

# **SHARY: a supervision system adapted to Human-Robot Interaction**

Aurélie Clodic, Hung Cao, Samir Alili, Vincent Montreuil, Rachid Alami,  
Raja Chatila

► **To cite this version:**

Aurélie Clodic, Hung Cao, Samir Alili, Vincent Montreuil, Rachid Alami, et al.. SHARY: a supervision system adapted to Human-Robot Interaction. In: Khatib O., Kumar V., Pappas G.J. (eds) Experimental Robotics. Springer Tracts in Advanced Robotics, vol 54. Springer, Berlin, Heidelberg, pp.229-238, 2009. hal-01979196

**HAL Id: hal-01979196**

**<https://hal.laas.fr/hal-01979196>**

Submitted on 12 Jan 2019

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

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

---

# SHARY: a supervision system adapted to Human-Robot Interaction

A. Clodic, H. Cao, S. Alili, V. Montreuil, R. Alami, and R. Chatila

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

**Summary.** Human-Robot Interaction (HRI) brings new challenges to robotics. We focus in this paper on the decisional issues of HRI enabled robots. We propose a control architecture specifically designed for HRI and present an implemented system that illustrates its main components and their interaction. These components provide integrated abilities to support human-robot collaborative task achievement as well as capacities to elaborate task plans involving humans and robots and to produce legible and socially acceptable behavior.

**Key words:** Human-Robot Interaction, Control Architectures, Robot Supervision, Task Planning

## 1 Introduction

One challenge in robotics research is to develop robots able to operate and help humans in everyday life. One main consequence is that humans should be taken into account during robot task realization. Indeed, several levels of interaction may intervene, from monitoring the human's intention to intricate collaboration in joint task achievement. This is precisely the context of the work presented here.

The human presence brings new requirements for robot's abilities both at the functional and at the deliberative levels[24]. For the former, the topics involve motion[25, 7, 26], navigation[5, 35], manipulation[23] in presence of humans as well as perception of human activities[9, 10]. For the latter, when interacting with humans, robots need to incorporate communication and collaboration abilities. Several theories dealing with collaboration [16, 22, 11] emphasize that collaborative tasks have specific requirements compared to individual ones e.g. since the robot and the person share a common goal, they have to agree on the manner to realize it, they must show their commitment to the goal during execution, etc. Several robotic systems have already been built based on these theories [29, 32, 36, 8] and they all have shown benefits of

this approach. They have also shown how difficult it is to manage turn-taking between communication partners and to interleave task realization and communication in a generic way. Finally, today only few systems [18, 8, 33] take humans into account at all levels.

Designing a supervision system for an interactive robot raises several new challenges. In fact, when performing tasks in interaction with humans, the robot supervision system is not only responsible for the refinement and the correct execution of the robot plan, but also for the appropriate set of communications and monitoring activities within and around task realization. It is also in charge of monitoring human commitment and activities in order to provide appropriate response based on the current context (which is included in the fact database). Furthermore, robotics context implies that execution can fail. Since the robot is evolving in the real world, hardware and software failure may occur frequently. To ensure human safety, the supervision system must be capable of recovering from failure by stopping current activities and re-planning task execution.

The work presented here consists of an approach and an implemented system which address a number of the above mentioned challenges. The contribution focuses on the robot’s decisional abilities. It involves a control architecture and several key components specially designed to embed HRI abilities and constraints. Indeed, the decisional layer involves a HRI-enabled supervision system, called SHARY<sup>1</sup> and a task planner called HATP<sup>2</sup>[4]. These components integrate abilities to support human-robot collaborative task achievement as well as capacities to produce legible and socially acceptable behavior.

Section §2 presents the robot control architecture. Then, sections §2.1, §2.2 and §2.3 give a short description of the components involved in the decisional layer. Section §3 deals with the implemented system and illustrates its use through several examples involving a mobile manipulator acting in interaction with a person. Finally, section §4 concludes and discusses future work.

## 2 A control architecture dedicated to HRI

We have devised a control architecture dedicated to robot decision and action in a human context (Figure 1). It has been developed as an instance of the generic the LAAS Architecture for Autonomous Systems[1].

The decisional layer consists of three components:

- The **Task Agenda** which embeds all decisional activities related to high-level robot goals.
- **SHARY** which constitutes the decisional kernel. It is based on an incremental context-based task refinement in a human context.

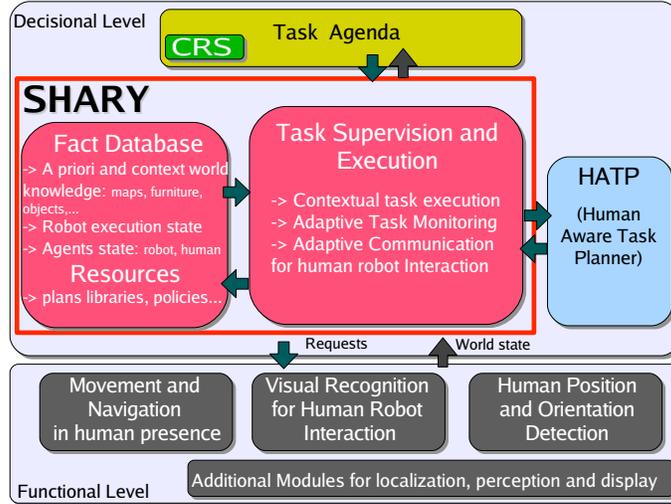
---

<sup>1</sup> SHARY: Supervision for Human Aware Robot Ynteraction

<sup>2</sup> HATP: Human Aware Task Planner

- **HATP**: A task planning system that is able to synthesize socially acceptable robot plans that may involve human-robot collaborative action.

We will now explain the different parts of this architecture.



**Fig. 1.** The proposed control architecture for interactive robots: The decisional layer consists of three components. The central component, SHARY, achieves task supervision and execution as well as human-activity monitoring. Other decisional abilities are provided by HATP, a task planner, and Task Agenda which manages high-level robot goals in a human context.

### 2.1 The Task Agenda

The role of the Task Agenda is to manage high-level robot goals and their associated tasks. It maintains an ordered list of high-level tasks and embeds a mechanism that permits the robot to exhibit a proactive behaviour, for instance taking the initiative to serve a drink or to behave as a “curious” robot that decides to acquire information about the state of the environment (e.g. exploration of new objects placed by a person on a table).

The task agenda relies on **CRS chronicle recognition system**[17], in order to detect a conjunction of partially ordered events that might call for a robot action, or more precisely for the creation of a new task. In the current implementation, CRS has been essentially used to interpret activity of the persons in the robot vicinity.

The Task Agenda generates tasks depending on requests from the users (through multi-modal dialog) and on chronicles recognized by CRS. Tasks are then scheduled based on priority and on the current context.

The Task Agenda maintains several TODO lists that specify the robot tasks. These tasks are high level tasks like: serve-a-drink, explore-new-objects, throw-bottle-in-the-garbage, cleanup-a-table, ... In response to inputs from CRS or dialog, the Agenda adds a high-level goal into the appropriate TODO list if the goal is not already present. This means that if the robot detects the need to serve a drink and takes the initiative to do it, and if at the same time the person requests to serve him a drink, the task will be added only once.

Besides, The Task Agenda generates tasks to satisfy the goals stored in TODO lists. It uses a task queue for scheduling tasks. It is able to suspend an ongoing task in order to execute tasks with a higher priority. Suspending a task means cancelling the task execution and keeping the goal in its current TODO list.

## 2.2 SHARY : The supervision and execution system

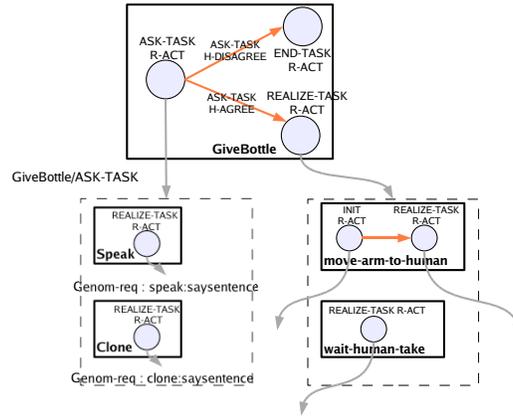
SHARY'S originality, as a supervision system, lies in its ability to take into account not only the task achievement but also communication and monitoring needed to support interactive task achievement in a flexible way. SHARY allows to define a task or a hierarchy of tasks linked to more or less elaborated "communication policies" that enable to execute tasks given the possibility to deal with contingencies that could occur during its execution (or even to stop the task in case of unexpected events).

As illustrated in figure 2, a communication scheme, for a given joint task, represents all possible turn taking steps and synchronisations between the robot and its human partner [15]. Each time a state is visited the corresponding task recipe or atomic task is launched.

**Communication scheme:** From a practical point of view, a communication scheme is a finite state automaton where :

- each **transition** corresponds to information from the world state (human state, robot state). This condition is modelled as a communication act and corresponds to monitored conditions during the state execution.
- each state is a communication act that would be carried out by the concerned agent (robot or human).

A communication act is the central notion in the formalism. It plays the role of state in the communication scheme, it plays also the role of transition condition. Consequently, a communication act represents an information exchange between two agents. This exchange can be realized through dialog or by an expressive motion or a combination of the two. It enables each agent to communicate his belief about the task to be realized in order to share mutual knowledge and to agree on execution plans. Communication acts are defined by a name which characterizes the object of the communication and by a type which defines the evolution of this object during the interaction [12]. E.g. **ASK-TASK** evolution could be: R-ACT (the robot asks the human



**Fig. 2.** This figure illustrates two strongly-related notions in SHARY: “tasks” (illustrated by a box) and “communication policies” (graph of circles). In one hand, each task is associated to a communication scheme and in the other hand, each communication scheme is composed of acts of which the execution consists in executing a recipe, e.g. a set of tasks. The example shows a concrete case for the give-bottle task. The communication scheme associated to this task is composed of 3 communication acts: ASK-TASK, END-TASK and REALIZE-TASK. This communication scheme allows the robot to ask for human’s agreement before realizing the task: ASK-TASK R-ACT, at the same time a monitoring is done on ASK-TASK H-AGREE or H-DISAGREE, i.e. does the human answer positively or not to the task request. So the final robot behavior could be the sequence of ASK-TASK/REALIZE-TASK or ASK-TASK/END-TASK. The gray arrows (vertical) indicate the decomposition link of a communication scheme act to a recipe or an atomic request (Genom-req) and the orange arrows (horizontal) correspond to transitions inside the scheme.

if he agrees to do the task), H-AGREE (the human agrees to do the task) or H-DISAGREE (he does not agree).

We have defined a set of communication acts that we found mandatory in the framework of task achievement [13]. At any time, both the user and the robot can propose the following task-based acts:

- ASK-TASK: proposing a task.
- PROPOSE-PLAN: proposing a plan (recipe) for a given task.
- MODIFY-PLAN: proposing a modification of the current plan for a given task.
- GIVE-UP: gives up a task (e.g., because the task becomes impossible). For the robot this is a way to announce that it is unable to achieve the task.
- CANCEL: cancellation of a task (voluntary give-up).
- END-TASK: announces that the task has been done.
- REALIZE-TASK: announces that the task performance will start.

This set takes inspiration from Joint Intention Theory ([16]) that states that each partner should be informed of the beginning, realization and ending of a joint task.

An act is a data structure used by a communication scheme description in a generic way. An act inside a task, i.e. an instantiation of an act within a task, is defined as a particular data structure: `act_X_task`. It is the data manipulated by the executive engine and to which will be attached a recipe, an execution state, etc. For example, when the robot is proposing to give the human an object, it is realizing the `act_X_task` defined by the `Give Object` task and the `ASK-TASK` act.

**Task Recipe:** Task recipes are methods that compute the partially ordered list of subtasks of an `act_X_task`. This sub-task tree contains both a set of tasks needed for the achievement of the `act_X_task` but also a list of tasks required for monitoring the execution. Recipes can be scripts, i.e. provided by the programmer, or can be synthesized by a planner such as HATP [27] presented in the next section.

**Fact database:** The fact database contains robot world representation with a priori information (map, objects, furniture...) and contextual data (positions of robots, humans, objects,...). It contains also an updated model of all agents involved in the task. This model is obtained through data fusion of different perception modalities.

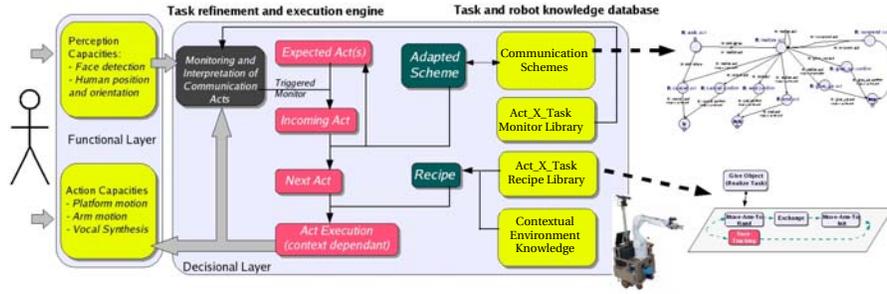
**SHARY execution stream:** To program a task in SHARY, you need to define a communication scheme. Then, for each communication act, it is necessary to define a recipe or to choose an atomic task. Figure 3 describes SHARY execution at a given task level and exhibits the incremental context-based task refinement process which results in a dynamic hierarchical task tree.

### 2.3 Human Aware Task Planner - HATP

In order to devise a task planner adapted for activities involving robots and humans, we have identified four challenges: *Intentions and state expression*, *Agent abilities and preferences*, *Social aspects of the plan*, *Plan negotiation*, *Real time constraints*.

We have elaborated a planner that is based on hierarchical task planning[20] and integration of behavior rules that drive the robot decisions and produce social plans. Let us give a brief description of HATP domain, define the socials rules and explain which kinds of rules we choose.

In HATP, the world description is represented by a set of entities  $Wb = \langle En_1, En_2, En_3, \dots, En_n \rangle$ . Each entity is unique and is implemented as an object in C++ or Java languages. The attributes associated to these entities are



**Fig. 3.** General Description of Shary (at a given task level inside a hierarchy of tasks): when the task is created, a communication scheme associated to the task is instantiated according to the task, the context and the concerned agent = *Adapted Scheme*. This scheme gives the first act to execute. The recipe corresponding to that act (precisely to this act\_X\_task) is instantiated by the help of a recipes library: *Recipe*. During *Act Execution*, communication and execution monitoring is done through wait on *Expected Acts*. When a monitor is triggered *Incoming Act*, i.e. when an expected act happens, the current act is stopped and the answer is instantiated given the communication scheme *Next Act*. And so on...

```

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;
}
Jido = new Agent;
DoorPlace = new Place
Jido.type = "robot";
Jido.maxObjects = 2;
Jido.posTopo = DoorPlace;
Jido.full = false;
    
```

Entity declaration (points to the first two lines)  
 Attributes declaration (points to the curly braces and their contents)  
 Entity type declaration (points to the Jido and DoorPlace declarations)  
 Entity attributes initialization (points to the Jido attribute assignments)

**Fig. 4.** HATP world representation: In this example, we define new entities *Agent* and *Place*. For the entity *Agent*, we define a set of attributes such as type, maxObjects, posTop (Topological Position),...

similar to predicate or state variables in classical planning. Figure 4 illustrates how an entity is represented in HATP.

HATP domain is represented by a pair  $D = \langle T, R \rangle$ , where  $T$  is a set of tasks and  $R$  is set of rules. We can distinguish two types of tasks in  $T = \langle Op, M \rangle$ : *basic primitives* (or *operators*)  $Op$  and non-primitive tasks  $M$  (called *methods*). A basic primitive represents an action that can be executed directly while a non-primitive task must be decomposed into sub-tasks  $ST \subset T$ . The set of rules  $R$  also called social rules. Each  $r_i \in R$  is represented by a tuple  $r_i = \langle B_i, Pe_i^{Ag}, pref_i \rangle$  where  $B_i$  is the description of the rule (it is defined as patterns to recognize in the plan structure),  $Pe_i^{Ag}$  is a penalty

(cost) added to plan score if the rule  $r_i$  is violated in the solution plan, this cost depends on the violated rule  $r_i$  and the agent  $Ag$  who did it.  $pref_i$  represents the weight of the rule.

Non atomic tasks (see figure 5) can be decomposed into sub-tasks allowing to build a task hierarchy like in SharedPlans[22, 21] or in SHOP [28].

```

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) action

```

method taskname(En1, En2, ..., Enn) method ReachObject(Agent Ag, Object Obj)
{
  empty
  {
    OR
    {formula1}
    :
    {formulan};
  }
  // decomposition 1
  {
    preconditions
    {Obj.owned == true; Ag.posTopo == Obj.owner.posTopo; }
    subtasks
    { 1:ReachAgent(Ag, Obj.owner); };
  }
  // decomposition 2
  {
    preconditions
    { Obj.owned == false; Ag.posTopo == Obj.furniture.posTopo; };
  }
  subtasks
  { Id1:taskname(parameter);
    :
    Idn:taskname(parameter);
  };
  :
  :
  // decomposition n
  {}
}

```

(b) method

Fig. 5. HATP method and action representation

In HATP, a set of so-called “social rules” are introduced in order to allow the planner to produce plans that satisfy a number of social conventions. We have defined six types of rules that, we believe, allow to synthesize robot plans that can be more easily accepted and whose intention can be easily inferred by humans. Depending on the context, the programmer can select or inhibit some of them. The current rules deal with:

- **Undesirable state:** a specification of a set of world states that should be avoided if possible (e.g. leaving the fridge door open)
- **Undesirable sequence:** sequence that can conduct to a feeling of unpleasantness for the human. For example, a plan in which the robot puts down an object and its human partner picks it up immediately after is

less preferred to a plan where the robot hands directly the object to the human.

- **Bad decomposition:** if the planner has to select a decomposition among others of a method, he will prefer the one that is nearer to human behavior. For example, if the robot has to put down an object for someone, it is better to put it on a furniture instead of putting it inside a furniture.
- **Effort balancing:** we prefer plans where robot effort is higher than human one because we consider that the robot is here to help human in everyday life task.
- **Timeouts:** the idea is to prevent a long waiting time between two actions done by a same agent.
- **Crossed links:** the idea is to apply a penalty to plans involving intricate human-robot actions since they might cause too much burden to the human

HATP planning process is composed of two threads. One thread is responsible of the plan refinement. When a possible valid plan is found, it is transmitted to a thread responsible of complete evaluation and storage of plans. The overall search process is given a bounded time.

HATP algorithm is largely inspired from SHOP2 procedure [28]. The main differences between them are:

- HATP manipulates task trees.
- HATP 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 [27]).
- HATP makes multi-criterion plan evaluation .

The thread responsible of plan evaluation uses a metric based on the Decision by Objectives theory, more precisely The Analytic Hierarchy Process (AHP)[19]. it is a generic method [31, 30], 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.

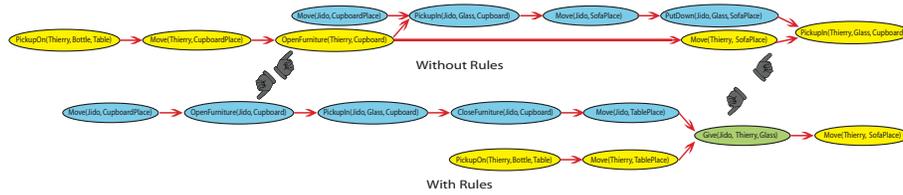
Now, let us give examples of HATP plans. Figure 6, illustrates the influence of social rules on the quality of the plan produced.

Figure 7 illustrates a series of plans, achieving the same goal, that are synthesised by HATP in a given context or after an execution contingency.

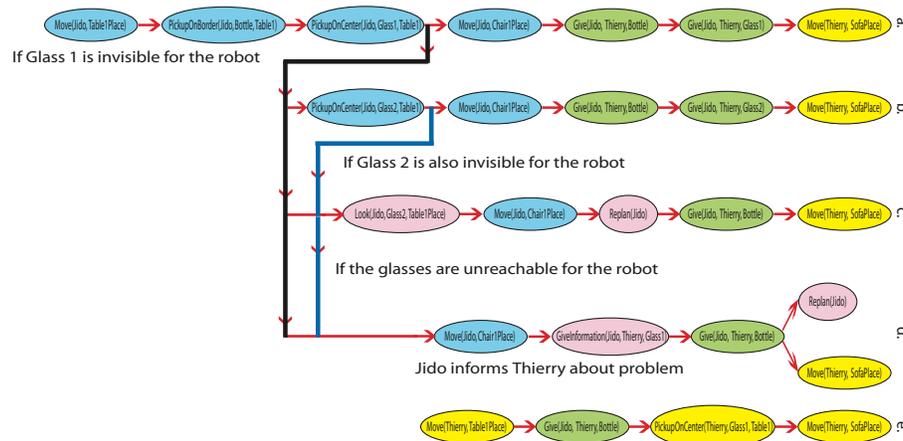
### 3 Experiments

A first version of the complete system has been implemented and integrated on a fully equipped mobile manipulator called “Jido”.

The goal of the Jido robotic platform is to demonstrate the use and the benefit of a robot in our daily life. The experimental environment, shown



**Fig. 6. Effect of social rules on the plans:** In this example the goal is the fact that the human has a glass, a bottle and is seated on the sofa. The rules introduced here are: (1) no object on the sofa (2) do not let cupboard door open (3) no sequence as **Move**, immediately followed by an other **Move** executed by a same agent (4) no sequence as **Putdown** done by robot, immediately followed by **Pickup** done by human (5) preference to plans where the actions **Open/Close** cupboard door are done by a same agent (6) preference to plans where the effort delivered by the human is reduced, (7) preference to plans where there is not time-out and minimum of joint actions.



**Fig. 7. HATP failure and contingency handling:** In this example the goal is the fact that the human has a glass, a bottle and is seated in the sofa. Plan (a) is the plan where everything is right. Plan (b) is the alternative when the object Glass1 is invisible to the robot, the planner replaces the object Glass1 by Glass2. Plan (c) is the alternative when the two glasses are invisible, the robot will take look at the last known position. Plan (d) is the alternative when the two glasses are unreachable for the robot. HATP decides to request human help. If the human agrees, it produces the last plan where the human intervenes. HATP might also insert **replan** action that indicates to the supervision system that it should ask a new plan after a non-deterministic action.

figure 8, simulates a living-room with furniture (tables, chairs) and objects (colored bottles).

At the functional level, Jido is equipped with a set of functionalities such as a laser-based in-doors localisation and navigation system as well as simple functions for detecting, localising and manipulating a set of objects (colored bottles). Jido is also equipped with functions that are specifically oriented towards human robot interaction: a navigation and motion planner that produces human-friendly motion [35, 34], a set of perception primitives allowing the robot to detect and track humans in the vicinity of the robot as well to localize in 3D the head and the hands of a person facing the robot [10], a set of speech related primitives (recognition [10] and synthesis [6]).

Equipped with these functions and with the decisional capabilities presented above, Jido is able to perform a number of tasks:

- serving a drink to a person.
- cleaning a piece of furniture or a bottle (i.e. put a used bottle in the trash, put unused ones on the store table).
- maintaining an updated knowledge of the state of the world (detecting and tracking persons in the room, detecting and recognising new objects placed on tables by the persons. . . .

The challenge for Jido is to perform these tasks in a robust fashion and to exhibit social and interactive abilities i.e. safe, legible and user acceptable behaviour. We present, in the sequel, several illustrative examples of Jido capabilities.



(a) A “living-room”



(b) 3D model used by the navigation and manipulation planners

**Fig. 8.** Jido working environment: Several colored bottles (yellow, green, orange, blue, ..), humans, tables, 2 chairs and a trash.

**Example 1: Picking a bottle, failure and re-planning in a HRI context**

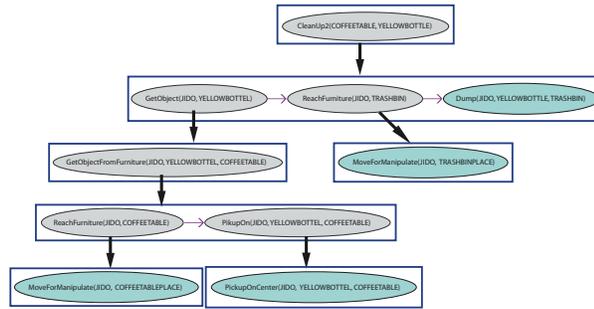
Through this first scenario, we would like to show the ability of the system to recover from a contingent situation that calls for re-planning in HRI context. The task consists for Jido to clear out the living-room table by picking a bottle and throwing it into the trash. The failure here comes from the fact that the bottle is too far to be accessed by Jido. Jido will try to find a different way to get the bottle by asking help to a person present in the environment.

Figure 9 illustrates the plan produced by HATP as well as a snapshot of a current task refinement decomposition performed by SHARY.

Figure 10 illustrates the different steps including request to the planner

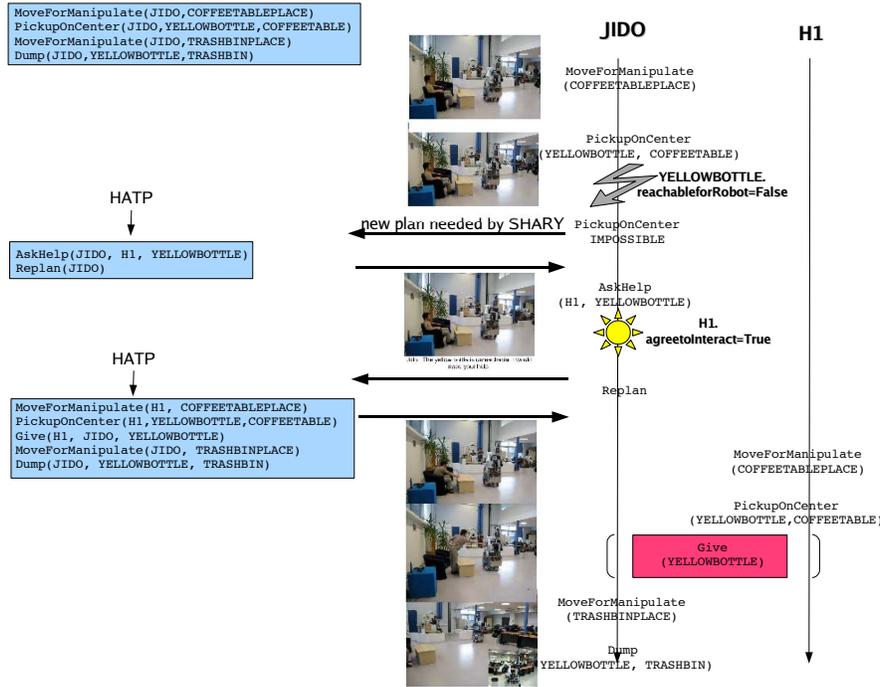


(a) Current execution task stack in SHARY: Boxes are tasks, circles and diamonds shapes are act\_X\_tasks (RT is the abbreviation of **REALIZE-TASK**), gray arrows represent decomposition links and dotted arrows are transitions between act\_X\_tasks inside a communication scheme. Blue color corresponds to *achieved* tasks and acts while green color means that they are being executed.



(b) Hierarchical plan from HATP for the **clean-up** task (c) Snapshot from the experiment

**Fig. 9. Example 1 (First trial):** Achieving **clean-up(yellow-bottle)** consists mainly in achieving its act\_X\_task RT or its plan HATP as well. The HATP plan stops at a given abstract level in task decomposition ( 9(b)). Consequently, SHARY needs to further refine these tasks corresponding to the leaves in the HATP plan tree. This is illustrated in 9(a) for **MoveForManipulate** task.



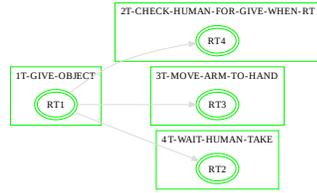
**Fig. 10. Example 1 (First trial and recovery plan):** clean-up task execution: At the top left of the figure, we see a simple version of the first HATP plan computed to achieve the `clean-up`. In the middle and at the right side of the figure, we see the execution stream corresponding to this plan execution. This first plan failed due to robot inability to catch the bottle (even when it has perceived it). Then SHARY asks 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.

**Example 2: Give task and reaction to human commitment**

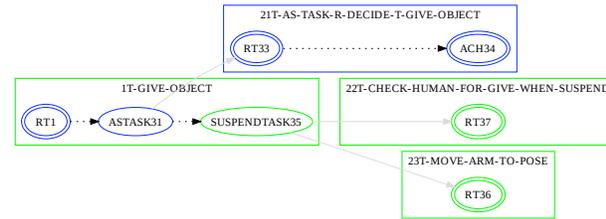
As explained above, monitoring changes in human commitment or focus of attention is a must for a robot that has to act in coordination and/or in collaboration with human partners. This capability is illustrated by the `give-bottle` task presented in figure 11.

**Example 3: Serve a drink and Explore new objects : Initiative taking**

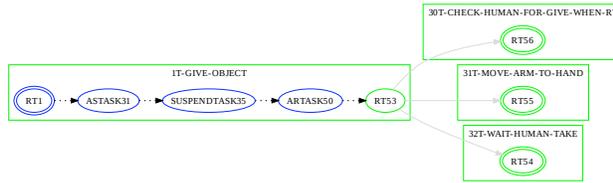
As mentioned in section 2.1 the Task Agenda implements the ability for the robot to manage its high-level goals and to generate new tasks leading to a



(a) The task is started: it consists of 3 tasks including the monitoring task T-CHECK-HUMAN-FOR-GIVE-WHEN-RT



(b) Human has lost attention. The act\_X\_task REALIZE-TASK (or RT) is stopped and



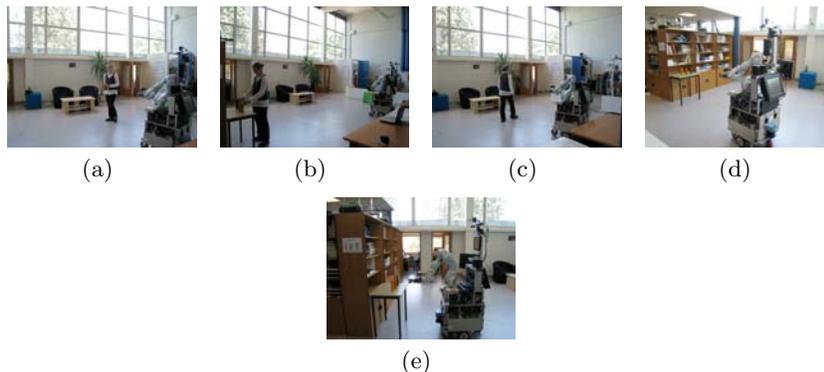
(c) Human looks again at the robot



(d) Jido resumes the task

**Fig. 11. Example 1: Handing a bottle to person.** The give-bottle task consists in handing over a bottle to a given person standing or sitting in the robot vicinity. The sequence shows the ability of the robot to monitor human activity thanks to an instantiation of a communication scheme in the context of this task. In the example, the person has lost intention because of a phone call while Jido was moving the arm towards him. In consequence, the robot suspends the task and waits until the human turns his attention again toward the robot or abandons the task.

kind of initiative taking. This is based on the recognition of chronicles representing human activity. Initiative taking is illustrated by the scenario where the robot detects the need to serve a drink to a person or when the robot decides to acquire information about objects that have been put by a person on a table. Another situation that can be recognized is the will to interact.



**Fig. 12. Example 3: Initiative taking.** These snapshots illustrate the ability of the robot to recognize a chronicle: a person approaches the table near the cupboard and stays still for a moment before leaving. This induces the fact that the person might have put or taken bottles. Jido takes the initiative to approach the table and to update its knowledge using its perception functions.

## 4 Conclusion and Future work

In this paper, we have a robot control architecture dedicated to robot action in presence or in interaction with humans ([2, 4, 3, 14]). We have discussed the main decisional issues involved in such a context and described how they have been implemented in our system.

We have demonstrated through several implemented scenarios various assets of the system: (re)planning abilities, human monitoring management, communication consideration during task achievement,...

Concerning SHARY, future work will consist in the introduction of more flexibility depending of the task or the agent concerned and more generally on the context. This flexibility is two-folds: defining several communication schemes for a same task given the context (e.g. letting the robot to avoid some checks with a known person), defining several recipes for a given act (e.g. for greeting someone, you can choose between smiling or saying hello). The system design allows this flexibility but it is not yet exploited. Indeed,

the difficulty lies in the ability to develop a decisional process at this level. Learning is also an option to be investigated here.

Concerning HATP, future work will be on various aspects. The first is the development of heuristics in the refinement process in order to explore the most promising parts of the solution space first and so to increase HATP speed. Another aspect is about the improvement of temporal constraints management.

Concerning the task Agenda, task scheduling is currently based on priority and each task has a fixed priority. This is not sufficient for a rational and natural behavior. In fact, priority should be dynamic depending on several aspects such as, for instance, task progress. Moreover, we would like to build a system allowing to give and to manage not only short-term but also long-term view/plan to the robot..

**Acknowledgments:** The work described here has been partially conducted within the EU Integrated Project CHRIS (Cooperative Human Robot Interaction Systems) funded by the E.C. Division FP7-IST under Contract 215805. It would have not been realized without the work of a number LAAS robotics team members and more particularly Sara Fleury, Matthieu Herrb, Akin Sisbot, Luis Felipe Marin, Brice Burger, Muhammad Ali, Xavier Broquere, Nizar Sallem.

## References

1. R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand. An architecture for autonomy. *International Journal of robotics Research, Special Issue on Integrated Architectures for Robot Control and Programming*, 17(4), 1998.
2. R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, and R. Chatila. Task planning for human-robot interaction. *The Smart Objects and Ambient Intelligence Conference, sOc-EuSAI 05, Grenoble, France, 2005*.
3. R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, and R. Chatila. Towards human-aware cognitive robotics. In *Cogrob 2006, The 5th International Cognitive Robotics Workshop (The AAAI-06 Workshop on Cognitive Robotics)*, Boston, USA, July 16-17 2006.
4. R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, and Chatila R. Toward human-aware robot task planning. In *AAAI Spring Symposium "To boldly go where no human-robot team has gone before"*, Stanford, USA, March 27-29 2006.
5. P. Althaus, H. Ishiguro, T. Kanda, T. Miyashita, and H. I. Christensen. Navigation for human-robot interaction tasks. *Proc. in IEEE Int. Conf. on Robotics & Automation, New Orleans, USA, 2004*.
6. G. Bailly, M. Brar, F. Elisei, and M. Odisio. Audiovisual speech synthesis. *International Journal of Speech Technology*, 2003.
7. Jur P. van den Berg and M. Overmars. Roadmap-based motion planning in dynamic environments. Technical report, Utrecht University, NL, 2004.
8. C. Breazeal. Towards sociable robots. *Robotics and Autonomous Systems*, pages 167–175, 2003.

9. C. Breazeal, A. Edsinger, P. Fitzpatrick, and B. Scassellati. Active vision for sociable robots. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 31(5):443–453, September 2001.
10. B. Burger, I. Ferrane, and F. Lerasle. Multimodal interaction abilities for a robot companion. In *ICVS*, pages 549–558, 2008.
11. H. H. Clark. *Using Language*. Cambridge University Press, 1996.
12. A. Clodic. *Supervision pour un robot interactif : Action et Interaction pour un robot autonome en environnement humain*. PhD thesis, University of Toulouse, 2007.
13. A. Clodic, R. Alami, V. Montreuil, S. Li, B. Wrede, and A. Swadzba. A study of interaction between dialog and decision for human-robot collaborative task achievement. In *16th IEEE International Symposium on Robot and Human interactive Communication, RO-MAN 2007*, 2007.
14. A. Clodic, V. Montreuil, R. Alami, and R. Chatila. A decisional framework for autonomous robots interacting with humans. *IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN, Nashville, USA)*, August 13-15 2005.
15. A. Clodic, M. Ransan, R. Alami, and V. Montreuil. A management of mutual belief for human-robot interaction. In *IEEE Int. Conf. on Systems, Man and Cybernetics (SMC)*, 2007.
16. P. R. Cohen and H. J. Levesque. Teamwork. *Nous*, 25(4):487–512, 1991.
17. C. Dousson, P. Gaborit, and M. Ghallab. Situation recognition: Representation and algorithms. In *proc. of the 13th IJCAI*, pages 166–172, 1993.
18. T. W. Fong, C. Kunz, L. Hiatt, and M. Bugajska. The human-robot interaction operating system. In *2006 Human-Robot Interaction Conference*. ACM, March 2006.
19. E. Forman and M. A. Selly. *Decision By Objectives (How to convince others that you are right)*. World Scientific, 2001.
20. M. Ghallab, D. Nau, and P. Traverso. *Automated Planning - theory and practice*. Morgan Kaufmann Publishers, 2004.
21. B. Grosz and S. Kraus. The evolution of sharedplans. *Foundations and Theories of Rational Agencies*, 1999.
22. B. J. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86:269–358, 1996.
23. C.C. Kemp and E. Edsinger, A. Torres-Jara. Challenges for robot manipulation in human environments. *Robotics & Automation Magazine, IEEE*, 2007.
24. G. Klein, J. M. Woods, D. D. and Bradshaw, and P. J. Hoffman, R. R. and Fel-tovich. Ten challenges for making automation a "team player" in joint human-agent activity. *IEEE Intelligent Systems*, 19(6):91–95, Nov-Dec 2004.
25. D. Kulic and E. Croft. Pre-collision safety strategies for human-robot interaction. *Autonomous Robots*, 22(2):149–164, 2007.
26. K. Madhava Krishna, R. Alami, and Simeon T. Safe proactive plans and their execution. *Robotics and Autonomous Systems*, 54(3):244–255, March 2006.
27. V. Montreuil, A. Clodic, and R. Alami. Planning human centered robot activities. In *IEEE Int. Conf. on Systems, Man and Cybernetics (SMC)*, 2007.
28. D. Nau, T. C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman. Shop2: An htn planning system. *Journal of Artificial Intelligence Research*, pages 379–404, december 2003.
29. C. Rich and C. L. Sidner. Collagen: When agents collaborate with people. *Proceedings of the first international conference on Autonomous Agents*, 1997.

30. T. L Saaty. *Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World*. RWS Publications, 1999-05-01.
31. T. L Saaty. *Fundamentals of the analytic hierarchy process*. RWS Publications, 2000.
32. C. L. Sidner, C. Lee, C. Kidd, N. Lesh, and C. Rich. Explorations in engagement for humans and robots. *Artificial Intelligence*, 166(1-2):140–164, 2005.
33. E. A. Sisbot, A. Clodic, R. Alami, and M. Ransan. Supervision and motion planning for a mobile manipulator interacting with humans. 2008.
34. E.A. Sisbot, L.F. Marin-Urias, and R. Alami. Spatial reasoning for human robot interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2007*, 2007.
35. E.A. Sisbot, L.F. Marin-Urias, R. Alami, and T. Simeon. A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 23(5):874–883, october 2007.
36. M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.