SAPHARI

SAFE AND AUTONOMOUS PHYSICAL HUMAN-AWARE ROBOT INTERACTION

SEVENTH FRAMEWORK PROGRAMME

COGNITIVE SYSTEMS & ROBOTICS

Project funded by the European Community's 7th Framework Programme (FP7-ICT-2011-7)
Grant Agreement ICT–287513

Deliverable D7.5.2

Prototype of the human-aware robot controller

Report due date: 1 August 2015	**Actual submission date:** 01/08/2015
Start date of project: 1 November 2011	**Duration:** 45 months
Lead beneficiary for this deliverable: UNINA	**Revision:** Final

www.saphari.eu

Executive Summary

This deliverable of workpackage WP7 describes the prototype of the human-aware robot controller specified in the D.7.5.1. The implemented controller permits flexible and interactive execution of structured cooperative plans of actions. Specifically, the human-aware controller integrates cooperative plan execution, human-monitoring, multimodal interaction, attentional regulation, and dialogue management. In this document, we detail the overall implemented prototype, we discuss the system performance in a simulated environment, and finally, we describe the implemented prototype at work in the AIRBUS scenario.

Table of contents

1 The Executive System within the SAPHARI architecture

In Figure 1, we provide an overview of the SAPHARI architecture providing details about the WPs and the modules involved in the design and development of the human-aware robot controller (executive system). More specifically, the human activity is monitored by the classifiers developed in the WP5, the results of this interpretation are provided to the decisional kernel designed and developed in the WP7. This module integrates the dialogue manager, the attentional system (executive and behaviour-based), and the human-aware planner. The actual execution of the robot activities is managed by the flexible reactive processes developed in the WP6, which are continuously monitored by the perceptive modules provided by the WP4.

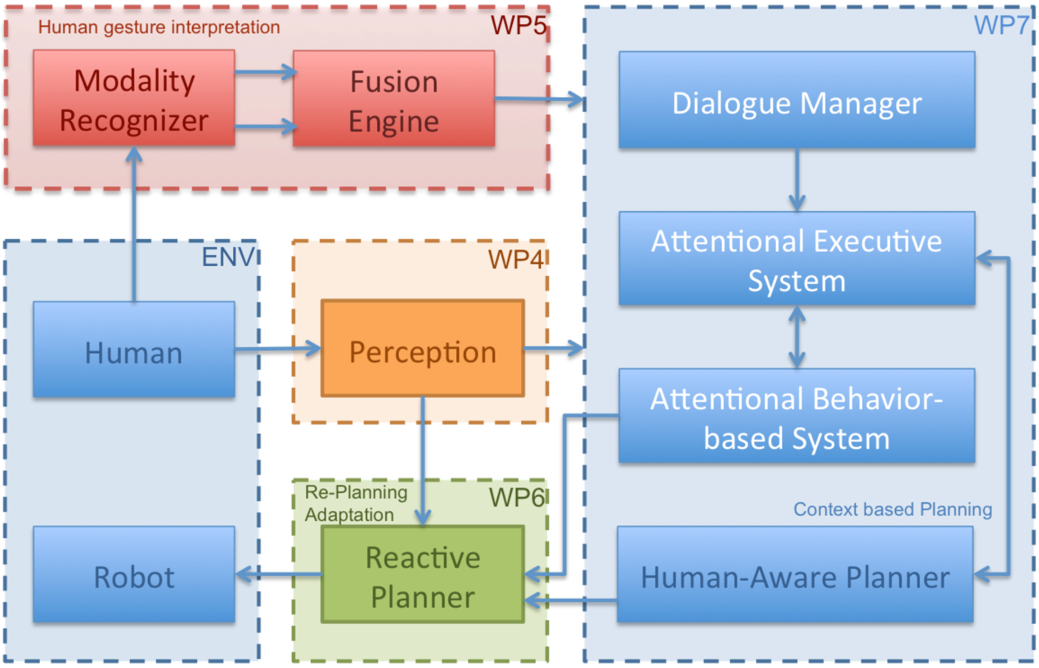

Figure 1: The executive system within the WPs organization in SAPHARI.

2 The Human-Aware Executive System

In this section, we illustrate the human-aware executive system within the overall system architecture (see Figure 2) describing its main components and their interactions.

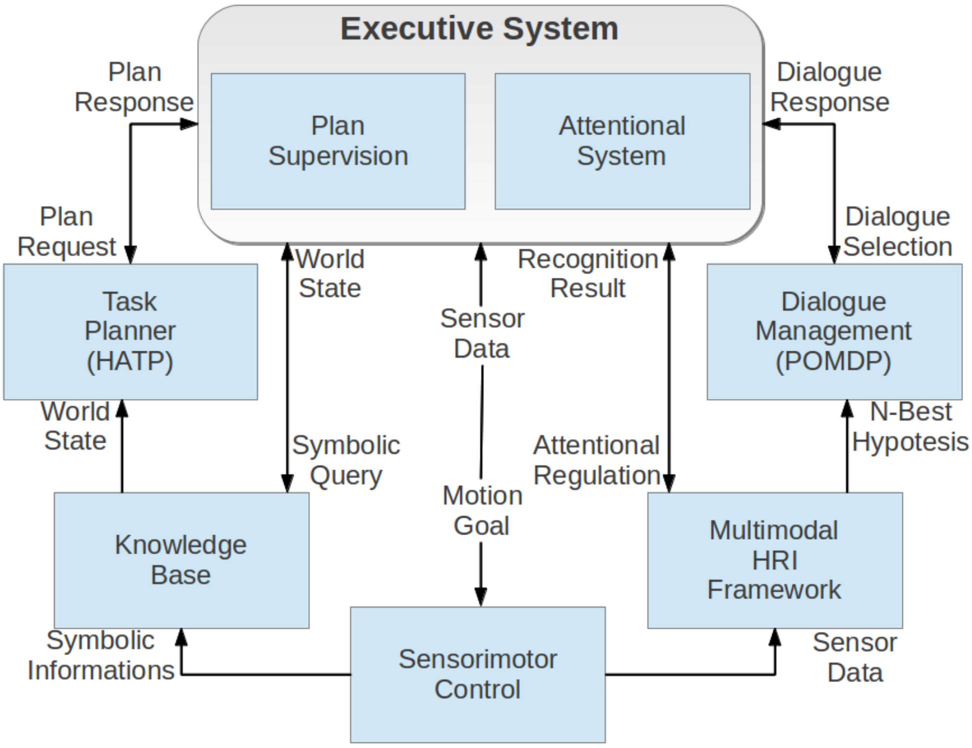

Figure 2: Architecture of the integrated system.

The overall framework combines human-aware planning, attentional plan execution, human monitoring, interaction, and replanning. In this system, a human aware planner generates a shared human-robot cooperative plan, while plan execution is flexibly regulated and adapted by an executive system that supervises the human actions and the environmental changes. This way, the system can deploy replanning only if the adaptation is not possible, e.g. when the human behavior significantly diverges from the computed plan, or in case of execution failures. This process will be better detailed in the following.

1) Supervision System: The supervision system interacts with the modules involved in the overall executive cycle. More specifically, the system is responsible for the following tasks: 1) *Goal management* and interaction with the task planner to generate a multi-agent plan; 2) *Plan monitoring* and execution interacting with the attentional system for action selection and regulation; 3) *Interaction with the motion planner* for action execution; 4) *Plan repair* both at the action level, trying new strategies to perform the current action, and at the plan level, invoking replanning. The replan activities are performed when the robot is not able to perform a selected action or when the generated plan cannot be adapted to the behavior of another agent (e.g. the human involved in the interaction). More details about the supervision system can be found in [Fiore2014].

2) Human Aware Task Planner: The system is endowed with a Human-Aware Task Planner (HATP) [Lalle14], which is based on a Hierarchical Task Networks (HTN) and a SHOP-like [Nau99] refinement process. HATP is able to produce hierarchical plans for multi-agent systems, including humans. The planning problem is defined as a 3-tuple $(\mathrm{g}, \mathrm{s}_0, \mathrm{D})$, which are respectively, the goal, the initial state, and the planning domain. The latter is defined by the pair (A, M), where A is a finite set of operators and M is a finite set of methods.

A method in M is a 4-tuple $(\mathrm{m}, \mathrm{t}, \mathrm{p}, \mathrm{b})$, where m is the name of the method, t is the task/goal, p is a precondition specifying when the method is applicable, and b describes a sequence of operators or methods. In HATP, each operator $\mathrm{A}_{\mathrm{k}}^{\mathrm{a}}$ for an agent a can be associated with a duration $\mathrm{D}_{\mathrm{k}}^{\mathrm{a}}$ and a cost function $\mathrm{C}_{\mathrm{k}}^{\mathrm{ctxt}}$. Moreover, HATP permits to define specific social rules along with a cost for their violation $< \mathrm{S}_{\mathrm{k}}, \mathrm{P}_{\mathrm{k}}^{\mathrm{ctxt}} >$. This way, a plan P is associated with a cost: $\mathrm{Cost(P)} = \sum_{a_i \in P} C_{a_i}^{ctxt} + \sum_{s_k \in P} P_{s_i}^{ctxt}$, where a_i is an action of the plan P and s_k is a social rule. By setting a different range of parameters, the generated plans can be tuned to adapt the robot behavior to the desired level of cooperation.

HATP is able to produce a different stream of actions for each agents, where each stream is a sequential list of actions. Each action has a finite number of precondition links to other actions, which can be part of any stream.

3) Attentional System: The attentional system (see [Cac14a, Cac14b] for details) manages the cognitive control cycle [Pos75, Bot01] and provides the attentional regulations [Norm86, Coop00] during the intractive execution. It receives the generated plan from the supervision system and selects/regulates the robot activities exploiting attention bottom-up (stimuli-oriented) and top-down (task-oriented) influences. This process is managed by a control cycle that continuously updates the working memory (WM) and a set of behaviors (BP) exploiting the task structure defined in the long term memory (LTM) (see Figure 3). The LTM represents the hierarchical cooperative tasks defined in the HATP domain. Each method of HATP is indeed associated with a schema defined by the couple $(\mathrm{name}, [\,\mathrm{ss}_1, \mathrm{ss}_2, \dots, \mathrm{ss}_\mathrm{n}])$, where name is the schema identifier, and $[\,\mathrm{ss}_1, \mathrm{ss}_2, \dots, \mathrm{ss}_\mathrm{n}]$ is a list of sub-schemata; the primitive operators are directly associated with primitive behaviors.

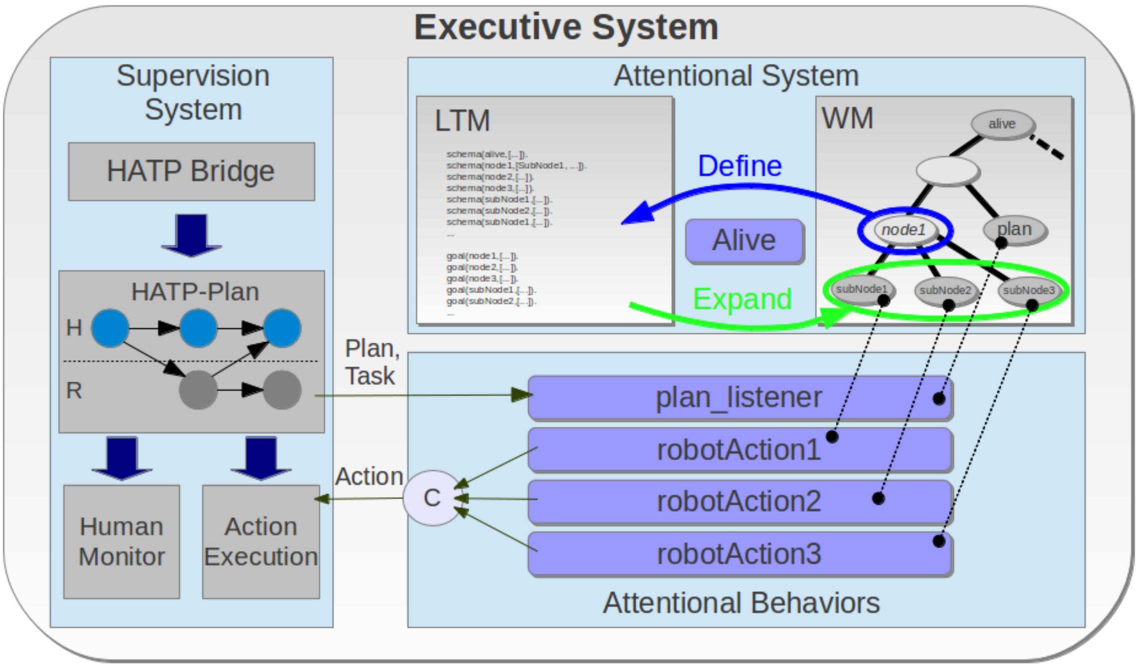

Figure 3: Executive System Architecture: Plan Supervision and Attentional System.

The supervision system provides the attentional system with the generated plan along with an identifier of the planned cooperative task, which is described in the LTM as specified above. Once the task is selected for the execution, its structure is suitably allocated in the **working memory** (WM) and represented as an annotated subtree of the current execution tree represented in the WM. The tree nodes can be either **concrete**, if they are related to specific primitive behaviors, or **abstract**, if they are associated with methods to be hierarchically decomposed. Each concrete behavior represents an active sensorimotor process endowed with a releaser and an attentional mechanism that regulates the temporal resolution at which the process is monitored and controlled. The releaser enables/disables the behavioral effects, while the clock regulates the sensory sampling rate and the frequency of the activations (see [Bur10, Bro14] and Deliverable D.7.5.1 for details). This regulation depends on top-down and bottom-up influences. The first one is directly affected by the perceptual stimuli and the inner state of the behavior, while the second one represents the influence due to the task structure and the current state of the plan execution.

4) Situation Assessment: The situation assessment module produces symbolic facts from sensory data [Millez14] updating the system knowledge base. It receives inputs from different sensors and performs geometrical and temporal reasoning on objects and agents (human or robot). The system can assess distances, orientations, velocities, etc.. Following a perspective taking approach, the situation assessment module can also maintain a different belief state for each agent involved in the scene depending on its knowledge and perception. HATP is able to exploit this information while planning including, when necessary, explicit communications among the agents.

5) Dialogue Management: The dialogue managers integrates the interpreted multimodal action of the human [Ross13] into a coherent dialogue flow with an associated dialogue policy. The dialogue manager is endowed with a set of dialogue models which are provided as graph-based specifications where multiple dialogue flows can be combined in order to build a unique model in a modular and extensible manner (see [Luc13] and D.7.5.1). The resulting dialogue model is represented by a POMDP, which can cast the inherent ambiguity due to noise on the input channels, misunderstanding of human actions or commands, multiple interpretations of a particular observation or non-deterministic effects of robot actions. The solution of the POMDP is a robust dialogue policy for a specific interaction model: it provides a (multimodal) machine action for each belief state of the dialogue. This machine action is then associated with a task to be allocated in WM whose execution is modulated by top-down and bottom-up attentional mechanisms. This way, the machine action in the dialogue policy can be instantiated with contextual and task-related subtasks and arguments and suitably regulated by the attentional system.

3 Executive System Validation

In this section, we discuss the flexibility of the executive system at work in a simulated scenario where a robot is to execute pick, carry, and place tasks. In this context, our aim is to assess the system behaviour in the presence of structured tasks, incomplete/ambiguous commands, and decisional conflicts.

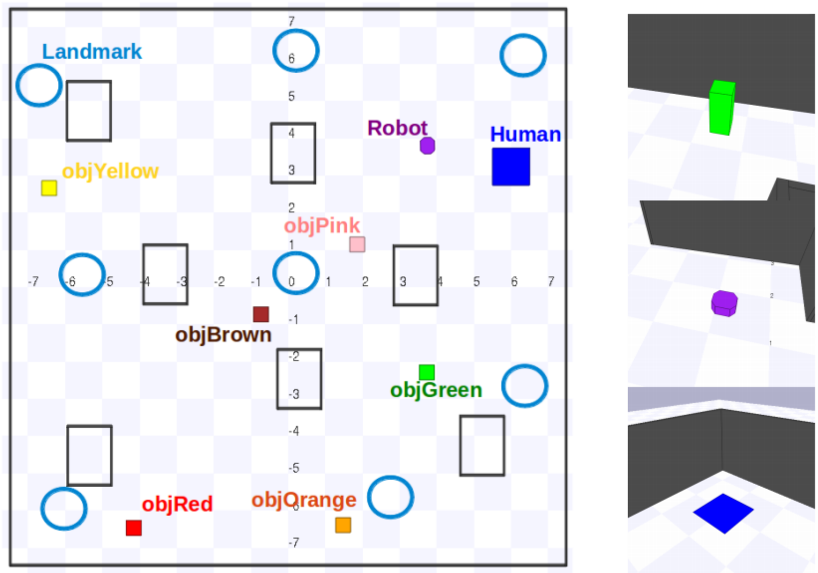

Figure 4: Simulated environment: we have colored objects (small cubes), obstacles (black), the human position (large flat square), and the mobile robot (purple).

Figure 4 illustrates a simulated scenario where a robot is to accomplish pick, carry, and place tasks while interacting with a human. The simulated robotic platform setting is a Pioneer 3 DX mobile robot provided with ultrasonic sensors and a gripper. In this environment, we have several coloured objects that can be taken and carried by the robot. We assume that the robot can hold several objects at the same time.

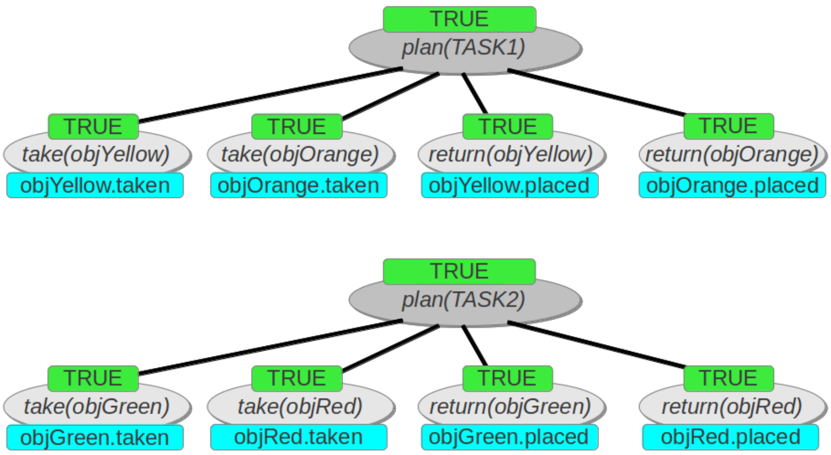

Figure 5: Two conflicting plans represented in the WM. The ordering of the plan actions is from left to right.

In this case study, the aim is to assess how the framework is capable of flexibly interleaving the execution of multiple tasks depending on opportunities. In particular, we assume that the two concurrent plans depicted in Figure 5 are already loaded in the WM and ready for the execution. Each plan represents a sequence of four actions, but their order is not directly enforced by the plan structure: when the preconditions are enabled the execution of the associated actions depends on the attentional state, hence sequence violations are permitted. Notice that the execution of the sequence can be easily constrained by the plan structure, however, in this setting action orchestration (including sequencing) is fully regulated by the emphasis (top-down attention) in order to allow flexible execution. Our aim is to compare the performance of our plan execution mechanism with respect to the best choices, i.e. the ones associated with a minimal total path for the robot. Indeed, the execution of the actions in the two plans should be suitably interleaved trading-off plan completion (top-down influence) and opportunities (bottom-up stimuli) in order to minimize the overall cost. Specifically, during the execution, we evaluate the executive system decisions as follows:

- *True-positives:* active actions, selected by the system, which minimizes the total path while respecting the plans sequences (i.e. the selected action is the best choice).
- *True-negatives:* active actions, not selected, and not expected to be selected, even if enabled and bottom-up stimulated (i.e. competing actions correctly defeated).
- *False-positives:* active actions enabled by the system, which are not suboptimal choices.
- *False-negatives:* active actions which are good choices, but disabled by the system.

TEST	true-positives	true-negatives	false-positive	false-negative
avg	7.5	2.6	0.5	0.1
std	0.8	1.7	0.8	0.3
min	6	0	0	0
max	8	6	2	1
TOTALS	75	26	5	1

Figure 6: TRUE/FALSE POSITIVE/NEGATIVE over 10 runs.

In Figure 6 and Figure 7, we report the results obtained with the execution of the two plans over 10 trials. For each trial, the two plans are concurrently executed with randomly defined positions of the objects (cubes and base). Figure 6 shows that the two plans are usually executed as expected (7.5 with respect to 8 best actions) with few suboptimal choices (0.5) and sparse missed opportunities (0.1) despite distractive alternatives (2.6 true-negatives).

Performance			Error	
Accuracy	**Precision**	**Recall**	**Violation**	**Worse**
0.9439	0.9375	0.9868	0.8333	0.3333

Figure 7: Performance during the multiple plan tests.

This is better illustrated in Figure 7 where we can observe that the system makes over 94% of correct choices (accuracy) with an executed action (precision) which is usually the expected one (93%). Moreover, the wrong choices are usually (83,3%) due to a violation of the plan sequence (Violation in Figure 7), while suboptimal choices appear more rare (Worse in Figure 7), sometimes associated with violations (sequence violations and suboptimal choices are not mutually exclusive). The sequence violation is usually obtained when the top-down boost towards plan completion is not sufficient to induce the activation order because the bottom-up influence is too strong (e.g. proximity of an object to be picked after in the plan). Nevertheless, the probability of making a good choice (expected actions) among the possible choices (enabled actions), is high (recall over 98%).

4 Plan Execution and Attentional Regulations

The plan-execution cycle is managed by the interaction of the supervision and the attentional system. Given a task to be executed the supervision system invokes HATP to generate a multi-agent plan. The attentional system receives the plan, which is represented as a sequence of actions $(a_1, \ldots, a_n)$ for each agent involved in the interaction, together with the requested task. The attentional system allocates a new tree in WM representing the hierarchical structure of the requested task, while the generated HATP plan is exploited as a guidance for the action selection and execution (see Figure 8). Notice that this approach allows us to keep the overall structure of the hierarchical task, which is managed by the attentional system, together with the goal-oriented sequence of action generated by the planner. Indeed, the task tree allocated in the WM contains alternative methods and action primitives (e.g. $take$ and $receive$ are two alternative ways to get an object) that permits a flexible adaptation depending on the current executive state (e.g. choosing $take$ instead of $receive$ if an object is close). Moreover, the generated plan is here used as a soft guidance towards the goal accomplishment exploiting attentional regulation. Contentions between alternative possible actions can be solved using both the top-down and bottom-up regulations. The bottom-up stimulation emphasizes actions that are more accessible to the robot, while the top-down stimulation exploits the task structure and the HATP plan to emphasize the task-related and goal-oriented actions.

More specifically, the attentional state of each behavioral schema in the leaves of task tree is defined by the couple (λ, μ), where $\lambda \in (0,1]$ (frequency) represents the contribution of the bottom-up stimuli, while $\mu \geq 1$ (magnitude) is for the top-down influence. For each node in the WM, we assume $\mu = 1$ when no top-down influence is present, while a magnitude change for a node is top-down inherited by all its descendant. The magnitude variations are associated with WM updates. Namely, when a subgoal is accomplished, the parent emphasis of the node is increased by a default constant value k_g; analogously, when an action of the plan is allocated in the WM, the associated behavior in WM is increased by k_p. The overall attentional regulation for a specific behavior is given by the emphasis $e = \frac{\mu}{\lambda}$ (which represents the activation period of the associated clock).

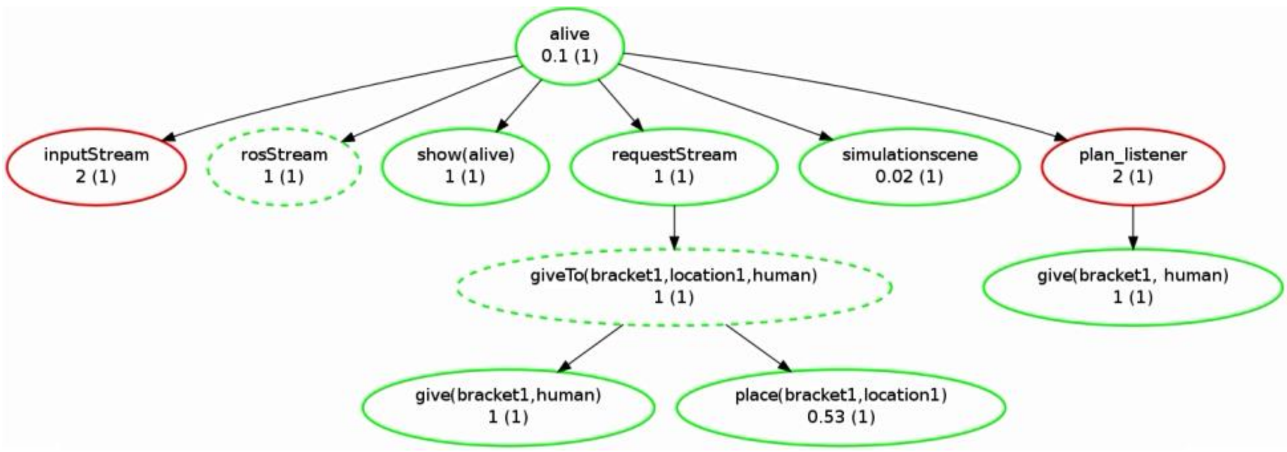

Figure 8: Example of a task tree in the WM: inactive nodes are in red, dotted nodes represent abstract behaviors.

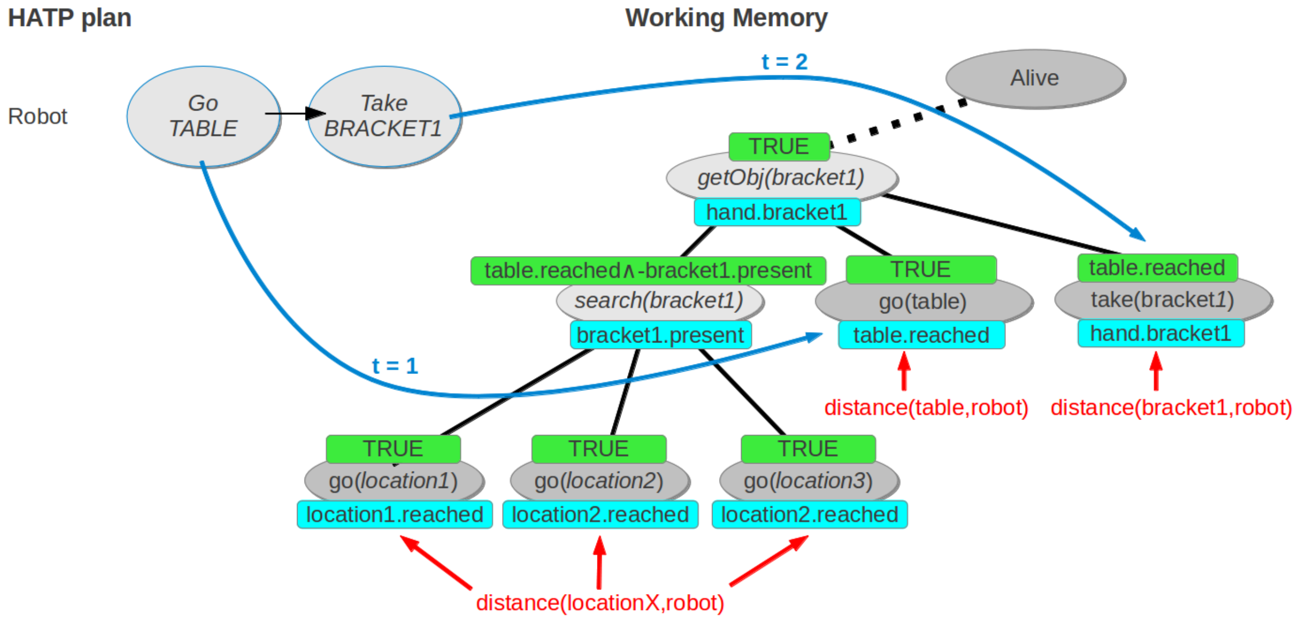

Figure 9: Plan-based (top-down) and environmental influence (bottom-up) during the execution. Green and light blue solids are, respectively, preconditions and goals of tasks/schemata. Top-down regulations are provided by the actions of the HATP plan (blue arrows), while bottom-up regulations are induced by environmental stimuli (red arrows).

Figure 9 shows the combined effects of plan-based (top-down) and environmental (bottom-up) influences. Here, the task of getting an object (e.g. $\mathrm{getObj(bracket1)}$) can be performed either by directly going to the table, if the object is detected, or by searching for it otherwise. In this context, a bottom-up stimulus like object proximity affects specific concrete behaviors (e.g. $\mathrm{take(bracket1)}$ is enhanced if $\mathrm{bracket1}$ is close to the robot), while the planned actions provide top-down regulations (e.g. the command "$go\ table$" enhances the magnitude of the $go(table)$ behavior). The emphasis combines these effects and provides an action selection criterion. This way, in case of opportunities and unexpected events, the attentional system can retrieve alternative methods from the task definition without replanning.

5 System Implementation

In this section, we describe the main tools and methodologies exploited for the implementation of the executive system and the communication among the modules.

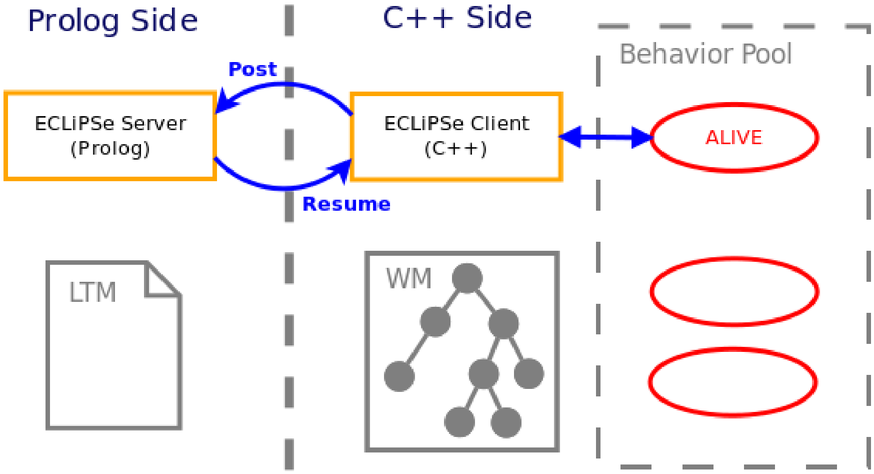

Figure 11: Sketch of the implemented prototype: the LTM is managed by an ECLiPSe Prolog server; each behavior is implemented by a thread, while a special thread (alive) manages the WM updates.

The executive system is developed on a Linux platform (Ubuntu 12.10), implemented as a C++ multithreaded application that interacts with an ECLiPSe prolog engine. ECLiPSe prolog is an open-source software system for the cost-effective development and deployment of constraint logic programming applications (e.g. in the areas of planning, scheduling, resource allocation, timetabling, transport etc.). ECLiPSe combines logic programming and constraint solvers providing a high-level modelling and control language, interfaces to third-party solvers, an integrated development environment, and several interfaces to host environments.

Figure 10 illustrate the interaction with the ECLiPSe solver. The ECLiPSe-based server is exploited by the alive process to expand the tree nodes in the WM according to the definition of the schemata provided in the LTM. Indeed, the LTM (Figure 12) is a repository of prolog predicates that can represent either goals or schemata for concrete and abstract behaviours. The main predicates of the LTM are:

- **schema(+SchemaInstance, +SonList)** which associates a schema instance (e.g. *goto(X,Y)*) to the related list of sub-schemata *[*$schema_1$*,*$schema_2$*, ..., *$schema_n$*]*, and any sub-schema is a 3-list *[name, emph, releaser]*, containing the name, the default level of emphasis and the releasing formula.
- **goal(+SchemaInstance, +GoalList)** which provides the schema of the goal formula.

The client-server communication between LTM and the rest of the system is implemented by the post-resume paradigm provided by the ECLiPSe prolog SDK. Specifically, when a new node is allocated in WM, the *alive* process –the one that manages the overall executive cycle- asks for the list of the associated sub-schemata in the LTM (post phase); in order to serve this request, the prolog server queries the LTM and sends back the results to the client (resume phase).

```
schema(alive,[
        [sonarStream,1,["TRUE"]],
        [engineStream,1,["TRUE"]],
        [blobStream,1,["TRUE"]],
        [inputStream,1,["TRUE"]],
        [requestStream,1,["TRUE"]]]).

schema(goto(X,Y),[
        [avoid,1,["TRUE"]],
        [gotoxy(X,Y),1,["TRUE"]]]).

schema(followColor(Color),[
        [avoid,1,["TRUE"]],
        [reachColor(Color),1,["TRUE"]]]).

schema(searchColor(X),[
        [avoid,1,[-X.near]],
        [wander,1,[-X.present]],
        [reachColor(X),1,[X.present]]]).

schema(explorexy(X,Y),[
        [avoid,1,["TRUE"]],
        [gotoxy(X,Y),1,["TRUE"]],
        [wander,1,[gotoxy(X,Y).arrived]]]).

schema(gotoColor(X,Y,Color),[
        [explorexy(X,Y),1,[-Color.present]],
        [followColor(Color),1,[Color.present]]]).

schema(pickup(Obj),[
        [place(Obj,C,knapsack),1,["TRUE"]]]):-color(Obj,C).
```

Figure 12: Prolog-based representation of schemata in the LTM.

The rest of the system (including WM and attentional behaviours) is implemented as a multithreaded C++ application wrapped in a specific ROS node. The I/O communication exploits ROS topics.

Each allocated behavior is executed by a specific thread, while two main data structures (WM and WMV) are defined in the shared memory:

- The **Working Memory (WM)**, which is a tree data structure where each node represents a specific behaviour (behaviour activations, attentional state, goal status, etc.) and the parenthood relation is defined according to the LTM predicates.
- The **Working Memory Variables (WMV),** which is implemented as a hash table that associates a variable name to the related value.

The communication between the attentional system and the supervision/planning system (OPRS-based) is managed through an OPRS bridge, which permits the bidirectional exchange of string messages.

6 Real-World Case Study

The integrated system has been tested in a case study inspired by a human-robot co-working domain (see [Ala14]). In this setting, the aim is to test whether the integrated system can adapt its execution to unexpected behaviors of the human, avoiding the computation of a new plan, while enabling a smoother and more naturalinteraction. A demonstration video of the scenario is available at the following link http://wpage.unina.it/jonathan.cacace/Media/icsr2015.mp4.

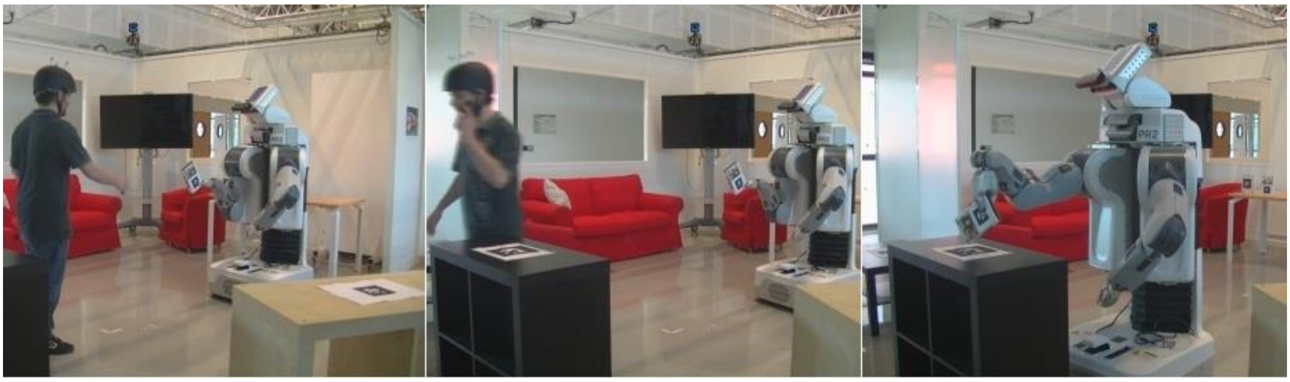

Figure 13: Experimental scenario: the robot switches from handover to place.

The environment is set as follows: there are three working locations, each containing a slot, and a table with a set of objects including a glue bottle and some brackets (see Figure 13). The user and the robot must cooperatively install the brackets in the slots, by first cleaning the slot, then applying the glue in the area, and finally fixing the bracket on it (see Figure 14).

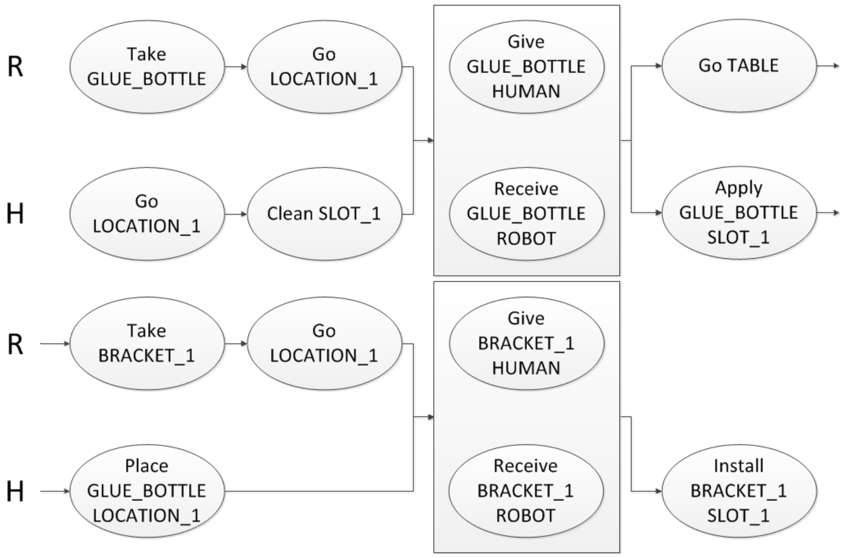

Figure 14: HATP plans: (up) the human is to apply the glue on the SLOT 1; (down) the human is to install the bracket. R and H represent the Robot and the Human actions, respectively; the box highlights the collaborative actions.

At the start, the supervision system invokes HAPT to obtain a plan of actions. In Figure 14, we find an example of a generated HATP plan: the ROBOT should bring the GLUE_BOTTLE and the BRACKET_1 to the HUMAN agent, who is to glue the SLOT_1 position and install the bracket on it. The plan is composed of two streams, one for each agent, but some actions (i.e. handovers) are collaborative and involve both the agents. The supervision system provides the plan and the associated task to the attentional systems, then it starts monitoring the human actions and the environment while executing the actions proposed by the attentional system.

We now describe the system at work by illustrating some significant segments of the scenario explained above. In particular, we show how the integrated system can flexibly adapt the execution to unexpected behaviors of the human, in so avoiding the computation of a new plan while enabling a smoother and more natural interaction.

1) Handover to Search: In this segment the robot is waiting to receive a bracket from the human in order to install it. Indeed, in the planned sequence the human should bring the object to the robot, however, the human remains idle and does not interact as expected. According to the plan, the robot should keep waiting for the human, however, the activations of the receive behavior, in the absence of external stimuli, decrease with time. Therefore, after some seconds of waiting, since the human does not cooperate in the task, the activations of the search behavior become dominant (as illustrated in Figure 15 A), hence the robot abandons the handover and starts searching for the object by itself (Figure 15 A).

2) Take to Search: In this segment the robot should get the BRACKET_1 and give it to the human, which is to fix it on a slot. As suggested by the plan, the robot travels to the TABLE in order to take the object, but cannot find it there. Therefore, analogously to the previous case, the attentional system switches to a search behavior (the activations are illustrated in Figure 15, B) where the robot inspects other locations looking for BRACKET_1. As soon as the object is found, the take action becomes dominant again allowing the robot to continue the plan (Figure 15, B).

3) Handover to Place: In this segment, the human is to obtain the GLUE_BOTTLE in order to glue SLOT_1. Following the HATP plan, the robot tries to perform a handover, but the human moves away from the working space. The supervision system cannot replan since the human has not performed its action and the plan is still valid. However, the attentional system can solve the impasse without waiting for the human initiative. Indeed, the bottom-up stimulation of the *give* decreases as the robot-human distance increases, while the *place* activation remains stable (because affected by the table position). When *place* wins the contention (see activations in Figure 15, C), the robot places the object on the work location allowing the plan continuation. Once the *place* is accomplished, the supervision system can manage the action substitution changing the monitored human action from *receive* to *take* (Figure 15, C).

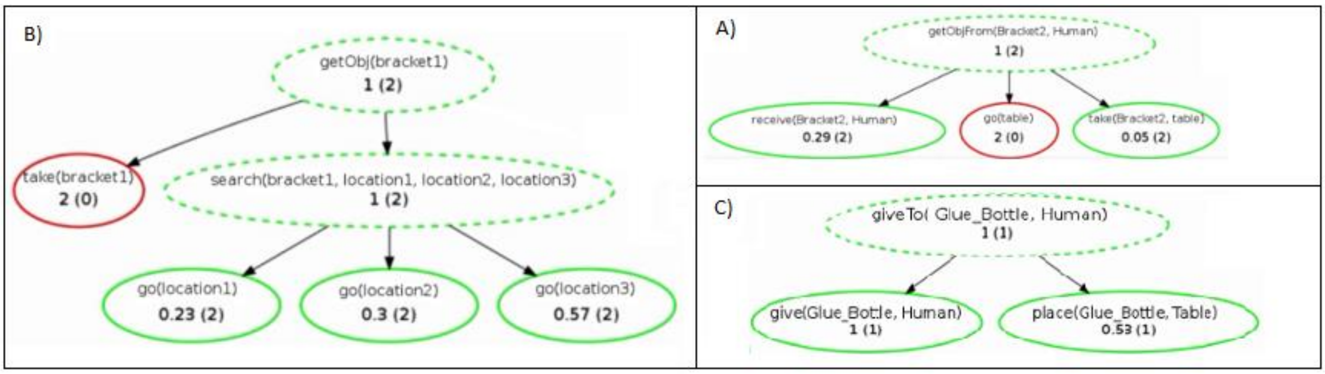

Figure 15: Examples of top-down and bottom-up in influences during the interaction; for each node n, (m) represent 1/e and μ respectively, where e is the emphasis and μ is the magnitude.

7 Conclusions

In this document, we detailed the design and the development of a human-aware controller that permits flexible and cooperative execution of structured collaborative tasks. The system exploits attentional regulation to orchestrate and regulate the robot actions with respect to the human initiative and the cooperative task. We described the overall implemented system focusing on the executive control cycle. We tested the implemented prototype both in simulation and in real-world case studies form the AIRBUS domain. In a simulated case study, we tested flexible and interactive execution of multiple and collaborative plans showing how plan guidance and attentional regulation can be exploited to solve decisional impasses driving the system towards the task/plan accomplishment. In the real-world scenario, we tested the overall developed prototype showing how the proposed executive system permits a flexible plan execution and smooth recover in the presence of unexpected behaviors and events.

8 References

[Ala14] Gharbi, Benjamin Vadant, Raphael Lallement and Adolfo Suarez. On human-aware task and motion planning abilities for a teammate robot. Rachid Alami, Mamoun. In Proc. of WSRSS 2014.

[Bot01] M. M. Botvinick, T. S. Braver, D. M. Barch, C. S. Carter, and J. D. Cohen. Conflict monitoring and cognitive control. Psychological review, vol. 108, no. 3, pp. 624-52, 2001.

[Bro14] X. Broquere, A. Finzi, J. Mainprice, S. Rossi, D. Sidobre and M. Staffa. An Attentional Approach to Human-Robot Interactive Manipulation. International Journal of Social Robotics, Springer, pp. 1-21, 2014.

[Bur10] E. Burattini, S. Rossi, A. Finzi, and M. C. Staffa. Attentional modulation of mutually dependent behaviors. In Proc. of SAB 2010, pp. 283–292, 2010.

[Cac14a] R. Caccavale, A. Finzi, L. Lucignano, S.Rossi, M. Staffa. Attentional Top-down Regulation and Dialogue Management in Human-robot Interaction. In Proc. of HRI-14, pp. 130-131, 2014.

[Cac14b] R. Caccavale, E. Leone, L. Lucignano, S. Rossi, M. Staffa, A. Finzi. Attentional regulations in a situated human-robot dialogue. In Proc.of Ro-MAN-14, pp. 844 – 849, 2014.

[Coop00] R. Cooper and T. Shallice. Contention scheduling and the control of routine activities. Cognitive Neuropsychology, vol. 17, pp. 297–338, 2000.

[Fiore14] M. Fiore, A. Clodic and R. Alami. On Planning and Task achievement Modalities for Human-Robot Collaboration. In Proc. of ISER 2014.

[Lalle14] R. Lallement, L. de Silva, and R. Alami, HATP: An HTN Planner for Robotics. In 2nd ICAPS Workshop on Planning and Robotics (PlanRob 2014), 2014.

[Luc13] L. Lucignano, F. Cutugno, S. Rossi, and A. Finzi. A dialogue system for multimodal human-robot interaction. In Proc. of ICMI-13, pp. 197–204, 2013.

[Mainp11]. J. Mainprice, E.A. Sisbot, L. Jaillet, J. Cortes, R. Alami, and T. Simeon. Planning human-aware motions using a sampling-based costmap planner. In Proc. of ICRA-11, pp. 5012 – 5017, 2011.

[Millez14] G. Milliez, M. Warnier, A. Clodic, and R. Alami. A framework for endowing interactive robot with reasoning capabilities about perspective taking and belief management. In Proc. of Ro-Man-14, pp. 1103 – 1109, 2014.

[Nau99] D. S. Nau, Y. Cao, A. Lotem, and H. Muñoz-Avila. SHOP: Simple hierarchical ordered planner. In T. Dean, editor, Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99), pp. 968–973, 1999

[Norm86] D. A. Norman and T. Shallice. Attention to action: Willed and automatic control of behaviour. In Consciousness and self-regulation: Advances in research and theory, vol. 4, pp. 1–18, 1986.

[Pos75] M. I. Posner and C. R. R. Snyder. Attention and cognitive control. In Information Processing and Cognition, pp. 55–85, 1975.

[Ros13] S. Rossi, E. Leone, M. Fiore, A. Finzi, and F. Cutugno. An extensible architecture for robust multimodal human-robot communication. In In Proc. of IROS-2013, pp. 2208 – 2213, 2013.