UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

PH.D.THESIS IN

INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

FLEXIBLE TASK EXECUTION AND COGNITIVE CONTROL IN HUMAN-ROBOT INTERACTION

RICCARDO CACCAVALE

TUTOR: PROF. ALBERTO FINZI

XXIX CICLO

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

Contents

This page intentionally left blank.

List of Acronyms

The following acronyms are used throughout this text.

HRI Human-Robot Interaction

WM Working Memory

LTM Long Term Memory

SAS Supervisory Attentional System

This page intentionally left blank.

List of Tables

This page intentionally left blank.

List of Figures

Introduction

The flexible organization of activities is one of the most relevant issue in the development of autonomous robotic systems that are capable to naturally interact with the environment and the humans. The dominant approach in this context relies on plan-based autonomous systems, where planning and replanning processes are exploited to execute goal-oriented tasks, while reacting to external changes or human interventions. Although these frameworks can effectively handle many aspects of mixed-initiative human-robot interaction, in real-world cooperative scenarios, where unpredictable human behaviors or events are frequent, this paradigm can impair the robot performance and reduce the quality of the interaction. Following this perspective, the aim of this thesis is to develop an executive framework supporting robot autonomy and human-robot cooperation, that also enables flexible execution of structured and goal-oriented tasks. In contrast with the plan-based approach, in this work, we consider the cognitive psychology perspective on the human capability of properly orchestrating actions and tasks according to the executive context and the purpose [110]. In particular, we rely on the concept of cognitive control [26], which is the human ability to flexible adapt the execution of the activities in order to carry on multiple tasks simultaneously, switching from one to another, while reacting to external events, either disturbances or opportunities. In this respect, attentional processes play a crucial role in action monitoring, modulation, and coordination of cognitive and executive processes. One of the most influential model of cognitive control is indeed the Supervisory Attentional System (SAS) [110], where the cognitive control is obtained as the interaction of two different mechanisms: contention scheduling and supervisory attention. The first one manages routinized activities, while the latter provides top-down attentional modulation in order to overcome conflicts/impasses and smoothly drive goal-oriented behaviors [45].

The framework proposed in this thesis, inspired by the SAS model of human cognitive control [45, 26, 46], proposes attentional regulation mechanisms for the orchestration of robotic sensorimotor processes. In this setting, the task structure and the high-level processes (e.g. planning, reasoning or heuristics) can top-down affect the execution of low-level actions or sensorimotor behaviors, providing goal-oriented drives. On the other hand, low-level actions and monitoring processes can bottom-up influence the systems behaviors according to the environmental and the executive state. In the proposed framework, the attentional state of each robotic process is represented by an *emphasis* value that regulates the competition to acquire shared resources. Following this approach, as long as the robotic tasks are not interfering, they can take place in parallel, otherwise, in case of conflicts or unexpected events, the attentional regulations permit to rapidly switch from one task to another in order to overcome the conflicting or novel situations.

We discuss the system at work in different robotic contexts. In a first scenario the human-robot interaction capabilities of the proposed approach are explored. We then tackle the problem of flexible and cooperative execution of structured tasks discussing multiple plan execution, human-robot cooperation through attention manipulation [36, 42], and mechanisms for plan execution, monitoring and repair [41].

In this context, we discuss different case studies, inspired by the EU FP7 SAPHARI project, where flexible, human-guided and plan-guided task execution are considered. Furthermore, the proposed attentional system can be also exploited for the design of attentive adaptive interfaces [33]. We explore this approach in a human-multidrone interaction setting inspired by the EU FP7 SHERPA project. In this context, the system is to filter irrelevant informations, hence emphasizing the prominent ones with respect to the executive and the human state. Notice that the SAPHARI and the SHERPA domains provides two different HRI settings where attentional supervision can be exploited: in the first case it enables flexible and collaborative task execution; in the second case, it support human situation awareness in a multirobot context.

Finally, we show how the proposed framework can integrate learning-by-demonstration methods. We propose an application where attentional guidance and interaction by cueing can be exploited to support task teaching. The proposed framework integrates kinesthetic teaching and task-based attentional guidance.

The rest of this thesis is organized as follows. In Chapter 1 an overview of the related works is provided with a special focus on the cognitive models and concepts which inspired the system design. In Chapter 2, the cognitive control framework is introduced detailing its components. In Chapter 3 the application of the proposed system in the context of human-robot interaction is presented. Chapters 4 focuses on flexible plan execution, while Chapter 5 discusses the execution of human-robot collaborative plans. This solution has been proposed in the context of the EU FP7 project SAPHARI. Chapter 6 describes an adaptive attentional interface that adopts the proposed attentional framework to filter the information provided by a set of drones to a human operator. This solution is proposed in the context of the EU FP7 project SHERPA. In Chapter 7, we discuss adaptive methods proposing a learning-by-demonstration framework that combines attentional supervision and kinesthetic teaching. The final chapter summarizes the results and discusses possible lines of future research.

Chapter 1

Background and Related Works

In the context of human-robot interaction, a robotic system should be capable of dynamically execute complex activities, react to human interventions or environmental changes and also continuously learn novel tasks. These issues are 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 robotics operating in domestic environments). Following this perspective, several frameworks has been proposed in the robotic literature to conciliate some of these aspects with a natural human-robot interaction. Some systems rely on deliberative approaches where planning and replanning processes manage the task execution and the adaptation to environmental changes or human actions, while alternative works involve reactive or hybrid reactive/deliberative systems to allow faster responses to external events. Other works directly exploit cognitive modeling and cognitive architectures. In this case robotic actions are selected by bio-inspired systems in order to improve the naturalness of the human-robot interaction.

In the following sections we firstly introduce some widespread robotic approaches regarding deliberative and reactive systems, discussing their features and limitations, and providing some relevant examples and related works (Section 1.1). In a second section we focus on cognitive systems, with particular attention on the robotic applications in human-robot interaction and cooperation (Section 1.2). Finally, some relevant concepts of cognitive science which are related to the framework proposed in this thesis, are introduced (Section 1.3).

1.1 Deliberative and Reactive Systems

The problem of flexible and interactive task execution in robotics has been largely addressed by the artificial intelligence research. Relevant works in this perspective rely on the three layer architectures [63] where *deliberative layer* and *control layer* are interleaved by a middle layer that manages plan execution and schedules the associated executive/monitoring processes. Some plan-based systems emphasize the role of the deliberative layer exploiting replanning and plan adjustment processes to overcome unexpected events [62, 91, 56, 145, 69]. For instance in [62] the tasks to be executed are represented in a queue, which maintains for each task a set of alternative methods along with their contextual constraints (specifying when methods are applicable). Therefore, in case of failures, the plan can be adjusted deploying alternative methods. A more structured executive framework is proposed in the CLARAty architecture [56, 107]. Here, the executive functions are are managed through the TDL (Task Description Language) system that decomposes planned task into "task trees" which provide execution monitoring, synchronization and exception handling.

In the aforementioned frameworks external changes are mainly managed at the deliberative layer or, otherwise, by proposing alternative methods that do not interfere with planned activities. However, deliberative processes are often characterized by high variance performance, which may conflict with real-time robotic requirements [102, 57, 97].

Following the plan-based approach, in the context of human-robot interaction and cooperation, flexible and interactive task 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 [23, 13, 148]. For instance, in human-aware planning [87, 99] structured hierarchical plans are generated for both the human and the 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 [58, 44, 76]. These plan-based methods are effective in mixed-initiative planning, on the other hand, the associated continuous planning and replanning process usually impairs the naturalness and effectiveness of the interaction with the humans and the environment.

In contrast with these approaches, reactive and behavior-based systems [12] have been proposed to on-line adapt the task execution to the environmental changes or unexpected events [11, 108, 129]. A relevant example is proposed in [118] where the iB2C architecture integrates simple behaviors in complex robotic tasks allowing behaviors interaction, coordination and hierarchical abstraction. Another hierarchical behavior-based system has been proposed in [108] where also deliberative processes are considered. Differently, other works like [136] deploy behavior-based systems to learn tasks and objects representation. Other bio-inspired approaches include also simple attentional mechanisms for the dynamic control of behaviors [129, 130, 135]. For instance, in [129], attentional mechanisms are mentioned as behavior orchestration mechanisms and deployed in a case study to detect lack of progress towards the target. In contrast, in [130], attention is mainly used to orient and focus the system perception. However, in the latter frameworks, structured tasks execution and plan-based control are not considered. A hybrid architecture that integrates a planner is proposed by [135], but the generated plan is exploited for behavior configuration and monitoring, while the behaviors are mainly bottom-up influenced by motivational drives. The framework proposed in [66] exploits a behavior representation of actions for mobile robots. In this case parallel behaviors are managed by arbiters [134] that fuse their output. These systems have been largely applied in mobile robotics, but relevant applications in the context of human-robot interaction are more rare; moreover, an effective mechanism to orchestrate low-level behaviors and processes is still missing. Alternative frameworks, instead, exploit human-robot interaction to learn novel structured tasks and adapt/integrate pre-programmed actions and behaviors to the executive context. In [109] explicit verbal communication allows the robotic system to learn structured tasks starting from basic motion skills, while in [10, 52] the high-level task representations are learned from the observation of the human activities. This approach is useful for the human activity prediction and for intuitive robotic teaching, but the learned task structure is often rigid and unable to be flexibly adapted to the executive context.

1.2 Cognitive Systems

All the previous systems address specific aspects of the human-robot cooperation. Other approaches, exploit cognitive architectures and systems for the development of flexible and human-friendly robotic frameworks [53, 88, 47, 65, 124]. The *cognitive architectures*, in particular, are computational models that integrate various concepts from cognitive psychology and neuroscience, and they are able to emulate or interpret a large number of human capabilities and behaviors. These architectures can be exploited in a robotic setting in order to execute complex tasks. For example, the SOAR cognitive architecture [84] is applied in different robotic contexts such as pick-and-place with robotic arm [86], mobile robot planning/execution [85] and navigation [83]. Moreover, cognitive architectures can be deployed in social and cognitive robots increasing the naturalness of the interaction and supporting the execution of human-like robotic behaviors. In this context a notable example is given by [143], where the proposed framework exploits the ACT-R cognitive architecture [8] to learn the hide-and-seek game starting from children demonstration. Further applications of the above work include the spatial representation and reasoning of human-robot team [79], and the prediction of human actions in human-robot cooperative patrolling [141].

Despite these robotic applications, the architectures presented above are mostly focused on the validation of theories from psychology rather on the development of effective robotic frameworks. In contrast with the classic cognitive systems, alternative architectures have been proposed to be specific for robotic applications [5, 21, 78, 64, 150]. An interesting example is the 4D/RCS architecture [4, 5, 6, 131]. The system provides a hierarchical representation of the tasks where the nodes of the hierarchy represent both sensory and executive processes and allows planning/replanning cycles at different level of abstraction [6]. This architecture has also been applied in robotic-assisted constructions [94] in order to support human teleoperation. Another similar robotic architecture is ADAPT [21, 20], a SOAR-based framework endowed with a hierarchical representation of tasks. These systems provides robotic applications, but the planning/replanning approach can be uneffective in dynamic and human dwelled environments. Indeed, relevant human-robot interactive scenarios are not considered. Other applications of cognitive architecture in robotics are proposed in [78, 131] but mainly related to object manipulation. Further details about architectures and their applications can be found in [81].

The aim of this thesis is to propose an executive framework that facilitates human-robot cooperation in dynamic environments, supporting flexible task execution, scheduling, learning, and continuous monitoring of humans and environment. To this end, in contrast with some of the mentioned systems, these issues are tackled from a different perspective exploiting the concept of cognitive control [113, 110, 46, 26] introduced in cognitive psychology and neuroscience. The cognitive control includes mechanisms and functions needed to support flexible and adaptive responses, which underly the execution of complex goal-directed cognitive processes and behaviors. Among the most influential models for cognitive control, the one proposed by Norman and Shallice [110] assumes that two main processes are involved in activity orchestration and execution: *contention scheduling* and *supervisory attentional system*. Contention scheduling is a low-level reactive process in charge of executing habitual and routinized activities, hence 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 cognitive model, attentional regulations play a central role in action selection. Indeed, each process 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 [45, 46]. On the other hand, in the robotics literature the deployment of these mechanisms is quite rare. In the human-robot interaction literature, attentional mechanisms are usually related to visual perception and considered as important means of implicit nonverbal communication [30, 101] which are involved in joint attention [75, 27, 104], anticipatory mechanisms [68], perspective taking [142, 29], etc.. In contrast, in this thesis we are concerned with attentional executive frameworks, which are pretty rare in the robotic literature [31, 60, 77] and usually not suitable for the execution of complex structured tasks. For example, in [77] the authors proposes 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. In this case, executive attention and top-down attentional regulations are not considered. Instead, executive attention is investigated in [60] 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, the framework proposed in this thesis allows the integration of planning mechanisms, 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.

1.3 Cognitive Control

In this section, we focus on the cognitive control notion discussing related concepts and models.

Central Executive and Working Memory The *working memory* (WM) is a short-term repository that enables information storage and manipulation to support reasoning, decision making, and goal-oriented behaviors [16, 14]. In [16], Baddeley and Hitch introduce this notion as follows:

> A brain system that provides temporary storage and manipulation of the information necessary for such complex cognitive tasks as language comprehension, learning, and reasoning. [Baddeley and Hitch 1975]

The term working memory was firstly introduced by Miller, Galanter, and Pribram [98] to identify the short-term repository of the temporary information related to executing tasks and cognitive processes. The model of working memory was further refined by Baddeley and Hitch during their study on the amnesia [17]. In this work they hypothesize the existence of a temporary storage, different from the short-term memory, specific for task-relevant information and able to support reasoning processes [16]. From the experimental evidences, Baddeley and Hitch formulate a model of the working memory represented by a master-slave architecture where a *central executive* (master) manage and integrates the information form underlaying modules (slave), maintaining a shared representation which is independent from the source (schemata [9]).

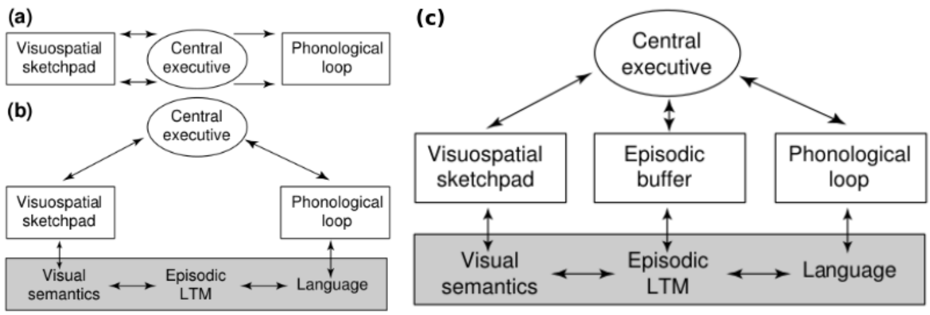

Figure 1.1. Evolution of the working memory model (Baddeley and Hitch 1975-2000).

The number and the functions of the slaves modules hypothesized by Baddeley and Hitch changed during years (see Figure 1.1) up to comprise three buffers: the *visuospatial sketchpad* collects vision-based inputs, the *episodic buffer* that contains episodes from the past which are recalled from the long-term memory, and the *phonological loop* that allows humans to rehearse task-specific informations [15]. The role of these buffers is to store informations from/to outer systems (visive, linguistic, and long-term memory) while the central executive exploits the attentional supervision to emphasize informations and related tasks [14]. In this context cognitive processes (such as reasoning or planning) exploit the shared representation provided by the working memory and influence the human behaviors, in so enabling cognitive control.

Attention and Conflict In cognitive science, the humans ability to select and successfully monitor behaviors and processes is included under the term of *cognitive control* [26] and involves a large number of cognitive processes (*executive functions* [55, 51]) such as attentional control, working memory, task switching, reasoning and planning. In a more general sense the cognitive control can refer to processes that allow information processing and behaviors to real-time adapt depending on the current goals and the environmental state. Cognitive control can be characterized as the:

> ability to configure itself for the performance of specific tasks through appropriate adjustments in perceptual selection, response biasing, and on-line maintenance of contextual information. [Botvinick et al. 2001]

The aim of this thesis is to develop an architecture which allows cognitive control of robotic processes during the execution of structured tasks. Studies of human action selection [110, 26] emphasize to the role of *attention* in the cognitive control of actions during the execution of structured or complex tasks. The studies of attention date back to the late nineteenth century (James 1890 [73]):

> Everyone knows what attention is. It is the taking possession by the mind, in clear and vivid form, of one out of what seem several simultaneously possible objects or trains of thought. Focalization, concentration, of consciousness are of its essence. It implies withdrawal from some things in order to deal effectively with others, and is a condition which has a real opposite in the confused, dazed, scatterbrained state which in French is called distraction, and Zerstreutheit in German. [James 1890]

However, there is still an open debate about its main models, mechanisms, and functions [114, 153]. Indeed, attention is a core element of many relevant phenomenas of cognition, e.g. Stroop effect [137], ideational/conceptual apraxia [48, 45], attention-deficit and hyperactivity disorder [18] to mention only few examples. In the context of cognitive control, attentional regulations effects task-relevant actions and processes allowing goal-oriented behaviors.

Relevant studies about attention in human activities are about visual attention [144, 115, 72]. Here attentional mechanisms such as selection, focusing and switch, drive the eyes orientation and the fixation of the salient elements in the scene. Of particular interest in this context are the mechanisms for suppression/selection of visual stimuli [74, 100]. Indeed, the presence of multiple attractors can interfere in the attentional allocation, hence only relevant stimuli have to be selected, ignoring the distracting ones. These conflicting situations, also known as *crosstalk interference* [100], are managed by cognitive control exploiting attentional supervision where high-level cognitive processes are so that high-level cognitive processes can modulate and emphasize goal-oriented stimuli and behaviors [49].

Attentional regulations and conflict management have also been studied specifically in the context of action selection. In the attention-to-action model [110] (also known as Norman-Shallice model) two distinct attentional process can be employed. The attentional process selection depends on how the actions to execute are simple and familiar. On the one hand the routinized actions, triggered in response to well-known environmental stimuli, can be easily managed through automatic attentional processes with low impact on the human cognitive load. On the other hand, novel and unfamiliar actions, related to unique or unexpected situations, requires more cognitive resources to be correctly scheduled and executed, hence controlled attentional processes are employed. In the case of routinized actions, a mechanism of *contention scheduling* (CS) [45] is involved in the selection process. Here non-interfering actions, which do not require shared resources, can be directly executed, while conflicting actions have to compete each other: only the most active actions are selected for the execution. Conversely, when novel or unexpected events occur, contention scheduling is not sufficient to overcome the new situation, hence a more complex attentional mechanism is needed. In this case a *supervisory attentional system* (SAS) [110] is employed to oversee the contention scheduling and regulate the actions and processes activation according to the executive context. Through this attentional regulation the SAS exploits deliberative processes to solve the impasses due to the novelty, and provides a mechanism for the effective and goal-oriented action scheduling. This mechanism is better detailed below.

Contention Scheduling and Supervisory Attention In the literature, one of the most relevant model for the action selection is the *Interactive Activation Network* (IAN) proposed by Cooper and Shallice [45].

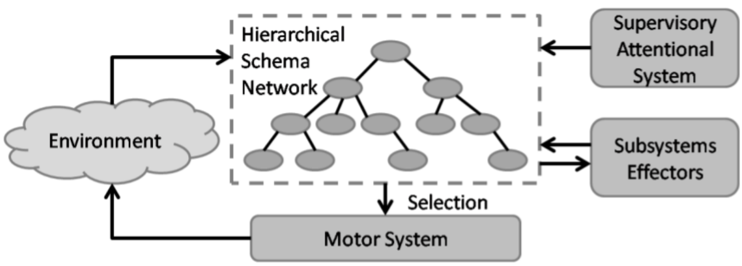

Figure 1.2. Abstract representation of the Interactive Activation Network (Cooper e Shallice 2000). The framework comprises a hierarchical schema network, which integrates and monitors subsystems and effectors (also represented as a schema network) and is top-down regulated by the supervisory attentional system.

In the IAN model a hierarchical representations of the actions is proposed: actions are associated with hierarchical schemata [9, 67] which are decomposed in sub-schemata to decrease the level of abstraction. This hierarchical decomposition stands until the concrete schemata (which generates the motions of the related actions) are reached. Following this approach, the schemata execution is regulated by the SAS which allocates the attentional resources on task-specific elements of the hierarchy:

> The scheduling is, therefore, quite simple and direct. No direct attentional control of selection is required (or allowed). Deliberate attention exerts itself indirectly through its effect on activation values. [Norman and Shallice 1980]

The action schemata of the hierarchy are then allowed to acquire the shared resources (actuators and objects) through the *contention scheduling*. Specifically, each action is associated with an activation level which is affected by the following influences:

- The *self influence* represents the contribution of the schema itself. In case a schema is not selected by the attentional system it increases its activations value in order to be selected in future.
- The *lateral influence* decreases the activation of a schema according to the number and the activations of competitors.
- The *internal influence* given by the source of a schema (i.e. the upper node in the hierarchy) that increases a sub-schema activation when selected by the attentional system.
- The *external influence* regulates the the activation value with respect to external features/stimuli.

Another relevant model of the human cognitive control and action selection is the *simple recurrent network* (SRN) proposed by Botvinick and Plaut in 2004 [25]. In the SRN the role of the supervisory attentional system is still of crucial importance but, in soft contrast with the above IAN model, the actions are represented by recursive networks instead of a static hierarchy of schemata:

> Although a given schema may be associated with multiple higher-level schemata [...], there is typically no mechanism that allows the details of the lower level schema to change depending on the higher-level schema that recruits it. [Botvinick and Plaut 2004]

The presented cognitive control models have been proposed to capture relevant features of the human cognition processes, therefore their application and adaptation to the orchestration of complex robotic frameworks is not direct. In this work, we aim at providing a cognitive control framework suitable for robotic system which is inspired by these models and mechanisms.

This page intentionally left blank.

Chapter 2

The Attentional System

Flexible plan execution is a crucial issue in cognitive robotics and human-robot interaction. Indeed, a robotic system should be capable of dynamically executing complex tasks, while reacting to human interventions and environmental changes. These capabilities are particularly relevant in service and social robotics, where the execution of structured and well defined activities (e.g. human-robot interaction in a factory or in a domestic environments) should preserve the naturalness and the flexibility always needed when a human is involved.

As discussed in the previous chapter, in cognitive neuroscience these capabilities are associated with the concept of cognitive control [26] which comprises the executive mechanisms/functions needed to flexibly orchestrate and switch between different cognitive and executive processes, while keeping an overall coherent and finalized behavior. In this context, activity control is usually believed to be hierarchically organized [89, 130] and several models have been proposed to provide an account for action selection and execution. As already explained, in the influential model of Norman and Shallice [110, 46] action control is obtained as the interaction of two main processes: contention scheduling and supervisory attentional system.

The first one, is a low-level mechanism that reactively selects and regulates concurrent sensorimotor processes representing habits and routinized activities; the second one is a higher-level mechanism that coordinates and monitors schemata in order to manage novel and complex tasks. In this process, attentional regulations play a central role. Inspired by these cognitive control models, we propose a framework that integrates planning, executive, and interactive processes providing flexible mechanisms for hierarchical task execution, regulation, and switching. While approaches to dynamic control of hierarchical activities have been proposed in the robotics literature [108, 129], the integration of plan execution and attentional control appears as a less explored topic. In this chapter, we present our approach and show how the proposed framework permits flexible and interactive execution of multiple plans by exploiting top-down (task-oriented) and bottom-up (stimuli-oriented) attentional regulations. Specifically, attentional mechanisms are here deployed to smoothly regulate the activations of multiple hierarchical robotic behaviors by emphasizing the ones coherent with respect to the task and the environment.

2.1 Cognitive Control Cycle

In this section, we introduce our framework along with its main components. 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 [16], the WM maintains and manages short-term data for online processing and execution, supported by a LTM which provides a vast, long-term storage of information [121]. 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

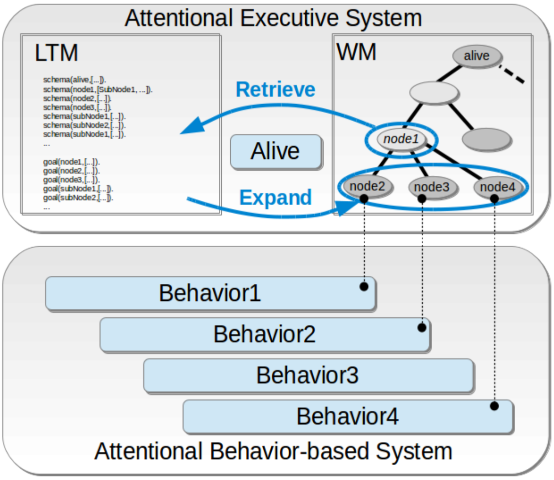

Figure 2.1. System architecture. The cognitive control cycle (blue arrows and ovals) is managed by the process *alive* that updates the *Working Memory* (WM) exploiting the behavior schemata stored in the *Long Term Memory* (LTM).

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 and ovals in Figure 2.1) 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.

2.2 Working Memory

In line with a typical approach in AI and cognitive psychology [89, 122, 110, 46], we assume a hierarchical organization for tasks and activities. In our framework, this hierarchy is represented in the WM as tree structure (see Figure 2.2) 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 Figure 2.2), instead an abstract behavior identifies a complex activity that needs to be hierarchically decomposed into different sub-activities in order to be executed (e.g. $take(objRed)$ in Figure 2.2).

2.2.1 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 $\mathbf{schema}(m, l, e)$, where m is the name of the behavior, l is a list of sub-behaviors associated with enabling conditions (releasers), i.e. $l{=}\langle(m_1, r_1), \ldots, (m_n, r_n)\rangle$, 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:

$$\begin{aligned}&\mathbf{schema}(take(Obj),\\&\qquad\langle(goto(Obj), true), (pickup(Obj), Obj.reached)\rangle,\\&\qquad Obj.taken),\end{aligned}$$

$$\begin{aligned}&\mathbf{schema}(goto(Obj),\\&\qquad\langle(explore(X, Y), true), (followColor(Obj), true)\rangle,\\&\qquad Obj.reached).\end{aligned}$$

In the first schema, the behavior of taking an object *take*(*Obj*) is composed of two sub-behaviors: reach the object (*goto*(*Obj*)) and 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 Figure 2.2, 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.

2.2.2 Working Memory Structure

The WM is represented by an annotated tree, whose nodes represent allocated processes/behaviors, while the edges represent parental relations among sub-processes/sub-behaviors. Each node, either concrete or abstract, represents an instance of a LTM behavior schema, hence it is associated with a name, a set of sub-behaviors, a post-condition, and a releaser. Both the releaser and the post-condition are represented as boolean expressions to be satisfied. For instance, in Figure 2.2, *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 Figure 2.2, *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.

2.2.3 Working Memory 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 object, the *interaction* behavior allocates a $take(objRed)$ node, that will be expanded by the *alive* process (see Figure 2.2) selecting the sub-behaviors involved in the hierarchy, as specified in the $take(Obj)$ schema presented above.

Algorithm 1 The 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
```

2.3 Behaviors and Attentional Regulations

Following a schema theory representation [9, 32], 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 Figure 2.3). The releaser enables/disables the motor schema. As already stated above, it is expressed as the conjunction of two boolean expressions to be satisfied: the internal releaser and 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

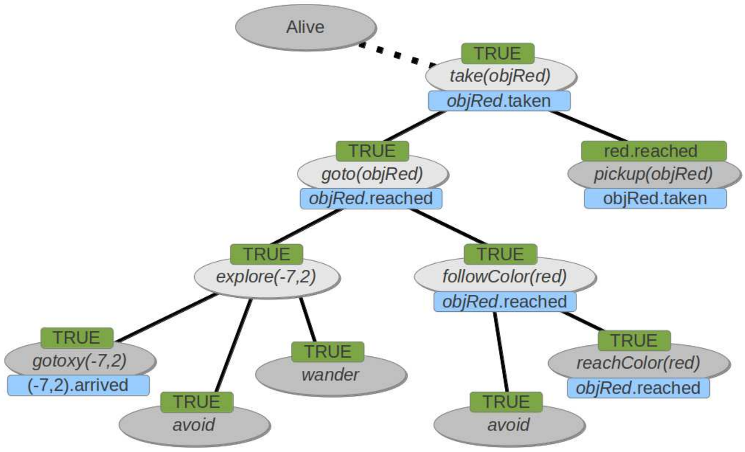

Figure 2.2. Hierarchical tasks in the WM: releasers, behaviors, and post-conditions are, respectively, green, gray, and blue.

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 [32, 31], while related methods for parameters setting are discussed in [50].

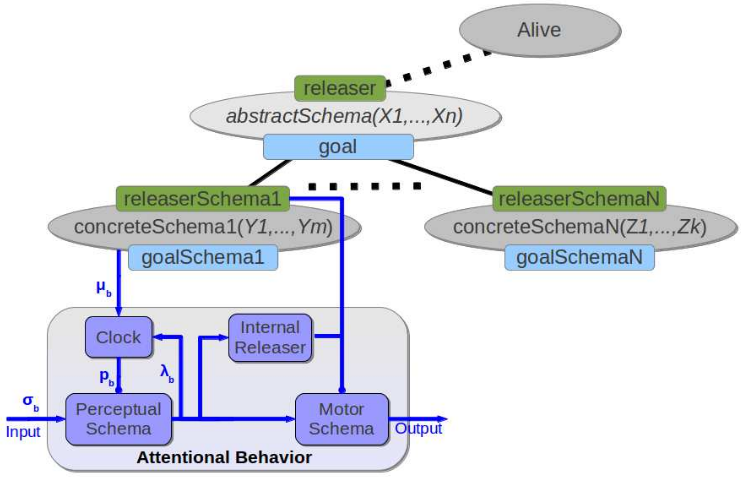

Figure 2.3. Tree structure and detail of a concrete behavior. The clock period p_b is top-down (μ_b) and bottom-up (λ_b) regulated, while the internal and the external releasers depend on internal and external properties, respectively. The external releaser for a concrete behavior is the conjunction of the releasers of the nodes along the path to *alive*.

2.4 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 Figure 2.4). 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* [100]. These conflicts and impasses can be either prohibited by construction or solved by means of suitable evaluation functions [26].

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.

2.4.1 Bottom-up Regulation

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:

$$g(\sigma_{avd}, \varepsilon_{avd}) = \begin{cases} \lambda^{min} & \text{if } \sigma_{avd} \leq r^{min} \\ \lambda^{max} & \text{if } \sigma_{avd} \geq r^{max} \\ \alpha \cdot \sigma_{avd} + \beta & \text{otherwise,} \end{cases} \tag{2.1}$$

characterized by two parameters r^{min} and r^{max}, while α and β are used to describe the linear increase of g for $\sigma_{avd} \in [r^{min}, r^{max}]$ as follows:

$$\alpha = \frac{\lambda^{max} - \lambda^{min}}{r^{max} - r^{min}} \tag{2.2a}$$

$$\beta = \lambda^{min} - \alpha \cdot r^{min} \tag{2.2b}$$

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 [31]).

2.4.2 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.

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.

procedure BEHAVIORCYCLE(b)
    while s is allocated in the WM **do**
        activate the perceptual schema of b
        if rel_b^i and rel_b^e are satisfied **then**
            activate the motor schema of b
        end if
        update λ_b and μ_b
        wait for $1/e_b$
    end while
    end b
end procedure

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 (line 2-3), while the motor schema is active only if its releasers are satisfied (line 4-6); at the end of the cycle the new clock period is defined (line 7-8). 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:

$$f(\sigma_b, \varepsilon) = \frac{g(\sigma_b, \varepsilon_b)}{\mu_b} = p_b \tag{2.3}$$

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 towards 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 towards its active successors. As a side effect, this mechanism induces a lateral influence among the behaviors involved in the same task.

2.4.3 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 non-mutually 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 non-mutually 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: acceleration of the clock and 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 towards the goal accomplishment, goal-oriented behaviors become dominant, in so overcoming decisional impasses. In the following chapter,

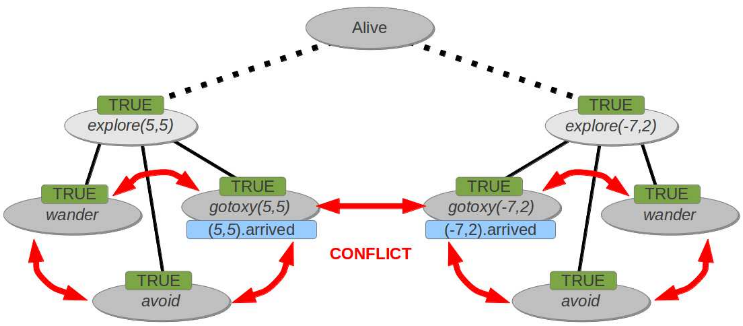

Figure 2.4. Example of conflicting tasks: the two *explore* tasks are in conflicts and associated with mutually exclusive (e.g. $gotoxy(5,5)$ vs $gotoxy(-7,2)$) and not mutually exclusive (e.g. *avoid* vs $gotoxy(5,5)$) concrete behaviors.

we will discuss how this framework can be exploited in a human-robot interaction setting.

This page intentionally left blank.

Chapter 3

Attentional Regulations in Human-Robot Interaction

In this chapter we propose an extension of the framework that accounts for the human interaction. A natural and effective interaction between humans and robots can be modeled as a multimodal dialogue flow, involving speech, gaze orientation, gestures, while attentional mechanisms can be used to orient and focus the robotic and the human perceptive and cognitive processes during the interaction. Some authors addressed these issues considering visual attention during human-robot conversation [101, 111] detecting the human to interact with or the task to be executed. Other authors mainly focused on joint attention and perspective taking methods for HRI [128, 29]. Differently from these approaches, which are mainly concerned with visual attention, in this thesis we focus on executive attentional mechanisms regulating the human-robot dialogical interaction. More specifically, we aim at defining a system that can manage and regulate the multimodal dialogue between the human and the robot by exploiting top-down and bottom-up attentional regulations. In this chapter, we propose and discuss a multimodal real-time HRI system integrating a dialogue manager and a hybrid cognitive control architecture. The dialogue between the human and the robot is modeled as a Partially Observable Markov Decision Process (POMDP) [92] 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 that are not coherent. We illustrate the system at work in simple scenarios where the human and the robot have to interact in order to accomplish a cooperative tasks.

Notice that an overview of the work reported in this chapter is published in [39], while further details can be found in [40].

3.1 System Architecture

The HRI architecture proposed in this work is depicted in Figure 3.1. In particular, the hierarchical representation of the executing tasks is maintained in WM while the human-robot dialogue and the multimodal interaction is managed by the HRI module.

3.1.1 Cognitive Control Cycle

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*, etc.). Each allocated behavior can affect the WM by inserting new nodes. For example, if the *interaction block* allocates a $take(objRed)$

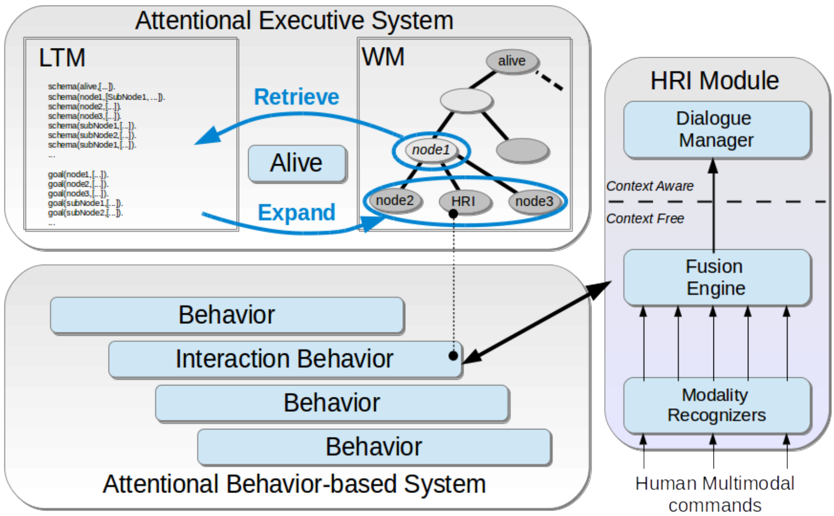

Figure 3.1. The HRI architecture

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 2.2 in the previous chapter) allocating other nodes as specified in the LTM. Moreover, in case of ambiguous commands or misclassification, multiple concurrent behavior can be allocated in WM in order to be flexibly executed by the system exploiting bottom-up and top-down attentional regulations. In particular, human multimodal commands or actions can drive the robot execution by top-down emphasizing specific behaviors. These features will be better detailed in Section 3.2.

3.1.2 Dialogue Management and Multimodal Framework

The HRI module is appointed to recognize the multiple human commands and actions, such as utterances, gaze directions, gestures or body postures, and to provide an interpretation of user's intentions according to the dialogue context. It is integrated in the overall architecture as a special behavior (the *interaction behavior* in figure 3.1) and it is composed of three layers: the lower layer contains the *classifiers* of the single modalities; the middle layer, the *fusion engine*, performs a Support Vector Machine (SVM)-based late fusion and provides a context-free integration of the multiple inputs [123]; the upper layer, the *dialogue manager* [93], 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 interaction model. The dialogue models are provided as graph-based specifications (see Figure 3.2). Multiple dialogue flows can be combined in order to build a dialogue model in a modular and extensible manner [93]. 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 off-line generated for that interaction model. For additional details we refer the reader to [93, 123].

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.

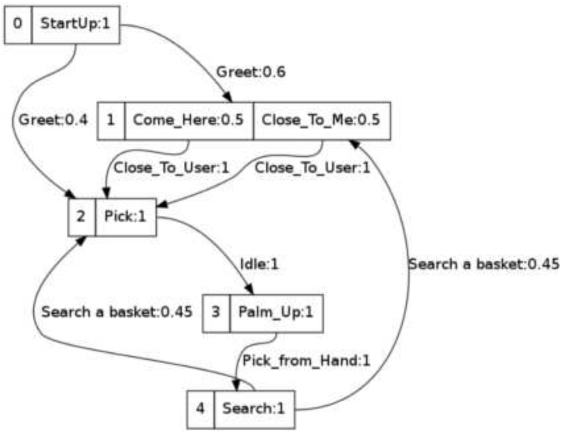

Figure 3.2. Excerpt extracted from dialogue models: node 1 has the two possible interpretations "Come Here" and "Close to Me". In both these cases, the robot action is to go close to the human from where, in the node 2, the robot expects that the user asks to pick something.

3.2 Case Studies

We now discuss the system behavior considering simple case studies.

3.2.1 Mobile Robot Scenario

The robot shares the workspace with several users which can interact with the system 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 4.4 (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 the robot has a list of user dialogue models describing possible patterns of commands or movements. Each gesture is linked to one or more meanings, so 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 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.

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 3.1. Execution time of a generic *take* in different contexts.

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 reference 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. 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 2.4) 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 3.3 (up) 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 3.3 (up) 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 Tab. 3.1, we illustrate 10 runs where the robot interprets and executes an unreferenced *take* given the dialogue model and the belief state (see [93]). 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 3.1, left); in the second one we have three objects (red, green, and yellow in Tab. 3.1, right). For each scenario we report the executed sequence of tasks and the time needed to accomplish the goal (minutes). The *executed*

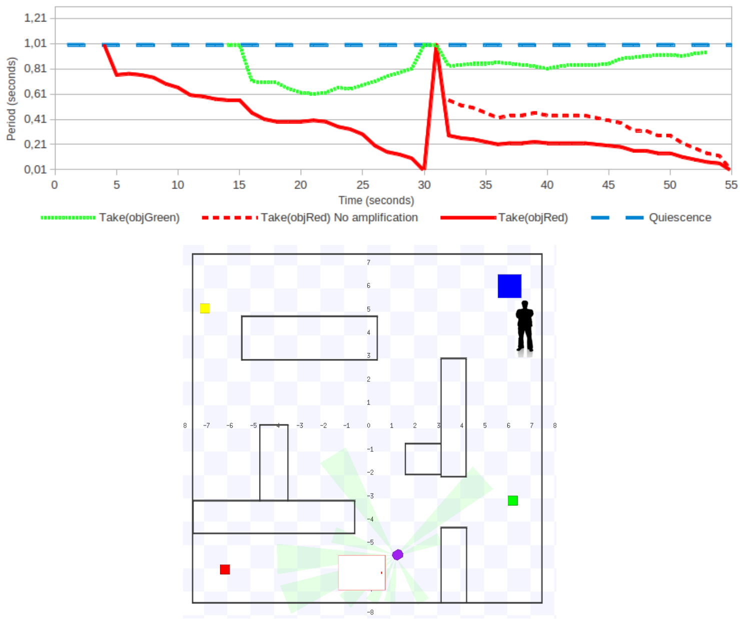

Figure 3.3. (up) Period modulation during a conflicting situation in a lab scenario: *take*(*objRed*) is amplified, hence the frequency and the outputs are enhanced driving the robot towards the red target; (down) lab scenario.

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 Tab. 3.1 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 Tab. 3.2 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., $take(green)$). By comparing the average values at the end of Tab. 3.1 with the values in Tab. 3.2 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 3.2. Execution time of the specific *take*.

3.2.2 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 [45]) 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 3.4) by *alive* once a suitable stimulus is detected (e.g. human command mentioning the coffee). Here, the human can either

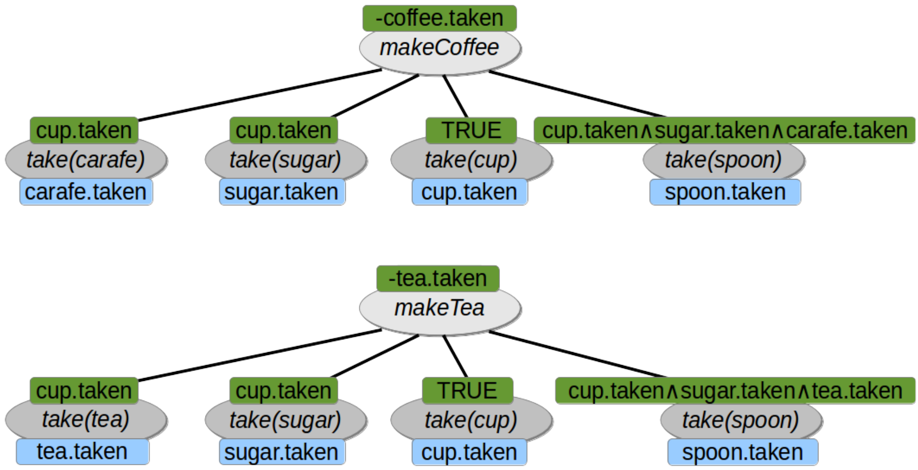

Figure 3.4. MakeCoffee task in the WM.

take an object or get it form the robot (see Figure 3.5). The actions of the robot are simulated: these are only declared, but not implemented (we only introduce a 10 sec sleep that simulates the action execution), while the human actions and objects are monitored by a kinect-based visual system that provides object recognition/tracking and hand tracking. Analogously to the previous case study, the user can interact with the system using gestures and speech. A human gesture can be either interpreted as a command or as an object manipulation action depending on the target of the human gesture and the proximity of objects. Therefore, the system can either respond to a human request or take an 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 3.5, 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 3.5, center) which is the only action enabled since the $take(carafe)$ was already executed by the human. Finally, the human can conclude the task with the $take(spoon)$ action (Figure 3.5, right). Figure 3.6 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 monitored by concrete attentional behaviors whose frequencies are regulated by a function of the tracked features (e.g. in Figure 3.6 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.

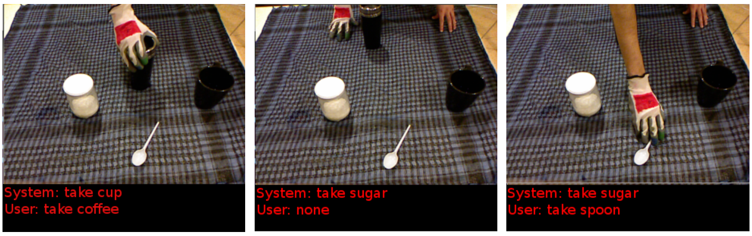

Figure 3.5. (left) the system selects the cup, then the user takes the carafe, (center) the system selects the sugar (cup and carafe already taken), (right) the system takes the sugar, then the user takes the spoon.

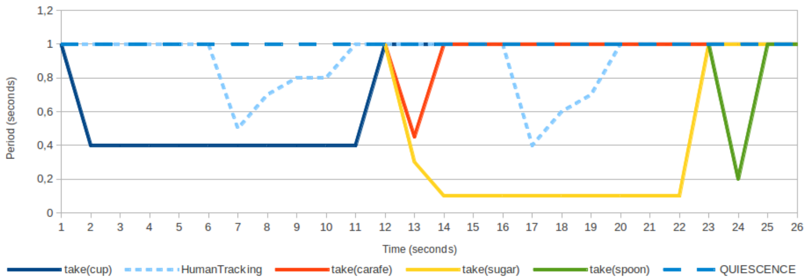

Figure 3.6. Period modulation profile in the coffee scenario. Both human (dotted line period) and robot (solid line period) behaviors are tracked by the attentional system.

3.2.3 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 3.4 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 to correct; this allows us to test how the system can deal with this additional ambiguity and misinterpretations. The scenario is depicted in Figure 3.7 where the following objects are disposed on the table: a cup, a spoon, sugar box, a tea box, and a coffee box.

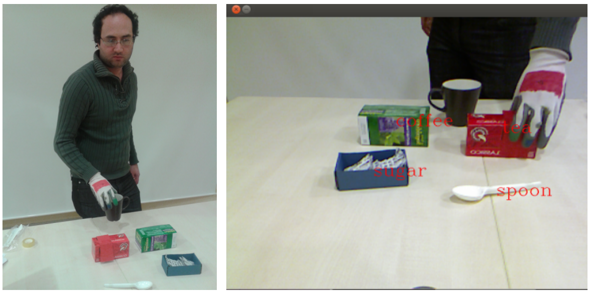

Figure 3.7. (left) The human takes a cup (right) the system detects the human taking the tea. In the second case, the task is disambiguated by the first action of the human.

In this context, as preliminary test, we asked 10 subjects (grad. students; 6 males, 4 females) 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 Tab. 3.3 show that, despite the inherent ambiguity of the domain, the task can be accomplished in 83.3% of the cases, considering both directly successful interactions (robot initiative correct w.r.t. the human intention) or interactions where human explicit corrections are needed (correction). Moreover, the robot initiative seems effective in reducing the human actions needed to execute the task.

	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 3.3. 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.

This page intentionally left blank.

Chapter 4

Flexible Plan Execution

In this chapter we propose an integrated framework that combines hierarchical planning, flexible execution of multiple structured tasks, and human-robot interaction 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 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 [75] (e.g. gestures, utterance, object manipulation, etc.) to indirectly bias the robot behavior towards the execution of the required activities. Notice that attention-based interaction is also very relevant for social communication [140], but in this work 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.

An early version of the work reported in this chapter was published in [36], while the full version, enriched with further experiments and details, can be found in [42].

4.1 System Overview

The framework proposed in this chapter 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 the proposed attentional system to affects and orients sensory processing and behaviors activations according to the human actions, the active tasks, and the environmental stimuli. The overall human-robot architecture (see Figure 4.1) integrates the multimodal interaction module detailed in the previous chapter (*HRI module*), a hierarchical task planner (*Planner*), and the executive system. The latter can be subdivided in two components: the *Attentional Executive System*, that manages behavior allocation and provides top-down regulations, and an *Attentional Behavior-based System*, that collects the allocated sensorimotor process, 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 fused (*Fusion Engine*) by the *HRI module* in order to recognize the human activities and inten-

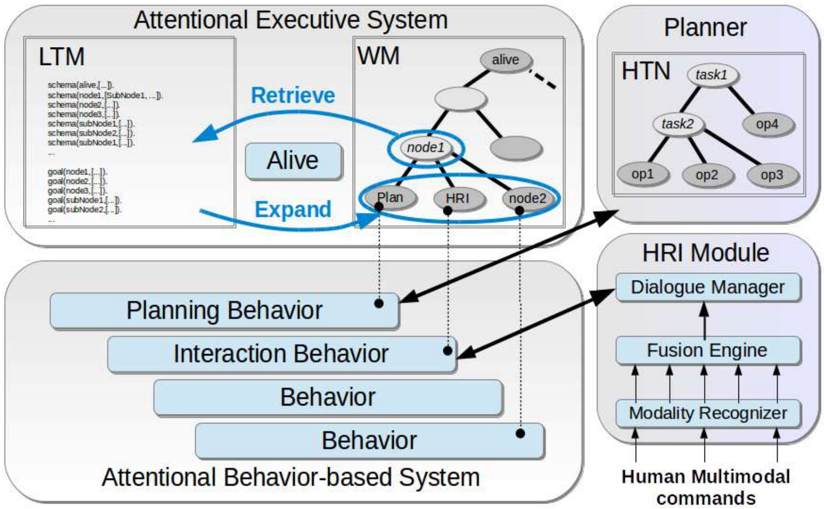

Figure 4.1. System architecture: the *HRI module* permits multimodal interaction between the human and the robotic system; the *Planner* generates plans - represented as *hierarchical task networks* (HTN) - to be executed and monitored by the *Attentional Executive System* providing top-down and bottom-up regulations. The executive control cycle (blue arrows and ovals) is managed by the process *alive* that updates the *Working Memory* (WM) exploiting the behavior schemata stored in the *Long Term Memory* (LTM).

tions. As a final stage of the multimodal interaction process, the *Dialogue Manager* generates a behavior, that can be instantiated and continuously adjusted by the attentional executive system with respect to the environmental and the operative context (see Chapter 3 for details). On the other hand, a task planner (*Planner*) can generate plans of actions, represented as *hierarchical task networks* (HTN) [105], where both the human and the robot may be involved. The integrated planning and execution system will be better detailed in Section 4.3.

4.2 Executive System and 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 [106] HTN (Hierarchical Task Network) framework. A 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 [106].

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 **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 sub-methods $\langle m_1, \ldots, m_n \rangle$ 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 accomplished. 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 $(take(Obj), true, b)$, with $b{=}\langle goto(Obj), pickup(Obj)\rangle$ 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 Chapter 2, 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 on-line executed (Algorithm 1) or off-line planned (HTN planning) and then executed.

The interaction between the planner and the WM is managed by a concrete behavior, called *planning* (see Figure 4.1), 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.

4.3 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 Figure 4.2). 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 towards one of them in order to stimulate the activations of the associated tasks already allocated in the WM.

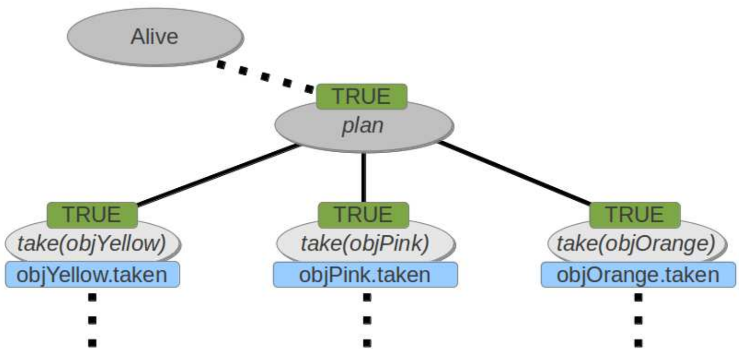

Figure 4.2. A generated plan is allocated in the WM and the associated abstract behaviors are hierarchically expanded by the cognitive control cycle.

4.4 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

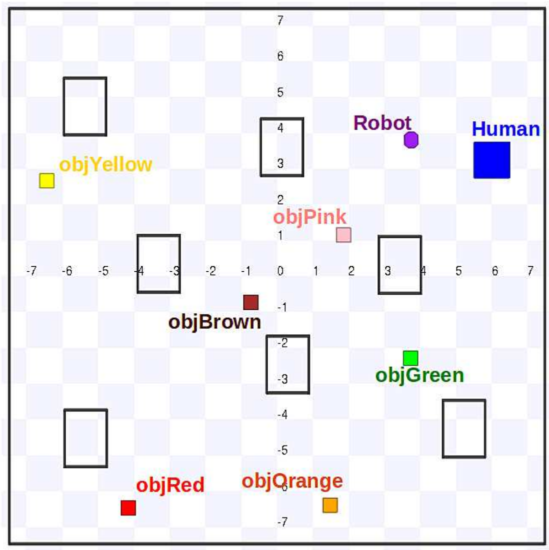

Figure 4.3. Simulated environment: we have colored objects (small cubes), obstacles (rectangles), the human position (large flat square), and the mobile robot (purple).

human has to influence the execution of multiple tasks through attention manipulation.

4.4.1 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 (Figure 4.3). As a robotic platform, we consider a 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 Figure 4.4 we can observe a competition of two

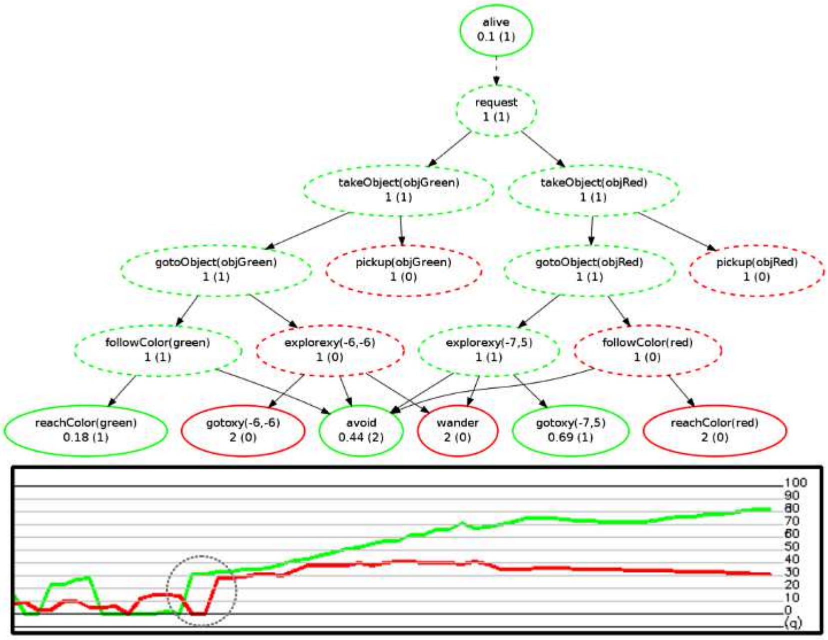

Figure 4.4. Conflicting tasks in the WM (top) and emphasis plots (below) for the two conflicting behaviors *reachColor* and *gotoxy* associated with the abstract behaviors $takeObject(green)$ and $takeObject(red)$ respectively. After the conflict (dotted circle in the plot) the robot heads towards the 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.

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 towards 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.

4.4.2 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*, *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 towards 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 towards 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 seconds 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*; $[1, 10]$ for *goto*, *pointo*, and *reachColor*.

4.4.3 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.

Specifically, we assume that the two concurrent plans depicted in Figure 4.5 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 towards task completion (top-down).

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

- *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).
- *True-negatives:* competing active actions, not executed by the sys-

tem, and not expected to be executed (i.e. actions correctly defeated).

- *False-positives:* executed actions, not expected to be executed (i.e. suboptimal choices).
- *False-negatives:* competing best actions, which are not executed (i.e. missed best actions).

Table 4.1. True/false positive/negative over 10 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

We tested the system with 10 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 Table 4.1 and Table 4.2.

Table 4.1 reports the system performance: the two concurrent plans are executed with an effective selection of the correct actions (7.5 with respect to 8 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 4.2. Measures of performance

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

Table 4.2 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 sub-task 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 [45]). 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%).

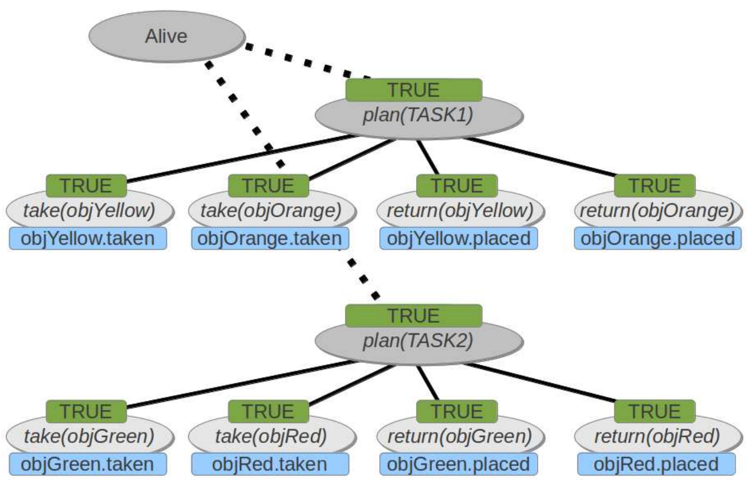

Figure 4.5. Conflicting plans in the working memory. The planned activities are sequenced from left to right.

4.4.4 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 towards 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 towards 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 towards 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:

- *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.

- *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 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 Figure 4.5, 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 10 graduated students in these tests (7 males and 3 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 4.3, 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.

4.4.5 Case Study 3: Interaction with a 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 Figure 4.3) in a realistic 3D environment provided by the robotic simulator *v-rep* (see Figure 4.6). In particular, the simulated robot is a kuka omnirob, equipped with 4 mecanum wheels, a kuka LBR 4+ manipulator, a baxter gripper, 2 laser scans (SICK S300),

Table 4.3. Interventions and execution time (4 and 6 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"

and a RGB-D 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.

A 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 towards some directions in the 3D simulated environment. For instance, in Figure 4.6, the human is indicating the yellow object in the simulated environment.

The adopted multimodal interaction framework is the one described in [123, 93]. 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 Figure 4.3) with respect to ones collected in a more realistic environ-

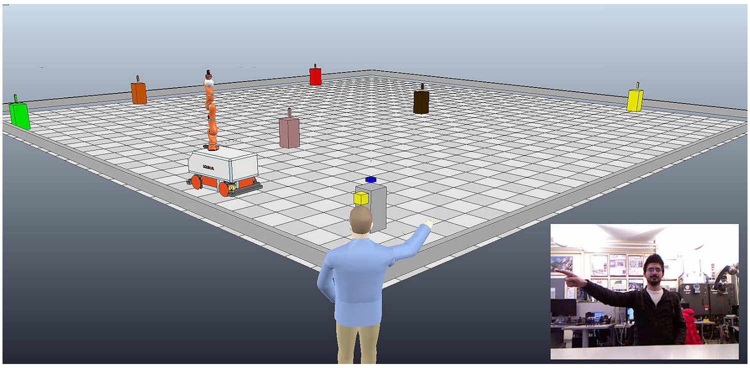

Figure 4.6. Interaction with a simulated robot: the human - illustrated in the small windows at the bottom-right - points towards the simulated yellow objects influencing the attentional state and the behavior of the simulated robot.

ment. 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 4.4).

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.

In these tests, we involved another group of 10 graduated students (6 males and 4 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 towards 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 4.5, 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 4.4, which is structured as follows:

- 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;

- an *interaction assessment* section with questions used to rate the user experience on a 5-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 [151] and human experience in human-robot interaction [31, 54, 133], selecting and adapting the entries with respect to the specificities of our interaction scenario.

Table 4.4. 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 collected results are illustrated in Table 4.6. 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

Table 4.5. 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 4.6. 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 4.7. 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

the first rows of Table 4.7), 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 4.7, 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*.

Chapter 5

Attentional Supervision of Collaborative Plans

In this chapter, we extend the proposed framework exploiting attentional supervision and contention scheduling to combine human-aware planning, plan execution, and natural human-robot interaction. Specifically, in the proposed approach, hierarchical cooperative plans are exploited as top-down attentional guidance for the robotic executive system, which can flexibly orchestrate the task activities while reacting to environmental stimuli and human behaviors. We describe the overall framework discussing some case studies in human-robot collaborative scenarios.

A preliminary version of the work reported in this chapter is published in [35] while the extended version can be found in [41]. This work is farmed in the context of the EU Fp7 SAPHARI Project.

5.1 Integrated System Architecture

In this section, we illustrate the overall architecture (see Figure 5.1) describing its main components along with their interactions. The multi-

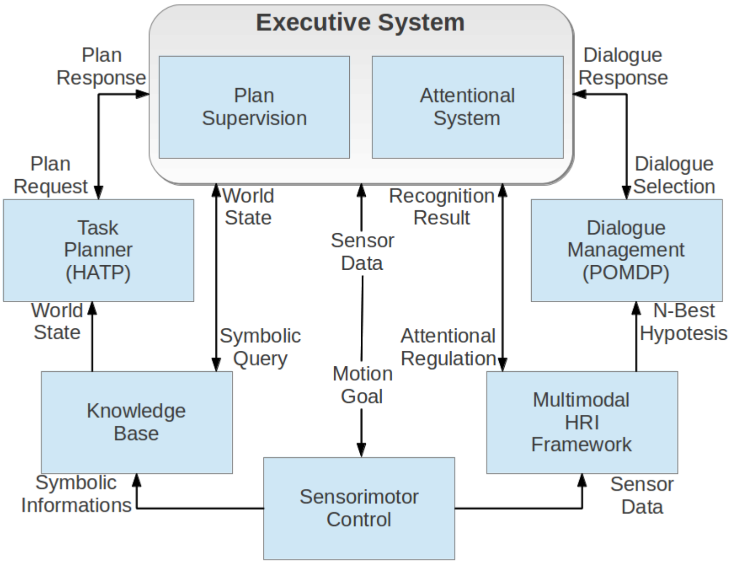

Figure 5.1. The overall human-robot interaction architecture.

modal HRI framework is appointed to recognize the multiple human commands and actions, such as utterances, gaze directions, gestures or body postures, and to provide an interpretation of user intentions according to the dialogue context [123]. The system is endowed with a Human-Aware Task Planner (HATP) [87] which is based on a Hierarchical Task Networks (HTN) and a SHOP-like [105] refinement process. HATP is able to produce hierarchical plans for multi-agent systems (including humans), generating different sequences for each agent. The executive process is managed by two subsystems: the supervision system and the the attentional system. The first one is to interact with the task planner, monitor the plan execution and formulate replanning requests. The second one exploits bottom-up (stimuli-oriented) and top-down (task-oriented) influences to regulate the plan execution and the dialogue policy.

5.2 Human Aware Task Planning

The Human-Aware Task Planner (HATP) [87] is based on a Hierarchical Task Networks (HTN) and is able to produce hierarchical plans for multi-agent systems, including humans. Analogously to SHOP [105], the HTN planning problem is defined as a 3-tuple $\langle g, s_0, D\rangle$, which are respectively, the goal, the initial state, and the planning domain. The latter is defined by the pair (A, M), where A is a finite set of operators and M is a finite set of methods. A method in M is a 4-tuple (m, t, p, b) where m is the name of the method, t is the task/goal, p is a precondition specifying when the method is applicable, and b describes a sequence of operators or methods. The set of operators A is denoted by a STRIPS-like representation. In HATP, each operator A_k^a for an agent a can be associated with a duration D_k^a and a cost function C_k^{ctxt}. Moreover, HATP permits to define specific *social rules* along with a cost for their violation $\langle S_k, P_k^{ctxt}\rangle$. This way, a plan P is associated with a cost: $Cost(P) = \Sigma_{a_i \in P} C_{a_i}^{ctxt} + \Sigma_{s_k \in P} P_{s_k}^{ctxt}$, where a_i is an action of the plan P, s_k is a social rule. By setting a different range of parameters the plans can be tuned to adapt the robot behavior to the desired level of cooperation. Moreover, HATP is able to produce a different stream of actions for each agents, where each stream is a sequential list of actions. Each action has a finite number of precondition links to other actions, which can be part of any stream.

5.3 Cognitive Control and Attention

The attentional system receives the generated plan from the supervision system and selects/regulates the robot activities exploiting bottom-up (stimuli-oriented) and top-down (task-oriented) influences [36]. This process is managed by the cognitive control cycle that continuously updates the internal hierarchical structure (WM), and the set of *behaviors* representing the overall processes involved in the execution (see Figure 5.2) exploiting schemata specifications (LTM). The LTM collects the declarative representations of all the possible behaviors and tasks available to the robot, including the executive schemata associated with the methods M and the operators A defined in the HATP domain (see Section 4.2 for further details about the integration of SHOP-like methods/operators).

In the following, we introduce a more formal description of the WM data structure. The WM represents the executive state of the system as an annotated tree structure, whose nodes $s \in S$ represent processes/behaviors, while the edges represent parental relations among sub-processes/sub-behaviors. Indeed, the nodes $S \in S_c \cup S_a$ are partitioned in *concrete* and *abstract*, where the concrete nodes in S_c represent real sensorimotor processes, while the abstract ones in S_a represent complex behaviors to be hierarchically decomposed. Each node $s \in S$ is denoted by a 6-tuple (s, t, x, q, v, e), where s is the name of a behavior, t is the task, x represents the set of the associated sub-behaviors of s, q represents a releaser, v is a set of state variables representing the executive state of s, while e is a post-condition used to check the success of s. In this context, when allocated for the execution, methods $m \in M$ are represented by associated abstract nodes/behaviors $s_m \in S_a$ in WM, with x list of sub-behaviors associated to b and the releaser q used to monitor the precondition p during the execution. Analogously, each $op \in A$ can be associated with a concrete or abstract nodes, depending on the executive schema represented in the LTM. If the releaser q of an allocated node is satisfied, all its sub-nodes x can be also allocated in the WM; conversely, if a behavior is accomplished or dismissed, this is removed from the WM along with its hierarchical decomposition. In this framework, an allocated behavior is active when its releaser is enabled along with the releasers of all its ancestors. The WM update process is managed by the process *alive* $\in S_c$ which is also the root of the WM tree. For instance, in Figure 5.3, we have a representation of the WM once the task $giveTo$ (abstract behavior) is expanded into *give* and *place* (concrete behaviors), which can be directly executed. Here, $giveTo$ is an abstract node that represents the execution of a method, *give* and *place* are concrete nodes representing running operators, while other nodes represent running/suspended low-level processes which are not represented in the planning domain. It is worth noticing that, not only multiple tasks can be allocated in the WM, but also multiple methods for the same tasks may compete for the actual execution. The orchestration of multiple tasks/activities, possibly in conflicts, is obtained by exploiting attentional processes.

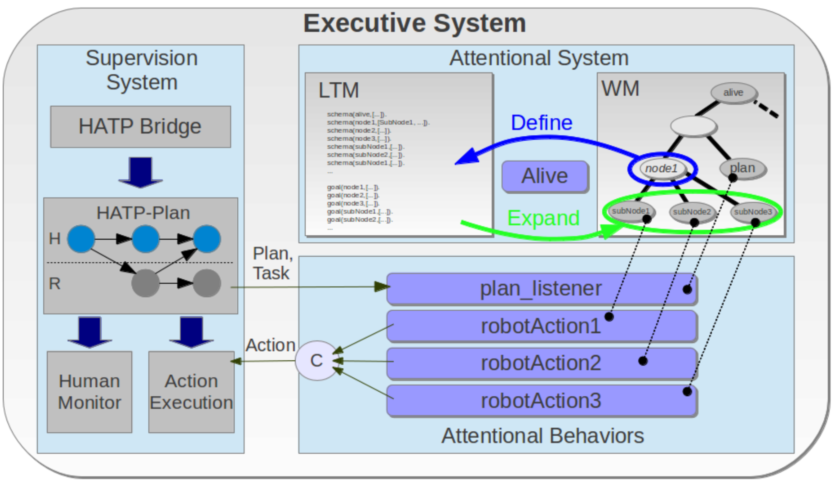

Figure 5.2. The executive system integrates a supervision system and an attentional system. The latter permits a flexible execution of cooperative plans.

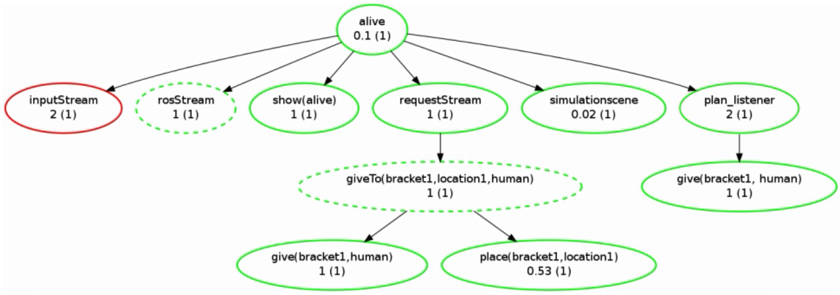

Figure 5.3. Tasks in the working memory: dotted and solid ovals are for abstract and concrete behaviors; green and red ovals represent active and not active processes; n and (m) are for, respectively, the clock frequency (inverse of the *emphasis*) and the associated magnitude.

5.4 Plan Execution and Attentional Regulation

The plan-execution cycle is managed by the interaction of the supervision and the attentional system. Given a task t to be executed, the supervision system invokes HATP to generate a multi-agent plan. This is represented by a set of sequences of actions $\pi = (s_1, .., s_n)$, one for each agent involved in the interaction. In this context, we assume $\pi = (s_R, s_H)$, where s_R is for the robotic activities and s_H is for the interactive human. Once generated, the plan π, together with the associated task t, is received by the attentional system, through the *planListener* behavior (see Figure 5.2) which then allocates in the WM the enabled behaviors for t and π. Here, the task t is hierarchically expanded by the *alive* process into a hierarchy of behaviors, from abstract to concrete, while the plan π is exploited as a guidance for action selection and execution exploiting attentional regulations. At the executive level, primitive human actions are implemented by human monitoring behaviors suitably specified in the LTM.

The plan listening cycle is described by Algorithm 3 and works as follows. Once a new HATP plan π is generated, the behaviors associated with the task t are allocated in the WM, then a monitoring cycle starts and remains active as far as the plan is available and a replanning activity is not invoked. Within this cycle, the next plan action is selected according to the plan sequence; this action is then associated with a corresponding concrete behavior p_a which is top-down enhanced by a suitable constant factor k used to facilitate its execution. Once p_a is allocated in the WM, if p_a is accomplished (i.e. its post condition is satisfied) the next plan action is selected for the execution. Otherwise, if the action α selected by the attentional system is different from the planned one, a plan adjustment procedure is started. When this adjustment is not possible, replan is set to true and a replanning step is then invoked. The plan adjustment strategy checks whether there exists a common ancestor in the WM tree for the selected α and the planned p_a in order to find an alternative decomposition and then modify the plan π accordingly. Notice that that several refinement strategies are also possible in this framework. For instance, following a conservative approach, plan adjustment may be limited to primitive actions only. External plan repair methods, similar to [13, 148], may also be deployed. Otherwise, following a different approach, since the generated plan is here used as a top-down guidance, plan adjustments may be postponed: the executive system may also keep active an inconsistent plan until its attentional disturbance reaches a suitable threshold.

In the proposed plan execution approach, both the generated plan π and the hierarchical decomposition of task t are used for the execution. The actions in π are used to stimulated the attentional system towards the execution of the associated concrete behaviors, which are allocated, activated, and regulated during the expansion of t. This way, not only the execution of different non-conflicting behaviors/tasks may be interleaved with the planned activities, but also alternative expansions of t can be exploited for on-line plan repair actions. Indeed, the task tree allocated in the WM can maintain alternative methods and action primitives in competition/conflicts (e.g. *take* and *receive* are two alternative ways to get an object) in order to permit flexible adaptation of the task depending on the current executive and attentional state (e.g. choosing *take* instead of *receive* if an object is close).

Algorithm 3 Plan listening cycle

procedure PLANLISTENER(plan π,task t)
    add task t to the WM
    select next action p_a from the plan π
    set $Replan$ to $false$
    while ($\neg empty\ \pi$ and $\neg Replan$) **do**
        if (p_a is allocated in the WM) **then**
            if (p_a is accomplished) **then**
                remove p_a from WM and from π
                select next plan action p_a from π
            else
                get most active α action in WM for t
                if (α exists and $\alpha \neq p_a$) **then**
                    set π to $adjust(p_a, \alpha, \pi, t)$
                end if
                if (α does not exist or π is not valid) **then**
                    set $Replan$ to $true$
                end if
            end if
        else
            add p_a to the WM
            emphasize p_a by a constant factor k
        end if
    end while
end procedure

A simple example of the integrated effect of plan-based, task-based, and environmental influences is provided in Figure 5.4. In this case, the task is to get an object ($getObj(bracket_1)$) and the generated plan states that the robot should first reach the table and then take it (HAPT plan in Figure 5.4). However, the task *getObj* is associated with two methods: the robot

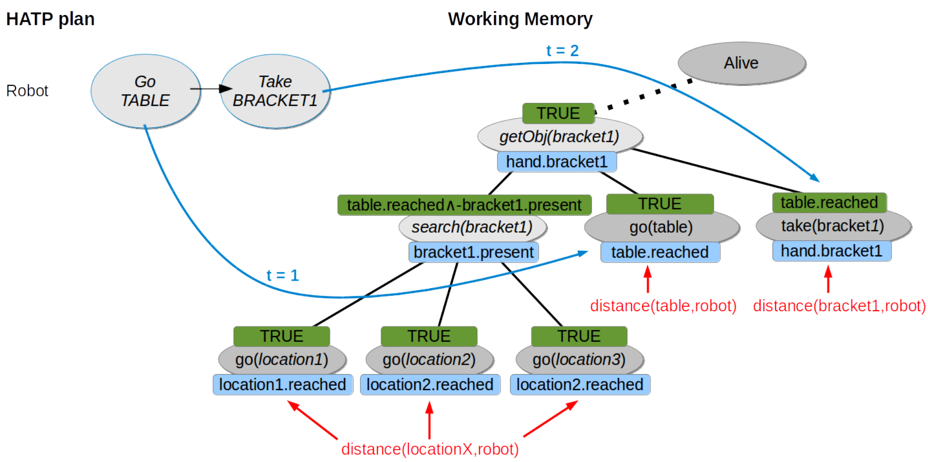

Figure 5.4. Regulations in WM. Green and light blue solids are, respectively, preconditions and goals of tasks/schemata. Top-down regulations are provided by the actions of the HATP plan (blue arrows), while bottom-up regulations are influenced by environmental stimuli (red arrows).

should go towards the table ($go(table)$) and, either take it ($take(bracket_1)$), as planned, or search for it ($search(bracket_1)$), e.g. if the object is not present or object detection fails. At the plan start, the $go(table)$ behavior is allocated in the WM, enabled, and aroused both by the plan (top-down regulation) and by a the table distance (bottom-up regulation). Then, once the table is reached, the action plan is removed and the behavior is disabled, while $take(bracket_1)$ becomes active and can be enhanced by the current planned action and the proximity of the bracket. The emphasis combines these effects and provides an action selection criterion. However, when the *bracket* is not present the bottom-up stimuli does not support the planned action *take* and alternative enabled behaviors may become dominant, in this case $search(bracket_1)$. This alternative execution is then followed by a plan adjustment. This way, in contrast with rigid and sequential activity dispatching, in case of opportunities and unexpected events, the attentional system may retrieve alternative methods from the task definition avoiding a continuous replanning process.

5.5 Case Studies

The integrated system has been tested in a case study inspired by a human-robot co-working scenario where a collaborative robot should assist a human operator during a bracket assembling process [3]. The overall test-bed is inspired by the one proposed in the EU FP7 SAPHARI Project. In this context, we discuss and analyze the system behavior presenting both simulated experiments and a real-world robotic demonstrator.

5.5.1 Simulated Tests

In this section, we illustrate experimental results collected in a simulated scenario. Our aim is to show how the proposed framework permits flexible plan execution when the human behavior diverges from the expected one.

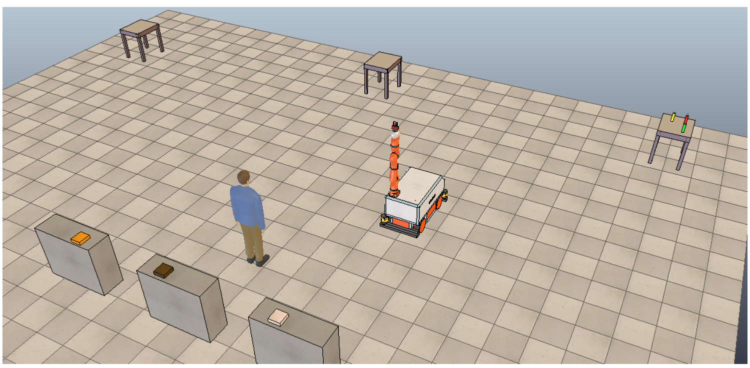

Figure 5.5. Simulated scenario for human-robot cooperation.

Experimental set-up The overall environment is simulated in v-rep interfaced via ROS to our planning and execution framework. We assume a simulated human and simulated robot (kuka omnirob, kuka LBR 4+ manipulator endowed with a baxter gripper, laser scan, and RGB-D camera) that moves within an environment of 15×15 m with a maximum speed of 0.4 m/s. As a computational platform we used a laptop $i5$ 4 core, 4 gb ram. In the simulated environment, we have 3 bracket, positioned in 3 locations, to be installed in 3 panels (see Figure 5.5). Both the robot and the human can *go* towards the predefined locations, or *take*, *place*, *give*, *receive* objects, while only the human can *install* the brackets into the panels. In this context, the task is a sequence of 3 subtasks $install(bracket_1)$, $install(bracket_2)$, $install(bracket_3)$. For each test, we consider an already generated HATP plan where each subtask is to be executed as follows: the robot takes $bracket_i$ from an expected location, navigates towards the human, gives $bracket_i$ to the human, which then installs it in $slot_i$. In order to test on-line flexible adaptation of plan execution, we introduce random changes during plan execution. In particular, since the human behavior is simulated, we can move the operator in and out of the working space to disturb the execution of the cooperative plan. Indeed, if the human moves away before receiving an object, the handover task cannot be executed, hence plan refinement or replanning steps are needed. Additionally, the bracket positions can randomly change in 3 possible locations. Random changes are introduced at the start of each bracket installation subtasks with uniform distributions on the human (in-/out) and the bracket positions respectively. In these tests, we assume that all the manipulation actions are reliable (*take*, *place*, *give*, *receive*), hence the only sources of uncertainty are restricted to the human behavior and the object positions. As for the attentional regulations, the bottom-up frequency associated with the concrete behavior b depends on the distance $dist(target_b) \in [min_b, max_b]$ of the associated target (e.g. distance of *bracket* to activate *take*). Here we assume a very simple setting: within a suitable interval $[min_b, max_b]$ the frequency is increased/decreased proportionally to the reduction/increment of the distance, otherwise, we have a linear decrease of frequency when the stimulus is stable or removed (see [31] for examples of more complex regulations in similar settings). As for top-down influence, the initial value of magnitude for each behavior is $\mu_b = 1$, while for the plan guidance k, we considered two possible setting $k_h = 4$ and $k_l = 0.2$, where the first is considered high and the second low with respect to the default magnitude.

Experimental Results In this setting, the aim is to test the plan execution performance considering successes or failures, time to accomplish the task, repair and replanning episodes during the execution. In order to assess the system performance, we tested 30 times the simulated plan execution in different conditions. First of all, we considered a nominal situation (*baseline*) where the human behaves as expected and the objects are not moved during the execution; in this case no replanning and no repair is needed. In a second experiment, we introduced a randomized situation, where the human moves in and out of the working space, while the objects (brackets) can change their position (*high plan guidance* with k_h). Finally, we repeated the experiments in the randomized setting the top-down plan guidance set to a low value (*low plan guidance* with k_l), in this case the executive system is mainly affected by the task structure in the WM, with a minimal plan influence. The collected results are summarized in Table 5.1 and Table 5.2; we never obtained task failures, therefore these data are not explicitly reported in the tables. In the case of high plan guidance (*hpg*), we can observe that the time to accomplish the task is comparable with respect to the one of the baseline test (*bsl*), where everything works as expected in the plan, indeed the replanning episodes are pretty rare, while the system can on-the-fly find alternative executions and plan adjustments. These plan repairs are usually due to the absence of the human during a planned handover or the absence of an expected object in a planned location. In Table 5.2, in order to show the impact of replanning on the overall performance, we consider the results of the high plan guidance (*hpg*) tests distinguishing between cases with or without replanning episodes. In correspondence to replanning episodes the time performance is significantly worst ($p < .0001$ with a two-tailed t-test). In the last row of Table 5.1, we consider the case of a reduced top-down plan guidance (*lpg*) and compare the performance with respect to the *hpg* tests. Here, as expected, we observe a significant increase of the replanning ($p < .0001$) and plan adjustment episodes (p value $< .04$) that also affects the task execution time ($p < .0001$). On the other hand, even though the overall performance is reduced, the robotic system is able to accomplish the task despite a randomized situation and a weakened plan guidance. This seems to suggest that plan guidance may also be relaxed and modulated when necessary (e.g. more reactive interaction) as a leashing mechanism that affects the overall plan-oriented behavior.

Table 5.1. Experimental results collected in the three settings.

Tests	**time**		**replan**		**refine**	
	avg	std	avg	std	avg	std
hpg	3'53"	0'47"	0.3	0.46	1.25	0.69
lpg	4'23"	0'20"	1.6	0.49	1.8	0.98
bsl	3'49"	0'40"	-	-	-	-

Table 5.2. Experimental results collected with high plan guidance.

Replan				No-Replan			
time		**refine**		**time**		**refine**	
avg	std	avg	std	avg	std	avg	std
4'3"	0'52"	1.33	0.75	3'48"	0'46"	1.21	0.67

5.5.2 Robotic Demonstration

We now describe the system at work in a real-world robotic scenario that implements and extends the simulated setting discussed above. The

Figure 5.6. Experimental scenario: the robot switches from handover to place. A demonstration is available at the following link: http://wpage.unina.it/riccardo.caccavale/media/roman2016.mp4

real set-up extends the simulated one as follows (see Figure 5.6). There are three work locations, each containing a slot and a table that supports a set of objects including a glue bottle and some brackets. The user and the robot must cooperatively install the brackets in the slots, differently from the simulated experiment, in order to install the bracket, the human should first clean the slot, then applying the glue in the slot. In this scenario, a PR2 robot can help the human bringing the appropriate objects. The overall scene is monitored by an OptiTrack motion capture system that provides the positions of the human and the objects, the PR2 is provided with rgdb camera and a laser scan. At the start, the supervision system invokes the HAPT planner in order to obtain a suitable collaborative plan. In this scenario we consider a plan where the *ROBOT* first brings the *GLUE_BOTTLE* and the *BRACKET_1* to the *HUMAN* agent, who is to glue the $SLOT\_1$ position and install the bracket on it. In the following, we describe and discuss some typical situations where the attentional system refines the plan during the execution.

Handover to Take In the planned sequence the human should bring the object to the robot, however, in this case the human remains idle and does not interact as expected. According to the plan, the robot should keep waiting for the human, however, the attentional regulation mechanisms comes here into play to solve the impasse. Indeed, since the target stimulus (human distance) does not change, the bottom-up activations of the *receive* behavior, decrease with time. Therefore, if an alternative method ($take(bracket)$) is enabled by the proximity of a bracket, after some seconds of waiting the associated activations become dominant (the lower $1/e$ is selected) and can be selected as a plan adjustment. In Figure 5.7 (up), we illustrate an excerpt of the WM after the task switch due to the emphasis value and the associated adjusted plan (down).

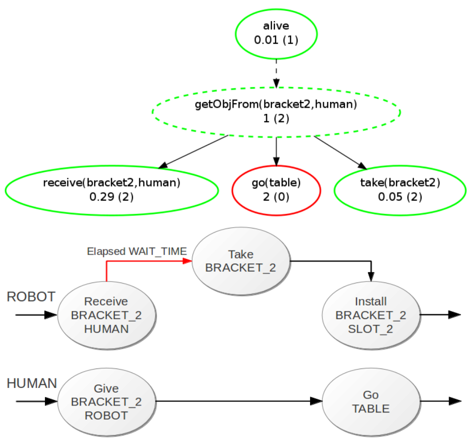

Figure 5.7. Handover to take: (up) if the human does not behave as expected, the *take* behavior becomes dominant, hence it is selected by the attentional system; (down) the associated plan is modified accordingly

Take to Search In a second scenario, the robot should get the bracket and give it to the operator to finalize the installation. In this case, as suggested by the plan, the robot goes towards a target table to take the bracket, however, once arrived the bracket cannot be found. Therefore, the take action cannot be executed, while an alternative method *search* is enabled and becomes dominant. The attentional system can then select the *search* behavior (the activations are illustrated in Figure 5.8, up) and the plan can then suitably modified. The robot can then inspect other locations looking for the bracket.

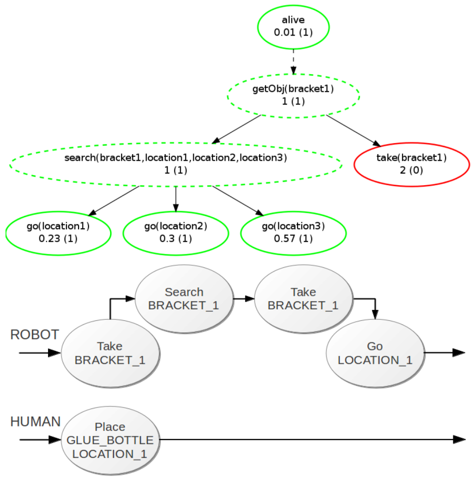

Figure 5.8. Take to Search: (up) the *take* behavior is not enabled because the target object is not available, however, the alternative method is available in the WM hence it is selected by the attentional system; (down) the *search* action is then introduced in the plan.

Handover to Place In this scenario, the human is to obtain the glue bottle ($GLUE\_BOTTLE$) in order to glue the slot ($SLOT\_1$). Following the HATP plan, the robot tries to perform a handover, but the human moves away from the working space during the interaction. Also in this case, the attentional system can solve the impasse without waiting for the human initiative. Indeed, the bottom-up stimulation of the *give* decreases as the robot-human distance increases, while the alternative method *place* is enabled with the associated bottom-up stimuli activated by the table distance. When *place* wins the contention, (see activations in Figure 5.9), the robot can start placing the object on the work location allowing for plan continuation. In this case, plan refinement is also associated with a substitution of a monitored human action from *receive* to *take*.

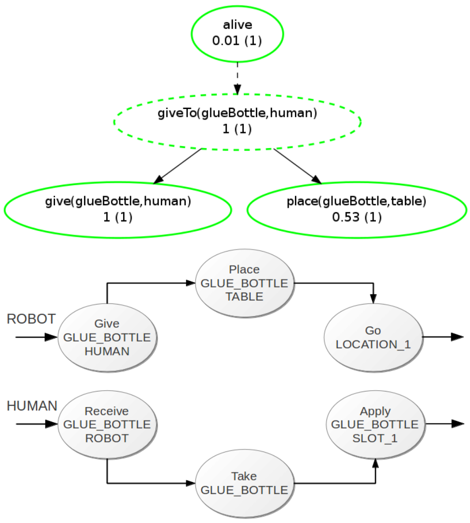

Figure 5.9. Handover to Place: (up) when *place* wins the competition (emphasis $0.53 < 1$) with *give* it is selected and (down) the plan is refined.

5.6 Conclusions

We proposed an integrated system for human-robot cooperation where top-down and bottom-up attentional modulations are used to flexibly execute human aware plans. The framework is to adapt plan execution with respect to the human behavior and the environmental changes reducing replanning activities while enabling a natural and smooth interaction. In this context, the overall execution is managed by an attentional system while a generated cooperative plan is used as a top-down attentional guidance that stimulates the system towards task accomplishment. This approach allows us to combine accessibility (bottom-up influence on enable activities) and facilitation (top-down task/plan based regulations); these mechanisms are here deployed to support flexible activity execution, reactive robotic interventions, and natural human-robot interaction. We described and discussed the proposed system in a human-robot co-working scenario considering both simulated and real-world experiments. In these contexts, we illustrated how plan guidance and attentional regulation allow us to solve decisional impasses and reduce replanning episodes while driving the system towards task and plan accomplishment.

Chapter 6

Attentional Filtering and Adaptive Interfaces

In the previous chapters we have shown the proposed cognitive control framework in different robotic applications including human-robot interaction and flexible plan/task execution. In this chapter we propose a different human-robot interaction setting where the system is employed to regulate the communication between the user and a team of robots. In particular, our aim here is to exploit the attentional regulations to filter the informations provided by the multiple platforms and increase the naturalness and the easiness of the interaction. Specifically, we present a multimodal attentional interface suitable for a human operator that monitors and controls the activities of a team of drones during search and rescue missions. We consider a scenario where the operator is a component of the rescue team, hence not fully dedicated to the robots, but only able to interact with them with sparse and incomplete commands. In this context, an adaptive interface is needed to support the user situation awareness and to enable an effective interaction with the drones. In this work, we propose a multimodal attention-based interface designed for this domain. This framework is to filter the information flow towards the operator selecting and adapting the communication mode according to the context and the human state. We illustrate the features of the adaptive system along with an initial assessment in a simulated scenario.

The work reported in this chapter is exploited in the context of the EU Fp7 SHERPA project [1] and published in [33]. A short version can also be found in [34].

6.1 SHERPA Domain

We consider search and rescue mission scenario where a team of rescuers is supported by a team of aerial robotic platforms. In this context, a special rescuer, involved in the search mission, is endowed with wearable devices to interact with the robots and to monitor their behaviors. Specifically, the human is equipped with a tablet, a headset to vocally communicate with the robots, a *Thalmic Myo Armband* (8 Steel EMG and 9 DOF IMU) for gesture-based interaction, and a band used to monitor his/her status (galvanic skin response, heart rate monitor, skin temperature, GPS, 3-axis accelerometer). This way, the user can interact with the robot through fast and natural multimodal commands involving gesture, voice, touch-based commands, while receiving audio, video, and vibrotactile feedbacks. In particular, the tablet provides a graphical interface that allows the user to monitor the robots status, tasks, paths, and video streams of the associated on-board cameras. In this context, an adaptive user interface should select the information to be provided to the human attention and the communication channel (audio, video, tactile), depending on the saliency of the event along with the human and mission state (see Figure 6.1). Concerning the robotic platforms, we assume quadrotors with standard specification (flight time 25 min., max. airspeed 15 m/s, max. climb rate 8 m/s, etc.) equipped with standard sensors including an on-board camera used by the operator to remotely inspect the environment, and an avalanche transceiver (ARVA) used to detect missing persons under an avalanche.

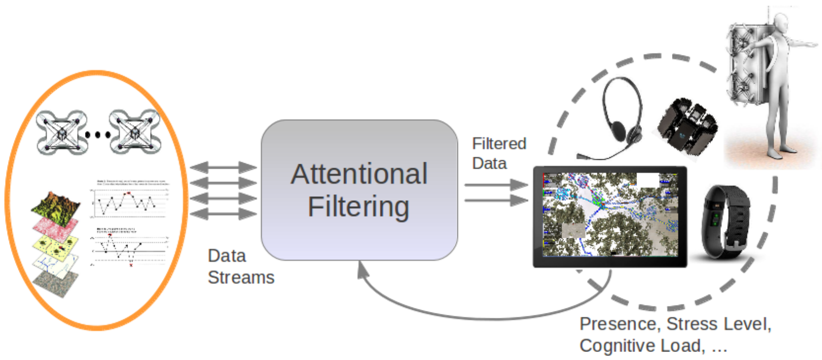

Figure 6.1. Attentional Filtering: the operator is equipped with wearable devices enabling multimodal communication with the drones. The information provided by the robot is filtered by the attentional system that selects relevant data/events defining the associated timing and presentation mode.

6.2 Attentional Filtering

Depending on the current task, the environmental context, and the human emotional/cognitive state, the robotic system should provide the operator with suitable information along with a cognitively adequate interface adapting the associated human-robot interaction schema. For instance, in the crucial phases of the mission, the operator should be provided with focused information and a task-specific interface that minimizes his/her cognitive load, while the SHERPA robots should avoid disturbing interactions by enhancing their proactive and autonomous behavior. The responder is assumed to be in a specific mission context that requires a particular type of information in its interface and an associated interaction mode with the robotic system. The information provided on the interface should assist the responder during the task. The interaction can be explicit or implicit. In the first case, the operator can directly query the interface to get additional details about the mission state. In the second case, the context and the state of the operator directly preshapes the information presented on the interface.

6.2.1 System Design

In order to design the adaptive human-robot interface and the information filtering system, we adopted an attentive paradigm [70, 146] tailored for the peculiarities of our domain. Indeed, in our case the human can be deeply involved in the rescue scenario and can communicate with the drones in a multimodal manner through wearable devices, while his/her level of attention, physical and cognitive stress, can affect the interaction mode. Hence, the system should support the user by selecting salient and task-relevant data, deciding when and how to present the information to the user (timing, frequency, channel, modality), taking into account the human state, the operational context, and the effort of divided attention and attention/task shifting. In order to address all these issues, we propose an approach where the interface is regulated by cognitive control mechanisms supporting the human cognitive and executive processes during the operations. In the context of an adaptive interface, the interaction between top-down (task-oriented) and bottom-up (stimuli-oriented) attentional processes permits to emphasize mission relevant information according to the human and the environmental state, while contention scheduling is to manage conflicts [26] due to limited resources constraints (i.e. working memory, cognitive load, multiple activities, etc.). This framework allows us to uniformly manage divided attention, attentional switch, and multichannel distribution of the information.

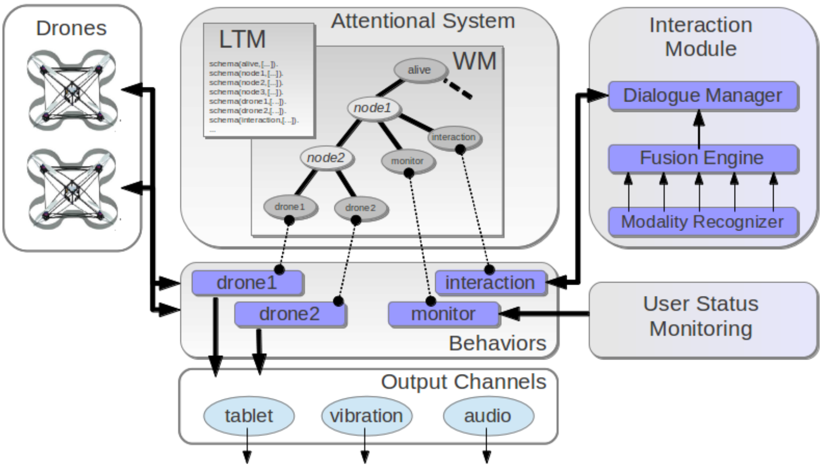

Figure 6.2. System Architecture: the supervisory attentional system manages the adaptive interface providing filtered and task-related data, while reacting to the human commands and status.

6.2.2 System Architecture

The overall system architecture is depicted in Figure 6.2. The interaction between the human and the robots is mediated by a supervisory attentional system that monitors the robot and the human states along with the generated environmental data. The human can communicate with the system in a multimodal manner, deploying gestures, voice, and tablet-based interaction; these multimodal inputs are first fused [123] and then interpreted according to the dialogue context [93]. On the other hand, the supervisory attentional system [36] selects the relevant data and the presentation mode exploiting different channels (audio, video, vibro-tactile). In order to track the mission state, the system WM maintains a hierarchical representation of the active tasks of the agents (robots and humans) involved in the scene (see Figure 6.6). This structure is continuously updated by the cognitive control cycle exploiting behavioral schemata specifications collected in the LTM. The abstract behaviors represent hierarchical activities to be further decomposed ($search(Area_x)$), while concrete behaviors are for real sensorimotor processes ($navigate(wayPoint_y)$). These are denoted by a perceptual schema, a motor schema, a releasing mechanisms, and associated with an activation level represented by an adaptive clock. While the releasing mechanism enables/disables the behavior, the clock regulates the behavior arousal and the frequency of its activations [32]. This frequency is affected by bottom-up and top-down influences. We recall that this frequency is affected by an emphasis value that combine bottom-up and top-down influences (see Sections 2.3 and 2.4). The emphasis value is used to solve contentions among multiple behaviors, indeed, following a winner-take-all approach, the access to mutually exclusive resources is prioritized according to the emphasis value. Notice that, this frequency-based regulation provides us with mechanisms for process selection (attentional filtering), multiple task monitoring (attention allocation and divided attention), task-switching (executive attention). For additional details about this framework we refer the reader to [36, 42].

6.2.3 Adaptive Interface

The supervisory attentional system described so far should track the mission state, monitor the actors' status (robots and humans), and provide the human operator with focused information suitably distributed on the audio (headset), vibro-tactile (armband), and video (tablet interface) channels. The status of the human and the drones are continuously monitored by concrete behaviors in the WM, whose activation level (emphasis) is regulated by the active hierarchical tasks and the salient events/stimuli. For each drone, we consider standardized exploration missions composed of the following subtasks: *takeoff*, *navigate*, *explore*, *inspect*, *return*, *land*. These activities are represented in the WM and further decomposed into lower level processes including the associated monitoring and communication behaviors (see Figure 6.5). In particular, for each drone, we introduce *drone monitoring* behaviors to manage the presentation of the following data: pose, altitude, speed, battery level, intensity of the ARVA signal, current task, video stream of the on-board camera. As for the operator, a *human monitor* behavior inspects the operator activities along with his/her physical and cognitive stress. In particular, GPS and wearable sensors (armband) are used to estimate the human activity (idle, walking, jogging, running, ascending, descending), while, depending on these activities, adaptive thresholds have been defined to detect anomalies and to infer the level of physical/cognitive stress [132]. Since in this work we are interested in presenting and assessing a rich multimodal interaction during the execution/monitoring of multiple tasks, we directly assume that the human has low physical stress (not moving or walking slowly) and high cognitive load (see the next section). The estimated human activity can be directly used to set constraints and preferences on the communication channels. Indeed, if the human is moving fast or under physical stress, we assume that the tablet cannot be directly inspected, hence only vocal or vibration signals can be employed. On the other hand, if the human is idling, the tablet-based interaction becomes an available channel that can be exploited depending on the type and saliency of the information. In the setting considered in this work, we assume that vibration is mainly used to alert the user about important events that can be further detailed either vocally or in the tablet interface. Analogously, the vocal communication is mainly used to gather the user attention on relevant context-switching situations like: change of tasks, victim detection, low-battery, lost connections, malfunctioning, proactive help requests from the drones. Notice that the usage of this channel should be minimized in order to allow the operator to communicate with the other SHERPA actors including the other rescuers involved in the mission.

Contention mechanism In order to regulate the interaction with the user, we define a contention mechanism where the drones compete to acquire communication slots for each channel. In the vocal and vibration channels, we assume only one slot available at each time, while for the tablet we introduce the possibility of parallel communication in m communication slots $s_1, s_2, ..., s_m$ (see Figure 6.5), with m depending on the estimated user cognitive load and physical stress: low stress is associated with a maximum number of available slots, high stress with a minimal one. Since more then one drone can try to acquire a single communication slot, the competition is regulated using the emphasis mechanism described above: the active processes with the higher emphasis acquire the lock of the slots following a N-winner-take-all approach. Notice that the emphasis defines also the frequency of the update, therefore information with high priority is also directly associated with more frequent updates.

Tablet interface The tablet interface is depicted in Figure 6.3. It is structured in two layers: background and foreground. The background displays the map of the search area (a digital elevation map with semantical annotations) and general information about the mission: robot positions, the planned and executed path for each robot, and salient information generated by the drones (e.g. intensity of the ARVA signal). In the foreground, each active robot can provide a status info-boxes where the user can inspect data about the drones. Notice that, while the tablet background can be associated with a multiple target monitoring task [7], each info-box in foreground requires the exclusive access to specific locations, with an additional effort needed to shift attention and elaborate the data [116].

Moreover, the info-box can be displayed in different modes (see Figure 6.4), each associated with a different representation of the status. This representation depends on the relevance of the drone activity. We consider

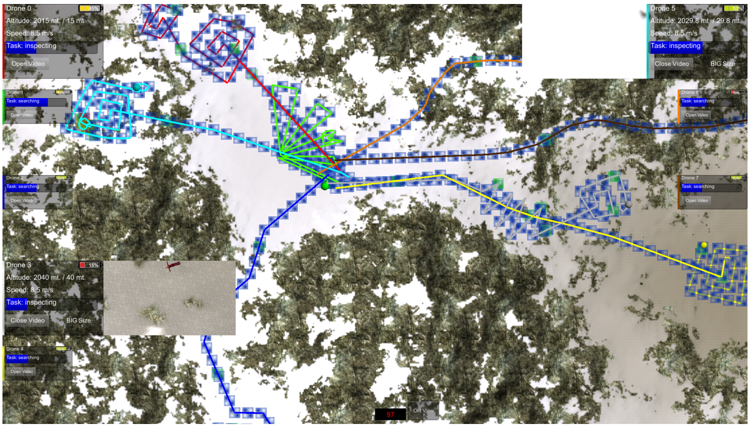

Figure 6.3. Tablet interface. The background illustrates the map of the area with positions, trajectories, and data generated by the robots. Each robot is associated with a color and an info-box in foreground that provides information about the robots status and video streams.

the following three modes:

- *Not relevant:* The info-box is set at the minimum dimension. In this case only the task name, the task progress, and the battery level are displayed. This representation is associated with routinized activities that can be monitored with minimal effort, e.g. when the drone is autonomously navigating towards a search area or during the execution of a predefined search pattern.

- *Weakly relevant:* The info-box is set to a medium dimension and the altitude value is also displayed. This representation is related with not nominal situations, where a human intervention/teleoperation may be needed, e.g. during critical navigation episodes, low battery, malfunctioning, etc..

- *Relevant:* The info-box is set at the maximum dimension (to gather the attention of the user) displaying the complete status. This mode

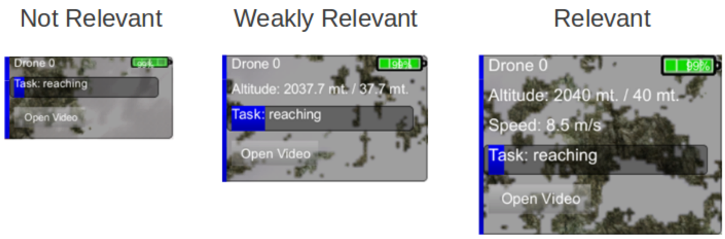

Figure 6.4. Status info-box representation. We consider three different presentation mode associated with not relevant, weakly relevant, and relevant information.

is provided when a careful inspection of the user is needed, e.g. when the ARVA sensor detects the presence of a human.

Events Management We introduce additional cueing mechanisms to draw the attention of the user upon a specific drone or a specific element of a drone info-box. Indeed, when a relevant event occurs during the execution of a task (subtask accomplishment, low-battery, lost connection, navigation problem, malfunctioning, victim detection alert), the related drone can inform the user exploiting either the visual channel, by flashing his status info-box, or by sending signals on the audio and vibro channels.

In the setting considered in this work, we introduce the following simple cueing mechanism. Whenever a drone acquires a slot to communicate new information, both vibro and audio signals are sent to the user to gather his/her attention, while the related info-box starts flashing to allow the attentional shift towards the related info-box. This flashing depends on the frequency of the associated process and continues until the emphasis is higher than a fixed threshold or others drones, with more relevant/urgent data, get the control of the available slots (see the Algorithm 4). In order to facilitate this process and improve the rate of relevant information provided

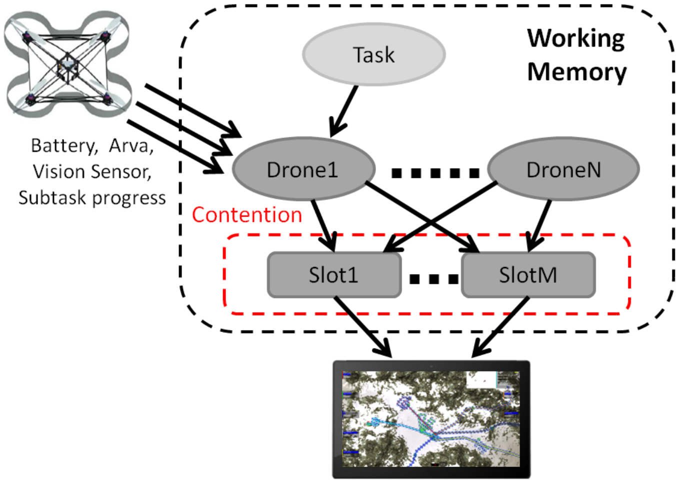

Figure 6.5. Competition among different processes to present the information. Contentions are solved with a N-winner-take-all approach that allows the access to the most emphasized behaviors.

to the user, we introduce additional time constraints for the exclusive lock of the slot and the associated flashing:

- *Reaction time:* The flashing period must persist for a suitable interval t_{react} that allows the user to reorient the attention, perceive and elaborate the information [116, 117]. If other n info-boxes are already flashing when the new one occurs, the new flashing period should be suitably enlarged; here, we assume $t_{react} = \theta_{react} + \delta_{react} \times n$.

- *Attentional blink:* When the information changes on the same info-box, we introduce a no-flashing period t_{blink} in order to signal the new info enabling context switch and elaboration. Empirical evidences on rapid serial visual presentation indicate a temporal suppression on visual processing during a specific interval [120].

These temporal constraints are enforced through frequency modulations, indeed, whenever a new process acquires a slot, it receives an ad-

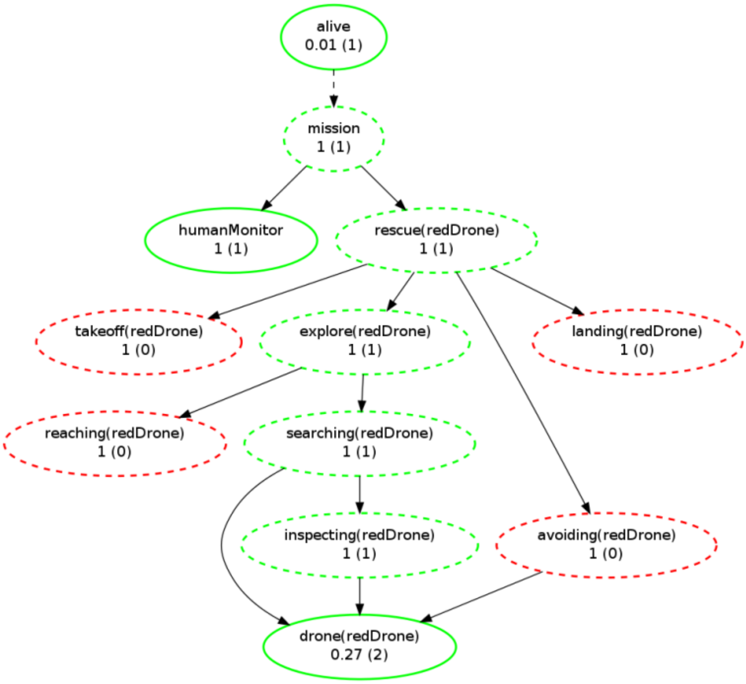

Figure 6.6. Activities of the red drone represented in the WM. Green, red, dotted, and solid ovals are for active, disabled, abstract, and concrete behaviors respectively. The red drone is here inspecting during an exploration subtask, hence its monitoring behavior is emphasized.

ditional emphasis value that prevents task switching during the latency t_{react} (see Figure 6.7). A pseudocode of the overall monitoring cycle associated with a generic drone is provided in Algorithm 4, where emphasis regulation, contentions, cueing mechanisms, and temporal constraints are illustrated. Specifically, when a monitor acquires a new slot, flashing starts after t_{blink} sustained by an enhanced emphasis during t_{react} and reinforced by multimodal cueing signals; when a slot is released, flashing ends.

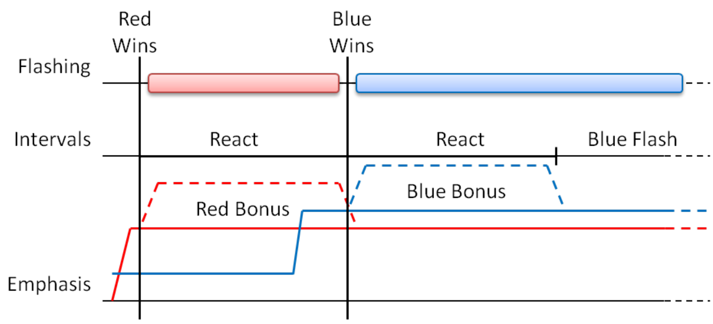

Figure 6.7. Temporal constraints for task switching. The overall latency represents the minimal time to react.

6.3 Case Study

In this section, we present a case study used to assess the adaptive interface described so far. In particular, we consider a dual-task scenario [149] where the operator has to monitor a group of drones involved in a search and rescue mission, while performing a secondary distracting task. This task has been introduced to simulate the operator involvement in the search activities, as envisioned in the SHERPA domain. The two tasks are set as follows. In the primary *interface-monitoring* task (IM) the user has to monitor 8 drones during their search activities, attending to salient events (navigation problems, possible missing detection, low battery alerts, etc.) and looking for victims by inspecting the video stream provided by the on-board cameras of the drones. In a secondary task, the user should perform a *word-counting* (WC) activity, i.e. the user should count the letters contained in each word of a list; this operation is executed on a different computer. Moreover, as a situation awareness test, whenever an event is signaled, the tester is to check the interface and declare the reason of the signal (e.g. low battery, task switch, etc.). In order to assess the effectiveness of the adaptive interface, we set up two instances of the system. In the first one (*adaptive*), we deploy the adaptive version of

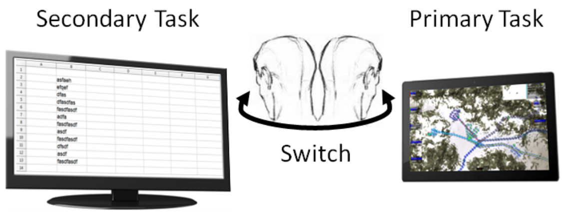

Figure 6.8. In the dual-task scenario the user have to switch between WC and IM tasks trying to maximize the number of words correctly counted and the number of victims/salient events detected.

the interface where the number of communication slots is reduced to one (maximum cognitive load), therefore only one drone is able to signal an event per latency. In a second instance (*non-adaptive*), we define a statical interface where the size of all the info-boxes are fixed to the maximum one and multiple parallel communications are allowed. For the tests, we involved a group of 16 participants (10 males), all undergraduate students of an engineering school in our university. In this setting, each tester is to perform two sessions of the dual-task test using both the adaptive and the non-adaptive interface. In order to decouple the experimental results from the users practice, we defined three search and rescue mission scenarios: one for fast training, and the other two for testing the adaptive and the non adaptive case. In addition, these two tests have been alternated to prevent a learning effect. In these two testing scenarios, we assume the tasks of the drones already loaded in the WM along with the associate subtasks, pre-planned search paths, defined positions of victims, and simulated logs of salient events. Testers are endowed with the tablet, headset, armbands and are asked to find a maximum number of victims while executing the WC secondary task. Testers are not aware that we have 5 missing people for each scenario and are not instructed about priorities between the two tasks. During the test, for each audio/vibro alert received, the user is instructed to verbally classify the type of event (task change, battery, obstacles, etc.).

Each test session lasts about 7 minutes.

Table 6.1. Measures of performance.

	Events Occurred		**Events Missed**		**Victim Missed**	
	avg	std	avg	std	avg	std
no-Adapt	19.22	1.69	3.67	3	0.33	0.25
Adapt	11.44	2.28	1.22	0.69	0.11	0.11
p-value	< 0.0001		0.0034		0.0031	

Table 6.1 provides a summary of the results collected with the two tested modalities. The first entry presents the salient events (average and standard deviation) actually communicated by the interface to the user via visual/audio signals, while the other entries represent, in order, events and victims available on the interface, but missed by the users. These results show that, despite the average number of communicated events decrease by 40.6% in the adaptive mode, the average number of missed victims and events decrease, suggesting that the adaptive interface, not only does not affect victim/events detection, but also slightly improves it with a significant reduction of the information flow. In Table 6.2 we

Table 6.2. Rate of missing events.

no-Adaptive		**Adaptive**		**Improvement**	
avg	std	avg	std	avg	std
19.4%	1%	10.6%	0.4%	8.9%	0.4%

present the average and the standard deviation of missing events per trial (missing events over total events) expressed in percentage. Here, we can observe that, in our tests, the proposed adaptive interface decreases the probability of missing a single event by 8.9%.

Algorithm 4 Drone monitoring cycle

```
procedure DRONEMONITOR(d)
    while drone is active do
        update interface
        try to get a slot
        if slot acquired then
            if just acquired then
                enhance the emphasis
                if update after flashing info-box then
                    set t_blink
                    not flashing
                end if
                set t_react
                if slot audio/video acquired then
                    send audio/vibro signal
                end if
            end if
            if time t_blink elapsed then
                start flashing
            end if
            if time t_react elapsed then
                reduce the emphasis
            end if
        else
            not flashing
        end if
        set next period p
        wait(p)
    end while
end procedure
```

Chapter 7

Learning Tasks from Demonstrations

In this chapter, we present a framework that allows a robot manipulator to learn structured tasks from human demonstrations and to execute them in cooperation with a human co-worker. The proposed system combines physical human-robot interaction with attentional supervision in order to support kinesthetic teaching, incremental learning, and cooperative execution of hierarchically structured tasks. We describe the overall system architecture and detail how cooperative tasks are learned and executed. The proposed approach is evaluated in a human-robot co-working scenario, showing that the robot is effectively able to rapidly learn and flexibly execute structured tasks.

The work reported in this chapter is developed in cooperation with the TUM university of Munich (Germany). A preliminary version of this work is published in [43, 38], while the extended version is currently under review.

7.1 Motivation and Background

The integration of robotic devices in human populated environments requires the ability of the robot to continuously learn novel tasks and to adapt their execution to the human intentions and behaviors. In a human-dwelled environment, indeed, the robot will be asked to execute incrementally complex activities both autonomously or in cooperation with human co-workers. In this scenario, the interaction with the human should be natural and fluent during both task execution and task learning. In this work, we propose a framework which allows natural human-robot interaction along with incremental teaching and autonomous or cooperative execution of structured tasks.

A structured task, like preparing a certain recipe, can be hierarchically decomposed in different subtasks involving multiple primitive actions and manipulated objects. Actions have to be performed in a coherent manner, meaning that the actions have to be executed on certain objects with a particular order. For example, to pour water in a cup, the robot has to take the bottle, reach the cup, and then pour the liquid. In order to make a robot able to learn and execute structured tasks, our approach integrates multimodal interaction [123], attentional supervision [110, 46, 36, 42], and kinesthetic teaching [90, 125]. In our framework, the human operator can naturally interact with the robot using gestures, voice, and physical guidance, while a supervisory attentional system [110, 46] continuously monitors and tracks the human-robot interactive activities during both training and execution sessions.

Attentional mechanisms that are suitable for human-robot task teaching have been explored in the robotic literature, mainly in the context of visual attention [103, 28, 24]; in contrast, in this work we focus on attentional supervision and physical interaction. Namely, in course of a kinesthetic teaching session, the human can physically interact with the robot to demonstrate the execution of the actions, while the supervisory system is exploited to interpret the human guidance in the context of a structured task. In this setting, the supervisory attentional system supports implicit non-verbal communication and permits to track the human demonstration at different levels of abstraction (tasks, sub-tasks, actions and motions primitives).

More specifically, human demonstrations are automatically segmented into basic movements, or motion primitives, exploiting contextual information (e.g. the relative distance between the robot, the objects to manipulate, explicit human commands, etc.). The generated primitives are simultaneously monitored by the attentional system, which relates them to the associated task structure exploiting top-down (task-based) and bottom-up (stimuli-driven) attentional mechanisms. These mechanisms enable also a natural interaction of the robot with the teacher, which can exploit attention manipulation (object and verbal cueing, pointing gestures, etc.) to facilitate the learning process [109]. Notice also that in the proposed framework, action segmentation, annotation, and (task-based) contextual interpretation are one-shot and automatic, hence they do not require any manual post-processing of the collected data.

The rest of the chapter is organized as follows. Section 7.2 presents and discusses related work. The proposed architecture for multimodal teaching/execution is detailed in Section 7.3. Section 7.4 describes how structured tasks are learned and executed using the proposed architecture. Experiments in a real word scenario are presented in Section 7.5. Finally, Section 7.5.2 states the conclusions and proposes further extensions.

7.2 Related Works

Kinesthetic teaching is a natural and intuitive way to teach elementary robotic motions [90, 125]. The goal of kinesthetic teaching is to physically guide the robot to show the desired behavior. In this setting, collected demonstrations are used to learn and reproduce the elementary motions. On the other hand, structured robotic tasks consist of several elementary motions, which have to be sequenced and executed in a coherent manner. The work in [82, 138] focus on segmenting demonstrated movements in order to create a dictionary of basic motions, which can then be combined in more complex behaviors. In [82], the authors propose motion graphs to combine the learned primitives, but object manipulation and goal-oriented activities are not considered.

The problem of deciding the next motion to execute can be considered as a classification problem. The approach in [112] uses nearest neighbor classification to determine the motion to execute. In [96], a graph is used to represent transitions between elementary motions. A classifier associated with each node in the graph determines when a transition occurs, i.e., when a motion is finished and the robot can execute the next one. These approaches permit to learn and reproduce complex robotic tasks from human demonstrations, but they do not consider the possibility of executing learned tasks in cooperation with the human.

Alternative works have focused on the problem of learning high-level task representations from human observations [10]. In [139, 154], sequential constraints (like reaching an object and then grasping it) are used to determine a set of semantic rules that elementary motions (or symbols) have to satisfy; these rules determine the sequence of actions to perform. Semantic rules are also used by [119] to learn, recognize and reproduce human activities from video sequences. These recognized activities are then matched with a set of pre-programmed motion primitives and executed by the robot. The problem of task learning from human activity observations is also faced by [52]. Here, the human demonstration is used to generate a robot-independent task structure associated with robot-specific primitives. Also in this case, learning is performed at the task level, while robot primitives are predefined. A similar approach is proposed by [109], where the task structure is generated from a set of available robotic primitives (skills). In this case, during the user demonstration, the robotic system monitors the skills activations and extracts the graph of the executed actions. In other works [95, 80], probabilistic task representations are learned from human observations and then used to recognize the current activity and infer future human actions. In these settings, human activity anticipation can be used by the robot to generate the right response to the human behavior [80].

Aforementioned approaches mainly focus on task learning from a set of pre-programmed motion primitives, in contrast we are interested in learning both primitives and the associated task structure. Indeed, we propose a framework that enables incremental task teaching and cooperative task execution at different levels of abstraction. Moreover, we are interested in natural and smooth human-robot interaction that supports cooperative task execution and incremental adaptation.

In this respect, related to our work, in [2] the users demonstration is exploited to build a semantic representation of the task. The framework integrates visual perception learning and imitation learning to learn the sequence of actions and the primitive skills. This work is mainly focused on sequential tasks. Alternatively, in [109] the teacher can use simple verbal cues to facilitate the learning process. In particular, the authors propose explicit verbal instructions to bias the learner attention to relevant aspects of the demonstration, but an attentional framework is not deployed. Differently from this approach, we propose to deploy a supervisory attentional system that enables more complex attention-base interaction (verbal, non-verbal, explicit, implicit, etc.) during both the teaching and the execution phase. Social attentional mechanisms for non-verbal task teaching are proposed and investigated by [28]. In this case, the authors mainly focus on visual attention and gaze direction. In particular, they show the effectiveness of spatial scaffolding cues during interactive task demonstration. Visual attention mechanisms for robot learning are also proposed by [103, 24, 19]. In contrast with these works, we focus on executive attention and cognitive control mechanisms supporting kinesthetic task teaching. Cognitive control frameworks for robotic system have been proposed [77, 42], but not in a learning-by-demonstration context.

7.3 Robotic Arm Interface

The overall system architecture is depicted in Figure 7.1. The human can interact with the robot in a multimodal manner with gestures, speech, and physical guidance during both task execution and kinesthetic teaching sessions. The attentional system supervises both the human and the robot activities (*Attentional Behavior-based System*) and manages high-level tasks monitoring and execution (*Attentional Executive System*). On the other hand, the *Robot Manager* is responsible for low-level tasks supervision, execution and learning. These components will be better detailed below.

7.3.1 Robot Manager

The Robot Manager (RM) handles low-level aspects of the human-robot interaction and it is responsible for a correct task execution. In particular, RM is responsible for: *i)* smooth transition between teaching and execution modes; *ii)* demonstrated task segmentation into basic motion primitives [71]; *iii)* scene monitoring (objects classification and tracking); and *iv)* robot state monitoring (robot-objects distance, motion primitives learned or executed). Task teaching is performed by means of kinesthetic teaching [90]. In this work, we use the gravity compensation control to make the robot ideally massless, guaranteeing an easy and safe physical guidance. High level tasks are represented as a set of point-to-point mo-

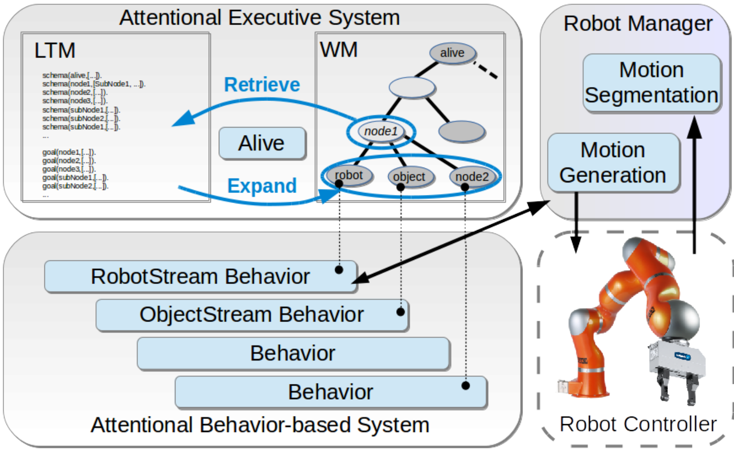

Figure 7.1. The overall architecture for teaching and execution. The attentional system supervises task execution and learning, while the Robot Manager enables activity segmentation, kinesthetic teaching, primitive action learning and execution. The attentional system manages the execution of high-level tasks (*Attentional Executive System*) and low-level sensorimotor processes (*Attentional Behavior-based System*).

tion primitives (reaching and manipulating objects), learned from human demonstrations. RM adopts stable dynamical systems (DS) to compactly represent motion primitives and to generate motor commands in the execution phase. DS are well-suited for point-to-point motion generation since they are guaranteed to converge towards a given target, and they can rapidly adapt to external perturbations, like changes in the initial/target location and unforeseen obstacles [126, 127].

7.3.2 Attentional System Deployment

The attentional system proposed in this thesis provides the cognitive control mechanisms needed to flexibly orchestrate the execution of complex tasks and to monitor the human activities. In this case, the attentional framework is exploited to support both task execution and teaching. In our human-robot interaction setting, the attentional system exploits hierarchical task representations to supervise and regulate the robot actions, while interacting with the human.

More Specifically, we deploy the attentional regulation to select, among the multiple conflicting structured tasks allocated in WM, the activities that are involved in the learning-by-demonstration session. In this case (analogously to previous chapters) we exploit top-down and bottom-up regulations to selects the behaviors according to the environment and the task structure. The selection mechanism along with the teaching and execution processes will be better explained in the following sections.

7.4 Kinesthetic Teaching of Structured Tasks

The framework proposed in this chapter supports human-robot interaction during both task demonstration and task execution. In order to enable natural interaction and incremental task learning, the system can anytime switch between teaching and execution. The teaching phase can start from the human or the robot initiative. In the first case, the human can explicitly switch to a demonstration session through a command (either vocal and/or gestural) and directly show the execution of a task. Otherwise, in the second case, the robot can wait for the human assistance when not able to execute an activity. This happens when a task under execution is not linked to concrete sensorimotor behaviors. In this case, the system waits for the user guidance in order to learn how to perform the missing subtasks.

During the teaching phase, the human can physically guide the robot in order to demonstrate the correct execution of the task. This kinesthetic teaching session is supervised by the attentional system, which has to connect the segmented training motions to the related tasks and subtasks. The attentional system tracks and monitors both the human and

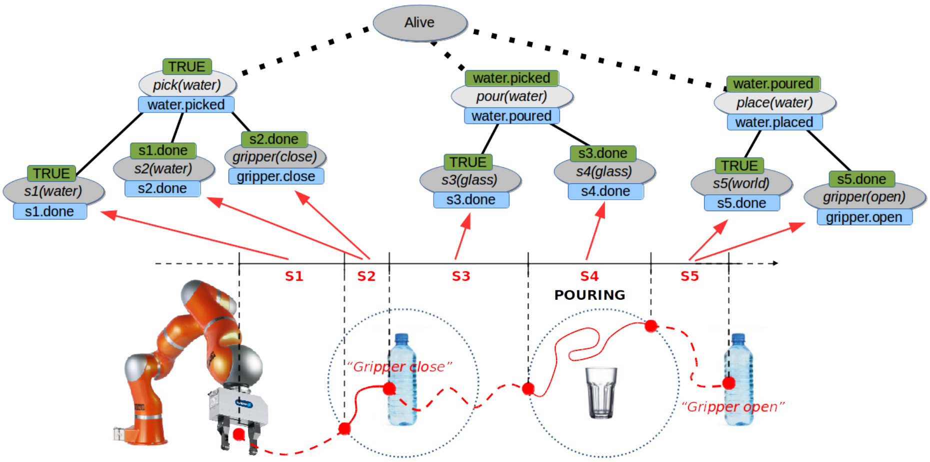

Figure 7.2. Action segmentation and hierarchical task decomposition during kinesthetic teaching of a pouring task. The robot has to pick-up the bottle (*pick(water)*), reach the glass, pour the water (*pour(water)*) and place the bottle (*place(water)*). The Robot Manager (down) performs action segmentation (S_1, S_2, ..., S_5) and learns the associated motion primitives, while the attentional system (up) connects the generated segments and primitives to the task structure ($s1(water)$, $s2(water)$, and $gripper(close)$ connected to $pick(water)$; $s3(glass)$ and $s4(glass)$ connected to $pour(water)$, etc.). The green and blue labels represent, respectively, releasers and post-conditions.

the robot task execution. This way, the low-level robotic actions taught by the user through kinesthetic teaching are labeled by the higher level tasks/sub-tasks managed by the attentional system. For instance, Figure 7.2 illustrates the action segmentation of a water-pouring task along with the associated hierarchical task decomposition. During the teaching mode, the RM provides action segmentation and motion primitive learning, while the attentional system monitors the subtasks to be fulfilled (*pick(water)*, *pour(water)* and *place(water)*). When a new segment is recognized by the system (S_1, S_2, ..., S_5), a new node in the tree is generated ($S_1(water)$, $S_2(water)$, ..., $S_5(word)$, $gripper(open)$) and linked to the most emphasized subtask.

During the demonstration, the human can also facilitate the learning process by providing additional verbal and non-verbal cues to the robot (such as object handling, pointing, vocal commands, etc.). These cues can affect the attentional state of the robot, hence they can influence task/subtask contentions and segments associations. Moreover, the human can always inspect the result of a training session by invoking the repetition of learned tasks and subtasks. Indeed, if the learned activities are not satisfactory, task demonstrations can be repeated. In the rest of the section we detail the overall learning process.

7.4.1 Action Segmentation

The demonstrated task has to be segmented into elementary actions, or motion primitives, that the robot executes. An effective segmentation strategy has to be fast enough to work in real-time, consistent across different demonstrations of the same task, and complete, meaning that the generated segments represent the entire task. In [82, 138] effective strategies are proposed for human motion segmentation into atomic motion units. The data stream is split into smaller units of fixed length [138] or using a moving window of fixed size [82]. Hidden Markov models are used to recognize and reproduce the motion units. A popular approach [59, 119] suggests to segment an action stream looking at the zeros in the joint velocities. Velocities smaller than a given threshold value are considered as zero. These approaches are effective in segmenting free human motions, but they do not provide a matching between action segments and objects in the scene. The relation between objects in the scene and actions to perform is considered in [147], where object distances are exploited to split the demonstrated task into action segments. The approach can effectively generate basic actions with associated objects, but it requires a library of predefined object-action complexes [152] to reproduce the segmented task with a real robot.

In this work, we propose a simple and effective action segmentation mechanism, which is based on object proximity and explicit human commands. Following the approach by [147], each object in the environment is associated with a proximity area, i.e., a sphere of radius r around each object (we set $r = 120\,\mathrm{mm}$). When the end-effector of the robot enters or leaves the proximity area of an object, a new segment is generated. Analogously, when a human command (e.g. open/close gripper) is executed a new low-level action is created. The attentional system can then automatically connect the generated action segments to the task structure (see Figure 7.2,), while the Robot Manager uses the robot's trajectories to learn a motion primitive for each action segment. Human commands are also included in the task structure, in order to control the gripper when the robot executes the task. We distinguish between two classes of actions:

- Near-Object-Action (NOA): the action is segmented inside the proximity area of an object. In this case, we exploit Dynamic Movement Primitives (DMP) [71] to compute a robust approximation of the observed trajectory in order to accurately reproduce the motion.

- Far-Object-Action (FOA): the action is segmented out of the proximity area of any object. In this case, only the end-point of the observed trajectory is considered. The action is then reproduced with a point-to-point motion, generated with a linear dynamical system. This way, the robot reaches the proximity area always with the same pose, and executes NOA actions starting from a state which is consistent with the demonstration.

The proposed segmentation mechanism allows the system to reproduce complex actions involving two or more objects. For example, the pouring action (NOA) illustrated in Figure 7.3 has been trained with high accuracy and associated with the $pour(water)$ primitive behavior within the abstract task of pouring.

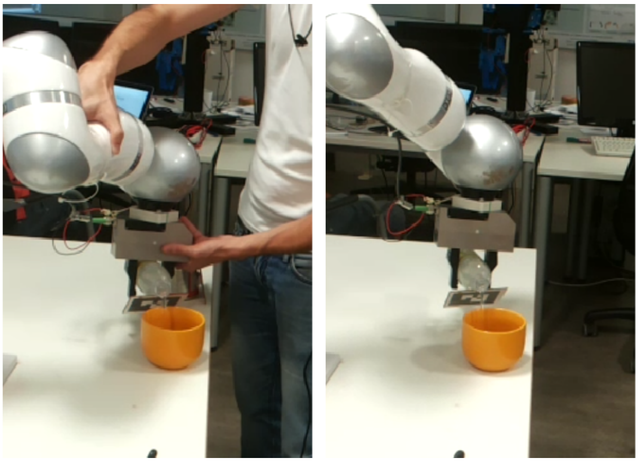

Figure 7.3. Teaching and execution of the *pouring* action (NOA). In the teaching phase (left) the user drives the robot near the cup and pour water, while in the execution phase (right) the robot reproduces the movement.

7.4.2 Task Learning

In this section, we illustrate how the generated segments and motion primitives are connected to high-level task structures. This process is managed by the attentional system while monitoring the human demonstration. When a teaching phase starts, we assume that the task to be learned is already hierarchically decomposed and allocated in the WM. Notice that multiple tasks and subtasks can be allocated in the WM, hence the system has to manage conflicting interpretations of the learning task. During the teaching process, the attentional system has to connect the concrete nodes in the WM to the segments and motion primitives generated by the RM. In order to describe this process, we consider the demonstration of a water-pouring task (see Figure 7.4). This task is hierarchically decomposed in the *take-water* and *pour-water* subtasks (frame $t1$), which are denoted in

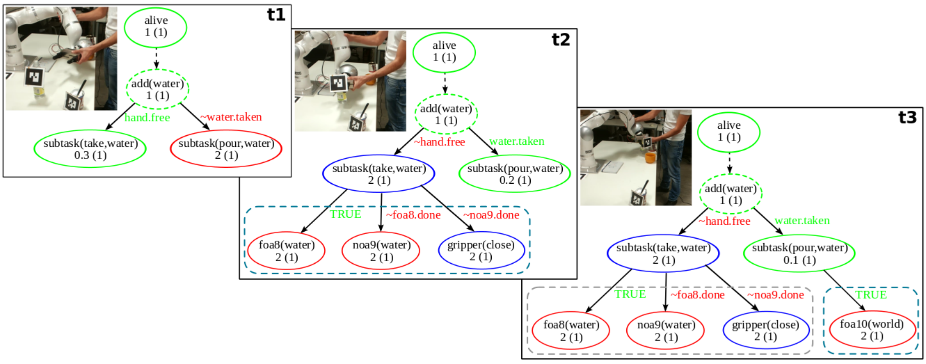

Figure 7.4. Representation of the WM update during the *pouring* task. The system starts from a simple structure for the *add*(*water*) task ($t1$). During the user demonstration new segments are added to the *take-water* subtask ($t2$) along with their releaser (labels on the arrows). When the new *pour-water* subtask is selected ($t3$) a new FOA is linked with a *true* releaser. Here, green and red ovals represent enabled and disabled behaviors (satisfied and unsatisfied releasers), blue ovals are for accomplished behaviors (satisfied postconditions), dotted ovals are for abstract behaviors. For each behavioral node, the values n (m) represent the inverse of emphasis $1/e_b$ (i.e. the activation period) and magnitude μ_b (top-down influence) respectively.

the attentional system LTM by the following schemata:

$$\begin{aligned}
\mathbf{schema}(&add(Obj),\\
&\langle(subtask(take,Obj),hand.free),\\
&(subtask(pour,Obj),Obj.taken)\rangle,\\
&Obj.used)
\end{aligned} \tag{7.1}$$

$$\mathbf{schema}(subtask(take,Obj),\langle\,\rangle,Obj.taken)$$

$$\mathbf{schema}(subtask(pour,Obj),\langle\,\rangle,Obj.used)$$

Here, the pick-and-pour task can be instantiated with different objects (Obj). The subtask *take* is enabled when the hand is free (releaser $hand.free$)

and associated with the *Obj.taken* post-condition, while the *pour* subtask is enabled when the object is taken (releaser *Obj.taken*) and related to the *Obj.used* post-condition.

In order to be executed, this task has to be allocated in the WM. However, the two subtasks (*pour* and *take*) are not linked to concrete sensorimotor processes, which are provided by the motion primitives segmented and learned through the kinesthetic teaching process. Since each subtask is a concrete WM node, it is associated with an activation level, which is (bottom-up) affected by the proximity of the objects in the scene (see equation (2.1)) and (top-down) modulated by the overall tasks allocated and enabled in the WM. Therefore, during the human demonstration, the attentional system enhances the activations of the subtasks which are accessible (i.e. closer to the associated target objects) and task relevant (i.e. top-down stimulated through the task structure). These activation values are then used to link the concrete subtasks to the generated segments and motion primitives, as described in Algorithm 5.

Algorithm 5 Allocation of a new segment in the task hierarchy.

```
while true do
    if exists a new segment s then
        get the most emphasized subtask sub from WM
        add segment s to subtask sub in WM
        add segment s to subtask sub in LTM
        if sub is a new and s is FOA then
            set releaser of s as true
        else
            get post-condition p of the previous segment
            set releaser of s to p
        end if
    end if
end while
```

In particular, when a new segment is created by the Robot Manager (line 2), all the enabled subtasks of the WM compete to add the segment as a new child node (line 3). In our framework, this competition is managed by a winner-takes-all approach where the most emphasized subtask acquires the new segment and the WM is updated accordingly (lines 4,5) (see also Figure 7.4, frame $t2$, where the linked segments are indicated by the dotted line). When a new segment is linked, we have to define its releaser and post-condition (lines 6-12). The releaser is always enabled ($true$) if a FOA segment is added to a subtask with no other child nodes (lines 6,7). Otherwise, in the case of NOA segments, the execution of the motion primitives has to be chained, hence the post-conditions of the previous segment is exploited as the releaser of the current one (lines 9,10). Notice that the chaining constraint is introduced for segments belonging to the same subtask and only for the motion primitives annotated by NOA segments, which require the fixed starting point provided by the previous segment. On the other hand, if the new subtask starts with a FOA segment, we can keep it decoupled from the previous subtask, enabling reusability and flexible execution of the associated subtask. Indeed, at the execution time, all the enabled segments of the WM compete to acquire the control of the robotic platform. Hence, multiple independent tasks and subtasks can be executed in a flexible manner, diverging from the sequence shown during the demonstration.

The overall method is exemplified in Figure 7.4. Once the user drives the robot to the bottle and grasp it, the system generates 3 new segments: $foa8(water)$ when the robot enters the objects area, $noa9(water)$ and *gripper(close)* when the bottle is reached and grasped. These segments are attached to the *take-water* subtask, which is the only one available in this context, while the associated enabling conditions are needed to ensure the sequence of the segments (i.e. $noa9$ executed after $foa8$, and $gripper(close)$ after $noa9$). Afterwards, when the robot is driven towards the cup, the novel segment $foa10(world)$ is generated and linked to the *pour-water* subtask which becomes active after the bottle grasping. In this case, the motion between the bottle and the cup represents a FOA and the new generated segment is associated with a *true* releaser (i.e. always enabled).

7.5 Experiments and Discussions

In this section, we propose some experiments to show that the proposed approach can be effectively applied for *i)* incremental learning and execution of structured tasks, *ii)* execution of learned tasks in cooperation with the human, and *iii)* reuse of acquired knowledge in different contexts. To this end, we consider two typical tasks of a kitchen scenario; namely prepare coffee and prepare tea. The robot is a KUKA LWR IV+ [22], equipped with a WSG50 2-fingers gripper. As illustrated in Figure 7.5, objects are recognized and tracked using markers and a RGB-D camera (following the approach by [61]). The marker close to the robot base is used to compute the coordinate transformation between the camera frame and the robot base. Due to possible marker occlusions during the teaching, we estimate the robot-camera transformation and the pose of the cup at the beginning of each experiment and keep them constant during the execution. All the other objects, instead, are continuously tracked at 30 Hz. The user initiates a kinesthetic teaching session via the speech command *teach*. The teaching session is terminated by the speech command *done*. The user can interrupt/restart the execution of a learned task using the speech commands *stop/repeat*.

7.5.1 Experiments

Pouring a drink In the first experiment we teach the robot how to pour water in a cup. The pouring task consists of two subtasks: *take-water* and *pour-water* (see Figure 7.4, t1). The teaching process is simple and intu-

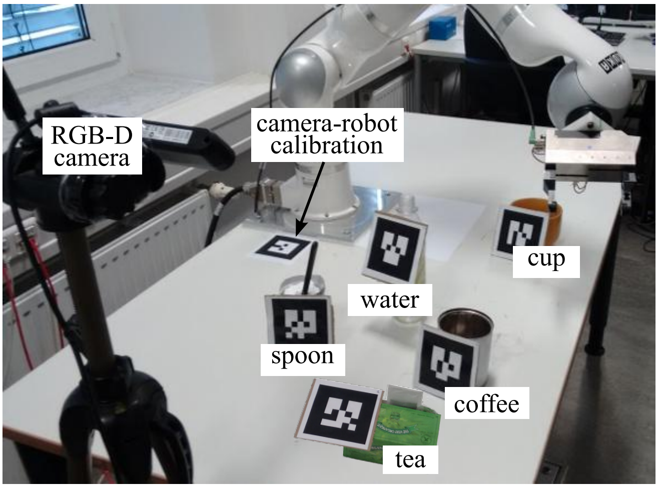

Figure 7.5. Experimental setup.

itive even for an untrained and inexpert user. The teacher has to simply guide the robot towards the task execution, providing sparse speech commands (*open*/*close*) to control the gripper. In Figure 7.6, we illustrate the WM state after a one-shot teaching session. At the end of the demonstration, we can find nine generated primitive actions, which are linked to the associated subtasks. These new elements are also associated with preconditions, effects, and activation values. As detailed in Algorithm 5, these generated elements are also stored in the system LTM to be re-used at execution time. Once learned, the task can be executed. In this situation, the attentional system first selects the subtask *take-water*, which is enabled when the robot has no object in its gripper (*hand.free*). The segments linked to the same subtask are executed in the order shown during the demonstration. For example, in order to perform the *take-water* subtask, the robot executes *foa1(water)*, *noa2(water)* and then *gripper(close)*.

In order to quantitatively evaluate the effectiveness of the proposed approach, we measured teaching and execution times over ten repetitions of the task. Moreover, in order to show the robustness of our approach

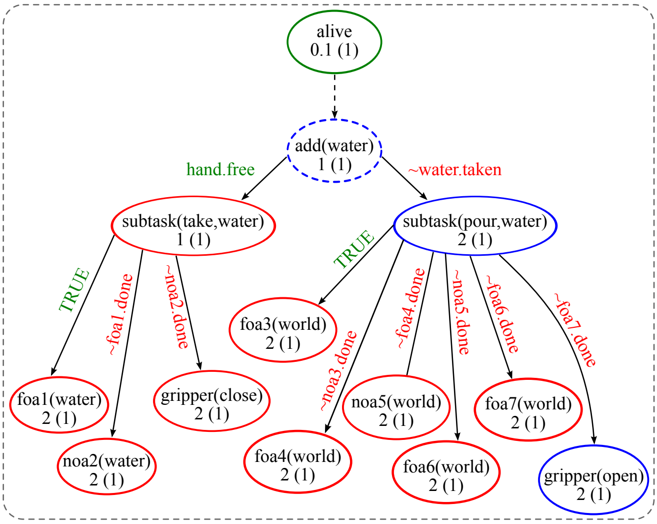

Figure 7.6. The WM state after the pouring task demonstration. Nine generated segments are linked to the associated subtasks.

with respect to the initial conditions, we performed ten repetitions of the task with the bottle placed at random positions, measuring the *success rate* as the number of correct executions over the total executions number. A trial is considered successful if the robot grasps the bottle and pours the water within the cup. In this case we obtained a 90% of success rate, hence one execution failed, with an average teaching time always less then 1 minute.

Coffee Preparation This experiment shows how a complex, structured task is learned and executed using the proposed framework. We consider the task of preparing a coffee, in which the robot has to: *i)* pour the water in the cup, *ii)* add the coffee powder, and *iii)* mix water and coffee powder with a spoon. Before learning, the WM only contains the three behaviors *add(water)*, *add(coffee)* and *use(spoon)* without any link to motion primitives.The action primitives and segments are automatically added during

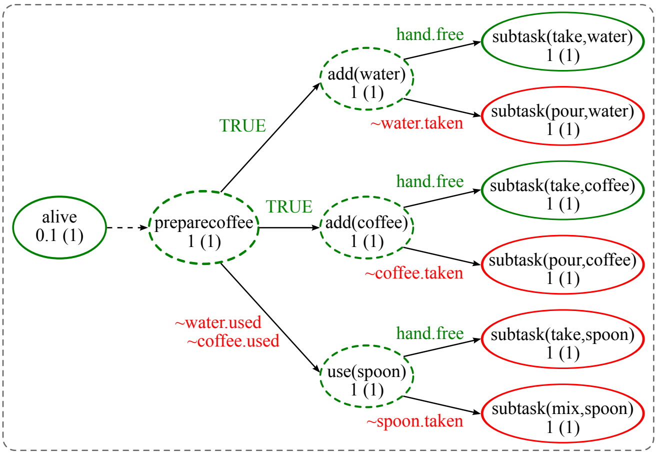

Figure 7.7. The WM before learning how to prepare a coffee. The task *preparecoffee* has three child nodes, namely *add(water)*, *add(coffee)* and *use(spoon)*. *add(water)* and *add(coffee)* can be executed in any order (*true* releaser), while *use(spoon)* requires that both the water and the coffee powder are added. Initially, both *subtask(take,water)* and *subtask(take,coffee)* are enabled, hence they compete for the initial segments.

the kinesthetic teaching and then used to reproduce the task. Note that the order of execution of *add(water)* and *add(coffee)* is not relevant for task learning and execution, indeed, they are both enabled when the task starts. In this case, task selection only depends on the attentional regulations. Here, the user can directly teach the overall *prepare coffee* task and then execute it, otherwise the task can be step by step demonstrated and executed (see the *prepare tea* task in Paragraph 7.5.1). Also in this case we perform ten repetition of the teaching and execution process. Analogously to the previous section the success rate is 90%, while the overall teaching time increase linearly to the complexity of the task and is always less then 3 minutes.

Cooperative Coffee Preparation The proposed framework permits a cooperative execution of the learned tasks. As a proof of concepts, we consider the coffee task described in the previous experiment and two cooperative scenarios. In the first case, the human helps the robot to fulfill the task by adding the water himself. To show the on-line capabilities of the attentional system, the user intentionally takes the bottle, while the robot is approaching it, (i.e., while it is executing the *foa8(water)* action in Figure 7.8) and pours the water. In this situation, the system has to rapidly adapt task execution with respect to the human behavior. Since the water is not anymore available in the scene, the *add(water)* behavior becomes less attractive for the robot that starts to execute the *add(coffee)* (which is available and enabled in the WM). At the same time, the system can monitor the human behavior and assign the *add(water)* execution to the human. In this setting, for the sake of simplicity, the above assignment is explicitly communicated by the human through a vocal utterance. Notice, however, that more complex activity recognition methods can be deployed for the same purpose [40]. We executed this cooperative task ten times, obtaining an average execution time lower than 3 minutes. Furthermore, cooperative executions are faster then non-cooperative by 20 seconds in average, this emphasize the beneficial of cooperation in term of execution times with a negligible time needed for the adjustment.

Tea Preparation and Task Reuse In the last experiment, we show that the acquired knowledge can be re-used to speed-up the acquisition of novel tasks. We consider the task of preparing a tea, where the robot has to pour the water in the cup and add a tea bag. The *add(water)* behavior is the same behavior used to prepare the coffee and can be re-used in this novel scenario. In other words, the already learned behavior can be loaded from the long term memory and instantiated in the working memory, while the user has only to teach how to add the tea bag. Once allocated in the WM, all the enabled and linked subtasks (e.g. *add(water)*) can be executed until the unlinked one (*add(tea)*) is selected. In this case

Figure 7.8. Cooperative execution of the prepare coffee task. The user takes the bottle and pours the water while the robot is approaching the bottle. Indeed, before the human intervention the most emphasized action segment is *foa8(water)*. When the human performs the action, the robotic task execution is on-line adapted: the most emphasized action segment becomes *foa1(coffee)* and the robot takes the coffee jar. A demonstration is available at the following link: http://wpage.unina.it/riccardo.caccavale/media/kin2016.mp4

the robot needs the human demonstration to learn how to complete the overall task. In order to assess the effectiveness of task re-usage, we run ten teaching sessions: in five runs the teacher has to demonstrate the entire task, while in the remaining five runs the operator only teaches the missing *add(tea)* behavior. In this scenario, task re-usage is effective and can reduce the teaching time of about 53%.

7.5.2 Discussions

We presented an approach to structured tasks learning that combines kinesthetic teaching and attentional supervision. In the proposed framework, a supervisory attentional system continuously monitors the human and the robot activities during both task execution and task learning. During kinesthetic teaching, the human demonstration is automatically segmented into motion primitives, while the attentional system relates them to the associated task structure exploiting top-down and bottom-up regulations. Specifically, contextual information is used to segment the human demonstration; each generated segment is associated with a stable dynamical system, which is then linked to a suitable node of an active task. This approach allows the system to learn how to executed structured task, from an intuitive, one-shot, user demonstration. In addition, the learned tasks can be stored in task a repository (the system long term memory) and then reused in different contexts. Indeed, the attentional system can flexible orchestrate the execution of the available tasks and subtasks, managing unexpected user interventions and environmental changes.

The proposed approach has been evaluated considering a robotic manipulator operating in a kitchen scenario. The experiments suggest that the proposed framework allows the robot to quickly learn and robustly execute typical structured activities that combine pick, place, and object manipulation actions. Moreover, we illustrated how learned tasks/subtasks can be reused in the context of novel task, in so enabling the acquisition of incrementally complex capabilities. Once learned, the tasks can be executed both autonomously or in cooperation with the human. We discussed the system at work during cooperative task execution showing that the human can effectively support the robot activity reducing the overall task execution time.

Chapter 8

Conclusions and Future Works

In this thesis we proposed a novel robotic cognitive control framework that facilitates human-robot cooperation in dynamic environments supporting flexible task execution, multimodal interaction, learning, and continuous regulation of sensorimotor processes. The proposed system 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 high level processes like plans, high-level commands, dialogue policy, etc. are used as a top-down attentional guidance that stimulates the system towards task accomplishment. This way, multiple hierarchical tasks can be flexibly orchestrated combining goal-oriented, reactive, and interactive behaviors.

8.1 Summary of Results

We proposed a robotic cognitive control framework for the execution of structured complex tasks with natural and flexible human-robot interaction [36, 42]. 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 towards 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 the 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, body postures are involved. In this setting, more sophisticated attentional regulation mechanisms (see e.g. [31]) can be introduced and assessed considering also visual attention, joint attention, and human intention recognition.

Following this perspective, we introduced a further scenario [35, 41] where multiple plans are to be adapted with respect to the human behavior and the environmental changes. We described and discussed the proposed system in a human-robot co-working scenario considering both simulated and real-world experiments. In these contexts, we illustrated how plan guidance and attentional regulation allow us to solve decisional impasses and reduce replanning episodes while driving the system towards task and plan accomplishment.

Moreover, we proposed an approach where a dialogue manager is also involved [39, 40]. In this context, the human-robot dialogue policy can be interpreted as a cognitive process that affects the overall attentional system. We assessed the system performance considering both task and communication ambiguities. The empirical results show that this integrated system is effective and can be exploited to further support the human-robot cooperation during the execution of shared structured plans.

We also deployed the proposed framework to design an attentive user interface suitable for monitoring the activities of multiple drones involved in search and rescue missions in the Alps [33]. The interface has been designed for an operator, involved in the rescue mission, that can interact with the drones exploiting different modalities. The attentive interface is managed by a supervisory attentional system that regulates contentions among multiple bottom-up stimuli depending on the active tasks and the human constraints. We described the overall system architecture along with the associated adaptive mechanisms. The interface has been assessed in a simulated case study by comparing user performance gathered with or without the adaptive interface. The collected results show that despite a significant reduction of the information provided to the user, the overall performance is not degraded, but slightly improved; this supports the hypothesis that the proposed framework is effective in filtering and selecting information relevant to the user.

Finally we reported on preliminary works exploiting the proposed system, along with its attentional regulations, in the context of task learning and refinement starting from human demonstrations. In this context we integrated kinesthetic teaching [43] with the attentional mechanisms. This framework continuously monitors the human and the robot activities during both task execution and task learning. During kinesthetic teaching, the human demonstration is automatically segmented into motion primitives, while the attentional system relates them to the associated task structure exploiting top-down and bottom-up regulations. Specifically, contextual information is used to segment the human demonstration; each generated segment is associated with a stable dynamical system, which is then linked to a suitable node of an active task. This approach allows the system to learn how to executed structured task, from an intuitive, one-shot, user demonstration. In addition, the learned tasks can be stored in a repository (the system long term memory) and then reused in different contexts. Indeed, the attentional system can flexibly orchestrate the execution of the available tasks and subtasks, managing unexpected user interventions and environmental changes.

8.2 Future Works

Different lines of future work can be defined. First of all, the framework can be applied to more complex real robotic scenarios, involving complex actuators and perceptual systems. In this case the perception of the human environment can be directly integrated in the control of the robot allowing more flexible and reactive motions, complex task execution and natural human-robot interaction. In particular, it is interesting to integrate additional vision, auditory and haptic attentional mechanisms. Indeed, joint-attention, object-based attention, visio-haptic attention are very relevant in human-robot interaction. For instance, joint attention and visio-haptic attention are involved during object manipulation and handover, while selective auditory attention is relevant during multimodal interaction. Moreover, more focused user studies in the context of human-robot interaction can be proposed, along with additional tests in realistic cooperative scenarios considering long-term autonomy settings. Another possible field of future research is the integration of the attentional regulations in problem solving and planning processes. The aim in this context is to adapt the reasoning process to the environmental changes, while the bottom-up and top-down stimulations can affect and drive the planning process in order to generate context related plans or to repair a misaligned plan. We started to address these topics in [41], but additional mechanisms should be considered and investigated.

Further works can be carried out in the context of task learning and adaptation. In this thesis, we started to investigate these issues in a learning-by-demonstration framework. The collected results encourage us to explore more complex attentional mechanisms supporting the learning process. In this regard, additional users studies can be proposed to asses the naturalness and the easiness of the teaching. Furthermore, other problems can be addressed. For instance, if we consider the force feedback provided by the human during kinesthetic teaching, we may explore how capabilities like drilling, pushing, cutting, can be demonstrated and transferred to the robotic system. Another interesting direction of research is the deployment of kinesthetic teaching for dual-arm manipulation. Additional challenging issues arise in this case, like arms synchronization, divided attention and the execution of more complex task structures.

We are also interested in on-line learning attentional regulation strategies from the human demonstration. In this perspective, a preliminary work has been presented in [37] where the regulations are refined following a neural-network approach based on error back-propagation. Our initial results suggest that this approach enables an effective interactive refinement of emphasis regulations for task monitoring and execution, but further experiments should be carried on considering several conflicting tasks to be learned and executed.

This page intentionally left blank.

Bibliography

[1] *Sherpa, eu collaborative project ict-600958*, http://www.sherpa-project.eu/sherpa/.

[2] Seyed Reza Ahmadzadeh, Ali Paikan, Fulvio Mastrogiovanni, Lorenzo Natale, Petar Kormushev, and Darwin G Caldwell, *Learning symbolic representations of actions from human demonstrations*, Robotics and Automation (ICRA), 2015 IEEE International Conference on, IEEE, 2015, pp. 3801–3808.

[3] Rachid Alami, Mamoun Gharbi, Benjamin Vadant, Raphaël Lallement, and Adolfo Suarez, *On human-aware task and motion planning abilities for a teammate robot*, RSS HRCIM, 2014.

[4] James S Albus, *The nist real-time control system (rcs): an approach to intelligent systems research*, Journal of Experimental & Theoretical Artificial Intelligence **9** (1997), no. 2-3, 157–174.

[5] ———, *4d/rcs: A reference model architecture for intelligent unmanned ground vehicles*, AeroSense 2002, International Society for Optics and Photonics, 2002, pp. 303–310.

[6] James S Albus and Anthony J Barbera, *Rcs: A cognitive architecture for intelligent multi-agent systems*, Annual Reviews in Control **29** (2005), no. 1, 87–99.

[7] G.A. Alvarez and P. Cavanagh, *Independent resources for attentional tracking in the left and right visual hemifields*, Psychological Science **16** (2005), no. 8, 637–643.

[8] John R Anderson, Michael Matessa, and Christian Lebiere, *Act-r: A theory of higher level cognition and its relation to visual attention*, Human-Computer Interaction **12** (1997), no. 4, 439–462.

[9] Michael A Arbib, Jane Hill, and Jeffrey Conklin, *From schema theory to language*, Oxford University Press, 1987.

[10] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning, *A survey of robot learning from demonstration*, Robot. Auton. Syst. **57** (2009), no. 5, 469–483.

[11] Ronald C Arkin, *Integrating behavioral, perceptual, and world knowledge in reactive navigation*, Robotics and autonomous systems **6** (1990), no. 1, 105–122.

[12] ______, *Behavior-based robotics*, MIT press, 1998.

[13] N.F. Ayan, U. Kuter, F. Yaman, and R. Goldman, *Hotride: Hierarchical ordered task replanning in dynamic environments*, Proc. of ICAPS WS on Planning and Plan Execution for Real-World Systems – Principles and Practices for Planning in Execution, 2007.

[14] Alan Baddeley, *Exploring the central executive*, The Quarterly Journal of Experimental Psychology: Section A **49** (1996), no. 1, 5–28.

[15] ______, *The episodic buffer: a new component of working memory?*, Trends in cognitive sciences **4** (2000), no. 11, 417–423.

[16] Alan D Baddeley and GJ Hitch, *Working memory*, The psychology of learning and motivation **8** (1975), 47–89.

[17] Alan D Baddeley and Elizabeth K Warrington, *Amnesia and the distinction between long-and short-term memory*, Journal of verbal learning and verbal behavior **9** (1970), no. 2, 176–189.

[18] Russell A Barkley, *Behavioral inhibition, sustained attention, and executive functions: constructing a unifying theory of adhd.*, Psychological bulletin **121** (1997), no. 1, 65.

[19] A. Belardinelli, F. Pirri, and A. Carbone, *Bottom-up gaze shifts and fixations learning by imitation*, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **37** (2007), no. 2, 256–271.

[20] D Paul Benjamin, *Designing a robot cognitive architecture with concurrency and active perception.*

[21] D Paul Benjamin, Damian M Lyons, and Deryle W Lonsdale, *Adapt: A cognitive architecture for robotics.*, ICCM, 2004, pp. 337–338.

[22] Rainer Bischoff, Johannes Kurth, Guenter Schreiber, Ralf Koeppe, Alin Albu-Schäffer, Alexander Beyer, Oliver Eiberger, Sami Haddadin, Andreas Stemmer, Gerhard Grunwald, and Gerhard Hirzinger, *The kuka-dlr lightweight robot arm - a new reference platform for robotics research and manufacturing*, International Symposium on Robotics, 2010, pp. 1–8.

[23] Guido Boella and Rossana Damiano, *A replanning algorithm for a reactive agent architecture*, Proc. of AIMSA 2002, 2002, pp. 183–192.

[24] Ali Borji, Majid Nili Ahmadabadi, Babak Nadjar Araabi, and Mandana Hamidi, *Online learning of task-driven object-based visual attention control*, Image Vision Comput. **28** (2010), no. 7, 1130–1145.

[25] Matthew Botvinick and David C. Plaut, *Doing without schema hierarchies: A recurrent connectionist approach to normal and impaired routine sequential action*, Psychological Review **111** (2004), 395–429.

[26] Matthew M Botvinick, Todd S Braver, Deanna M Barch, Cameron S Carter, and Jonathan D Cohen, *Conflict monitoring and cognitive control.*, Psychological review **108** (2001), no. 3, 624.

[27] Cynthia Breazeal, *Designing sociable robots*, MIT Press, 2002.

[28] Cynthia Breazeal and Matt Berlin, *Spatial scaffolding for sociable robot learning*, Proc. of AAAI-2008, 2008, pp. 1268–1273.

[29] Cynthia Breazeal, Matt Berlin, Andrew G. Brooks, Jesse Gray, and Andrea Lockerd Thomaz, *Using perspective taking to learn from ambiguous demonstrations*, Robotics and Autonomous Systems **54** (2006), no. 5, 385–393.

[30] Cynthia Breazeal, Cory D Kidd, Andrea Lockerd Thomaz, Guy Hoffman, and Matt Berlin, *Effects of nonverbal communication on efficiency and robustness in human-robot teamwork*, 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2005, pp. 708–713.

[31] Xavier Broquère, Alberto Finzi, Jim Mainprice, Silvia Rossi, Daniel Sidobre, and Mariacarla Staffa, *An attentional approach to human–robot interactive manipulation*, International Journal of Social Robotics (2014).

[32] E. Burattini, S. Rossi, Alberto Finzi, and Maria Carla Staffa, *Attentional modulation of mutually dependent behaviors*, Proc. of SAB 2010, 2010, pp. 283–292.

[33] Jonathan Cacace, Riccardo Caccavale, Alberto Finzi, and Vincenzo Lippiello, *Attentional multimodal interface for multidrone search in the alps*, Proc. of Proc. of IEEE International Conference on Systems, Man, and Cybernetics, 2016.

[34] ———, *A human-robot interaction framework for search and rescue in the alps*, Proc. of AIRO at AI*IA, 2016.

[35] Jonathan Cacace, Riccardo Caccavale, Michelangelo Fiore, Rachid Alami, and Alberto Finzi, *Attentional plan execution for human-robot cooperation*, Proc. of AIRO at AI*IA, 2015, pp. 19–28.

[36] R. Caccavale and A. Finzi, *Plan execution and attentional regulations for flexible human-robot interaction*, Proc. of IEEE International Conference on Systems, Man, and Cybernetics, 2015, pp. 2453–2458.

[37] ———, *Cognitive control and adaptive attentional regulations for robotic task execution*, euCognition Meeting in Vienna, Austria, 2016.

[38] R. Caccavale, A. Finzi, D. Lee, and M. Saveriano, *Integrated task learning and kinesthetic teaching for human-robot cooperation*, Proc. of AIRO at AI*IA, 2016.

[39] R. Caccavale, A. Finzi, L. Lucignano, S. Rossi, and M. Staffa, *Attentional top-down regulation and dialogue management in human-robot interaction*, Proc. of HRI 2014, 2014, pp. 129–130.

[40] R Caccavale, E Leone, L Lucignano, S Rossi, M Staffa, and A Finzi, *Attentional regulations in a situated human-robot dialogue*, RO-MAN, 2014, pp. 844–849.

[41] Riccardo Caccavale, Jonathan Cacace, Michelangelo Fiore, Rachid Alami, and Alberto Finzi, *Attentional supervision of human-robot collaborative plans*, Robot and Human Interactive Communication

(RO-MAN), 2016 25th IEEE International Symposium on, IEEE, 2016, pp. 867–873.

[42] Riccardo Caccavale and Alberto Finzi, *Flexible task execution and attentional regulations in human-robot interaction*, IEEE Transactions on Cognitive and Developmental Systems (2016).

[43] Riccardo Caccavale, Alberto Finzi, Dongheui Lee, Enrico Leone, Silvia Rossi, Matteo Saveriano, and Mariacarla Staffa, *Integrating multimodal interaction and kinesthetic teaching for flexible human-robot collaboration*, Proc. of HFR, 2015.

[44] Andrea Carbone, Alberto Finzi, Andrea Orlandini, and Fiora Pirri, *Model-based control architecture for attentive robots in rescue scenarios*, Autonomous Robots **24** (2008), no. 1, 87–120.

[45] Richard Cooper and Tim Shallice, *Contention scheduling and the control of routine activities*, Cognitive Neuropsychology **17** (2000), 297–338.

[46] Richard P. Cooper and Tim Shallice, *Hierarchical schemas and goals in the control of sequential behavior*, Psychological Review **113(4)** (2006), 887–916.

[47] Hugo De Garis, Chen Shuo, Ben Goertzel, and Lian Ruiting, *A world survey of artificial brain projects, part i: Large-scale brain simulations*, Neurocomputing **74** (2010), no. 1, 3–29.

[48] Eu De Renzi and F Lucchelli, *Ideational apraxia*, Brain **111** (1988), no. 5, 1173–1185.

[49] Robert Desimone and John Duncan, *Neural mechanisms of selective visual attention*, Annual review of neuroscience **18** (1995), no. 1, 193–222.

[50] Dario Di Nocera, Alberto Finzi, Silvia Rossi, and Mariacarla Staffa, *Attentional action selection using reinforcement learning*, International Conference on Simulation of Adaptive Behavior, Springer, 2012, pp. 371–380.

[51] Adele Diamond, *Executive functions*, Annual review of psychology **64** (2013), 135–168.

[52] Rüdiger Dillmann, *Teaching and learning of robot tasks via observation of human performance*, Robotics and Autonomous Systems **47** (2004), no. 2, 109–116.

[53] Wlodzislaw Duch, Richard Jayadi Oentaryo, and Michel Pasquier, *Cognitive architectures: Where do we go from here?*, Frontiers in artificial intelligence and applications **171** (2008), 122.

[54] Mihai Duguleana, Florin Grigorie Barbuceanu, and Gheorghe Mogan, *Evaluating human-robot interaction during a manipulation experiment conducted in immersive virtual reality*, International Conference on Virtual and Mixed Reality, Springer, 2011, pp. 164–173.

[55] Rebecca Elliott, *Executive functions and their disorders imaging in clinical neuroscience*, British Medical Bulletin **65** (2003), no. 1, 49–59.

[56] Tara Estlin, Rich Volpe, Issa Nesnas, Darren Mutz, Forest Fisher, Barbara Engelhardt, and Steve Chien, *Decision-making in a robotic architecture for autonomy*, Proceedings of the international symposium on artificial intelligence, robotics, and automation in space, 2001.

[57] Alberto Finzi, Félix Ingrand, and Nicola Muscettola, *Model-based executive control through reactive planning for autonomous rovers*, Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004

IEEE/RSJ International Conference on, vol. 1, IEEE, 2004, pp. 879–884.

[58] Alberto Finzi and Andrea Orlandini, *Human-robot interaction through mixed-initiative planning for rescue and search rovers*, Congress of the Italian Association for Artificial Intelligence, Springer, 2005, pp. 483–494.

[59] Ajo Fod, Maja J. Matarić, and Odest Chadwicke Jenkins, *Automated derivation of primitives for movement classification*, Autonomous Robots **12** (2002), no. 1, 39–54.

[60] Jason Garforth, Sue L. McHale, and Anthony Meehan, *Executive attention, task selection and attention-based learning in a neurally controlled simulated robot.*, Neurocomputing **69** (2006), no. 16-18, 1923–1945.

[61] S. Garrido-Jurado, R. Muñoz Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, *Automatic generation and detection of highly reliable fiducial markers under occlusion*, Pattern Recognition **47** (2014), no. 6, 2280–2292.

[62] Erann Gat, *Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots*, AAAi, vol. 1992, 1992, pp. 809–815.

[63] Erann Gat et al., *On three-layer architectures*, Artificial intelligence and mobile robots **195** (1998), 210.

[64] Ben Goertzel, Hugo De Garis, Cassio Pennachin, Nil Geisweiller, Samir Araujo, Joel Pitt, Shuo Chen, Ruiting Lian, Min Jiang, Ye Yang, et al., *Opencogbot: Achieving generally intelligent virtual agent control and humanoid robotics via cognitive synergy*, Proceedings of ICAI, vol. 10, Citeseer, 2010.

[65] Ben Goertzel, Ruiting Lian, Itamar Arel, Hugo de Garis, and Shuo Chen, *A world survey of artificial brain projects, part ii: Biologically inspired cognitive architectures*, Neurocomputing **74** (2010), no. 1, 30–49.

[66] Houcine Hassan, José Simó, and Alfons Crespo, *Flexible real-time mobile robotic architecture based on behavioural models*, Engineering Applications of Artificial Intelligence **14** (2001), no. 5, 685–702.

[67] Henry Head, *Aphasia and kindred disorders of speech.*, (1926).

[68] Guy Hoffman and Cynthia Breazeal, *Cost-based anticipatory action selection for human–robot fluency*, IEEE Transactions on Robotics **23** (2007), no. 5, 952–961.

[69] Andreas Hofmann, Enrique Fernandez, Justin Helbert, Scott Smith, and Brian Williams, *Reactive integrated motion planning and execution*, AAAI Press/International Joint Conferences on Artificial Intelligence, 2015.

[70] Eric Horvitz, Andy Jacobs, and David Hovel, *Attention-sensitive alerting*, Proc. of UAI-99, 1999, pp. 305–313.

[71] A. Ijspeert, J. Nakanishi, P Pastor, H. Hoffmann, and S. Schaal, *Dynamical Movement Primitives: learning attractor models for motor behaviors*, Neural Computation **25** (2013), no. 2, 328–373.

[72] Laurent Itti, Christof Koch, Ernst Niebur, et al., *A model of saliency-based visual attention for rapid scene analysis*, IEEE Transactions on pattern analysis and machine intelligence **20** (1998), no. 11, 1254–1259.

[73] William James, *The principles of psychology*, Digireads. com, 2011.

[74] Daniel Kahneman, *Attention and effort*, Prentice-Hall Inc., 1973.

[75] Frederic Kaplan and Verena V Hafner, *The challenges of joint attention*, Interaction Studies **7** (2006), no. 2, 135–169.

[76] Erez Karpas, Steven James Levine, Peng Yu, and Brian C Williams, *Robust execution of plans for human-robot teams.*, ICAPS, 2015, pp. 342–346.

[77] Kazuhiko Kawamura, Stephen M Gordon, Palis Ratanaswasd, Erdem Erdemir, and Joseph F Hall, *Implementation of cognitive control for a humanoid robot*, International Journal of Humanoid Robotics **5** (2008), no. 04, 547–586.

[78] Kazuhiko Kawamura, R ALAN PETERS II, Robert E Bodenheimer, Nilanjan Sarkar, Juyi Park, Charles A Clifton, Albert W Spratley, and Kimberly A Hambuchen, *A parallel distributed cognitive control system for a humanoid robot*, International Journal of Humanoid Robotics **1** (2004), no. 01, 65–93.

[79] William G Kennedy, Magdalena D Bugajska, Matthew Marge, William Adams, Benjamin R Fransen, Dennis Perzanowski, Alan C Schultz, and J Gregory Trafton, *Spatial representation and reasoning for human-robot collaboration*, AAAI, vol. 7, 2007, pp. 1554–1559.

[80] Hema S. Koppula and Ashutosh Saxena, *Anticipating human activities using object affordances for reactive robotic response*, Transactions on Pattern Analysis and Machine Intelligence **38** (2015), no. 1, 14–29.

[81] Iuliia Kotseruba, Oscar J Avella Gonzalez, and John K Tsotsos, *A review of 40 years of cognitive architecture research: Focus on perception, attention, learning and applications*, arXiv preprint arXiv:1610.08602 (2016).

[82] D Kulić, C Ott, D Lee, J Ishikawa, and Y Nakamura, *Incremental learning of full body motion primitives and their sequencing through human motion observation*, International Journal of Robotics Research **31** (2012), no. 3, 330–345.

[83] John E Laird, Keegan R Kinkade, Shiwali Mohan, and Joseph Z Xu, *Cognitive robotics using the soar cognitive architecture*, Cognitive Robotics AAAI Technical Report WS-12-06. Accessed (2012), 46–54.

[84] John E Laird, Allen Newell, and Paul S Rosenbloom, *Soar: An architecture for general intelligence*, Artificial intelligence **33** (1987), no. 1, 1–64.

[85] John E Laird and Paul S Rosenbloom, *Integrating execution, planning, and learning in soar for external environments*, Ann Arbor **1001** (1993), 48109–2110.

[86] John E Laird, Eric S Yager, Michael Hucka, and Christopher M Tuck, *Robo-soar: An integration of external interaction, planning, and learning using soar*, Robotics and Autonomous Systems **8** (1991), no. 1, 113–129.

[87] Raphaël Lallement, Lavindra de Silva, and Rachid Alami, *HATP: An HTN Planner for Robotics*, 2nd ICAPS Workshop on Planning and Robotics, PlanRob 2014, 2014.

[88] Pat Langley, John E Laird, and Seth Rogers, *Cognitive architectures: Research issues and challenges*, Cognitive Systems Research **10** (2009), no. 2, 141–160.

[89] Lashley, *The problem of serial order in behavior*, Cerebral Mechanisms in Behavior, Wiley, New York (L.A. In Jeffress, ed.), 1951.

[90] D. Lee and C. Ott, *Incremental kinesthetic teaching of motion primitives using the motion refinement tube*, Autonomous Robots **31** (2011), no. 2, 115–131.

[91] Mattias Lindstrom, Anders Oreback, and Henrik I Christensen, *Berra: A research architecture for service robots*, Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on, vol. 4, IEEE, 2000, pp. 3278–3283.

[92] Lorenzo Lucignano, Francesco Cutugno, Silvia Rossi, and Alberto Finzi, *A dialogue system for multimodal human-robot interaction*, Proceedings of the 15th ACM on International conference on multimodal interaction, ACM, 2013, pp. 197–204.

[93] ______, *A dialogue system for multimodal human-robot interaction*, In Proc. of ICMI-2013, 2013 (to appear).

[94] Alan M Lytle and Kamel S Saidi, *Nist research in autonomous construction*, Autonomous Robots **22** (2007), no. 3, 211–221.

[95] V. Magnanimo, M. Saveriano, S. Rossi, and Dongheui Lee, *A bayesian approach for task recognition and future human activity prediction*, Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on, Aug 2014, pp. 726–731.

[96] S. Manschitz, J. Kober, M. Gienger, and J. Peters, *Learning movement primitive attractor goals and sequential skills from kinesthetic demonstrations*, Robotics and Autonomous Systems **74, Part A** (2015), 97–107.

[97] Fulvio Mastrogiovanni, Ali Paikan, and Antonio Sgorbissa, *Semantic-aware real-time scheduling in robotics*, IEEE Transactions on Robotics **29** (2013), no. 1, 118–135.

[98] George Armitage Miller, Eugene Galanter, and Karl H Pribram, *Plans and the structure of behavior*, (1960).

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

[100] Michael C Mozer and Mark Sitton, *Computational modeling of spatial attention*, Attention **9** (1998), 341–393.

[101] Thomas Muller and Alois Knoll, *Attention driven visual processing for an interactive dialog robot*, Proc. of the 2009 ACM symp. on Applied Computing, 2009, pp. 1151–1155.

[102] David J Musliner, Edmund H Durfee, and Kang G Shin, *Circa: A cooperative intelligent real-time control architecture*, IEEE Transactions on Systems, Man, and Cybernetics **23** (1993), no. 6, 1561–1574.

[103] Yukie Nagai, *From bottom-up visual attention to robot action learning*, Proc. of International Conference on Development and Learning, 2009, pp. 1–6.

[104] Yukie Nagai, Koh Hosoda, Akio Morita, and Minoru Asada, *A constructive model for the development of joint attention*, Connection Science **15** (2003), no. 4, 211–229.

[105] Dana Nau, Yue Cao, Amnon Lotem, and Hector Muñoz-Avila, *Shop: Simple hierarchical ordered planner*, Proceedings of the 16th international joint conference on Artificial intelligence-Volume 2, Morgan Kaufmann Publishers Inc., 1999, pp. 968–973.

[106] Dana S. Nau, Tsz-Chiu Au, Okhtay Ilghami, Ugur Kuter, J. William Murdock, Dan Wu, and Fusun Yaman, *SHOP2: an HTN planning system*, J. Artif. Intell. Res. (JAIR) **20** (2003), 379–404.

[107] Issa A Nesnas, Anne Wright, Max Bajracharya, Reid Simmons, Tara Estlin, and Won Soo Kim, *Claraty: An architecture for reusable robotic software*, AeroSense 2003, International Society for Optics and Photonics, 2003, pp. 253–264.

[108] Monica N Nicolescu and Maja J Matarić, *A hierarchical architecture for behavior-based robots*, Proc. of AAMAS-02, 2002, pp. 227–233.

[109] Monica N Nicolescu and Maja J Mataric, *Natural methods for robot task learning: Instructive demonstrations, generalization and practice*, Proceedings of the second international joint conference on Autonomous agents and multiagent systems, ACM, 2003, pp. 241–248.

[110] Donald A Norman and Tim Shallice, *Attention to action: Willed and automatic control of behavior.*, Tech. report, DTIC Document, 1980.

[111] Hiroshi G Okuno, Kazuhiro Nakadai, KI Hidai, Hiroshi Mizoguchi, and Hiroaki Kitano, *Human-robot interaction through real-time auditory and visual multiple-talker tracking*, Proc. of IROS 2001, 2001, pp. 1402–1409.

[112] P Pastor, M Kalakrishnan, L Righetti, and S Schaal, *Towards associative skill memories*, International Conference on Humanoid Robots, 2012, pp. 309–315.

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

[114] Michael I Posner, *Cognitive neuroscience of attention*, Guilford Press, 2011.

[115] Michael I Posner and Yoav Cohen, *Components of visual orienting*, Attention and performance X: Control of language processes **32** (1984), 531–556.

[116] M.L. Posner, C.R.R. Snyder, and B. J. Davidson, *Attention and the detection of signals*, Journal of Experimental Psychology: General **109** (1980), 160–174.

[117] W. Prinzmetal, C. McCool, and S. Park, *Attention: reaction time and accuracy reveal different mechanisms*, Journal of Experimental Psychology: General **134** (2005), no. 1, 73–92.

[118] Martin Proetzsch, Tobias Luksch, and Karsten Berns, *Development of complex robotic systems using the behavior-based control architecture ib2c*, Robotics and Autonomous Systems **58** (2010), no. 1, 46–67.

[119] Karinne Ramirez-Amaro, Michael Beetz, and Gordon Cheng, *Understanding the intention of human activities through semantic perception: observation, understanding and execution on a humanoid robot*, Advanced Robotics **29** (2015), no. 5, 345–362.

[120] Jane E Raymond, Kimron L Shapiro, and Karen M Arnell, *Temporary suppression of visual processing in an rsvp task: An attentional blink?*, Journal of Experimental Psychology: Human Perception and Performance **18** (1992), no. 3, 849.

[121] Grega Repovš and Alan Baddeley, *The multi-component model of working memory: explorations in experimental cognitive psychology*, Neuroscience **139** (2006), no. 1, 5–21.

[122] David A. Rosenbaum, Rajal G. Cohen, Steven A. Jax, Daniel J. Weiss, and Robrecht van der Wel, *The problem of serial order in behavior: Lashley's legacy*, **26** (2007), no. 4, 525–554.

[123] Silvia Rossi, Enrico Leone, Michelangelo Fiore, Alberto Finzi, and Francesco Cutugno, *An extensible architecture for robust multimodal human-robot communication*, In Proc. of IROS-2013, 2013 (to appear).

[124] Alexei V Samsonovich, *Toward a unified catalog of implemented cognitive architectures.*, BICA **221** (2010), 195–244.

[125] Matteo Saveriano, Sang-ik An, and Dongheui Lee, *Incremental kinesthetic teaching of end-effector and null-space motion primitives*, ICRA 2015, 2015, pp. 3570–3575.

[126] Matteo Saveriano and Dongheui Lee, *Point cloud based dynamical system modulation for reactive avoidance of convex and concave obstacles*, International Conference on Intelligent Robots and Systems, 2013, pp. 5380–5387.

[127] Matteo Saveriano and Dongheui Lee, *Distance based dynamical system modulation for reactive avoidance of moving obstacles*, Proc. of ICRA-2014, 2014, pp. 5618–5623.

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

[129] Matthias Scheutz and Virgil Andronache, *Architectural mechanisms for dynamic changes of behavior selection strategies in behavior-based systems*, IEEE Transactions of System, Man, and Cybernetics Part B: Cybernetics **34** (2004), no. 6, 2377–2395.

[130] Matthias Scheutz, Jack Harris, and Paul Schermerhorn, *Systematic integration of cognitive and robotic architectures*, Advances in Cognitive Systems (2013), 277–296.

[131] Craig Schlenoff, Jim Albus, Elena Messina, Anthony J Barbera, Raj Madhavan, and Stephen Balakirsky, *Using 4d/rcs to address ai knowledge integration*, AI Magazine **27** (2006), no. 2, 71.

[132] C. Setz, B. Arnrich, J. Schumm, R.La Marca, G. Troster, and U. Ehlert, *Discriminating stress from cognitive load using a wearable eda device*, IEEE Transactions on, Information Technology in Biomedicine **14** (2010), no. 2, 410–417.

[133] Masahiro Shiomi, Takayuki Kanda, Satoshi Koizumi, Hiroshi Ishiguro, and Norihiro Hagita, *Group attention control for communication robots*, International Journal of Humanoid Robotics **5** (2008), no. 04, 587–608.

[134] JE Simó, A Crespo, F Blanes, and JL Posadas, *Vehicle guidance with distributed selection of emergent behaviour*, Proceedings of IFAC International Symposium on Artificial Intelligence in Real Time Control AIRTC, vol. 97, 1997.

[135] A. Stoytchev and R. Arkin, *Incorporating motivation in a hybrid robot architecture*, Journal of Advanced Computational Intelligence and Intelligent Informatics **8** (2004), no. 3, 269–274.

[136] Alexander Stoytchev, *Behavior-grounded representation of tool affordances*, Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, IEEE, 2005, pp. 3060–3065.

[137] J Ridley Stroop, *Studies of interference in serial verbal reactions.*, Journal of experimental psychology **18** (1935), no. 6, 643.

[138] W Takano and Y Nakamura, *Real-time unsupervised segmentation of human whole-body motion and its application to humanoid robot*

acquisition of motion symbols, Robotics and Autonomous Systems **75, Part B** (2016), 260–272.

[139] M Tenorth and M Beetz, *Knowrob – a knowledge processing infrastructure for cognition-enabled robots*, International Journal of Robotics Research **32** (2013), no. 5, 566–590.

[140] Michael Tomasello, *Origins of human communication*, MIT press, 2010.

[141] Greg Trafton, Laura Hiatt, Anthony Harrison, Frank Tamborello, Sangeet Khemlani, and Alan Schultz, *Act-r/e: An embodied cognitive architecture for human-robot interaction*, Journal of Human-Robot Interaction **2** (2013), no. 1, 30–55.

[142] J Gregory Trafton, Nicholas L Cassimatis, Magdalena D Bugajska, Derek P Brock, Farilee E Mintz, and Alan C Schultz, *Enabling effective human-robot interaction using perspective-taking in robots*, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans **35** (2005), no. 4, 460–470.

[143] J Gregory Trafton, Alan C Schultz, Dennis Perznowski, Magdalena D Bugajska, William Adams, Nicholas L Cassimatis, and Derek P Brock, *Children and robots learning to play hide and seek*, Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction, ACM, 2006, pp. 242–249.

[144] Anne M Treisman and Garry Gelade, *A feature-integration theory of attention*, Cognitive psychology **12** (1980), no. 1, 97–136.

[145] Manuela M Veloso, Joydeep Biswas, Brian Coltin, and Stephanie Rosenthal, *Cobots: Robust symbiotic autonomous mobile service robots.*, IJCAI, 2015, p. 4423.

[146] Roel Vertegaal, Jeffrey S. Shell, Daniel Chen, and Aadil Mamuji, *Designing for augmented attention: Towards a framework for attentive user interfaces*, Computers in Human Behavior **22** (2006), 771–789.

[147] M. Wächter, S. Schulz, T. Asfour, E. Aksoy, Wörgötter, and R. Dillmann, *Action sequence reproduction based on automatic segmentation and object-action complexes*, International Conference on Humanoid Robots, 2013, pp. 189–195.

[148] Ian Warfield, Chad Hogg, Stephen Lee-Urban, and Hector Munoz-Avila, *Adaptation of hierarchical task network plans*, Proc. of FLAIRS-07, 2007, pp. 429–434.

[149] Christopher D. Wickens, *Multiple resources and performance prediction*, Theoretical issues in ergonomics science **3** (2002), no. 2, 159–177.

[150] Jason R Wilson, Evan Krause, Matthias Scheutz, and Morgan Rivers, *Analogical generalization of actions from single exemplars in a robotic architecture*, Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2016, pp. 1015–1023.

[151] Bob G Witmer and Michael J Singer, *Measuring presence in virtual environments: A presence questionnaire*, Presence: Teleoperators and virtual environments **7** (1998), no. 3, 225–240.

[152] F. Wörgötter, A. Agostini, N. Krüger, N. Shylo, and B. Porr, *Cognitive agents — a procedural perspective relying on the predictability of object-action-complexes (oacs)*, Robotics and Autonomous Systems **57** (2009), no. 4, 420–432.

[153] Wayne Wu, *Attention as selection for action*, Attention: Philosophical and psychological essays (2011), 97–116.

[154] R Zoliner, M Pardowitz, S Knoop, and R Dillmann, *Towards cognitive robots: Building hierarchical task representations of manipulations from human demonstration*, International Conference on Robotics and Automation, 2005, pp. 1535–1540.