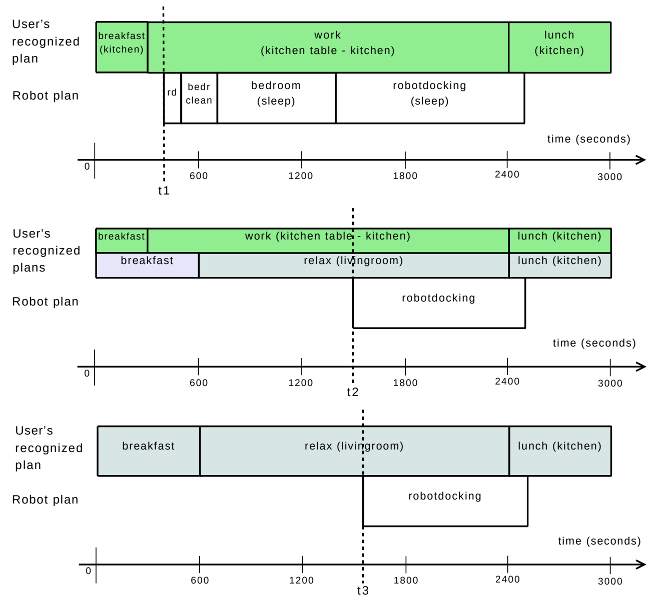

Fig. 7. Third scenario: situations at time points **t1**, **t2** and **t3**, when the plan recognition module provides new estimates of the human plans. At time **t1** (top) *workhome* is the only plan considered as possible for the user and the planner generates a corresponding plan for the robot. At time **t2** (middle) both *workhome* and *holiday1* are considered as possible. At time **t3** (bottom) only *holiday1* is possible. At both times **t2** and **t3** the planner is invoked, but it generates the simple plan to go at the docking station since in both cases the only goal left is to be at the docking station until the end of the morning.

At the beginning of the run, the user plan identified as the most probable is *workhome* and at time **t1**, the planner started to generate the actions for the robot (Figure 7, top diagram, shows the user recognized plan and the plan generated for the robot). As the user changed his pattern of actions, the probabilities of the user plans changed and the planner was invoked a second time, when *workhome* and *holiday1* were equally probable (time point **t2**, Figure 7, middle diagram), and a third time, when *holiday1* was recognized as the most likely (time point **t3**, Figure 7, bottom diagram).

As can be seen from Figure 7, the cleaning of the bedroom is executed after the first planning. Therefore, since the robot keeps track of the actions it performed and their outcome, in the following plans the only goal left is to stay at the docking station.

VI. Conclusions

The main contribution of this paper is to define a new type of human-aware robot task planning, in which the robot takes into account the (forecasted) future human activities when planning its own actions. Future human activities are considered in two ways by our planner. First, the robot's actions should be *compliant* with the human ones, where compliance is defined with respect to a set of constraints;

in our example, the vacuum cleaner should never clean a room where the human is located. Second, the robot should consider the *changes in status* caused by the human actions; in our example, the vacuum cleaner should plan to clean the kitchen after the human has used it.

We have tested our approach in a real system, in which human plans were recognized from real sensor data using a simple probabilistic technique, and the robot plans were executed on a small mobile robot. Although the scenarios were simple, these experiments prove the feasibility of our approach to perform human-aware planning in a real setting. Our future work will therefore focus on the extension of our framework to deal with more complex situations, including more complex robot tasks and human plans with non-deterministic actions. The use of a more sophisticated plan recognition module is also a priority in our agenda.

ACKNOWLEDGMENTS

This work has been supported by CUGS (Swedish national computer science graduate school) and by the Swedish KK foundation.

REFERENCES

[1] E. Akin Sisbot, A. Clodic, L.F. Marin, M. Fontmarty, L. Brèthes, and R. Alami. Implementing a human-aware robot system. In *Proc. of the ROMAN Symposium*, pages 727–732, 2006.

[2] R. Alami, A. Clodic, V. Montreuil, E.A. Sisbot, and R. Chatila. Task planning for human-robot interaction. In *sOc-EUSAI '05: Proc. of the 2005 joint conference on Smart objects and ambient intelligence*, pages 81–85, New York, NY, USA, 2005.

[3] R. Alami, A. Clodic, V. Montreuil, E.A. Sisbot, and R. Chatila. Toward human-aware robot task planning. In *AAAI Spring Symp 'To boldly go where no human-robot team has gone before'*, pages 39–46, 2006.

[4] S. Aoki, Y. Iwai, M. Onishi, A. Kojima, and K. Fukunaga. Learning and recognizing behavioral patterns using position and posture of human body and its application to detection of irregular states. *Syst. Comput. Japan*, 36(13):45–56, 2005.

[5] J. Blythe. Planning with external events. In *UAI '94: Proc. of the Tenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 94–101, 1994.

[6] A. Clodic, H. Cao, S. Alili, V. Montreuil, R. Alami, and R. Chatila. Shary: a supervision system adapted to human-robot interaction. In *Proc. of the 11th International Symposium on Experimental Robotics (ISER 08)*, 2008.

[7] T.H.J. Collett, B.A. MacDonald, and B.P. Gerkey. Player 2.0: Toward a practical robot programming framework. In *Proc. of the Australasian Conference on Robotics and Automation (ACRA)*, 2005.

[8] K. Dautenhahn, M. Walters, S. Woods, K. L. Koay, C. L. Nehaniv, A. Sisbot, R. Alami, and T. Siméon. How may I serve you? A robot companion approaching a seated person in a helping context. In *HRI '06: Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 172–179, New York, NY, USA, 2006.

[9] A. Farinelli, L. Iocchi, and D. Nardi. An analysis of coordination in multi-robot systems. *IEEE International Conference on Systems, Man and Cybernetics*, 2:1487–1492, Oct. 2003.

[10] C. Galindo, J.A. Fernández-Madrigal, and J. González. Multi-hierarchical interactive task planning. application to mobile robotics. *IEEE Transactions on Systems, Man, and Cybernetics part B*, 18(3):785–789, Jun 2008.

[11] B. Graf, M. Hans, and R.D. Schraft. Mobile robot assistants: issues for dependable operation in direct cooperation with humans. IEEE *Robotics and Automation Magazine*, 11(2):67–77, 2004.

[12] F. Gravot and R. Alami. An extension of the plan-merging paradigm for multi-robot coordination. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2929–2934, 2001.

[13] B.J. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[14] P. Haslum and H. Geffner. Heuristic planning with time and resources. In *Proc. 6th European Conference on Planning*, 2001.

[15] A. Hein and T. Kirste. Towards recognizing abstract activities: An unsupervised approach. In *BMI '08: Proc. of the 2nd Workshop on Behaviour Monitoring and Interpretation*, pages 102–114, 2008.

[16] K.V. Hindriks and M.B. Van Riemsdijk. Satisfying maintenance goals. In *Proc. of the 5th International DALT Workshop*. Springer-Verlag New York Inc, 2007.

[17] G. Hoffman and C. Breazeal. Cost-Based Anticipatory Action Selection for Human–Robot Fluency. *IEEE Transactions on Robotics*, 23(5):952–961, 2007.

[18] G. Hoffman and C. Breazeal. Effects of anticipatory action on human-robot teamwork – efficiency, fluency, and perception of team. In *HRI '07: Proc. of the ACM/IEEE International Conference on Human-Robot Interaction*, pages 1–8, 2007.

[19] L. Karlsson. Conditional progressive planning under uncertainty. In *Proc. of the 17th Int. Joint Conference on Artificial Intelligence (IJCAI)*, pages 431–438, 2001.

[20] H.L. Lou. Implementing the Viterbi algorithm. *Signal Processing Magazine, IEEE*, 12(5):42–52, 1995.

[21] Mausam and D.S. Weld. Planning with durative actions in stochastic domains. *Journal of Artificial Intelligence Research*, 31:33–28, 2008.

[22] V. Montreuil, A. Clodic, and R. Alami. Planning human centered robot activities. In *Proc. of the 2007 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2007.

[23] R. Muñoz-Salinas, E. Aguirre, and M. García-Silvente. People detection and tracking using stereo vision and color. *Image and Vision Computing*, 25(6):995–1007, 2007.

[24] D. Nau, Ghallab M., and P. Traverso. *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[25] A. Saffiotti and M. Broxvall. PEIS ecologies: Ambient intelligence meets autonomous robotics. In *Proc. of the Int Conf on Smart Objects and Ambient Intelligence (sOc-EUSAI)*, 2005.

[26] A. Saffiotti, M. Broxvall, M. Gritti, K. LeBlanc, R. Lundh, Rashid J., B.S. Seo, and Y.J. Cho. The PEIS-ecology project: vision and results. In *Proc of the IEEE/RSJ Int Conf on Intelligent Robots and Systems (IROS)*, Nice, France, 2008.

[27] E.A. Sisbot, L.F. Marin, and R. Alami. Spatial reasoning for human robot interaction. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2007*, pages 2281–2287, 2007.

[28] A. Tapus, M.J. Mataric, and B. Scassellati. The grand challenges in socially assistive robotics. IEEE *Robotics & Automation Magazine*, 14(1):35–42, Mar 2007.