SAPHARI

SAFE AND AUTONOMOUS PHYSICAL HUMAN-AWARE ROBOT INTERACTION

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

Deliverable D7.5.1

Specification of a human-aware robot controller

Report due date: 1 November 2014	**Actual submission date:** 1/11/2014
Start date of project: 1 November 2011	**Duration:** 36 months
Lead beneficiary for this deliverable: UNINA	**Revision:** Final

www.saphari.eu

Executive Summary

This deliverable of workpackage WP7 provides a detailed specification of a human-aware robot controller (decisional kernel). The process of human monitoring and the generation of robot behaviours are based on the primitives and sensory-motor processes developed in WP4, WP5, and WP6.

The human-aware controller developed in SAPHARI combines attentional regulation and dialogue management to monitor and coordinate the human-robot interaction (HRI) [Brea02]. A natural and effective interaction between humans and robots can be modeled as a multimodal dialogue flow, involving speech, gaze orientation, gestures, on the other hand, attentional mechanisms can be used to orient and focus the robotic and the human perceptive and cognitive processes during the interaction. The proposed executive system should manage and regulate the multimodal dialogue between the human and the robot by exploiting top-down and bottom-up attentional regulations. Inspired by attention and cognitive control literature in psychology and neuroscience [Pos75, Coh04], we assume that the attentional influence can be driven by both high level tasks (top-down) and external/internal stimuli (bottom-up). In this perspective, the role of attentional mechanisms is to orchestrate multiple processes, at different levels of abstraction, possibly in conflict. The human-aware robot controller proposed for the project integrates these mechanisms. More specifically, the framework proposed within SAPHARI combines a multimodal real-time HRI system, a dialogue manager, and a layered cognitive control architecture. The dialogue between the human and the robot is here modeled as a Partially Observable Markov Decision Process (POMDP) that can capture the inherent ambiguity of the situated communication. The generated dialogue policy provides an interaction multimodal template (involving not only speech, but also gestures, gaze directions, etc.) which can be instantiated and continuously adjusted with respect to the environmental and the operative context by the attentional system. Following this approach, the cognitive control cycle can modulate and polarize the robot execution by enhancing the attentional processes which are aligned with the operative (top-down) and environmental (bottom-up) state, while inhibiting the ones which are not coherent. We tested the system at work in interactive scenarios where the human and the robot have to interact in order to accomplish cooperative tasks.

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 which are involved in the attentional executive system. More specifically, the human activity is monitored by the classifiers provided by the WP5 and these results are provided to the decisional kernel designed and developed in the WP7. This is obtained by the interaction of 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 provided by the the WP4.

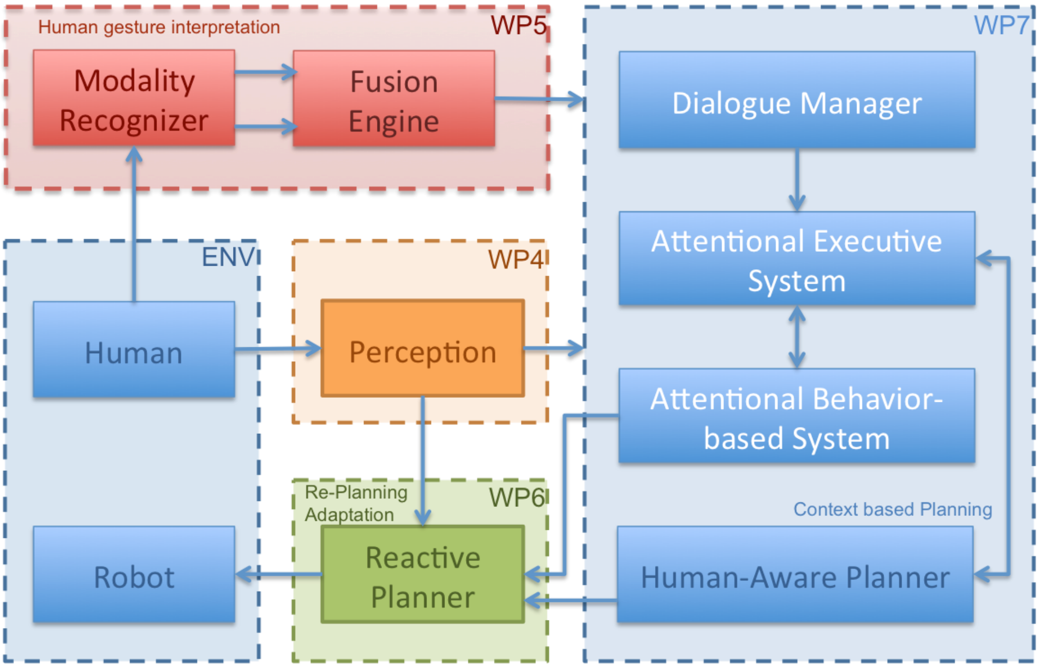

Figure 1: The Attentional System with respect to WPs organization in SAPHARI.
Here, we only detail the components involved in the attentional regulation and human-robot interaction.

2 The HRI Framework

We designed a supervisory system for human-robot collaborative interaction that integrates deliberative, executive, and reactive control. The framework is to modulate both reactive and deliberative processes taking into account the human-robot interactive scenario considering, in particular, user safety, interaction naturalness, and effectiveness. For this purpose, we designed and investigated a layered deliberative/reactive supervisory system endowed with attentional mechanisms, which focus sensory acquisitions/processing and regulate behaviours activations with respect to the human activities, tasks execution, and the environmental context. The aim is to provide the executive system with a supervisory attentional system [Norm86, Coop06] to suitably manage novel and stereotypical interactive situations by combining deliberative and reactive behaviours, while monitoring and regulating multiple concurrent and cooperative activities [Kahn73]. More specifically, the attentional system we propose is based on the following features:

• *Hybrid Control System*. We assume a hybrid control architecture integrating a behaviour-based reactive system, an executive control system, and a deliberative system.

• *Supervisory Attentional System.* The executive control combines reactive and goal-oriented behaviours using attentional mechanisms to orchestrate automatic reactions and activities that are scheduled on the basis of structured tasks (provided by the deliberative system and/or the dialogue manager).

• *Frequency-based Attentional Monitoring and Modulation.* Attentional mechanisms focus monitoring and behaviours activations on relevant activities and external stimuli. For each behaviour, the process of changing the sensory sampling rate and action activations is interpreted as an increase or decrease of attention towards internal and external processes: the higher the frequency, the higher the resolution at which an activity is monitored and regulated.

The HRI architecture proposed in this work is depicted in Figure 2. The cognitive control cycle involves three main modules: a behavior pool (BP), a working memory (WM) and a long term memory (LTM).

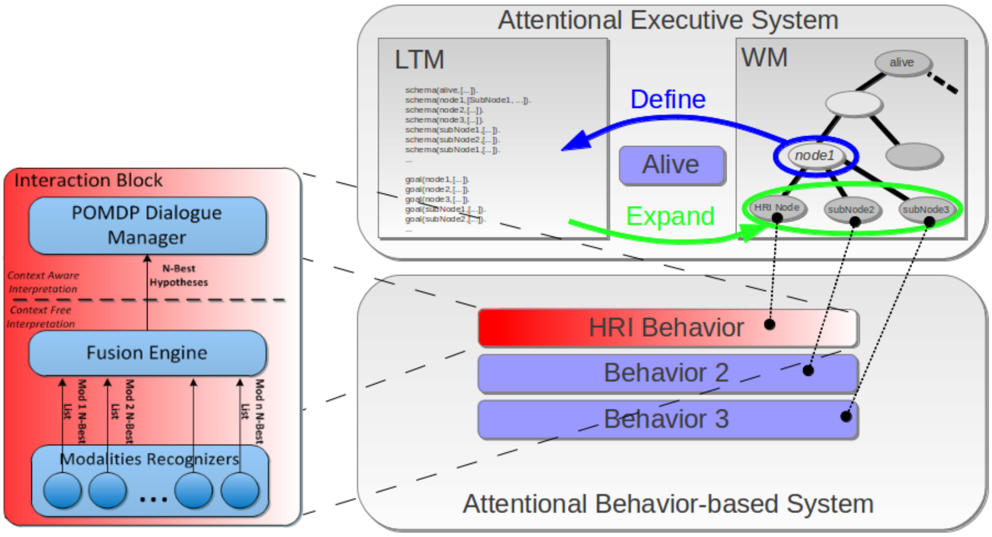

Figure 2: Attentional System and Multimodal Interaction.

The BP represents a set of active behaviors, which may contribute to the execution of a complex cognitive task. The WM contains a representation of the current executive state and a representation of the tasks in the attentional focus of the system. These include all the tasks the system is executing or willing to execute. Finally, the LTM is a repository where the definition of all the tasks available to the system are stored.

The cognitive control cycle is managed by a special behavior, called alive (see Figure 2), which continuously updates the behavior pool and the working memory exploiting the task definitions provided by the LTM. This process will be better detailed in the following.

1) **Attentional behaviors:** We assume that each behavior of the BP is structured as in [Bur10]. Specifically, following a schema theory approach, a behavior is composed of a Perceptual Schema [Arb87], which elaborates sensor data (behavior specific stimuli), a Motor Schema, producing the pattern of motor actions, and a control mechanism, based on a combination of a clock and a releaser. The releaser enables/disables the activation of the Motor Schema, while the clock regulates the sensory sampling rate and the behavior activations. This regulation is provided by an updating function, one for each behavior, which represents our main bottom-up attentional mechanism: it tunes the temporal resolution at which a behavior is monitored and controlled. We refer the reader to [Bur10] for additional details. Differently from [Bur10], here we assume a system where the clock frequency can be modulated not only by the perceptual stimuli (bottom-up), but also by the executive state of the system exploiting the structures available in the WM (top-down). Moreover, we introduce an additional external releasing mechanism that can enable the behavior activations depending on the executive state of the overall system (see Figure 3).

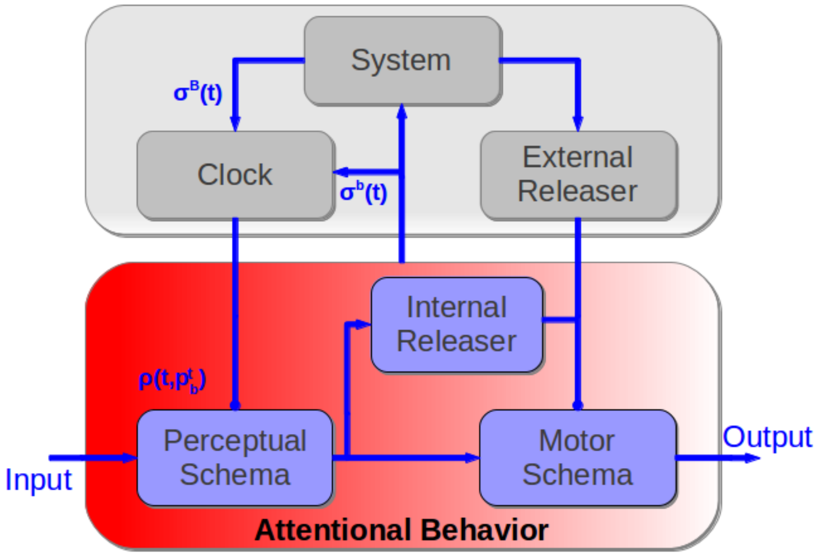

Figure 3: Schema theory representation of an attentional behavior.
Both the clocks and the releasers are top-down and bottom-up regulated.

2) **Working Memory:** The WM is a critical element of the system because it maintains the executive state and the structure of the tasks in the attentional focus of the system. The hierarchical tasks active in the WM are represented as tree [Coop06] (see Figure 3) enhanced with additional information about the behavior execution (clock frequency, releaser status, variables, etc.). The hierarchical structure of the taks follows a typical representation shared by both artificial and biological models of tasks [Rose07, Norm86], Coop00]. Each node of the WM can be classified in two categories:

- ***Concrete*,** representing an instance of an attentional behavior (e.g. *pickUp(objRed)* in Figure 3).
- ***Abstract*,** representing a chunk [Mill56] which may be hierarchically decomposed in subtasks (e.g., *take(objRed)* in Figure 4) .

Our cognitive cycle exploits the WM as follows. Initially, we assume a set of behaviors allocated to manage the basic system activities (e.g. alive, interaction block, etc.). Each behavior in the BP can affect the WM by inserting new nodes. For example, if the interaction block allocates a *take(objRed)* as a consequence of a human request, then *alive* (which is periodically activated to check for new nodes at each clock tic) will try to expand *take(objRed)* (see Figure 4) allocating other nodes as specified in the LTM (see Figure 5). The latter contains production rules representing hierarchical definitions of the available behaviors (analogously to [Lai87]). When a concrete node is allocated in the WM, the associated behavior is awakened by *alive*. The tree structure of WM, is also endowed with an *external releaser* (ER) for each node. These ERs (green in Figure 4) are boolean expressions representing guards to be satisfied to enable the execution of a behavior. Therefore, in order to activate a behavior, besides the internal releaser, not only its ER, but also the ERs of the ancestors must be satisfied. Finally, a node could be also provided with a goal, which is achieved after the completion of the behavior.

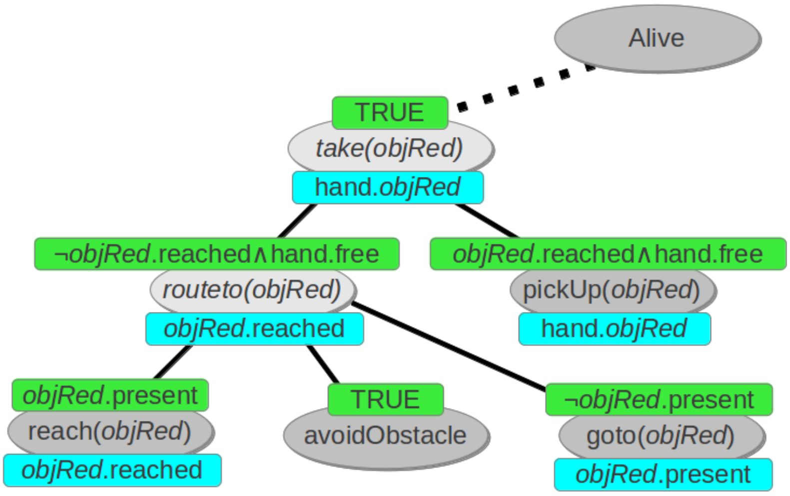

Figure 4: Hierarchical task in the WM: ER, behaviors, and goals are, respectively, in green, gray, and blue.

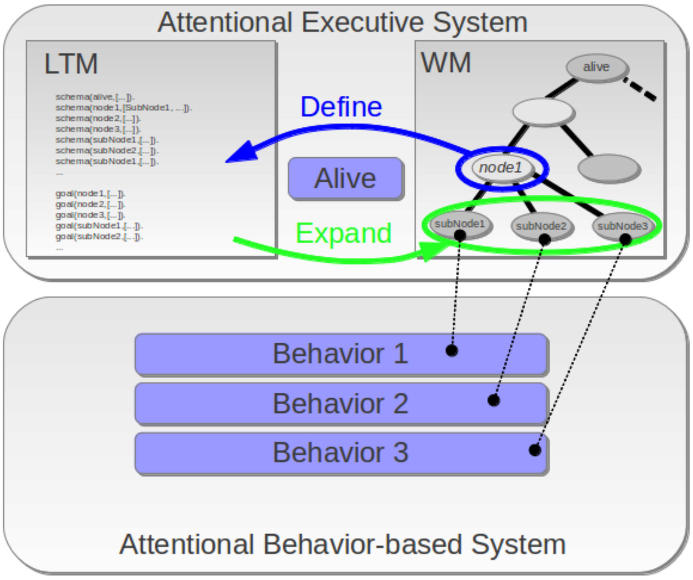

Figure 5: Node expansion in the Attentional Executive System.

3) **Emphasis:** The control cycle described above connects the execution of multiple behaviors with the hierarchical task structure provided in the WM, however, no explicit mechanism (beyond releasers) is

provided to avoid conflicts or erratic activities. For instance, many behaviors may access to a single resource in the WM, generating a crosstalk interference [Moz98]. These conflicts can be either prohibited by construction or solved by means of an evaluation function [Bot01]. We follow the latter, more flexible, approach by introducing a function that we call *emphasis*. This function provides a modulation mechanism that combines two types of influences on concrete behaviors:

– ***Frequency***: the behavioral clock period p_i affected by perceptual stimuli.
– ***Magnitude***: an externally provided values m_i representing the influence due to the WM status.

The first one is a bottom-up attentional mechanism, while the second captures top-down influences. Therefore, the attentional state of each behavioral schema in the tree can be represented by the couple (p_i, m_i). We define the *emphasis* as the frequency enhanced by the magnitude, that is $e_i = m_i/p_i$. By default, the *magnitude* is set to 1 for each node in the tree (i.e. this means that no top-down influence occurs and only the bottom-up value is active, namely $1/p_i$). If a node changes its magnitude, this updated value is inherited by all the child nodes. In order to change the magnitudes according to subgoal achievements we introduce a heuristic mechanism explained in the following. When a subgoal is accomplished, the emphasis of the parent node is increased by a constant value, which is then propagated to all the child nodes. This mechanism induces a soft teleological drive towards the completion of the open subtasks. The emphasis affects both the adaptive clock and the output values. More specifically, the clock period is reduced by m_i (with $m_i \geq 1$), hence the updated period is $p' = 1/e_i$. As for the output, given a (not mutually exclusive) variable v_i (e.g. the velocities of the motors) in the WM which is affected by the output of a set of behaviors, the emphasis can be exploited to weight and combine these multiple v_i contributions (one for each behavior) as follows: $v_i = \sum_i (e_i \times v_i) / \sum_i e_i$. The two effects of the emphasis (acceleration of the clock and modulation of the combined outputs) allows us to solve the conflicts in a smooth way: not only the emphasized behaviors provide more frequent updates, but also their contribution is amplified. Since the amplification is associated to a drive towards the goal accomplishment, the goal-oriented behaviors become dominant, hence overcoming behaviors contentions and decisional impasses (see Figure 6).

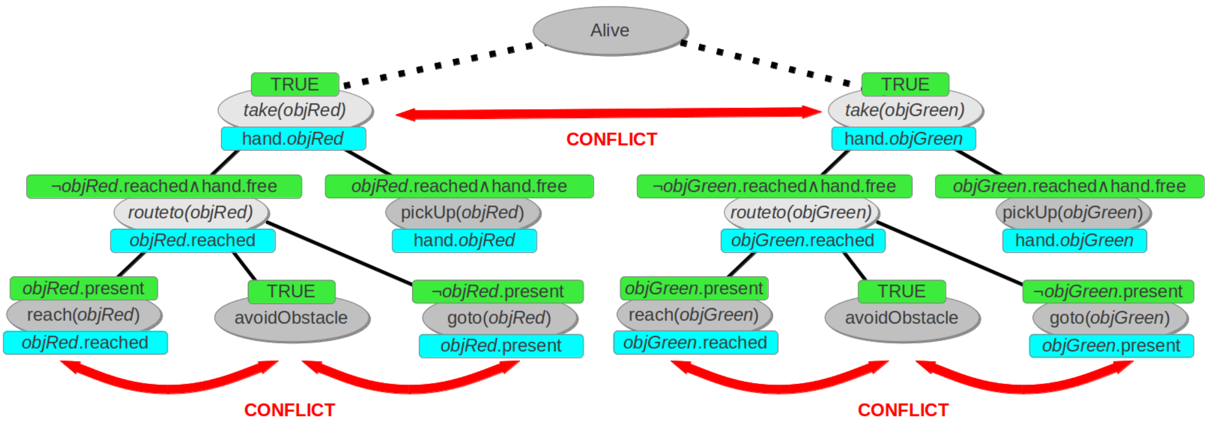

Figure 6: Example of conflicting tasks.

3 Dialogue Management System

The multimodal interaction module is to recognize the multiple human commands and actions, such as utterances, gaze directions, gestures or body postures, and to provide an interpretation of the user's intentions according to the dialogue context. The overall module is integrated in the overall architecture as a special behavior and it is composed of three layers: (1) the lower layer contains the classifiers of the single modalities; (2) the middle layer (fusion engine) performs a late fusion and provides a context-free integration of the multiple inputs [Ros13]; (3) the upper layer (dialogue manager [Luc13]) performs the coordination of the dialogue and accomplishes the semantic interpretation of the observations according to the context and the inner knowledge. The main feature of such structure is that the results of each layer are N-best lists of possible interpretations, which are fed to the next layer to solve in cascade the ambiguities at the upper layers of the system. The dialogue manager is the upper layer of the interaction block that provides the interaction policy depending on the iteraction model. The dialogue models are graph-based specifications (see Figure 7). Multiple dialogue flows can be combined in order to build a dialogue model in a modular and extensible manner [Luc13]. The resulting dialogue model is represented by a POMDP which can cast the inherent ambiguity due to noise on the 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 strategy generated for that interaction model.

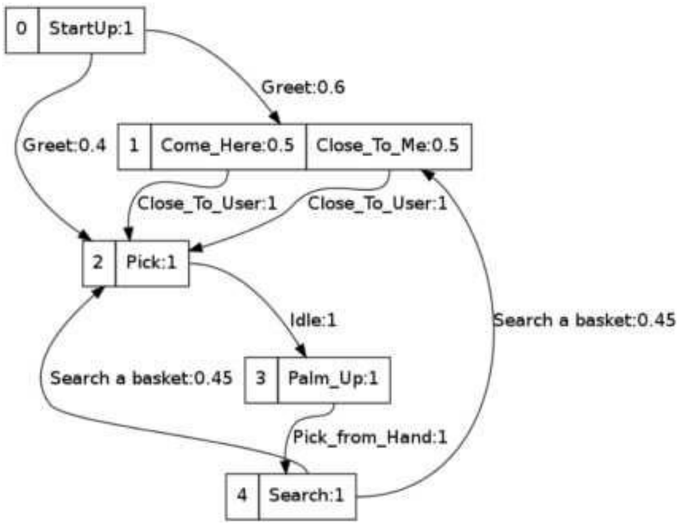

Figure 7: Excerpt extracted from dialogue models: node 1 has the two possible interpretations "Come Here" and "Close to Me". In both these cases, the robot action is to go close to the human from where, in the node 2, the robot expects that the user asks to pick something.

More specifically, the dialogue policy generated as a solution of the POMDP provides a machine action a_m 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; moreover, its execution can be modulated by the associated top-down attentional regulation mechanisms.

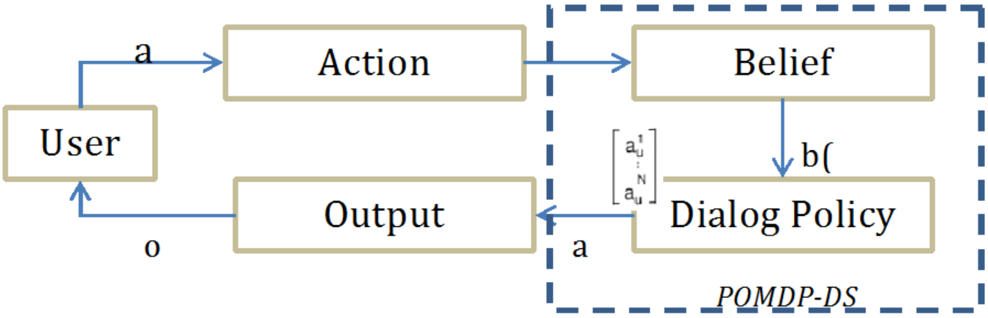

Figure 8: Dialogue Management. The user actions are interpreted, the dialogue context is then estimated and, depending on the human, environmental, and dialogue state, an action/communication is produced (dialogue policy) in output.

Interaction Models. In order to represent the knowledge about interaction, the system is provided with a set of interaction models, which describe how the dialogue can develop. Each model is named "dialogue flow" (see Figure 9), a graph where the nodes are states in which the conversation could be. From each node some user acts can be observed with the associated probability. Each observation is characterized by one machine action, which is expected by user and produces a transition to other state, which defines the edges of the graph. Edges between nodes, belonging to different graphs, are also allowed.

This model characterizes several features of the system:

- The real intention of the user is hidden;
- The results of classification are not error-free;
- The interpretation of a gesture could be multiple;
- An interpretation could lead to different system action, according to dialogue flow.

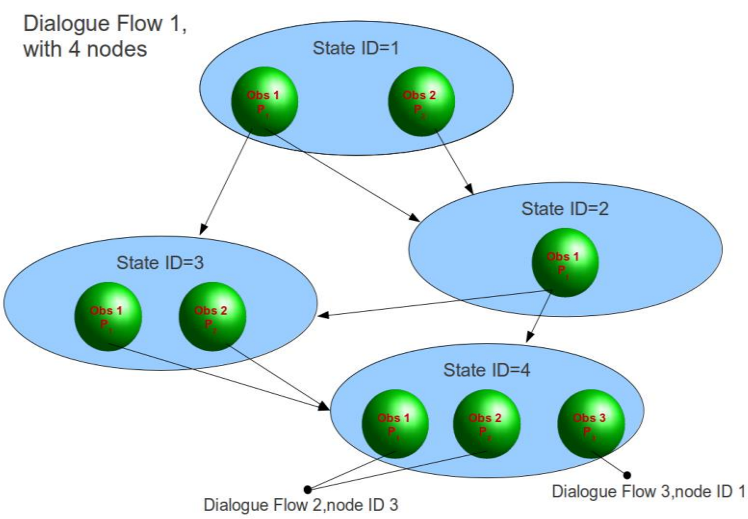

Figure 9: An example of a Dialogue Flow with four nodes.

The estimation of user act, the update dialogue state, and the choice of which machine action to perform can be formulated as a Partially Observable Markov Decision Process [You10,Luc13]. Formal details are provided below.

The MDM-POMDP is a tuple $(S, A_m, T, R, O, Z, \gamma, b_0)$, where:

- $S = S_{flow} \times S_{node} \times A_u$ is the set of states, or better a state is a triple $s =< s_{flow}, s_{node}, a_u >$, where s_{flow} is the ID of a dialogue flow, s_{node} is the ID of node in the dialogue flow and a_u is the last user action;
- A_m is the set of machine actions expected by the user and described in dialogue flows, in addition to control actions useful to get confirmation;
- T is the transition function $P(s' \,| s, a_m)$;
- R is the reward function $R(s, a_m) \in \mathbb{R}$;
- O is the set of observations that are N-best lists of hypothesis about user action $\bar{o}' = [< a_u^1, p_1 > \cdots < a_u^n, p_n >]$. The observation are provided by lower classification layer;
- Z is the observation function equal to $P(\bar{o}' | s, a_m)$;
- γ is the discount factor;
- b_0 is the initial belief state.

Since a probability distribution is maintained over the states, the belief state b represents this distribution, and b_s is the probability of being in state s. Following [Young10], we introduce some independence assumptions induced by the factorization of the state:

$$T(s', s, a_m) = P(s' \backslash s, a_m) \approx P(a_u' \backslash s_{flow}', s_{node}') \cdot P(s_{flow}', s_{node}' \backslash s_{flow}, s_{node}, a_u, a_m)$$

and

$$Z(s, \bar{o}) = P(\bar{o}' \backslash s', a_m) \approx P(\bar{o}' \backslash a_u')$$

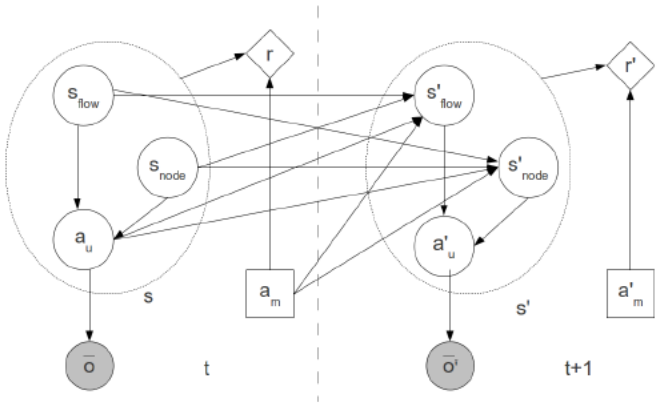

Figure 10: The POMDP model as a Bayesian Network. White circles are hidden variables, while the gray cricles are the observations.

In this model, when the system is in a belief state b and performs an action $a_m \in A_m$, it receives a reward and an observation, hence the update of the current belief is performed. The update belief function is defined as follows:

$$b(s^{t+1}_{flow}, s^{t+1}_{node}, a^{t+1}_{u}) = \underbrace{k}_{normalisation\,constant} \cdot \overbrace{P(\overline{o}^{t+1} \backslash a^{t+1}_{u})}^{observation\,model} \cdot \underbrace{P(a^{t+1}_{u} \backslash s^{t+1}_{flow}, s^{t+1}_{node})}_{user\,action\,model} \cdot$$
$$\sum_{(s^{t}_{flow}, s^{t}_{node}, a^{t}_{u}) \in b} \overbrace{P(s^{t+1}_{flow}, s^{t+1}_{node} \backslash s^{t}_{flow}, s^{t}_{node}, a^{t}_{u}, a^{t}_{m})}^{dialogue\,flow\,model} \cdot b(s^{t}_{flow}, s^{t}_{node}, a^{t}_{u}).$$

Given the model, we have to define a policy that selects a machine action $\pi(b) \rightarrow a_m$. A greedy policy is effective only when the system can infer the user intentions in a robust manner, but usually the explicit request of a confirmation helps to avoid misunderstanding. Since the POMDP could be very large, approximated solutions are more suitable than exact ones. Rather than optimizing the policy over all states, we follow a different approach based on the Point Based Value Iteration (PBVI), which consists in optimizing only over a finite set of belief points. However, in addition to PBVI that keeps bounded the complexity of planning, another improvement is to perform the optimization in a summary space rather than in the original space. The key idea here is that after the execution of an action, the user waits for a response, which could be either the execution of the most probable action or a control action. Indeed, we can generate the policy in a space where the only executable actions are reduced to the following: either do something linked to the top hypothesis or ask for an explanation. Following this approach, we defined a Summary MDP whose summary space that consists of three elements $S' = \{first, second, comp\}$, while the summary machine action set is $A'_m = \{Do\_act, Request\}$. The mapping from the original space to the summary one is done by assigning to $b(first)$ and to $b(second)$ respectively the probability of the two top hypothesis in original space, and $comp = 1$ if the machine actions from these hypothesis are similar, else $comp = 0$. The decoding of $a'_m \in A_m$ into $a\_m \in A\_m$ is done by performing the machine action linked to the top hypothesis if $a'_m = Do\_act$, otherwise, a request to the user is executed.

The whole optimization process is the following:

- Policy Optimization
  1. Select n summary belief points $B'_1 \dots B'_n$ B'$_1$_1 … B'$_n$
  2. Optimize over the selected points
  3. Return a policy that assigns an action to each selected point $a'^1_m \dots a'^n_m$

Then the action selection routine is as follows:

- Action selection routine, used at runtime
  1. Map the current belief into a summary belief point B'
  2. Find the index i of closest point among $B'_1 \dots B'_n$
  3. Decode and execute a'^i_m

We tested the dialogue system considering both simulated and real interactions. In particular, our aim was to assess robustness with respect to errors, the feasibility of the policy optimization process, and, finally, to estimate wether the user experience is natural. The results are presented in [Luc13].

4 Case Studies

In this section, we discuss the behavior of the previously presented HRI system considering simple case studies.

a) **Mobile Robot Scenario:** the robot shares the workspace with several users which can interact with the system in order to achieve some tasks such as picking or placing objects like bottles, or carrying paper sheets to other users. A representation of the environment is illustrated in Figure 10 (down). The robotic platform setting is the following: Pioneer 3 DX mobile robot provided with ultrasonic sensors and a gripper; RGB-D camera for users and gesture recognition and a High Definition camera for object detection; a microphone and a speech synthesizer. The users can interact with the robot by speaking or using gestures or body movements, while therobot has a list of user dialogue models describing possible patterns of commands or movements. Each gesture is linked to one or more meanings, hence ambiguities are possible. The meaning can be disambiguated according to the dialogue context. On the other hand, some user's acts are not explicit commands, therefore the system should interpret the human's intention supporting the human activity with a proactive behavior. We assume that the robot can pick up an object at a time, but it can carry a maximum of two objects. This scenario offers a wide variety of situations for testing the ability of the proposed framework in managing multiple requests and in solving the associated conflicts (pick different objects). Our aim is to assess the system behavior when the residual ambiguity in the dialogue policy and the associated decision conflicts should be resolved by the top-down and bottom-up attentional influences. For instance, if the human request is interpreted a generic take (without an explicit ref erence to the object to be taken) and a green and a red object are perceived by the robot during the navigation, the system should decide which object to take.

EXECUTION TIME			
Task Sequence	Time (min)	Task Sequence	Time (min)
TakeRed - TakeGreen		TakeRed - TakeGreen - TakeYellow	
Red Green Give	4.5	Red Green Give Yellow Give	9.19
Green Give Red Give	7.11	Green Give Red Give Yellow Give	8.19
Green Give Red Give	8.04	Red Green Give Yellow Give	7.21
Green Give Red Give	7.14	Yellow Give Green Give Red Give	9.08
Red Green Give	3.53	Yellow Green Give Red Give	7.28
Green Red Give	3.50	Red Green Give Yellow Give	6.41
Red Green Give	4.19	Red Green Give Yellow Give	7.02
Green Give Red Give	6.04	Red Green Give Yellow Give	7.05
Red Green Give	4.48	Yellow Give Green Give Red Give	9.43
Green Red Give	6.26	Red Green Give Yellow Give	8.48
AVG	STD	AVG	STD
5.48	1.64	7.93	1,07

Table I: Execution time of a generic take in different contexts.

In this case, the perceived affordances associated with the two detected objects can directly elicit two instances of a take task to be allocated as schemata in the WM (e.g., *take(objRed)* , *take(objGreen)*). These schemata are then decomposed in two subschemata (see Figure 6) representing the chunks associated with the task: reach the object, pick it up, and give it to the human. This way, these schemata/subschemata enter into the attentional focus of the robot along with the perceived objects and can be suitably top-down and bottom-up aroused. For instance, in Figure 10 (left) we can observe that, once a first red object is perceived by the robot, the *take(objRed)* task is bottom-up aroused by the activations of *reachColor(red)* (from 1 to 30) which is a concrete instance of *routeto(objRed)* in the WM. After 15 seconds the robot detects also a green object, therefore a decision conflict arises. However, in this case the robot heads towards the red object as an effect of the *reachColor(objRed)* dominant activations (bottom-up influence) with respect to *reachColor(objGreen)* since the red object is closer. Once the red object has been reached, the subtask can be accomplished by *pickUp(red)*. At this point the frequency of *take(objRed)* is relaxed (peak in the plot) because a new subtask *give(objRed)* is activated. This behavior receives the emphasis (top-down influence) from the partial achivement of the parent task *take(objRed)* that boosts *give(objRed)* towards the goal accomplishment. This effect is shown in Figure 10 (left) where, from time 30 to 55 we can see the restriction of the period (frequency enhancement) illustrating the modulation of the *give(objRed)* due to the bottom-up influence (dotted red line) and how it is reduced (frequency amplification) taking into account also the effect of the top-down emphasis (solid red line). In Table I, we illustrate 10 runs where the robot interprets and executes an unreferenced take given the dialogue model and the belief state. These data have been collected in two simulated scenarios: in the first one we have two objects to be taken (red and green in Table I, left); in the second one we have three objects (red, green, and yellow in Table I, right). For each scenario we report the executed sequence of tasks and the time needed to accomplish the goal (minutes).

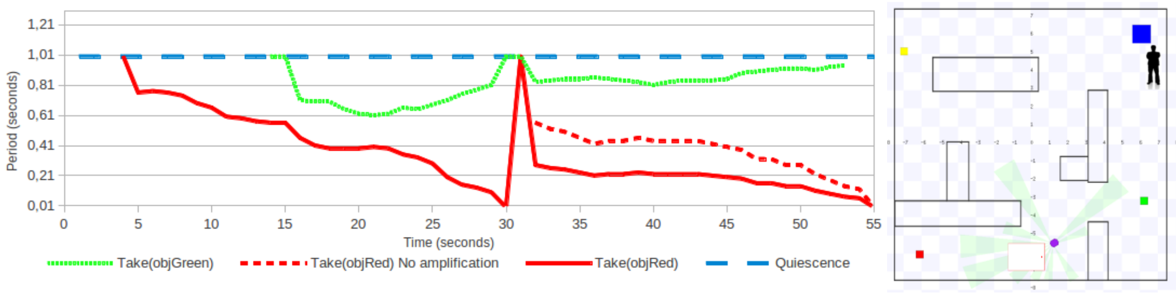

Figure 11: (left) Period modulation during a conflicting situation in a lab scenario: take(objRed) is amplified, hence the frequency and the outputs are enhanced driving the robot towards the red target; (right) lab scenario.

The executed sequence illustrates the subtasks sequence chosen by system (here Red , Green , etc. is an abbreviation for, respectively, reach and pick the object red, reach and pick the object green etc., while Give represents the delivery action that ends the task). A maximum of 10 minutes was provided for each run. To test the system in the ability of conflict resolution and flexible execution of multiple tasks, we allowed the robot to collect two items before the delivery. For instance, in the sequence “Red Green Give ” *take(objRed)* and *take(ObjGreen)* are interleaved, hence the robot first picks the red object, then it picks the green one, and finally it delivers the two objects to the human; in other cases, the task are sequentialized (e.g., in “Red Give Green Give ”). Notice that the parallel or sequential execution of the task is left to the system decisions and depends on the attentional mechanisms and environmental context. The results in Table I show that the system is always able to accomplish the goal, and when there is the opportunity it can interleave the execution of the tasks (6 times and 7 times in the first and the second scenario respectively), and, as expected, when this happens the temporal performance is enhanced. To better assess the temporal performance, in Table II we also report the average and the std of the values collected after the execution of 10 take tasks where the referenced object is provided (e.g., *take(ObjGreen)*). By comparing the average values at the end of Table I with the values in Table II we can observe that the mean time needed to accomplish the ambiguous requests is comparable with the mean time needed to achieve the tasks where the reference is explicitly defined. This seems to suggest that the conflict resolution mechanism is effective in managing the impasses.

Note that the proposed attentional mechanisms are here mainly elicited by the detection of gestures, speech, objects, colors however, additional, and more sophisticated mechanism (e.g., gaze detection and joint attention) can be easily incorporated in this framework.

EXECUTION TIME (min)					
Take-Red		Take-Green		Take-Yellow	
avg	std	avg	std	avg	std
3.99	0.28	1.48	0.36	2.04	0.27

Table II: Execution time of the specific take

b) **Coffee Scenario:** to show the system at work in a more interactive setting we introduce a second case study. We consider a coffee making scenario (inspired by the one in [Coop00]) where 4 objects are available on a table: a cup, coffee carafe, a sugar bowl, and a spoon. The human is to prepare the coffee by collecting these objects in a suitable order: first the cup, then the sugar and the carafe (any order is permitted), finally the spoon. This task is represented as a suitable schema in the LTM, which is activated in the WM (see Figure 12) by alive once a suitable stimulus is detected (e.g. human command mentioning the coffee). Here, the human initiative to help the human in accomplishing the task. Also in this case, the dialogue policy provides an abstract robot response to the human action (e.g. take something, ask for explanations, etc.) that should be completed and regulated by the attentional system. For example, if the human command is a generic *take* and all the objects are available on the table, the system has a decisional problem (each object is associated with a *take* affordance) which can be solved by a top-down attentional regulation: the cup is the first object to be taken in the coffee task, therefore the robot action *take**(**cup**)*** is emphasized and selected. Instead, if the human has already taken the cup, then the system is to decide among the other 3 objects. In this case, the top-down regulation emphasize both *take**(**carafe**)*** and *take**(**sugar**)***, while the bottom-up regulation enhances the action associated with the closer object. This decisional process is continuously influenced by the human commands and actions. For instance, in Figure 14, left, while the robot takes the cup, the human gets the coffee carafe. Once the cup is taken by the robot, the top-down attentional influence emphasizes the *take**(**sugar**)*** robot action (Figure 14, center) which is the only action enabled since the *take**(**sugar**)*** was already executed by the human. Finally, the human can conclude the task with the *take**(**spoon**)*** action (Figure 14, right). Figure 15 illustrates the period modulation profile associated with this successful sequence of robot (solid line) and human actions (dotted line). Since the robot actions are only simulated and the objects are not actually moved, the associated periods remain invariant. The green and red peaks arise when the system realizes that a subgoal is already accomplished by the human. Notice that, analogously the robot actions, also the human actions are

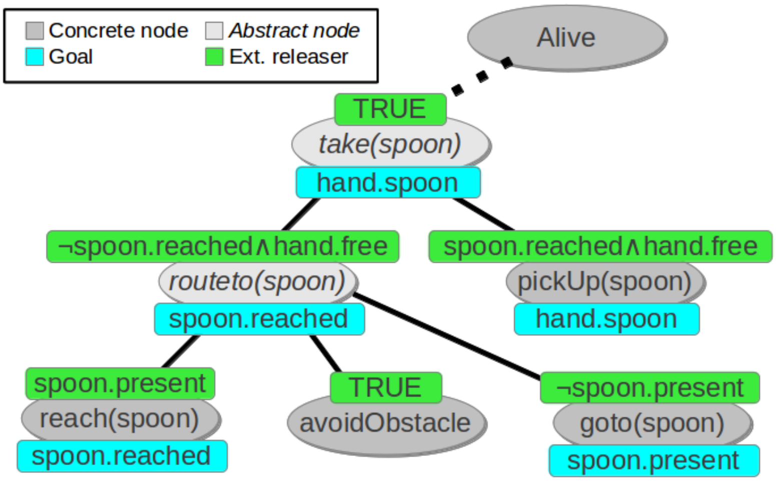

Figure 12: Representation of a task in the WM.

monitored by concrete attentional behaviors whose frequencies are regulated by a function of the tracked features (e.g. in Figure 15 the dotted period profile is associated with the velocity of the tracked hand). This simple domain shows how the proposed attentional framework permits a flexible and an adaptive execution of interactive tasks.

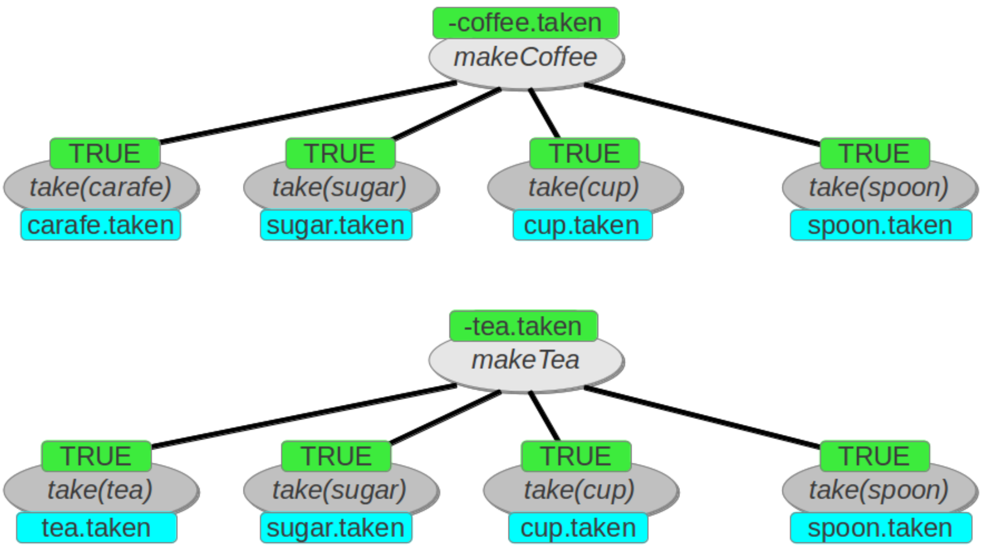

Figure 13: (up) make coffee schema, (down) make tea schema.

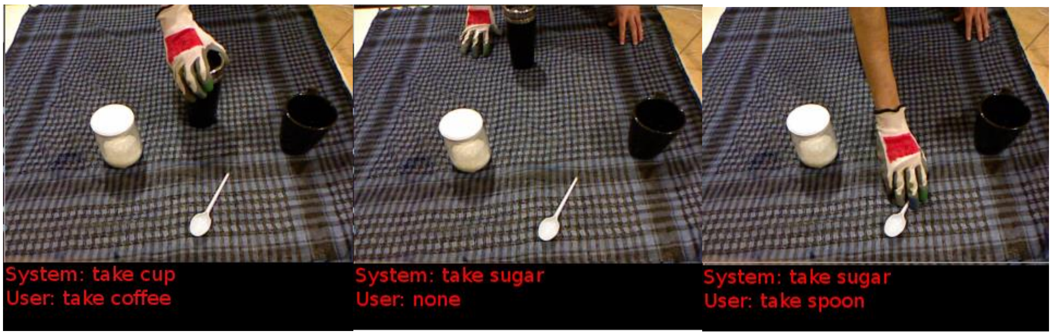

Figure 14: (left) the system selects the cup, then the user takes the carafe, (center) the system selects the sugar (cup and carafe already taken), (right) the system takes the sugar, then the user takes the spoon.

c) **Tea and Coffee:** we extended the previous scenario introducing also a tea making task. The associated schema is analogous to the coffee making one in Figure 10 with the tea used in the place of the coffee. This way, the robotic system is to interpret the intention of the human (coffee or tea?) depending on the human operations. A proactive interactive attitude of the robotic system can easily yield to an interpretation error hence the human can interact tocorrect; this allows us to test how the system can deal with this additional ambiguity and misinterpretations.

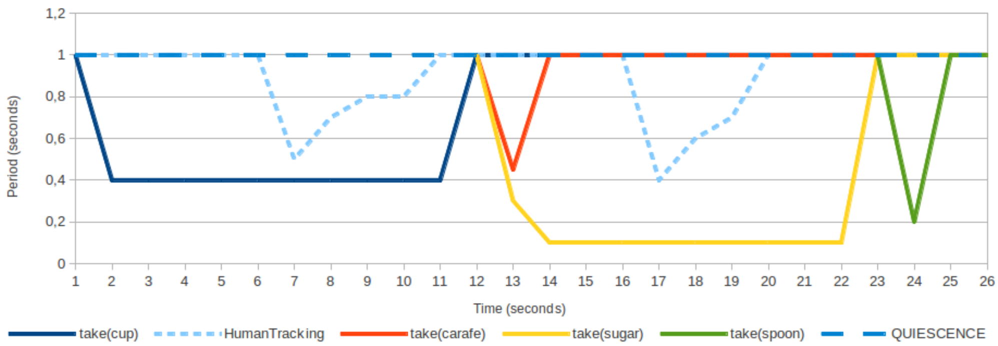

Figure 15: Period modulation profile in the coffee scenario. Both human (dotted line period) and robot (solid line period) behaviors are tracked by the attentional system.

	Success	Correction	Failure
avg:	56.6%	26.7%	16.7%
std:	0.67	0.42	0.52
Hum. Act. :	1.48	2.25	3.6

Table III: Successful executions, corrections, failures and mean number of human actions (out of the 4 actions needed to accomplish the task) in the coffee/tea domain.

The scenario is depicted in Figure 16 where the following objects are disposed on the table: a cup, a spoon, sugar box, a tea box, and a coffee box.

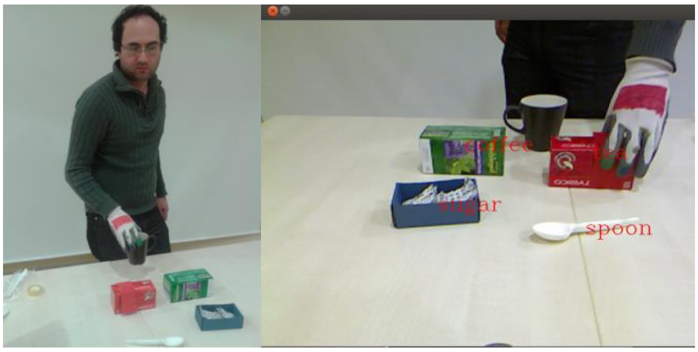

Figure 16: (left) The human takes a cup (right) the system detects the human taking the tea. In the second case, the task is disambiguated by the first action of the human.

In this context, as preliminary test, we asked 10 subjects (graduate students: 6 males, 4 femals) to execute 3 times one of the two tasks (tea or coffe) in cooperation with the system. For each execution we changed the disposition of the objects. We assumed both the tasks (*makeCoffee* and *makeTea*) already represented in the WM. The results illustrated in Table III show that, despite the inherent ambiguity of the domain, the task can be accomplished in 83:3% of the cases, considering both directly successfull interactions (robot initiative correct with respect to the human intention) or interaction where human explicit corrections are needed (correction). Moreover, the robot initiative seems effective in reducing the human actions needed to execute the task.

5 Planning and Execution

In this section, we illustrate how the attentional system described above is integrated within the Human-Robot Interaction framework developed at LAAS, CNRS [Fiore14]. The overall Human-Robot Interaction framework is depicted in Figure 17, while in Figure 18 we detail the Attentional System along with its interfaces with the rest of the architecture.

Human Aware Task Planning and Execution.

The Human-Robot Interaction system developed at LAAS [Fiore14] integrates a Human Aware Task Planner (HATP) a supervisory system, and a set of specialized motion planners. This system is composed of several layers (see Figure 17) which are detailed below.

- **SPARK.** The Spatial Reasoning and Knowledge component, responsible for geometric information gathering [Millez14]. SPARK embeds a number of decisional activities linked to abstraction (symbolic fact production) and geometric and temporal reasoning. SPARK maintains the geometric positions and configurations of agents, objects, and furniture coming from perception and previous or a priori knowledge. SPARK elaborates also perspective taking features [Brea06], enablig the system to reason on other agents' beliefs and capacities.
- **Knowledge Base.** The facts produced by SPARK are stored in a central symbolic knowledge base. This base mantains a different model for each agent, hence divergent beliefs can be mantained. For example, two agents can keep the information of two different positions referring to the same object.

- **HATP**. The Human-Aware Task Planner [Lalle14] is based on a Hierarchical Task Networks (HTN) refinement process where an iterative task de-composition is exploited to reach the atomic actions. HATP is able to produce plans for the robot as well as for the other participants (humans or robots). By setting a different range of parameters, the plans can be tuned to adapt the robot behavior to the desired level of cooperation. HATP is able to take into account the different beliefs of each agents when producing a plan, including actions that support and elicit joint attention [Scas99].
- **Collaborative Planners**. This set of planners are based on POMDP models which are used to estimate the user intentions in joint actions (e.g. handovers). The POMDP policy selects high level actions (like continue to plan or wait for the user), which are then adapted by the supervisory system to the current situation. More specifically, the supervisory system refines and executes each action in the HATP generated plan, using the collaborative planners to adapt its actions to those of the other agents during a joint action execution.
- **Supervisory System**. The supervisory system is to orchestrate the overall planning and execution cycle. Indeed, it manages plan generation and flexible execution of the plan while interacting and monitoring the human. This module integrates the attentional system illustrated above, where human monitoring and action execution functionalities are incapsulated into suitable behaviors. This integration will be detailed in the next subsection.
- **A set of Human aware motion, placement and manipulation planners**. These trajectory planners define the robot motions taking into account the environment and the agents constraints [Mainp11].

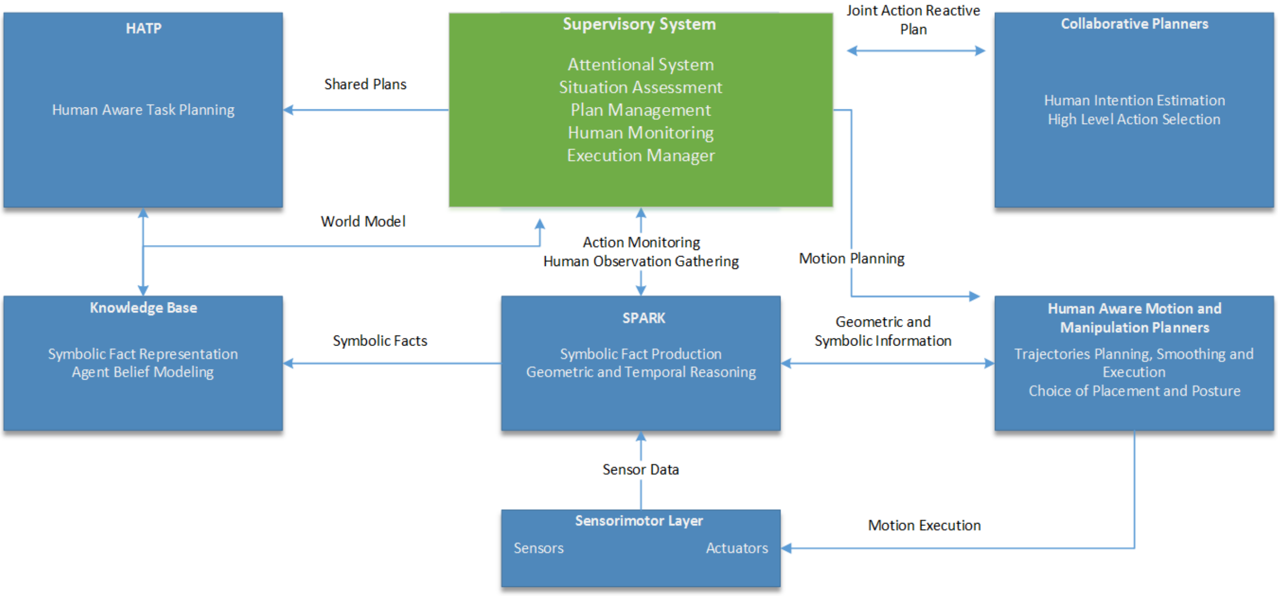

Figure 17: Human-Robot Interaction Architecture.
The detailed link between the Attentional System and the Supervisory System is shown in Figure 18.

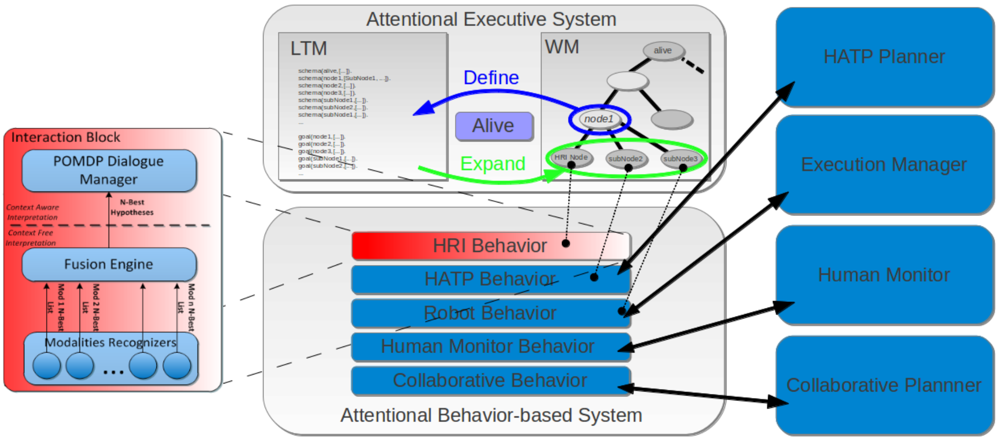

Figure 18: Integrated Attentional System.
The blocks on the right of the figure represent other procedures in the Supervisory System.

Attentional System and HATP Planning

In this subsection, we detail the integration of the Attentional System within the LAAS architecture presented above (see Figure 18). In order to integrate the HATP planner, the hierarchical tasks specified in the planning domain are to be also represented in the LTM as abstract or concrete schemata (corresponding, respectively, to methods and actions). In this context, the schemata features are explicitily represented in the planning domain, that is, releasers, effects, and goals are associated with, respectively, preconditions and effects of actions and methods. The planning activity can be invoked by the attentional system by exploiting a suitable interface behavior that interacts with the HATP system. It provides the WM state variable values to the HATP dynamic environment (initial state) and the planning requests (goals). As a result of the planning activity, the HATP behavior receives a plan of actions which can be performed either by the robot or by the human. Indeed, the plan is translated by the HATP interface behavior into a list of schemata and suitably allocated in WM in order to be expanded and executed by the system. If some actions are explicitly assigned to the human, these actions are replaced by specific task-specific human monitoring behaviors. During the execution of collaborative activities, specialized collaboration planners can be invoked to generate interactive behaviors, which are on-line regulated by the attentional system. Plan failures can be managed by the attentional system, which interacts with the dialogue manager in order to decide whether to replan, ask the help of the user, or to switch to a recovery task.

Case Study

The attentional framework has been implemented and tested in a case study inspired by the AIRBUS domain [Ala14]. The environment is set as follows: the user works in front of a table where various objects (screwdriver, screws, cloth, plate and glue) are located. The overall environment is also represented in v-rep that allows us to simulate the KUKA lightweight 7DOF manipulator (see Figure 19). The user has to interact with the robot in order to execute a set of tasks which can be planned by HATP. The human and the robot should interact in order to accomplish the task. During the execution, the robotic system is to adapt its actions according with the plan and the interpretation of the human behavior (gesture, voice, face, etc.) [Ien14].

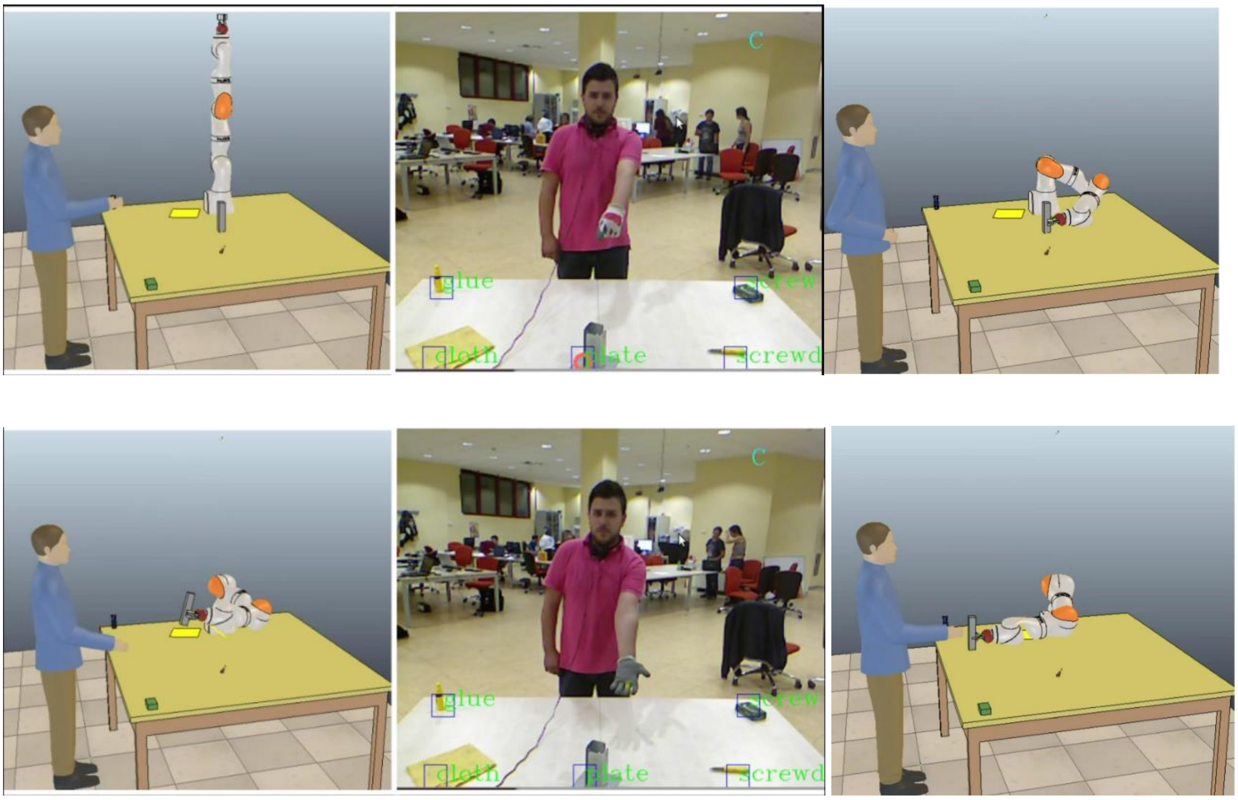

Figure 19: Simulated environment: the human interacts with a simulated KUKA arm in order to accomplish cooperative tasks.

6 Conclusions

We proposed a human-robot interaction system where top-down and bottom-up attentional modulations are used to contextualize the dialogue flow and solve ambiguous communications. We described the proposed HRI architecture, introducing simple interactive scenarios used to illustrate the system behavior in different situations. In particular, we have shown how both bottom-up and top-down attentional modulations allow the system to solve decisional impasses driving the system towards the task accomplishment. We have also discussed the integration within the overall HRI architecture focusing on planning and execution issues. We tested the approach with simple structured and interactive tasks, considering the system capability to adapt the execution with respect to the human behavior and the environmental opportunities.

In the near future we would like to further exploit the capacities of the supervision system, to have a more flexible interaction process. In particular, this system should be capable of interacting with humans in different modalities, by adapting to the user's preferences and switching from one modality to another in a seamless way, depending on the context. We highlight the following three modalities:

- **Human-based**. In this modality, the robot is not aware of the global plan or goal. The human decides when to ask the robot to perform specific tasks. The robot then acts by performing the requested task. Decisional autonomy here is limited to the requested task. On the other hand, the robot is to continuously monitor the human actions updating the state of the environment accordingly.
- **Robot-based**. In this case, a shared goal between the human and the robot is explicitly defined and shared. The robot generates its plan in order to achieve the goal, taking into account the world status, the abilities of the two agents and the preferences of the human. Once the plan is generated, it verbalizes it and achieves it by doing its 'part of the job' and monitoring the human activity. This modality corresponds to a fully agreed plan that can be built on-line or predefined and known to both agents.
- **Adaptive and Proactive Robot**. Also in this case, a joint goal between the human and the robot is defined as a common knowledge. The robot monitors the human behavior, and, whenever possible, tries to achieve an action or a set of actions that advances the plan towards the goal. This includes also the possibility for the robot to proactively facilitate the action of the human whenever this is possible.

7 References

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

[Arb87] M. A. Arbib, E. J. Conklin, and J. A. C. Hill, From schema theory to language. Oxford University Press, 1987.

[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, p. 624, 2001.

[Breaz02] C. Breazeal, Designing Sociable Robots. MIT Press, 2002.

[Brea06] C. Breazeal, M. Berlin, A. G. Brooks, J. Gray, and A. L. Thomaz, "Using perspective taking to learn from ambiguous demonstrations," Robotics and Autonomous Systems, vol. 54, no. 5, pp. 385–393, 2006.

[Bro14] An Attentional Approach to Human-Robot Interactive Manipulation. X. Broquere, A. Finzi, J. Mainprice, S. Rossi, D. Sidobre and M. Staffa. 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, 2010, pp. 283–292.

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

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

[Cac14c] Attentional Top-down Regulations in a Situated Human-Robot Dialogue, R. Caccavale, A. Finzi, L. Lucignano, S.Rossi, M. Staffa, in HRI-2014 workshop on "Attention Models in Robotics: Visual Systems for Better HRI".

[Coh04] J. Cohen, G. Aston-Jones, and M. Gilzenrat, “A systems-level perspective on attention and cognitive control,” Cognitive neuroscience of attention, p. 71, 2004.

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

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

[Ien14] Continuous Gesture Recognition of Flexible Human-Robot Interaction. S. Iengo, S. Rossi, M. Staffa and A. Finzi. In Prof. of ICRA 2014.

[Lai87] J. E. Laird, A. Newell, and P. S. Rosenbloom, “Soar: An architecture for general intelligence,” Artificial intelligence, vol. 33, no. 1, pp. 1–64, 1987.

[Lalle14] Raphael Lallement, Lavindra de Silva, and Rachid 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, 2013, pp. 197–204.

[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 IEEE International Conference on Robotics and Automation, 2011.

[Mill56] G. A. Miller, “The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information,” The Psychological Review, vol. 63, pp. 81–97, 1956.

[Millez14] Grégoire Milliez, Mathieu Warnier, Aurélie Clodic, and Rachid Alami. A framework for endowing interactive robot with reasoning capabilities about perspective taking and belief management. In Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication, 2014.

[Moz98] M. C. Mozer and M. Sitton, "Computational modeling of spatial attention," Attention, vol. 9, pp. 341–393, 1998.

[Noc14] The Role of Intrinsic Motivations in Attention Allocation and Shifting. D. Di Nocera, A. Finzi, S. Rossi and M. Staffa. Frontiers in Computer Science Journal. Front. Psychol. 5(273).

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

[Scas99] B. Scassellati, "Imitation and mechanisms of joint attention: a developmental structure for building social skills on a humanoid robot," in Computation for metaphors, analogy, and agents, 1999, pp. 176–195.

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

[Rose07] D. A. Rosenbaum, R. G. Cohen, S. A. Jax, D. J. Weiss, and R. van der Wel, "The problem of serial order in behavior: Lashley's legacy," vol. 26, no. 4, pp. 525–554, 2007.

[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, 2013.

[You10] The hidden information state model: A practical framework for pomdp-based spoken dialogue management. S. Young, M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. Computer Speech and Language, 24(2):150 – 174, 2010.