Flexible Task Execution and Attentional Regulations in Human–Robot Interaction

Riccardo Caccavale and Alberto Finzi

***Abstract*—A robotic system that interacts with humans is expected to flexibly execute structured cooperative tasks while reacting to unexpected events and behaviors. In this paper, we face these issues presenting a framework that integrates cognitive control, executive attention, and hierarchical plan execution. In the proposed approach, the execution of structured tasks is guided by top-down (task-oriented) and bottom-up (stimuli-driven) attentional processes that affect behavior selection and activation, while resolving conflicts and decisional impasses. Specifically, attention is here deployed to stimulate the activations of multiple hierarchical behaviors orienting them toward the execution of finalized and interactive activities. On the other hand, this framework allows a human to indirectly and smoothly influence the robotic task execution exploiting attention manipulation. We provide an overview of the overall system architecture discussing the framework at work in different case studies. In particular, we show that multiple concurrent tasks can be effectively orchestrated and interleaved in a flexible manner; moreover, in a human–robot interaction setting, we test and assess the effectiveness of attention manipulation for interactive plan guidance.**

***Index Terms*—Cognitive control, cognitive robotics, human–robot interaction, social robotics.**

I. Introduction

A ROBOTIC system should be capable of dynamically executing complex tasks, while reacting to human interventions and environmental changes. This issue is particularly relevant in cognitive and social robotics, where complex and structured activities should be flexibly executed by robots in cooperation with interacting humans (e.g., interaction with a robot-co-worker in a factory or health-care robots operating in domestic environments).

Several frameworks have been proposed in the robotics literature to conciliate natural human–robot interaction and the execution of structured tasks, the dominant approach relies on the planning and execution paradigm and deploys planning/replanning to adapt task execution to the behaviors of the agents involved in the interaction [1]–[5]. This paradigm is effective in mixed-initiative planning and execution, on the other hand, the associated continuous planning and execution process usually impairs the naturalness and effectiveness of the interaction with the humans and the environment.

In this paper, we tackle these issues from a different perspective exploiting the concept of cognitive control [6]–[9] introduced in cognitive psychology and neuroscience to denote the executive mechanisms/functions needed to support flexible, adaptive responses, and complex goal-directed cognitive processes and behaviors. In this context, complex activities are usually represented as hierarchically organized behavioral schemata [8], [10]–[12] and several accounts have been proposed for modeling action selection, sequencing, and execution. Among the most influential models for activity control, the one proposed by Norman and Shallice [7] assumes that two main processes are involved in activity orchestration and execution: 1) *contention scheduling* and 2) *supervisory attentional system*. Contention scheduling is a low-level reactive process in charge of the execution of habitual and routinized activities avoiding conflicts with competing behaviors; on the other hand, the supervisory attentional system is a higher-level mechanism that coordinates and monitors the behavioral schemata in order to manage novel and complex tasks. In this framework, attentional regulations play a central role in action selection. Indeed, each behavior is associated with an activation value that can be regulated and biased by the supervisory attentional system in order to facilitate the execution of desired behaviors and inhibit the undesired ones. Computational accounts for this model can be found in [8] and [13]. On the other hand, in the robotics literature the deployment of these mechanisms is pretty rare. While approaches to dynamic control of hierarchical activities [14], [15], and cognitive control frameworks [16] have been proposed, the deployment of executive attention for flexible plan execution remains a less explored topic.

Following a cognitive control paradigm, in this paper we propose an integrated framework that combines hierarchical planning, flexible execution of multiple structured tasks, and human–robot interaction (HRI) in cooperative activities.

In the proposed framework, top-down (task-oriented) and bottom-up (stimuli-driven) attentional processes are exploited to smoothly regulate the activations of hierarchical robotic behaviors by enhancing the ones related to the task and coherent with the environmental state, while reducing the ones in conflict. In this context, hierarchical plans are not

Manuscript received February 7, 2016; revised July 15, 2016; accepted September 1, 2016. Date of publication September 30, 2016; date of current version March 9, 2017. The research leading to these results was supported by the EU FP7 Project SHERPA (g.a. 600958) and by the EU FP7 ERC project RoDyMan (g.a. 320992). The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

The authors are with the Department of Electrical Engineering and Information Technology, University of Naples Federico II, 80125 Naples, Italy (e-mail: riccardo.caccavale@unina.it; alberto.finzi@unina.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCDS.2016.2614690

directly executed, but used to influence the attentional system and facilitate the execution of the associated behaviors. This smooth attentional guidance—along with the associated conflict resolution mechanisms—enables flexible execution of multiple concurrent tasks. Moreover, this setting is particularly suited for human–robot cooperative tasks, indeed, the human involved in the interaction can deploy attention manipulation [17] (e.g., gestures, utterance, object manipulation, etc.) to indirectly bias the robot behavior toward the execution of the required activities. Notice that attention-based interaction is also very relevant for social communication [18], but in this paper we mainly focus on cooperative task accomplishment.

We discuss the system at work in different simulated case studies introduced to assess flexible and interactive execution of multiple parallel tasks. The collected results show that the proposed cognitive control framework can effectively and flexibly manage multiple plan execution. In addition, in the case of cooperative activities, we show that attentional manipulation enables the human to interact with the robot in a natural and effective manner.

The rest of this paper is organized as follows. The next section provides an overview of the system architecture describing the overall control cycle along with the involved components and the associated attentional mechanisms. Section III introduces our approach to flexible plan execution based on the integration of hierarchical task planning and attentional regulation. Section IV describes the case studies and reports on the experimental work. Section V discusses the proposed system with respect to related works. Finally, Section VI draws the final conclusions and future works.

This paper presents an extended and further detailed version of the system introduced in [19] at the 2015 IEEE International Conference on Systems, Man, and Cybernetics.

II. System Architecture

The framework proposed in this paper integrates hierarchical planning, human–robot interaction, and attentional execution. It modulates both reactive and task-oriented processes in order to integrate human interventions and plan execution. This is mainly achieved by deploying an attentional system that affects and orients sensory processing and behaviors activations according to the human actions, the active tasks, and the environmental stimuli.

The overall human–robot interaction architecture (see Fig. 1) integrates a multimodal interaction module (*HRI module*), a hierarchical task planner (*Planner*), and an executive system. The latter can be subdivided in two components:

1) the *Attentional Executive System*, that manages behavior allocation and provides top-down regulations;
2) an *Attentional Behavior-based System*, that collects the allocated sensorimotor processes, which are affected by bottom-up influences.

The *HRI module* allows a human to naturally interact with the robot exploiting different modalities (e.g., speech, gestures, etc.). These multiple input channels are to be interpreted and

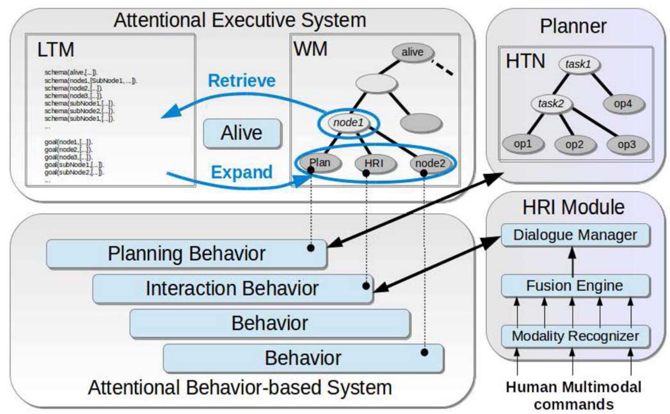

Fig. 1. System architecture. The *HRI module* permits multimodal interaction between the human and the robotic system. The *Planner* generates plans—represented as HTNs—to be executed and monitored by the *Attentional Executive System* providing top-down and bottom-up regulations. The executive control cycle (blue arrows and ovals) is managed by the process *alive* that updates the WM exploiting the behavior schemata stored in the LTM. The *Attentional Behavior-based System* collects the concrete sensorimotor processes allocated in WM (dotted lines). Among these, the *Planning* and the *Interaction* behaviors manage the interaction (black arrows) with the *Planner* and the *HRI module*, respectively.

fused (*Fusion Engine*) by the *HRI module* in order to recognize the human activities and intentions. As a final stage of the multimodal interaction process, a *Dialogue Manager* generates an interaction behavior, that can be instantiated and continuously adjusted by the attentional executive system with respect to the environmental and the operative context (see [20]–[22] for details). On the other hand, a task planner (*Planner*) can generate plans of actions, represented as hierarchical task networks (HTNs) [23], where both human and robot may be involved.

The integrated planning and execution system will be better detailed in Section III, while in the rest of this section we mainly focus on the executive system and the associated attentional mechanisms.

A. Cognitive Control Cycle

The attentional executive system manages a cognitive control cycle, which involves a set of attentional behaviors, a working memory (WM) and a long term memory (LTM). Following a central executive perspective [24], the WM maintains and manages short-term data for online processing and execution, supported by an LTM which provides a vast, long-term storage of information [25]. Specifically, in our framework, the attentional behaviors represent the sensorimotor processes that are currently involved in the execution of the robotic tasks. These are collected in the WM, which maintains the executive state of the system, including all the allocated hierarchical tasks along with the associated behaviors. Finally, the LTM is a long-term repository that collects the behavioral repertoire available to the system, including the definitions of all the abstract methods and the concrete actions the system can retrieve and instantiate for task execution. In this context, the cognitive control cycle is managed by a special process *alive* that periodically updates the WM (blue arrows

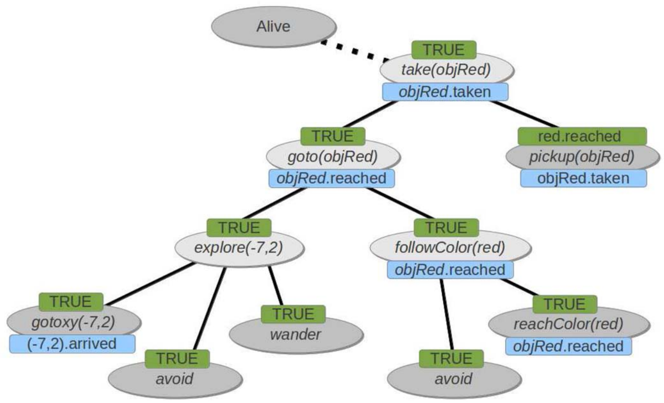

Fig. 2. Hierarchical tasks in the WM: releasers, behaviors, and post-conditions are, respectively, green, gray, and blue.

and ovals in Fig. 1) by allocating and deallocating behaviors, exploiting the associated denotations in the LTM. This process will be better explained below while describing the structure of the WM and the associated attentional mechanisms.

B. Working Memory

In line with a typical approach in artificial intelligence and cognitive psychology [7], [8], [10], [26], we assume a hierarchical organization for tasks and activities. In our framework, this hierarchy is represented in the WM as tree structure (see Fig. 2) that collects the tasks allocated for the execution. Each node of this tree represents a system behavior. In particular, we distinguish among *concrete* or *abstract* behaviors: a concrete behavior represents an allocated sensorimotor process [e.g., *reachColor*(*red*) in Fig. 2], instead an abstract behavior identifies a complex activity that needs to be hierarchically decomposed into different subactivities in order to be executed [e.g., take(objRed) in Fig. 2].

1) Behavior Schemata: Both abstract and concrete behaviors are represented in the LTM as behavior schemata. In particular, the LTM collects possible methods and actions encoded by a set of predicates of the form **schema**(m, l, e), where m is the name of the behavior, l is a list of sub-behaviors associated with enabling conditions (releasers), i.e., $l = \langle (m_1, r_1), \dots, (m_n, r_n) \rangle$, while e represents a post-condition used to check the accomplishment of the behavior. For instance, the abstract behaviors take and goto are specified in the LTM as follows:

$$\textbf{schema}\big(\text{take(Obj)}\\ \quad \langle(\text{goto(Obj)}, \text{true}), (\text{pickup(Obj)}, \text{Obj.reached})\rangle\\ \quad \text{Obj.taken}\big)$$

$$\textbf{schema}\big(\text{goto(Obj)}\\ \quad \langle(\text{explore}(X, Y), \text{true}), (\text{followColor(Obj)}, \text{true})\rangle\\ \quad \text{Obj.reached}\big).$$

In the first schema, the behavior of taking an object take(Obj) is composed of two sub-behaviors: 1) reach the object (goto(Obj)) and 2) pick it up (pickup(Obj)). The sub-behavior goto(Obj) is directly enabled once allocated, indeed its releaser is set to true; instead, pickup(Obj) is enabled when the robot is close to the target object (its releaser is enabled when Obj.reached is satisfied). After the successful execution of take(Obj), the robot holds the object (Obj.taken is the post-condition). Similarly, the behavior *goto* is composed of two sub-behaviors: the robot first explores an area (explore) and then follows the color associated with the object (followColor). Once allocated in the WM, these two behaviors are directly enabled (their releasers are both set to true). Finally, the *goto* behavior is successfully executed once the robot reaches the object position (Obj.reached is the post-condition). These definitions in the LTM are retrieved and exploited by the process *alive* to allocate the behaviors in the WM for their actual execution. For instance, in Fig. 2, the abstract behaviors take(objRed) and goto(objRed) are allocated in the WM and expanded using the two schemata introduced above. In the following, we better detail the WM structure and its dynamics.

2) WM Structure: The WM is represented by an annotated tree, whose nodes represent allocated processes/behaviors, while the edges represent parental relations among subprocesses/sub-behaviors. Each node, either concrete or abstract, represents an instance of an LTM behavior schema, hence it is associated with a name, a set of sub-behaviors, a post-condition, and a releaser. Both releaseres and post-conditions are represented as boolean expressions to be satisfied. For instance, in Fig. 2, pickUp(objRed) is enabled only if the releaser red.reached is satisfied, while its post-condition is objRed.taken. If the releaser of an allocated node is satisfied, then all the associated sub-behaviors can also be allocated in the WM. Conversely, if a behavior is accomplished or dismissed, it is removed from the WM along with its hierarchical decomposition. Notice that in this framework, an allocated concrete behavior is active when its releaser is enabled along with the releasers of all its ancestors. In this perspective, we distinguish between an *internal* and an *external* releaser: the first one represents the task-independent enabling condition for a concrete behavior, while the second one is the task-dependent enabling condition which is hierarchically inherited through the WM. For instance, in Fig. 2, red.reached is an external releaser (task-based) for pickUp(objRed), because it holds in the context of the task take(objRed), instead, the detection of a graspable object is an internal releaser (stimuli-driven) for the pickUp behavior; pickUp is activated when both these conditions are enabled.

3) WM Update: As already mentioned above, the *alive* behavior is periodically activated to update the tree, in so regulating the overall cognitive control cycle. This process is described in Algorithm 1. We assume that a set of concrete behaviors are always allocated and active in WM in order to manage the basic activities of the system (e.g., *interaction*, *avoidance*, *planning*, etc.). During the execution, each behavior is allowed to update the WM by inserting new nodes, which are then hierarchically expanded and instantiated by the *alive* behavior according to their specification in the LTM (lines 4–11 in Algorithm 1). In this setting, the human requests are managed by the *interaction* behavior that can suitably update the WM. For instance, if the human asks for a red

Algorithm 1 Alive Process Continuously Checks the WM by Allocating and Deallocating Nodes According to the Definitions in the LTM

```
procedure ALIVECYCLE
    while true do
        check the WM
        if there exists a node s to expand then
            search for the associated schema in the LTM
            if it exists then
                add the sub-behaviors of s to the WM
            else
                remove s from WM
            end if
        end if
    end while
end procedure
```

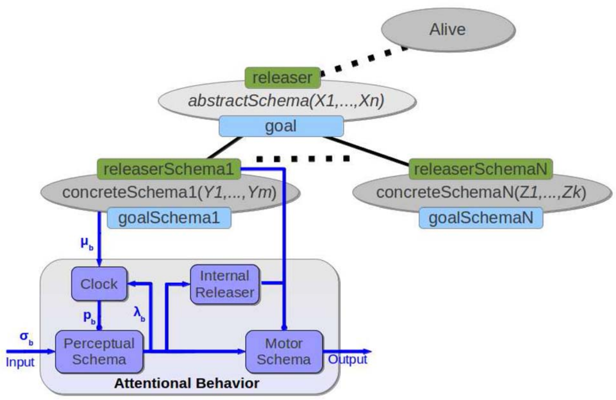

Fig. 3. Tree structure and detail of a concrete behavior. The clock period p_b is top-down (μ_b) and bottom-up (λ_b) regulated, while the internal and the external releasers depend on internal and external properties, respectively. The external releaser for a concrete behavior is the conjunction of the releasers of the nodes along the path to *alive*.

object, the *interaction* behavior allocates a take(objRed) node, that will be expanded by the *alive* process (see Fig. 2) selecting the sub-behaviors involved in the hierarchy, as specified in the take(Obj) schema presented above.

C. Behaviors and Attentional Regulations

Following a schema theory representation [27], [28], each concrete behavior is a sensorimotor process composed of a perceptual schema, which elaborates behavior-specific stimuli, a motor schema, that produces an associated pattern of motor actions, a releasing mechanism, and an adaptive clock which regulates the behavior activations (see Fig. 3). The releaser enables/disables the motor schema. As already stated above, it is expressed as the conjunction of two boolean expressions to be satisfied: 1) the internal releaser and 2) the external releaser. The clock is an adaptive mechanism that regulates the sensors sampling rate and the behavior activations. The clock period p_b is regulated by a behavior-specific *monitoring function* $f(\sigma_b, \varepsilon)$ according to the behavioral stimuli σ_b and the overall executive state of the system ε representing the current state of the WM (collecting the inner state of all the allocated behaviors along with their hierarchical relations). This way, each behavior becomes active after the latency period p_b, when the next clock period p'_b is adaptively redefined by the monitoring function. In our framework, this regulation represents a simple attentional mechanism that tunes the temporal resolution at which a behavior is monitored and controlled. For instance, given an obstacle avoidance behavior *avoid*, the behavioral stimuli σ_{avd} is the distance of the closer obstacle, the internal releaser triggers when σ_{avd} is less than a specific distance, while, in the absence of other top-down influences, the monitoring function regulates the clock frequency proportionally to the obstacle distance (bottom-up regulation). This way, the closer the obstacle, the higher is the frequency of the distance check, the more reactive is the avoidance behavior. On the other hand, an object to be grasped cannot be considered as an obstacle to be avoided, hence a top-down influence is needed to relax or inhibit a concomitant avoidance behavior (top-down regulation). The combined effect of top-down and bottom-up regulations will be illustrated below. Additional details on bottom-up clock regulation mechanisms can be found in [28] and [29], while related methods for parameters setting are discussed in [30].

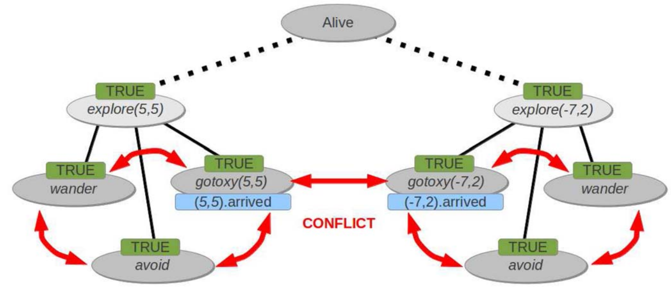

Fig. 4. Example of conflicting tasks: the two explore tasks are in conflicts and associated with mutually exclusive [e.g., *gotoxy*$(5, 5)$ versus *gotoxy*$(-7, 2)$] and not mutually exclusive [e.g., *avoid* versus *gotoxy*$(5, 5)$] concrete behaviors.

D. Top-Down Regulations and Conflict Resolution

The control cycle presented in Algorithm 1 illustrates how the hierarchical structure in the WM permits to recruit, allocate, activate, and regulate multiple behaviors for task execution. In this setting, arbitration mechanisms are needed to avoid conflicts or erratic activities (see Fig. 4). Indeed, in the proposed system, multiple tasks can be executed at the same time, several methods for the same tasks may compete in the WM, and many behaviors can try to access and modify a single resource generating *crosstalk interferences* [31]. These conflicts and impasses can be either prohibited by construction or solved by means of suitable evaluation functions [9].

In our framework, we follow the latter, more flexible, approach exploiting attentional regulations. For this purpose, we introduce an additional mechanism that combines bottom-up and top-down attentional regulations.

1) Bottom-Up Regulations: For each concrete behavior b, the bottom-up regulation is provided by a monitoring function $g(\sigma_b, \varepsilon_b) = \lambda_b$ that defines the behavioral clock period

λ_b in the absence of any top-down stimulation—hence only due to the behavior specific stimuli σ_b and the inner state of that behavior ε_b (internal state variables of the perceptive and motor schemata, previous clock regulation, internal releaser status, etc.). For instance, if we consider the avoidance behavior introduced above, given the stimulus σ_{avd} representing the minimal distance from an obstacle, we assume that the clock period λ_{avd}, varies in the interval $[\lambda^{\min}, \lambda^{\max}]$ and it is bottom-up regulated by the following saturating (and increasing) linear function:

$$g(\sigma_{\text{avd}}, \varepsilon_{\text{avd}}) = \begin{cases} \lambda^{\min} & \text{if } \ \sigma_{\text{avd}} \leq r^{\min} \\ \lambda^{\max} & \text{if } \ \sigma_{\text{avd}} \geq r^{\max} \\ \alpha \cdot \sigma_{\text{avd}} + \beta & \text{otherwise} \end{cases} \tag{1}$$

characterized by two parameters $r^{\min}$ and $r^{\max}$, with $\alpha = (\lambda^{\max} - \lambda^{\min})/(r^{\max} - r^{\min})$ and $\beta = \lambda^{\min} - \alpha \cdot r^{\min}$ used to describe the linear increase of g for σ_{avd} in $[r^{\min}, r^{\max}]$. Notice that in this simple example σ_{avd} directly affects λ_{avd}, however, in more complex settings this regulation can also depend on the behavior internal state (e.g., the new clock regulation can depend on the previous one, see [29]).

2) Top-Down Regulation: The top-down regulation is provided by a value μ_b, called *magnitude*, that summarizes the overall top-down (and lateral) influence of the WM status on the attentional state of a behavior. Thus, the attentional state of each concrete behavior depends on the couple (λ_b, μ_b), while the overall activation frequency of a specific behavior is defined by a value, called *emphasis*, that combines bottom-up and top-down influences as $e_b = \mu_b/\lambda_b$. Here, the bottom-up frequency, influenced by the behavioral stimuli, is directly modulated by the top-down magnitude that can enhance or reduce it. This way, the emphasis allows us to combine accessibility and facilitation: bottom-up stimuli emphasize actions that are more accessible to the robot (e.g., object affordances), while top-down stimulations exploit the task structures to facilitate the activations of task-related and goal-oriented actions. The overall sensorimotor cycle of a generic concrete behavior is summarized in Algorithm 2: once allocated in the WM, the behavioral perceptive schema is enabled (lines 2 and 3), while the motor schema is active only if its releasers are satisfied (lines 4–6); at the end of the cycle the new clock period is defined (lines 7 and 8). Notice that this clock period is the inverse of the *emphasis*, hence the overall *monitoring function* $f(\sigma, \varepsilon)$ introduced above can be characterized as follows:

$$f(\sigma_b, \varepsilon) = \frac{g(\sigma_b, \varepsilon_b)}{\mu_b} = p_b. \tag{2}$$

In this setting, the absence of a top-down influence is represented by $\mu_b = 1$, when the clock period is directly regulated by g_b, hence it equals λ_b. Otherwise, the value of μ_b depends on the overall state ε of the WM. Indeed, whenever a magnitude change happens for a node in the WM, this update is inherited by all its descendants, influencing the attentional state of all the associated concrete behaviors. Moreover, in order to induce a smooth drive toward task completion, we assume that when an activity is accomplished, the magnitude of the parent node is increased by a constant value k_b which is then propagated toward its active successors. As a side effect, this mechanism induces a lateral influence among the behaviors involved in the same task.

Algorithm 2 Each Concrete Behavior Is Endowed With a Perceptual and a Motor Schema, a Releasing Mechanism, and an Adaptive Clock That Regulates the Activation Frequencies. Here, rel_b^i and rel_b^e Are the Internal and External Releasers, μ_b Is the Top-Down Influence (*Magnitude*), λ_b the Bottom-Up Regulation of the Period, While $e_b = \mu_b/\lambda_b$ (*Emphasis*) Sets the Clock Frequency Integrating Top-Down and Bottom-Up Influences

1: **procedure** BEHAVIORCYCLE(b)
2:     **while** s is allocated in the WM **do**
3:         activate the perceptual schema of b
4:         **if** rel_b^i and rel_b^e are satisfied **then**
5:             activate the motor schema of b
6:         **end if**
7:         udate λ_b and μ_b
8:         wait for $1/e_b$
9:     **end while**
10:    end b
11: **end procedure**

3) Conflict Resolution: The combined effect of top-down and bottom-up regulations is then used to select the active behaviors and combine their activations. Contentions among alternative behaviors competing for mutually exclusive state variables can be solved using the *emphasis*: following a *winner-takes-all* approach, the most emphasized behavior is selected with the exclusive access. On the other hand, when a set of concurrent behaviors affects nonmutually exclusive variables, all of them are allowed to access and modify these variables, while the emphasis can be exploited to weight and combine these multiple contributions. Specifically, we assume that the overall contribution on a nonmutually exclusive variable is $v = \Sigma_b(e_b \times v_b)/\Sigma_b(e_b)$, where v_b and e_b are the values and the emphasis for each updating behavior. Notice that the *emphasis* has two combined affects here: 1) acceleration of the clock and 2) modulation of the combined outputs. This allows us to merge the multiple contributions in a smooth way. Indeed, not only the emphasized behaviors provide more frequent updates, but also their contributions are amplified. Since the amplification is associated with a drive toward the goal accomplishment, goal-oriented behaviors become dominant, in so overcoming decisional impasses.

III. Planning and Execution

In this section, we show how the executive system described above can be integrated with a hierarchical planning framework and deployed for plan execution. Specifically, we propose an approach to flexible and interactive plan execution, where a generated plan is not directly executed, but exploited as an attentional top-down guidance for the robotic behavior.

A. Hierarchical Task Planning

In our framework, we assume that the hierarchical tasks can be online generated by a suitable planning process. In particular, we refer to a SHOP-like [32] HTN framework. An HTN planning problem is defined by a goal g, an initial state s, and a planning domain $D = (A, M)$ that collects a set of primitive operators A and a set of methods M. Each method is represented by a triple $(m, p, b) \in M$, where m is the name of the method, p is a precondition that specifies when the method is applicable, while b describes a sequence of operators or methods. The primitive operators $a \in A$ are denoted by a STRIPS-like representation: each operator is characterized by a set of preconditions and effects. The HTN planning process selects applicable methods from M and applies them to abstract tasks in a depth-first manner until only primitive tasks are left. For additional details we refer the reader to [32].

B. Executive System and Planning

A generated HTN plan should be suitably executed in the WM by instantiating and allocating behavior schemata. For this purpose, the methods and the operators represented in the planning domain are to be associated with abstract and concrete behavior schemata in the LTM representing the corresponding executive processes. Specifically, primitive operators $a \in A$ can be associated to either a concrete or an abstract behavior schemata, while each method $(m, p, b) \in M$ is represented by an abstract $\textbf{schema}(m, l, e)$, with the same name m and a list of sub-behaviors $l = \langle (m_1, q_1), \ldots, (m_n, q_n) \rangle$ representing the submethods $\langle m_1, \ldots, m_n \rangle$ in b. Here, the q_i releasers in l extend the p_i preconditions of the m_i methods/operators with additional conditions to be satisfied during the execution, while, the post-condition e permits to monitor whether a behavior has been accomplishment. Indeed, the behavior schemata in LTM enrich the description of the associated methods and operators in the planning domain providing additional information needed at the execution time (i.e., releasers and post-conditions). For example, the take(Obj) schema introduced above, can be associated with a planning method $(\text{take(Obj)}, \text{true}, b)$, with $b = \langle \text{goto(Obj)}, \text{pickup(Obj)} \rangle$ as sub-behaviors and true as a precondition. This way, the hierarchical representation of tasks and actions is shared by the executive system and the planning system, therefore, the executive system can either directly apply task decomposition to update the WM, as described in Section II, or generate a hierarchical plan by invoking an external planner with a goal. For instance, since a take(Obj) behavior schema is also associated with a take(Obj) method for the HTN planner, this activity can be either online executed (Algorithm 1) or offline planned (HTN planning) and then executed.

The interaction between the planner and the WM is managed by a concrete behavior, called *planning* (see Fig. 1), that can activate planning/replanning processes providing the HTN planner with the initial state (obtained from the variables in the WM) and the planning requests (goals/tasks to be achieved). As a result of the planning activity, it receives the generated plan and then allocates it in the WM in order to be suitably expanded and executed by the cognitive control cycle.

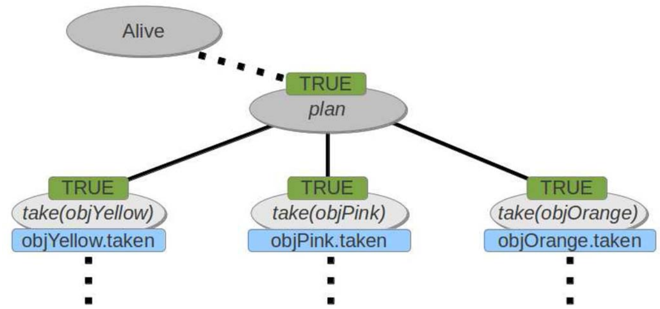

Fig. 5. The generated plan is allocated in the WM and the associated abstract behaviors are hierarchically expanded by the cognitive control cycle.

C. Plan Execution and Attentional Regulation

In our framework, a generated plan is treated as an extension of the tree in the WM and uniformly managed by the cognitive control cycle described above. This way, its execution can be flexibly regulated by the top-down and bottom-up attentional mechanisms influencing the execution of the associated abstract/concrete behaviors. More specifically, the generated plan is associated with a new node in the WM, this way the activities mentioned in the plan can be expanded by the associated behaviors, as specified by the schemata in the LTM (see Fig. 5). Moreover, since the system is endowed with *contention scheduling* mechanisms, multiple plans can be concurrently allocated and executed, while the actual execution of the associated concrete behaviors only depends on the releasing mechanisms and the attentional regulations provided by the hierarchical structure of the WM. In this setting, the cognitive control cycle is always active and ready to react to external events, including human requests and interventions. For instance, an interactive human can either directly induce task allocation with an explicit command (e.g., *take the red object*) managed by the *interactive* behavior, or implicitly influence plan execution by modifying the robot attentional state. Hence, in the presence of several objects, a human can point toward one of them in order to stimulate the activations of the associated tasks already allocated in the WM.

IV. Case Studies

In this section, we consider the system at work in a simulated scenario where a mobile robot can execute pick-carry-and-place tasks in the presence of multiple objects. We first test the system in the presence of multiple parallel plans in order to assess the system performance in flexible plan execution. Then, we consider two interactive scenarios where a human is to influence the execution of multiple tasks through attention manipulation.

A. Simulated Environment

We assume a 15×15 m simulated environment that contains several colored objects that can be taken and carried by a mobile robot (Fig. 6). As a robotic platform, we consider a

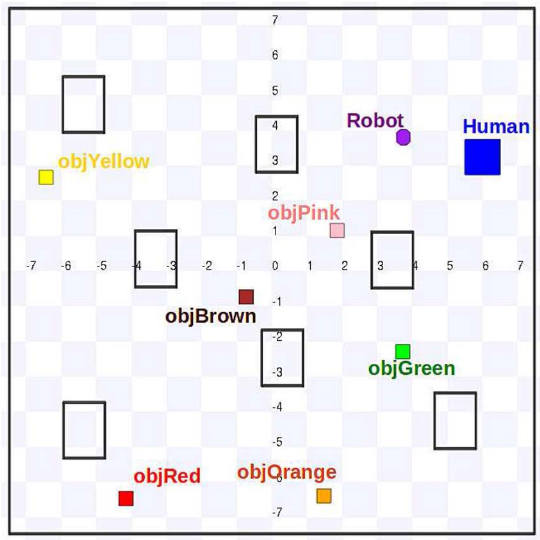

Fig. 6. Simulated environment: we have colored objects (small cubes), obstacles (rectangles), the human position (large flat square), and the mobile robot (purple).

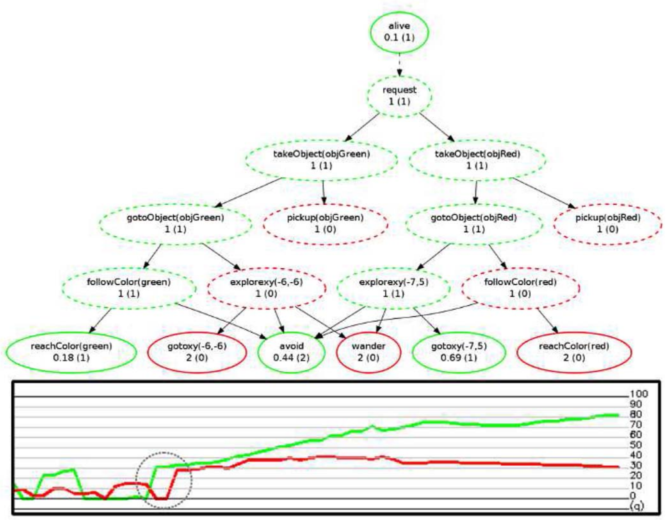

Fig. 7. Conflicting tasks in the WM (top) and emphasis plots (bottom) for the two conflicting behaviors *reachColor* and *gotoxy* associated with the abstract behaviors *takeObject*(*green*) and *takeObject*(*red*), respectively. After the conflict (dotted circle in the plot) the robot heads toward the object green. For each behavior node, n, (m) represents the clock period and the magnitude, respectively. Green (red) solid (dotted) ellipsis are for, respectively, active (inactive), concrete (abstract) behaviors.

simulated Pioneer 3 DX endowed with ultrasonic sensors, a gripper, and a camera for object detection. The robot can move with a maximum speed of 0.4 m/s and can pick up an object at a time, but it can hold several objects at the same time. This scenario enables us to assess the system behavior in the presence of an interactive human along with multiple structured tasks and the associated decisional conflicts. For instance, in Fig. 7 we can observe a competition of two tasks allocated in the WM as a consequence of an ambiguous human command (i.e., *take an object*). Indeed, two objects are perceived by the robot (i.e., red and green), thus two instances of the *takeObject* subtask are allocated and compete in the WM. In this case, the robot heads toward the green object as an effect of the *reachColor*(*green*) dominant activations ($p_b = 0.18$). Notice that, once the object is reached, the next active subtask will be pickUp(objGreen), with a top-down influence ($\mu_b = 2$) due to the *goToObject*(*objGreen*) subgoal accomplishment.

B. Behaviors Set-Up

The overall system is driven by the selection, activation, and modulation of concrete behaviors. In particular, we consider the following set: *wander*, *avoid*, *gotoxy*, *reachColor*, *place*, *pick*, *sonarStream*, *engineStream*, *blobStream*, and *pointTo*. When no task is provided in the WM, we assume that the robot behavior only depends on the wandering (*wander*) and obstacle avoidance (*avoid*) processes which regulate the robot linear and angular velocities (v, ω) interacting with the *engineStream* process. Here, *avoid* receives the obstacle distance as an input signal σ_{avd} from *sonarStream*. The *gotoxy* process drives the robot toward a final position (x, y) receiving the actual robot position provided by *engineStream* as the input σ_{gt}. The *pointTo* behavior is implemented in a similar manner stimulating the robot to move toward the pointed direction. On the other hand *place* and *pick* are in charge of the robot manipulation receiving as input the distance of the target place σ_{tp} and the target object σ_{to}, respectively. We assume here that object manipulation is reliable. The *reachColor* behavior searches for an object of a specific color, once the color is detected, it moves the robot in that direction. The associated input signal σ_{rc} is provided by *blobStream*.

The executive system manages selection, allocation, and orchestration of these concrete behaviors through the WM structure and the associated top-down and bottom-up attentional regulations. For the sake of simplicity we assume the following setting. The initial top-down influence is set to $\mu_b = 1$, the subtask magnitude increment is $k_b = 1$, while λ_b ranges from 0.01 to 1 s and, excluding *wander*, for all the other behaviors it is either increased or decreased proportionally to the input signal σ_b within an associated range $[r_b^{\min}, r_b^{\max}]$. On the other hand, we assume a linear decrease of frequency, when the stimulus is stable or removed (habituation and decay). More specifically, the period of *wander* is constant and set to 1 (i.e., maximum period and minimal influence); instead, for all the other behaviors, λ_b is regulated by g proportionally to σ_b [analogously to the *avoid* behavior, as specified in (1)], with the exception of *engineStream* whose period λ decreases with the robot linear velocity [similarly to (1), with $\alpha < 1$]. As for the $[r_b^{\min}, r_b^{\max}]$ ranges, these are set (in meters) as follows: [0.5, 1] for *avoid*, *pickup*, and *place*; [0, 3] for *sonarStream*, *engineStream*; [0, 10] for the *blobStream*; and [1, 10] for *goto*, *pointo*, and *reachColor*.

C. Case Study 1: Flexible Execution of Multiple Plans

We consider now a scenario where multiple plans should be concurrently and flexibly executed. The aim here is to assess how the proposed framework is capable of flexibly interleaving the execution of multiple tasks depending on the opportunities or the human requests.

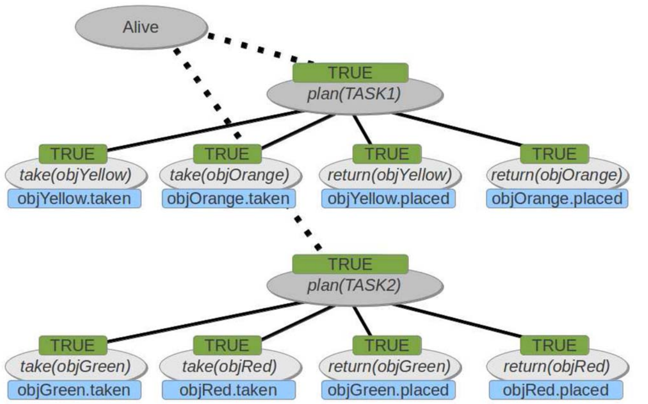

Fig. 8. Conflicting plans in the WM. The planned activities are sequenced from left to right.

TABLE I
TRUE/FALSE POSITIVE/NEGATIVE OVER TEN RUNS

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

Specifically, we assume that the two concurrent plans depicted in Fig. 8 are already loaded in the WM and ready for the execution. Each plan represents a sequence of four actions, but the execution order is not directly enforced by the plan structure. Indeed, here the releasers are deliberately enabled (i.e., set to true), in order to allow maximum flexibility in the action execution, which only depends on the attentional modulations. In this scenario, we aim at comparing the system performance with respect to the best choices, i.e., the decisions that guides the robots along the minimal total path. In this setting, path minimization can be achieved if the actions in the two plans are suitably interleaved trading-off action execution (bottom-up) and the drive toward task completion (top-down).

More specifically, during the tests we assess the executive system decisions considering the following items:

1) *True-Positives:* competing active actions, executed by the system, which respect the plan sequence and minimize the path cost (i.e., best choice among the active actions);
2) *True-Negatives:* competing active actions, not executed by the system, and not expected to be executed (i.e., actions correctly defeated);
3) *False-Positives:* executed actions, not expected to be executed (i.e., suboptimal choices);
4) *False-Negatives:* competing best actions, which are not executed (i.e., missed best actions).

We tested the system with ten trials. Each test consists of a concurrent execution of the two plans. At the beginning of each test the objects are randomly positioned in the environment. Each test ends with the two tasks accomplished. The collected results are reported in Tables I and II.

Table I reports the system performance: the two concurrent plans are executed with an effective selection of the correct actions (7.5 with respect to eight best actions) with few suboptimal choices (0.5) and rare missed opportunities (0.1) despite the influence of distractive alternatives (2.6 true-negatives).

TABLE II
MEASURES OF PERFORMANCE

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

Table II summarizes these results in terms of accuracy (true positive and negatives with respect to the possible selections), precision (true positives with respect to the selected actions), and recall (true positives with respect to the best selections). In the table we can observe that the system makes over 94% of correct choices (*accuracy*) in conflicting situations, with an executed action (*precision*) which is usually the expected one (93%). In this context, the few wrong choices are usually due to violations of the planned sequence (83.3%). This usually happens when the magnitude provided by the subtask achievement is not sufficient to contrast the bottom-up influence due to the proximity of an object associated with a future action (anticipation and utilization errors [13]). On the other hand, we can observe a high probability of making a good choice (expected actions) with respect to the available best choices (*recall* over 98%).

D. Case Study 2: Plan Execution and Human–Robot Interaction

In a second case study, we consider the presence of a human that can influence the execution of multiple tasks by manipulating the attentional state of the robot. In this context, the working hypothesis is that attentional manipulation can simplify human–robot interaction by reducing the human interventions needed for multiple task accomplishment. In order to assess this hypothesis, we designed a new scenario that extends the previous one enabling human interventions. In this case, a human can draw the robot attention by pointing toward an area of the map. This pointing is simulated by a mouse-click and associated with the concrete behavior *pointTo*, which is allocated in the WM and then top-down/bottom-up stimulated in order to drive the robot toward the target area. This behavior is similar to a *gotoxy* provided with a top-down enhanced impulse that represents the pointing intention of the human. This simple attentional manipulation mechanism is assessed considering the following task: the human is to drive the robot toward the execution of a desired pattern of actions (e.g., take the green object, then take the red one, and return to the base) with a minimal number of interventions. In this scenario, we compare the following two execution modes:

1) *Reactive Mode:* no planned task is available in WM, instead a set of subtasks are allocated to influence the robot to pick, carry, and place any object in the scene;
2) *Mixed Mode:* multiple structured plans are also present in the WM and compete in order to be executed.

We consider two experimental settings. In the first one, the task is the following: collect two objects (green and red) and

TABLE III
INTERVENTIONS AND EXECUTION TIME (FOUR AND SIX OBJECTS)

green-red-return-yellow-orange-return							
Reactive Mode				Mixed Mode			
Commands		**Time**		**Commands**		**Time**	
avg	std	avg	std	avg	std	avg	std
5.0	0.63	6'59"	0'12"	2.8	0.75	6'58"	0'38"

green-red-brown-return-yellow-pink-orange-return							
Reactive Mode				Mixed Mode			
Commands		**Time**		**Commands**		**Time**	
avg	std	avg	std	avg	std	avg	std
5.2	0.4	8'22"	0'15"	3.4	0.49	8'45"	0'31"

deliver them to the human; pick other two objects (yellow and orange) and bring them to the human again. These two rounds of pick and deliver are represented by the two plans depicted in Fig. 8, which are allocated in the WM in the *mixed* mode only. The second setting is similar to the previous one, but in this case the robot is to collect and deliver three objects in two sequences: first green, red, brown, then yellow, pink, and orange. Also in this case, we assume that in the *mixed* mode the two plans are already available in the WM, each representing one round of pick and delivery.

We involved ten graduated students in these tests (seven males and three females, with age varying from 25 to 34) asking them to execute the task with a minimum number of interventions in the two modes. No time limit was provided for each test.

In Table III, we illustrate the collected results considering the execution time and the human interventions (mouse clicks) needed to accomplish the task in the two modes. Failures are not reported because the task was always successfully accomplished by all the testers. In these tests, the advantage of the mixed mode clearly emerges from the relevant reduction of human interventions, on the other hand, the execution time is comparable in the two cases. These initial results seem to support the hypothesis that the top-down attentional guidance can effectively drive the robot behavior, while allowing sparse interventions of the human for corrections.

E. Case Study 3: Interaction With Simulated Robot

We now try to assess the effectiveness of the system in a similar, but more realistic human–robot interaction setting. For this purpose, we introduce a set-up where a real human can interact with a simulated robot exploiting gestures. The simulated scenario reproduces the abstract setting of the previous test (see Fig. 6) in a realistic 3-D environment provided by the robotic simulator *v-rep* (see Fig. 9). In particular, the simulated robot is a kuka omnirob, equipped with four mecanum wheels, a kuka LBR 4+ manipulator, a baxter gripper, two laser scans (SICK S300), and a rgdb camera mounted on the arm. The robot can move within an environment of 15×15 m with a maximum speed of 0.4 m/s.

An RGB-D sensor and a high definition camera are deployed for human monitoring and gesture recognition, this way a human operator can influence the behavior of a simulated robot by pointing toward some directions in the 3-D simulated environment. For instance, in Fig. 9, the human is indicating the yellow object in the simulated scenario.

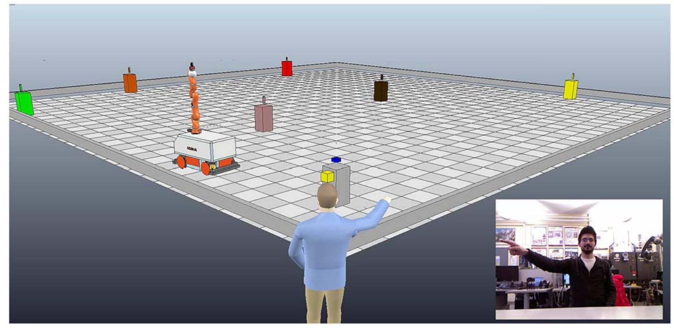

Fig. 9. Interaction with a simulated robot. The human—illustrated in the small windows at the bottom-right—points toward the simulated yellow objects influencing the attentional state and the behavior of the simulated robot.

TABLE IV
HRI QUESTIONNAIRE

Section	Question
Personal Information	Age? Gender? How familiarized are you with robotic applications?
Experience Assessment	**Controllability:** Please rate how easily could you control the robot behavior *[1 (very hardly) to 5 (very easily)]* **Interpretation:** Please rate the robot capability of interpreting your commands and intentions *[1 (very low) to 5 (very high)]* **Legibility:** Please rate how easily could you understand the robot behavior *[1 (very hardly) to 5 (very easily)]* **Docility:** Please rate how easily could you change/influence the robot behavior *[1 (very hardly) to 5 (very easily)]* **Supervision:** Please rate how much attention was needed in order to accomplish the task *[1 (very low) to 5 (very high)]*

The adopted multimodal interaction framework is the one described in [20] and [22]. In this context, analogously to the previous case study, we can consider again two competing plans of actions already in the WM, while the human can exploit real gestures for attention manipulation. Indeed, similarly to the previous case, the pointed direction is associated with a behavior *pointTo(x,y)* used to move the system focus on the scene close to the detected object, in so affecting the attentional regulations of the associated behaviors.

In this setting, we want to assess again the human performance considering both quantitative and qualitative evaluations. The quantitative evaluation allows us to compare the results obtained in the abstract setting (see Fig. 6) with respect to ones collected in a more realistic environment. The qualitative evaluation is used to assess the user perception of the interaction and is based on a questionnaire provided to the testers at the end of each test session (see Table IV).

Analogously to the previous case, we compared the performance of the users in the *reactive* and *mixed* mode described above. In both cases, the task to be accomplished was the longer one, i.e., first get the green, red, and brown objects and place them at the base; then get the yellow, pink, and orange objects and place them at the base.

TABLE V
MEASURES OF PERFORMANCE IN THE REACTIVE AND MIXED MODE

green-red-brown-return-yellow-pink-orange-return							
Reactive Mode				Mixed Mode			
Commands		**Time**		**Commands**		**Time**	
avg	std	avg	std	avg	std	avg	std
5.6	1.174	9'00"	0'34"	3.222	1.787	8'40"	0'12"

TABLE VI
MEASURES OF PERFORMANCE

Reactive Mode									
Controllability		**Interpretation**		**Legibility**		**Docility**		**Supervision**	
avg	std	avg	std	avg	std	avg	std	avg	std
3.55	1.01	3.67	0.5	3.78	1.09	4.22	0.67	4.0	1.32
Mixed Mode									
Controllability		**Interpretation**		**Legibility**		**Docility**		**Supervision**	
avg	std	avg	std	avg	std	avg	std	avg	std
4.44	0.73	4.56	0.73	4.78	0.44	4.67	0.5	1.55	0.53

TABLE VII
SIGNIFICANCE AND CORRELATIONS

Mixed vs Reactive: two tailed t-test P values									
Controllability		**Interpretation**		**Legibility**		**Docility**		**Supervision**	
$p < 0.05$		$p = 0.005$		$p = 0.015$		$p = 0.107$		$p < 10^{-4}$	
Quantitative vs Qualitative Correlation: Reactive									
Controllability		**Interpretation**		**Legibility**		**Docility**		**Supervision**	
r	p	r	p	r	p	r	p	r	p
-0.18	0.65	-0.47	0.19	-0.17	0.65	-0.17	0.67	0.38	0.31
Quantitative vs Qualitative Correlation: Mixed									
Controllability		**Interpretation**		**Legibility**		**Docility**		**Supervision**	
r	p	r	p	r	p	r	p	r	p
-0.70	0.03	-0.33	0.27	-0.66	0.05	-0.28	0.46	0.66	0.05

In these tests, we involved another group of ten graduated students (six males and four females, with ages varying from 23 to 35) asking them to accomplish the tasks with a minimum number of interventions in the two modes and no time limit for task accomplishment. The subjects were not specifically informed about the robot behavior. They were only told that the robot was equipped with certain skills/behaviors such as moving toward a position, picking or placing an object, and that their pointing gestures could influence the robot behavior by drawing its attention in that direction.

In Table V, we can observe the mean and standard deviation of the collected results in the two modalities. These results are aligned with the ones presented above, indeed also in this case the time performance is similar in the *reactive* and *mixed* case, on the other hand, the advantage in terms of command minimization seems confirmed in this more realistic environment ($p < 0.003$, two-tailed t-test comparing the samples collected in the reactive and the mixed mode). Moreover, the number of commands needed to accomplish the task in the realistic and abstract setting is comparable, despite the more complex interaction mode. Finally, a more accurate simulation of the robot operations justifies the longer durations of the tests in these experiments.

As for the qualitative assessment, at the end of each test we asked the participants to fill the questionnaire illustrated in Table IV, which is structured as follows:

1) A *personal information* section for the personal data and the technological competences of the participants. Here, we categorize subjects by their demographic attributes (age, sex), and their experience with robotics.
2) An *interaction assessment* section with questions used to rate the user experience on a five-point scale. Namely, the participants are asked to evaluate: ease of robot controllability, the docility of the interaction, the effort needed for the supervision, the system ability to interpret the human intentions, and the human ability to understand the robot behavior.

The proposed questionnaire is inspired by others introduced to assess presence/teleoperation [33] and human experience in human–robot interaction [29], [34], [35], selecting and adapting the entries with respect to the specificities of our interaction scenario.

The collected results are illustrated in Table VI. The improved performance of the *mixed* mode are confirmed by the user evaluations, indeed the testers could always perceive a more natural and readable behavior of the robot in this setting. This is particularly evident in the evaluation of the attentional effort needed to accomplish the task (*supervision*), which is significantly lower in the mixed case (see t-test two-tailed p values in the first rows of Table VII), and the capability of controlling (*controllability*) the robot that is assessed as quite lower in the reactive case. On the other hand, the capability of influencing the robot behavior (*docility*) seems comparable, with a slightly better rate for the mixed case, but not that significant. Moreover, in the *mixed* mode, the testers appreciated a more comprehensible robot behavior (*legibility*) associated with a significant improvement of the robot capability of understanding the human intention (*interpretation*). In Table VII, we can observe how these qualitative assessments are correlated with the number of commands needed to accomplish the task. As expected, both in the reactive and mixed mode *supervision* is positively correlated with the number of commands (less commands associated with lower attention needed to accomplish the task), while all the other entries are negatively correlated (less commands corresponding to higher rates). However, the significance of these correlations improves in the mixed mode, in particular, the improved performance of the participants is usually associated with a perception of an improved interaction in terms of *supervision*, *controllability*, and *legibility*.

V. DISCUSSION

The proposed approach to flexible and interactive plan execution combines concepts related to supervisory attentional system, contention scheduling, execution monitoring of hierarchical plans, and hierarchical behavior-based architectures.

The main idea in this paper is to deploy a supervisory attentional system to combine flexible task execution and human–robot interaction. In the planning and robotics literature, flexible and interactive plan execution is usually addressed by proposing integrated planning and execution frameworks, where the human initiatives and interventions are managed through replanning or plan repair mechanisms [36]–[38]. For instance, in human-aware planning [4], [39] structured hierarchical plans are generated for both human and robot involved in a cooperative activity and then generated again when the human behavior diverges form the expected one. Analogously, continuous replanning methods have been proposed to address mixed-initiative planning and execution in flexible temporal planning framework [1], [2]. Differently from these approaches, in our framework a plan is not directly executed, but loaded in the WM and then regulated by the associated attentional processes. In the context of behavior-based robotics, hierarchical behavior-based systems have been proposed [15], some of these include also simple attentional mechanisms for the dynamic control of behaviors [12], [14], [40]. For instance, in [14], attentional mechanisms are mentioned as behavior orchestration mechanisms and deployed in a case study to detect lack of progress toward the target. In contrast, in [12], attention is mainly used to orient and focus the system perception. However, in these frameworks structured tasks execution and plan-based control are not considered. A hybrid architecture that integrates a planner is proposed by [40], but the generated plan is exploited for behavior configuration and monitoring, while the behaviors are mainly bottom-up influenced by motivational drives. In the human–robot interaction literature, attentional mechanisms are usually related to visual perception and considered as important means of implicit nonverbal communication [41], [42] which are involved in joint attention [17], [43]–[45], anticipatory mechanisms [46], perspective taking [47], [48], etc. In contrast, in this paper we are concerned with attentional executive frameworks, which are pretty rare in the robotic literature [16], [29], [49] and usually not suitable for the execution of complex structured tasks. In this perspective, the work presented in this paper can be seen as a continuation of the one presented in [29], where only simple interactive manipulation tasks and bottom-up attentional regulations are considered. On the other hand, our framework naturally enables attention manipulation and attention-based interaction, which are crucial mechanisms for social communication [18]. The investigation of these issues is an interesting line of future research. Related to the system presented in this paper, Kawamura *et al.* [49] proposed the deployment of a cognitive control system for a humanoid involved in simple tasks. In this context, attention is mainly deployed to assign priority values to multiple sensory channels and to orient the focus of attention. Differently from our approach, executive attention and top-down attentional regulations are not considered. Instead, executive attention is investigated in [16] proposing an integrated neural architecture for a simulated autonomous robot that supports developmental learning. Here, hierarchical behaviors and top-down attentional modulations are considered, but only simple tasks are treated. In contrast, in our framework planning mechanisms are fully integrated within the overall architecture, hence complex structured plans can be generated and flexibly orchestrated. This integration allows us to scale the complexity of the robotic tasks and to tackle real-world human–robot interaction scenarios.

VI. CONCLUSION

In this paper, we proposed an integrated system that exploits cognitive control and executive attention to conciliate the execution of structured complex tasks with natural and flexible human–robot interaction. In this context, the overall execution is managed by top-down and bottom-up attentional mechanisms, while a hierarchical plan is used as a top-down attentional guidance that stimulates the system toward task accomplishment. This way, multiple hierarchical tasks can be flexibly orchestrated combining goal-oriented, reactive, and interactive behaviors. As far as the interaction with the human is concerned, the proposed attention-based control provides several interesting features: attentional human monitoring; flexible and interactive execution of complex plans; attentional manipulation (a human can influence the robotic behavior by orienting its attentional state). We discussed the proposed approach in a simulated robotic scenario considering different case studies. We first illustrated that the system can effectively orchestrate multiple tasks in a flexible manner, then we tested the efficacy of an attentional manipulation guidance during the execution of multiple structured tasks. In this context, we assessed both users' performance and their experience through questionnaires. The collected results suggest that the proposed system not only permits flexible and interactive execution of multiple tasks, but also enhances the naturalness of the user interaction. This assessment encourages us to test the system in real environments considering more complex cooperative tasks and long term interactions with real robots. In particular, we would like to investigate the effectiveness of the proposed interaction framework considering multimodal interaction scenarios, where utterance, gaze direction, physical interaction, and body postures are involved. In this setting, more sophisticated attentional regulation mechanisms (see [29]) can be introduced and assessed considering also visual attention, joint attention, and human intention recognition.

REFERENCES

[1] E. Karpas, S. J. Levine, P. Yu, and B. C. Williams, "Robust execution of plans for human-robot teams," in *Proc. ICAPS*, Jerusalem, Israel, 2015, pp. 342–346.

[2] A. Carbone, A. Finzi, A. Orlandini, and F. Pirri, "Model-based control architecture for attentive robots in rescue scenarios," *Auton. Robots*, vol. 24, no. 1, pp. 87–120, 2008.

[3] J. Shah, J. Wiken, B. Williams, and C. Breazeal, "Improved human-robot team performance using chaski, a human-inspired plan execution system," in *Proc. HRI*, Lausanne, Switzerland, 2011, pp. 29–36.

[4] R. Lallement, L. de Silva, and R. Alami, "HATP: An HTN planner for robotics," in *Proc. PlanRob*, Portsmouth, NH, USA, 2014.

[5] M. Fiore, A. Clodic, and R. Alami, "On planning and task achievement modalities for human-robot collaboration," in *Proc. Int. Symp. Exp. Robot.*, vol. 109. Marrakesh, Morocco, 2014, pp. 293–306.

[6] M. Posner and C. Snyder, "Attention and cognitive control," in *Information Processing and Cognition: The Loyola Symposium*. Hillsdale, NJ, USA: Erlbaum, 1975, pp. 55–82.

[7] 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*, vol. 4. New York, NY, USA: Plenum, 1986, pp. 1–18.

[8] R. P. Cooper and T. Shallice, "Hierarchical schemas and goals in the control of sequential behavior," *Psychol. Rev.*, vol. 113, no. 4, pp. 887–916, 2006.

[9] M. M. Botvinick, T. S. Braver, D. M. Barch, C. S. Carter, and J. D. Cohen, "Conflict monitoring and cognitive control," *Psychol. Rev.*, vol. 108, no. 3, pp. 624–652, 2001.

[10] K. S. Lashley, "The problem of serial order in behavior," in *Cerebral Mechanisms in Behavior*, L. A. Jeffress, Ed. New York, NY, USA: Wiley, 1951.

[11] M. M. Botvinick, “Hierarchical models of behavior and prefrontal function,” *Trends Cogn. Sci.*, vol. 12, no. 5, pp. 201–208, 2008.
[12] M. Scheutz, J. Harris, and P. Schermerhorn, “Systematic integration of cognitive and robotic architectures,” *Adv. Cogn. Syst.*, 2013, pp. 277–296.
[13] R. Cooper and T. Shallice, “Contention scheduling and the control of routine activities,” *Cogn. Neuropsychol.*, vol. 17, no. 4, pp. 297–338, 2000.
[14] M. Scheutz and V. Andronache, “Architectural mechanisms for dynamic changes of behavior selection strategies in behavior-based systems,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 6, pp. 2377–2395, Dec. 2004.
[15] M. N. Nicolescu and M. J. Matarić, “A hierarchical architecture for behavior-based robots,” in *Proc. AAMAS*, Bologna, Italy, 2002, pp. 227–233.
[16] J. Garforth, S. L. McHale, and A. Meehan, “Executive attention, task selection and attention-based learning in a neurally controlled simulated robot,” *Neurocomputing*, vol. 69, nos. 16–18, pp. 1923–1945, 2006.
[17] F. Kaplan and V. V. Hafner, “The challenges of joint attention,” *Interact. Stud.*, vol. 7, no. 2, pp. 135–169, 2006.
[18] M. Tomasello, *The Origins of Human Communication*. Cambridge, MA, USA: MIT Press, 2008.
[19] R. Caccavale and A. Finzi, “Plan execution and attentional regulations for flexible human-robot interaction,” in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Hong Kong, 2015, pp. 2453–2458.
[20] S. Rossi, E. Leone, M. Fiore, A. Finzi, and F. Cutugno, “An extensible architecture for robust multimodal human-robot communication,” in *Proc. IROS*, Tokyo, Japan, 2013, pp. 2208–2213.
[21] R. Caccavale *et al.*, “Attentional regulations in a situated human-robot dialogue,” in *Proc. RO-MAN*, Edinburgh, U.K., 2014, pp. 844–849.
[22] L. Lucignano, F. Cutugno, S. Rossi, and A. Finzi, “A dialogue system for multimodal human-robot interaction,” in *Proc. ICMI*, Sydney, NSW, Australia, 2013, pp. 197–204.
[23] D. Nau, Y. Cao, A. Lotem, and H. Muñoz-Avila, “SHOP: Simple hierarchical ordered planner,” in *Proc. IJCAI*, Stockholm, Sweden, 1999, pp. 968–973.
[24] A. D. Baddeley and G. J. Hitch, *Working Memory*, vol. 8. New York, NY, USA: Academic Press, 1974, pp. 47–90.
[25] G. Repovš and A. Baddeley, “The multi-component model of working memory: Explorations in experimental cognitive psychology,” *Neuroscience*, vol. 139, no. 1, pp. 5–21, 2006.
[26] 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,” *Human Movement Sci.*, vol. 26, no. 4, pp. 525–554, 2007.
[27] M. A. Arbib, E. J. Conklin, and J. A. C. Hill, *From Schema Theory to Language*. New York, NY, USA: Oxford Univ. Press, 1987.
[28] E. Burattini, S. Rossi, A. Finzi, and M. C. Staffa, “Attentional modulation of mutually dependent behaviors,” in *Proc. SAB*, Paris, France, 2010, pp. 283–292.
[29] X. Broquère *et al.*, “An attentional approach to human–robot interactive manipulation,” *Int. J. Soc. Robot.*, vol. 6, no. 4, pp. 533–553, 2014.
[30] D. D. Nocera, A. Finzi, S. Rossi, and M. Staffa, “Attentional action selection using reinforcement learning,” in *Proc. SAB*, Odense, Denmark, 2012, pp. 371–380.
[31] M. C. Mozer and M. Sitton, “Computational modeling of spatial attention,” in *Attention*. London, U.K.: Psychol. Press, 1998, pp. 341–393.
[32] D. S. Nau *et al.*, “SHOP2: An HTN planning system,” *J. Artif. Intell. Res.*, vol. 20, pp. 379–404, Dec. 2003.
[33] B. G. Witmer and M. J. Singer, “Measuring presence in virtual environments: A presence questionnaire,” *Presence Teleoperators Virtual Environ.*, vol. 7, no. 3, pp. 225–240, 1998.
[34] M. Duguleana, F. G. Barbuceanu, and G. Mogan, “Evaluating human-robot interaction during a manipulation experiment conducted in immersive virtual reality,” in *Proc. Int. Conf. Virtual Mix. Reality New Trends*, Orlando, FL, USA, 2011, pp. 164–173.
[35] M. Shiomi, T. Kanda, S. Koizumi, H. Ishiguro, and N. Hagita, “Group attention control for communication robots,” *Int. J. Humanoid Robot.*, vol. 5, no. 4, pp. 587–608, 2008.
[36] G. Boella and R. Damiano, “A replanning algorithm for a reactive agent architecture,” in *Proc. AIMSA*, Varna, Bulgaria, 2002, pp. 183–192.
[37] N. F. Ayan, U. Kuter, F. Yaman, and R. P. Goldman, “Hotride: Hierarchical ordered task replanning in dynamic environments,” in *Proc. Plan. Plan Execution Real World Syst. Principles Pract. Plan. Execution (ICAPS)*, Providence, RI, USA, 2007.
[38] I. Warfield, C. Hogg, S. Lee-Urban, and H. Muñoz-Avila, “Adaptation of hierarchical task network plans,” in *Proc. FLAIRS*, Key West, FL, USA, 2007, pp. 429–434.
[39] G. Milliez, M. Warnier, A. Clodic, and R. Alami, “A framework for endowing an interactive robot with reasoning capabilities about perspective-taking and belief management,” in *Proc. RO-MAN*, Edinburgh, U.K., 2014, pp. 1103–1109.
[40] A. Stoytchev and R. C. Arkin, “Incorporating motivation in a hybrid robot architecture,” *J. Adv. Comput. Intell. Intell. Inform.*, vol. 8, no. 3, pp. 269–274, 2004.
[41] C. Breazeal, C. D. Kidd, A. L. Thomaz, G. Hoffman, and M. Berlin, “Effects of nonverbal communication on efficiency and robustness in human-robot teamwork,” in *Proc. IROS*, Edmonton, AB, Canada, 2005, pp. 708–713.
[42] T. Müller and A. Knoll, “Attention driven visual processing for an interactive dialog robot,” in *Proc. ACM Symp. Appl. Comput.*, Honolulu, HI, USA, 2009, pp. 1151–1155.
[43] 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* (LNCS 1562). Heidelberg, Germany: Springer-Verlag, 1999, pp. 176–195.
[44] C. L. Breazeal, *Designing Sociable Robots*. Cambridge, MA, USA: MIT Press, 2002.
[45] Y. Nagai, K. Hosoda, A. Morita, and M. Asada, “A constructive model for the development of joint attention,” *Connection Sci.*, vol. 15, no. 4, pp. 211–229, 2003.
[46] G. Hoffman and C. Breazeal, “Cost-based anticipatory action selection for human–robot fluency,” *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 952–961, Oct. 2007.
[47] J. G. Trafton *et al.*, “Enabling effective human-robot interaction using perspective-taking in robots,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 4, pp. 460–470, Jul. 2005.
[48] C. Breazeal, M. Berlin, A. G. Brooks, J. Gray, and A. L. Thomaz, “Using perspective taking to learn from ambiguous demonstrations,” *Robot. Auton. Syst.*, vol. 54, no. 5, pp. 385–393, 2006.
[49] K. Kawamura, S. M. Gordon, P. Ratanaswasd, E. Erdemir, and J. F. Hall, “Implementation of cognitive control for a humanoid robot,” *Int. J. Humanoid Robot.*, vol. 5, no. 4, pp. 547–586, 2008.

Riccardo Caccavale is currently working toward the Ph.D. degree with the Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy.

He is involved in research projects sponsored by the European Community. His current research interests include cognitive robotics, human–robot interaction, and cognitive control.

Alberto Finzi received the Ph.D. degree in computer engineering from the Sapienza University of Rome, Rome, Italy.

He is an Assistant Professor with the Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy. He has been recently involved in several research projects sponsored by the European Community, the National Aeronautics and Space Administration, the European Space Agency, the Italian Space Agency, FWF (Austrian Science Fund), the Italian Ministry for University and Research, and private industries. His current research interests include cognitive robotics, human–robot interaction, executive and cognitive control, autonomous and adaptive systems, planning and scheduling systems, multiagent systems, and formal methods for autonomous systems.