RESEARCH ARTICLE

DiPRA: A Layered Agent Architecture which Integrates Practical Reasoning and Sensorimotor Schemas

Giovanni Pezzulo

Istituto di Linguistica Computazionale “Antonio Zampolli” - Consiglio Nazionale delle Ricerche, Via Giuseppe Moruzzi, 1 - 56124 Pisa, Italy
Istituto di Scienze e Tecnologie della Cognizione - Consiglio Nazionale Ricerche, Via S. Martino della Battaglia, 44 - 00185 Rome, Italy

(v2.0 released January 2008)

We introduce the layered agent architecture DiPRA (Distributed Practical Reasoning Architecture), composed of an ‘intentional’ layer, which includes beliefs, plans and goals, and a ‘sensorimotor’ layer, which includes schemas for situated action. DiPRA’s functioning is illustrated and evaluated in a simulated guards-and-thieves scenario. We also discuss the efficacy of the main features of DiPRA, such as the division of labour between off-line planning and on-line specification of action, the grounding of beliefs in sensorimotor interaction and anticipation, the use of bounded resources and knowledge, and the realization of deliberation and means-ends reasoning as intertwined processes.

Keywords:
layered architecture; practical reasoning; grounding; anticipation; schema

1. Introduction

Traditional theories in cognitive science distinguish between intentional (or willed) and automatic levels of action control. According to these theories, selection of appropriate action as well as perception of events result from two conflicting processes: goal-directed strategies and stimulus-driven action tendencies. Probably the best known example is the two-layered model of Norman and Shallice (1986) in which the higher layer, called ‘supervisory attentional system’ (SAS), is responsible for executive processes (mainly deliberation and planning intentional action) on the basis of available knowledge, and the lower level, which includes multiple, competing sensorimotor schemas and a ‘contention scheduling’ mechanism (CS), is responsible for situated action execution.

Based on such theoretical models, we have built a layered agent architecture that integrates intentional and sensorimotor behavior in two ways: 1) Courses of action planned in the intentional layer are realized by action schemas at the sensorimotor layer. 2) Schemas’ activity produces grounded beliefs, which are therefore used in the higher layer for practical reasoning. The architecture has been implemented and tested in a simulation of a guards-and-thieves scenario using the Irrlicht simulation engine (irrlicht.sourceforge.net) with realistic ODE physics (www.ode.org).
Schema training was performed, with results showing clear improvement over non-intentional approaches.

Our computational model relates to numerous others that utilize hierarchies of control. For example, Cooper and Shallice (2006) have implemented a computational system having a hierarchy of goals and schemas which compete for execution, and Glasspool and Cooper (2002) provided a model of higher level executive functions; see also (Botvinick 2008) for a recent review of alternative implementations of hierarchical control architectures. Hybrid cognitive architectures provide another well known example of systems that include executive processes and actuation. Most of these include two layers with different computational tasks: the higher layer is devoted to planning and reasoning, while the lower layer executes actions. Some examples of layered architectures are ACT-R (Anderson 1983), atlantis (Gat 1992) and CLARION (Sun 2002), all having two layers: ‘deliberative’ and ‘sensorimotor’. In these architectures the representational format is not homogeneous in the two layers. For example, ACT-R and CLARION adopt subsymbolic representations in the lower layer, and symbolic representations in the higher layer; this reflects a distinction popular in the psychological literature between procedural and declarative knowledge (Anderson 1983). All these systems have significant influence in the cognitive science community. However, they lack realistic ‘embodiment’: they have highly simplified perception and actuation, and the symbols they use for reasoning are not autonomously generated but predefined by their programmers. In examining the issue of the quality of cognitive models that lack embodiment and situatedness (Clark 1998), it is clear that, no matter how good these models are, they cannot be directly used for the realization of robots or agent architectures operating autonomously in open-ended environments.

Indeed, most of the robot architectures that integrate goal-directed and automatic control and operate in open-ended environments have simplified higher level cognitive capabilities. One popular example is the behavior networks architecture (BN) (Dorer 1999, Maes 1990), which is able to maintain competing goals and trigger them on the basis of continuously updated knowledge. In BN(s) the key elements are to maintain knowledge representation as simple and homogeneous as possible, so to speed up all processing, and to use a fixed policy for deliberation based on spreading activation. Although quite successful in limited applications, however, BNs were ultimately unsuited for scaling to open-ended domains, and for supporting sophisticated forms of reasoning and deliberation. Architectures with explicit layers of control offer another solution to the same problems, and are currently very popular thanks to the fact that they couple situated action with flexible knowledge manipulation. Unlike the BNs, those architectures include separated layers (typically two), one for reasoning and deliberation, and one for executing situated action, although the design of such layers vary. Like BNs, most layered architectures deal with the problem of communication between the two layers by using a homogeneous representation format, such as fuzzy logic, among layers (Bonarini et al. 2003, Pezzulo et al. 2005, Saffiotti and Wasik 2003). Although these systems have planning capabilities, they typically do not support sophisticated forms of reasoning and deliberation. One notable exception is described in (Parsons et al. 2000): an early attempt to hybridize a layered robot architecture with the BDI (belief, desire, intention) architecture for practical reasoning (Rao and Georgeff 1995), the former operating on sensory data and providing lower level navigation capabilities, and the latter operating on symbolic data and supporting deliberation and means-ends reasoning. Other examples of layered agent architectures are HMOSAIC (Haruno et al. 2003) and HAMMER (Demiris and Khadhouri 2005), whose layers include actions and plans specified at multiple, increasingly
abstract levels, and a homogeneous action or plan selection mechanism. These systems, however, put even less emphasis on higher level cognitive processing.

1.1. Aims and structure of the paper

The aim of this paper is to present an agent architecture, DiPRA (Distributed Practical Reasoning Architecture), which couples realistic embodiment with mechanisms of intentional action control that are more sophisticated than those offered by most robotic systems. DiPRA is a two-layer architecture. Higher level cognitive operations are realized in the intentional layer, whose design is inspired by Bratman et al.’s (1988) idea of practical reasoning, that is, a reasoning process that determines what one is to do, and steers action. Specifically, the higher layer realizes deliberation (i.e., the process of deciding what to do) and means-ends reasoning (i.e., the process of deciding how to do it), which jointly permit the selection of and planning of distal goals, or those not dictated by immediate affordances (Gibson 1979) in the environment. Moreover, it naturally implements replanning (changing plan to achieve the same goal), intention reconsideration (changing the intended goal), and commitment to intentions (pursuing intentions in the face of changing contextual conditions), which are fundamental abilities of practical reasoning systems. DiPRA includes also a sensorimotor layer composed of multiple sensorimotor schemas, roughly the equivalent of Norman and Shallice’s (1986) schemas. The sensorimotor layer is responsible for situated action in open-ended environments and, as we will discuss, is essential for ensuring grounding of knowledge manipulated at the higher layer.

The rest of the paper is structured as follows: sec. 2 offers a conceptual analysis of the main challenges faced in designing multilevel control architectures, and how DiPRA’s design takes them into account. Sec. 3 describes the DiPRA architecture, and sec. 4 illustrates the implementation of a DiPRA agent: a simulated robot (with realistic embodiment) that plays the role of a ‘thief’ in a guards-and-thieves scenario. Specifically, in an artificial ‘house’ (a 3D realistic virtual environment) with two floors and nineteen rooms, the thief has to locate and steal a ‘treasure’ and at the same time escape from one or more ‘guards’; see fig. 6. Two experiments are described in sec. 4 which demonstrate DiPRA’s ability to balance among levels of control and to acquire grounded knowledge which is then used to guide further reasoning and deliberation. Then, in sec. 5 we discuss the solutions we adopted in the design of DiPRA in the light of experimental results, and in sec. 6 we draw our conclusions.

2. Willed and automatic control of action: challenges and possible solutions

A schematic illustration of a generic layered architecture, which integrates willed and automatic levels of control, is provided in fig. 1.

In the lower layer (automatic control) actions are selected and executed whose outcomes satisfy the agent’s active drives by exploiting current affordances in the environment or producing novel ones to be exploited successively. At the same time, sensory feedback is used for triggering further action selection as well as for action control. This process is sufficient for an automatic selection of action and the satisfaction of a basic repertoire of drives. The higher layer (willed control) is instead responsible for intentional action. Here a (practical) reasoning process is responsible for selecting the most adequate plan on the basis of active goals, background information, and novel knowledge (which is continuously extracted from
the environment). Planned courses of action are then executed that achieve the agent’s goals—in most architectures this is done by reusing the same ‘machinery’ that supports automatic action.

Most of the layered architectures above described are (partially or totally) compliant with this basic architectural scheme, although of course the details of their implementation differ; for example, contention scheduling is just one of the possible realizations of action/schema selection. More important for our conceptual analysis is the fact that designers of layered architectures face another difficulty: ensuring the integration of components and processes in the two layers. As we have seen, some layered architectures adopt different representational formats in the two layers, which makes communication between them problematic and potentially prone to computational bottlenecks. This is not the main obstacle to integration, however. In the rest of this section we analyze the three main challenges of multilevel control of action: integration of levels of control, management of perceptual/attentional strategies, and knowledge extraction and grounding. Then, we briefly introduce DiPRA’s design solutions to tackle those problems.

2.1. Integration of levels of control

One problem that arises in the realization of architectures with multiple levels of control is how to manage their dependencies and possible conflicts. For example, how to balance automatic action selection processes and deliberate planning processes, which have different sources (stimulus-driven vs. goal-driven), operate on different time scales, and possibly rely on different representations and aim to achieve proximal or distal objectives. Too much top-down, goal-directed control makes the agent not opportunistic and unable to react promptly, while too much bottom-up, stimuli-driven control makes the agent reactive and unable to achieve distal goals, or those that are not dictated by current environmental affordances. Ideally, an intentional agent should be able to remain committed to its intentions without losing situatedness and opportunism. However, this raises several questions: how much reasoning should be performed before acting? How specified
should plans be, and how much should be left to on-line action control? How is it possible to be committed to the intended goal and plan, and at the same time to remain responsive to unexpected, potentially dangerous events which demand for an immediate reaction? It is easy to understand that all these problems are not simply solved by sharing a common representational format across layers, but it is necessary to ensure that they interact efficaciously and with coordinated timing.

2.2. **Managing perceptual and attentional processes**

There is a conflict between goal-directed and stimulus-driven processes, not only in the selection and control of action, but also in the guidance of perceptual and attentional strategies. Reactive agent architectures are typically designed, instructed or evolved to select stimuli from their environments that are relevant for their current action. In fig. 1 this is evident in the loop between action execution and sensory feedback. The former produces the latter, and the latter influences further action selection and control. In other words, actions are executed that unravel salient stimuli events, which in turn trigger further actions, and so on.

In intentional agent architectures this is not sufficient, however, since representing the current context is not enough. Recent research in neuroscience has shown that knowledge of future states, and in particular action outcomes and goals, is a crucial part of the cognitive representation that underlies action preparation (Jeannerod 1997), the ‘covert’ part of action execution. As a consequence, it is often necessary to direct attention to future states of affairs (e.g., obstacles in a trajectory), and to actively search for goal-related stimuli (Land 2006). Moreover, deliberative agents need information not only for acting here and now, but also for deciding what goal to achieve or which plan to follow; for instance, they (often) need to know what goal or plan preconditions are currently available. Overall, intentional agents need a multiplicity of anticipatory attentional and information-seeking strategies which can potentially conflict with the demands of situated action, and it is necessary to ensure that when they pay attention to distal events they do not disregard proximal ones.

2.3. **Knowledge extraction and grounding**

A related problem is that in most cases information required for making decisions and taking future action cannot simply be extracted from the perceptual environment and used on-line with action, but must be internally represented and maintained. In this vein, most layered or practical reasoning systems such as BDI explicitly represent and manipulate semantic knowledge in their higher layers (see the edge labeled ‘knowledge extraction’ in fig. 1). This poses yet another problem, however, that of guaranteeing the quality of such knowledge.

It has recently become evident in the cognitive science community that one central issue in the design of knowledge-based systems is the so-called symbol grounding problem (Harnad 1990), that is how representations of natural and artificial organisms acquire reference and aboutness. In other terms, it is important to guarantee that conceptual knowledge manipulated at the higher layer (either acquired au-

---

1This is reminiscent of the problem of intentionality (Brentano 1985); how an organism’s representations and knowledge are “of” or “about”. Note that here ‘intention’ is meant in the technical sense of ‘aboutness’, or the characteristic of representations or mental states ‘to be about something other’, not in the sense of intentional, goal-directed action. Similarly, Coradeschi and Saffiotti (2003) proposed the anchoring problem, that is how to assess the truth value of symbols explicitly manipulated on the basis of the current state of the sensors.
tomatically or manually inserted) refers appropriately to the world so that it can be used by the agent as a basis for decision and action. Several approaches have been proposed for finding adequate ‘correspondence rules’ to map knowledge in the two layers. For example, Harnad (1990) proposes a sensorimotor solution to the grounding problem, which consists in deriving (conceptual) entities belonging to the intentional level, including beliefs, from states of the sensors (often clustered with statistical or similar techniques). This approach is based on the idea that conceptual entities are caused by perceptual states, thus the former should be grounded in the latter. However, it is unclear how to ground beliefs such as expectations or abstract concepts, which cannot be simply inferred from the current state of the sensors. Other authors propose instead the idea of a ‘loop’ between external observations and internal knowledge revision processes in which the prediction of external events is a key component (Bouguerra et al. 2007, Pezzulo 2008a, Roy 2005)—an approach we will elaborate in this paper.

Providing an agent with knowledge leads to a number of additional problems. The first is related to the amount and relevance of knowledge that is internally manipulated. Although for most cognitive tasks artificial systems need to acquire and manipulate knowledge, it is unnecessary—and impossible—for them to internalize a complete explicit representation of the world (Clark 1998). Even if this were possible, they can use only a part of such knowledge, since they have bounded resources. Intuitively, only the relevant part of information has to be considered. Unfortunately, the well-known frame problem, originally proposed by McCarthy and Hayes (1969), indicates that it is extremely difficult for an artificial system to determine which part of its knowledge is relevant given its current context and goals. Furthermore, it is worth noting that for reasoning and deliberation processes it is useful to exploit knowledge about both present and future state of affairs. In order to be ‘attuned’ to the current situation or, in short, situated, an agent must be able to process knowledge that is up-to-date and acquired at the right time, otherwise it is useless or maladaptive, as pointed out by Brooks (1991) in his criticism of representation-based AI. Moreover, to take intentional action an agent has to know which of its possible actions are more likely to produce effects that will eventually achieve its goals. Although some means-ends relations are stable, it is often the case that the same action can produce different effects under different circumstances, so knowledge of action effects needs to be contextualized. Overall, this means that any knowledge-provided AI system has to continuously ensure the grounding and relevance of several kinds of knowledge, including the present state of affairs and the possible outcomes of its actions; otherwise its deliberative processes will be maladaptive.

2.4. DiPRA’s design principles

To tackle the aforementioned problems of integration of levels of control, management of perceptual and attentional processes, and knowledge extraction and grounding, in the design of DiPRA we introduced four key design principles, which we briefly introduce here.

2.4.1. Division of labour between off-line planning and on-line action

We have discussed how goal-directed action aims at realizing distal (desired) effects. However, in order to execute any voluntary action it is necessary to specify not only its goal but also the details of its execution. Since details vary depending on contextual conditions, they cannot be part of the goal specification, but must be filled in (or at least updated) later on. To do so, it has been proposed in the psy-
chronological and neuroscientific literature on planning and motor control that there is a division of labour between off-line planning and on-line specification (Hommel et al. 2001, Jeannerod 1997). This makes it possible to translate intentions into actual action, and at the same time to remain responsive to opportunities from the environment. To account for such division of labour, in DiPRA planned behavior is realized by the same sensorimotor schemas which subserve automatic action. Plans consist in the specification and activation of sequences of sensorimotor schemas, which are themselves responsible for on-line action specification, monitoring and fine-tuning; this means that environmental cues cause automatic adjustments and in some cases deviations from the planned courses of action.

2.4.2. Regulation of epistemic processes with a loop between the two layers

DiPRA’s attentional and epistemic dynamics are regulated by a continuous loop between the two layers, bottom-up (from the sensorimotor to the intentional layer) and top-down (vice-versa). Bottom-up processes permit the acquisition of grounded and up-to-date knowledge, and the translation of current action opportunities into knowledge to reason about. This is realized thanks to a continuous self-monitoring process: beliefs monitor the activity level of schemas, and knowledge is revised and updated when the sensorimotor context changes. At the same time, top-down, predictive processes can trigger epistemic actions, that is actions which aim to acquire information from the environment (Kirsh and Maglio 1994). Some examples include a shift in attention or mental simulations of possible actions (Jeannerod and Decety 1995); these strategies allow to check plans and actions preconditions, or to forecast their effects. Overall, we will argue, this attentional and epistemic loop permits agents to obtain knowledge both grounded in the current (or simulated) sensorimotor interaction and relevant for the agent’s goals. Essentially in DiPRA it is the grounding and relevance of knowledge, and not a homogeneous representational format, that permits layers integration despite their different demands and time scales.

2.4.3. Bounded resources allocation

Another important aspect of DiPRA is that the selection of goals, plans and schemas is regulated by a uniform mechanism that allocates bounded (computational and knowledge) resources on the basis of an (implicit) estimate of how reliably they can carry out their expected results—their achievability—and the desirability of such results. Inspiration for such mechanism comes from two sources. First, the biological literature suggests that decision and action selection are competitive processes (Prescott et al. 1999) based on a comparison of expected action outcomes (proximal and distal), and their value for the organism. Accordingly, in DiPRA the current motivational context, i.e. the value of active drives and goals, is responsible for the desirability dimension of selection: highly-valued drives and goals influence action by activating action schemas or plans respectively. Second, an authoritative view in motor control indicates that (sensory) prediction is the key mechanism for selection (Wolpert and Ghahramani 2004), since it provides a context-sensitive measure of the achievability of actions. Accordingly, in DiPRA a mechanism that favors goals, plans and actions whose predictions are correct is adopted in both layers to account for the achievability dimension of selection:

- Schemas in the sensorimotor layer compete on the basis of their prediction accuracy, that is, on the basis of how reliably they are expected to carry out their desired results in the current sensorimotor context.
- Similar to how action schemas compete on the basis of how much support they get from (predicted) sensory stimuli, goals and plans in the intentional
layer compete on the basis of how much support they get from (predicted) beliefs. Since beliefs are grounded in sensorimotor expectations, goals and plan ultimately compete on the basis of how reliably they are expected to realize desired outcomes through action.

The same mechanism of (bounded) resources allocation regulates how much priority is given to reasoning or sensorimotor processes and the shifting of levels of control (intentional vs. automatic) due to contextual conditions.

2.4.4. Realization of reasoning and deliberation as intertwined processes

Finally, unlike traditional models of practical reasoning such as BDI, in DiPRA means-ends reasoning and deliberation are intertwined processes. Means-ends chains for multiple candidate goals can be constructed in parallel that trigger the unraveling of novel information and feedback on the goal selection process when their achievability is assessed or novel (positive or negative) side-effects are taken into account. This is consistent with recent evidence indicating that the planning and specification of multiple possible courses of action and their selection occur concomitantly at the neural level as a competitive race (Cisek 2007, Cisek and Kalaska 2005). In order to avoid a combinatorial explosion, an incremental mechanism (based again on bounded resources) ensures a parsimonious exploration of the most promising means-ends chains first.

3. DiPRA: Distributed Practical Reasoning Architecture

DiPRA (Distributed Practical Reasoning Architecture) is a two-layer architecture; see fig 2. It includes an intentional layer performing practical reasoning (Bratman et al. 1988): selecting the current intention by choosing among multiple goals and sub-goals (such as ‘search the treasure’ or ‘reach room 11’) on the basis of current beliefs, and selecting a plan to adopt for the purpose of realizing the current intention. It also includes a sensorimotor layer, which provides interactive, situated capabilities by means of specialized schemas, each realizing an action (such as ‘grab the treasure’ or ‘avoid obstacle’), each having a variable activity level, and each including an anticipatory mechanism (internal forward model, see Wolpert and Kawato 1998) permitting to predict the effects of its actions, actual or potential1.

The two layers of DiPRA, higher and lower (above and below the dotted line in fig. 2), are seamlessly integrated and interact in two ways. First, plans belonging to the intentional layer can trigger sequences of schemas belonging to the sensorimotor layer (see the downward arrow in fig. 2). Moreover, knowledge manipulated in the intentional layer (e.g., a belief) is obtained by monitoring the activity level of schemas at the sensorimotor layer (see the upward arrow in fig. 2). For example, when the activity level of a schema for passing through doors is high (that is, when its accuracy in prediction is high), the belief ‘the door is open’ is derived. In the intentional level, beliefs can be obtained and manipulated not only when concerning the current sensorimotor context, but also about future expectations. In order to produce such expectations, schemas can be re-enacted ‘in simulation’, that is by chaining their predictions to produce long-term predictions for an arbitrary number of steps in the future. For example, the belief ‘the door is open’ can be

---

1All DiPRA components should be interpreted as belonging to a functional level of description, which is the typical level of schema-based architectures (Arbib 1992), and no attempt is done here to identify putative brain mechanisms or their anatomic equivalents (but see Wolpert et al. 1998 for a discussion of the brain substrate of forward and inverse models in sensorimotor schemas).
produced even if the agent is not passing the door, but (from a certain distance) simulates passing through it and the simulation runs successfully (if unsuccessful, the belief ‘the door is closed’ can be produced instead). This means that the belief ‘the door is open’ corresponds to an agent’s expectation that if it tries to pass through the door, its attempt will succeed.

3.1. An overview of the architecture

DiPRA is a modular and parallel system comprising several components. The components of the intentional layer are: the Reasoner, Goals, Plans, Beliefs, and Actions. The component of the sensorimotor layer are the Schemas. Each component is implemented as a concurrent module in the multi-thread framework AKIRA (www.akira-project.org). Each module has an arbitrary content and a variable activity level determining its computational resources: its thread’s priority, that in turn determines speed of execution and priority over sensors and actuators. Another relevant feature, that is essential for understanding DiPRA’s resources and knowledge boundedness, is that all modules share (and compete for) limited computational resources, stored in a component called the energy pool. At the beginning of each cycle, modules receive activation only if this is effectively available in the energy pool, otherwise they have to wait. This means that only some modules can be active at the same time. Finally, modules can communicate by exchanging symbolic messages (via a blackboard) or by exchanging activation (see Pezzulo and Calvi 2007a for details).
Before describing DiPRA’s components, here we sketch its functioning. In the *intentional layer* (above the dotted line in fig. 2) each Belief, Goal, Action and Plan is a module operating asynchronously with variable activity levels (see later) and is connected to other modules (such as: *Belief β supports Goal γ*). A special module, *the Reasoner*, maintains a consistent representation of the modules’ activity level and their relations by using a Fuzzy Cognitive Map (FCM) (Kosko 1986). As long as the agent acts or reasons new beliefs are added to the intentional layer. The main roles of the Reasoner are selecting one Intention among the alternative goals, adopting a suitable plan for achieving it, and finally activating actions. Consistently with the idea of practical reasoning, this is done on the basis of current beliefs (see Castelfranchi 1996).

Actions are then realized by specialized modules (schemas, e.g., ‘escape from guard’, ‘avoid obstacle’) that belong to the *sensorimotor layer* (below the dotted line in fig. 2). Schemas compete to send commands to the two actuators (camera and wheel controllers), and their activity level controls the ‘gating’ of their motor commands. A schema’s activity level increases under three conditions: (1) when the schema receives activation for plans (top-down way), (2) when the schema receives activation from active drives, and (3) when the schema’s forward model generates accurate predictions, indicating that it is appropriate to the current sensorimotor context (bottom-up way). For example, the schema ‘detect guard’ raises its activity level when it successfully predicts sensory evidence of a guard.

At the same time, schemas’ activity level determine the values of beliefs and other declarative components in the Reasoner: the activity level of schemas and their success or unsuccess are routinely checked by the corresponding declarative components of the intentional layer, such as plan conditions or other beliefs, which then update their values (and activity level) in the FCM. For instance, a Belief (e.g. ‘door is open’) can be verified by the success of a schema (e.g., ‘pass door’). Beliefs can also check their truth value by running schemas ‘in simulation’ (e.g., imagining passing through a door). The agent’s final behavior depends on an interplay of top-down pressures determined by plans and ‘mental simulations’ (in the intentional layer) and of bottom-up pressures of schemas which exploit current affordances in the environment.

As stated above, the whole system has a limited and fixed amount of computational resources, which means that all active processes compete to stay active. For example, when the FCM builds complex means-ends chains, it activates more modules in the intentional layer and as a side-effect the sensorimotor layer has fewer resources. Another side-effect of the boundedness of resources is that means-ends chains relative to currently active goals prevent others from being formed simultaneously—this means that the FCM avoids combinatorial explosion by exploring the more promising directions first. On the contrary, when events in the environment happen which demand rapid responses (e.g., dangerous events), the drives system ensures that more activation is supplied to schemas in the sensorimotor layer. Similarly, action schemas can be prepotentiated by current affordances. As a consequence, various contextual conditions determine shifts in the levels of control, intentional or automatic.

---

1For the sake of simplicity, in the current implementation, there is a 1-to-1 mapping between declarative components and schemas (e.g., the belief ‘guard in sight’ depends on the activity level of the schema ‘detect guard’). In principle, however, more complex mappings between an agent’s epistemic states and its actions can be designed or learned.
3.2. The intentional layer

The intentional layer includes modules that locally encode goals, plans, actions, and beliefs. As in the rest of the architecture, modules operate in parallel and asynchronously, and have a fixed life cycle with a variable amount of computational resources and speed, depending on their activity level. For example, goals continuously check their satisfaction, plans check their preconditions and activate actions, etc. For the sake of simplicity, a consistent representation of the intentional layer is maintained by a centralized component, the Reasoner.

Here we informally describe the main components of the intentional layer and their functioning; a formal description is provided in appendix A.

3.2.1. The Reasoner

The Reasoner maintains a consistent representation of the activity of all the modules and performs deliberation and means-ends analysis using an additive fuzzy system called FCM (Kosko 1986) whose nodes and edges represent the goals, plans, beliefs and their links respectively, and in which activation spreads among nodes. Sample FCM are shown in fig. 2 (above the dotted line), fig. 8 and fig. 9. The value of each edge is set according to the activity level of the module from which it originates, representing the impact of a given node over another. For example, if a belief sustains a goal (a positive link) and the value of the belief module is +0.7, the edge from the belief to the goal is set to +0.7. Some beliefs play the role of plan conditions, too, and sustain plans.

The values of nodes in the FCM represent their contextual relevance. The value of Beliefs (including goals and plan conditions) is fixed; the value of Plans and Goals is variable and calculated anew (by the FCM) at the beginning of each cycle of the Reasoner. Goals and plans therefore increase their value and their impact as long as they are increasingly supported by beliefs, or by other goals and plans. For example, an achievement goal that is close to satisfaction is likely to be linked to several supporting beliefs and highly active plans; therefore, its value increases and it increasingly inhibits the other goals. Overall, agents’ choices are determined by their beliefs, and intention-to-action hierarchies in the intentional layer lead to selection of its most appropriate plans and actions. Practical reasoning is then implemented as a dynamic interplay of means-ends reasoning (that builds causal chains in the FCM) and deliberation (that exploits them for assigning values to goals and plans, and then selecting them).

3.2.1.1. Cycle of the Reasoner. As illustrated in fig. 3, all Beliefs routinely check the activity level of corresponding sensorimotor schemas and update their values in the FCM; only those having a non-null value are considered. In turn, this leads to the selection of a plan, which therefore activates a sequence of schemas. Again, feedback from schemas’ activity further activates other beliefs.

The Reasoner runs concurrently with all other modules. Since it has a fixed activity level, it can never exceed a certain size. This feature is very useful for keeping means-ends analysis bounded: at the beginning, only top-level plans are considered in the FCM, which is filled in with subplans only as new modules become relevant due to current reasoning or action. Knowledge augments in a bounded way, too, as long as conditions and Beliefs related to active Plans, Actions and Goals are checked. The final result of computation in the intentional level is intending a goal and adopting a plan: the FCM continuously runs and sets the value of nodes. The most active goal becomes the Intention and the most active plan realizing that goal becomes the Adopted Plan. Fig. 4 shows that, depending on the current value
Figure 3. A sample loop between intentional and sensorimotor layers. Phase 1: in the intentional layer, two plans compete to be selected. Phase 2: a belief which supports Plan 1 monitors the activity level of its corresponding schema at the sensorimotor layer, getting a positive value and then determining the selection of Plan 1. Phase 3: Plan 1 in turn activates one sequence of schemas.

of an agent’s beliefs, different goals can win this race and possibly be intended. For example, in case 1, thanks to the fact that the belief guard in sight has value $-1$, the goal have V has value 0.63 and wins, thus being intended. In case 2, instead, guard in sight has value 1 and as a consequence the goal escape G wins and is intended.

Figure 4. Different intentions are selected on the basis of different beliefs. In the first case the goal to collect the valuable is selected (and becomes the new Intention). In the second case the goal to escape the guard is selected, due to a belief indicating the presence of a guard.

Intention selection has two consequences:

- Intended goals and adopted plans in the FCM receive recurrent connections, and thus self-reinforcement. This represents the fact the agent is committed to pursuing its intentions and goals despite slight contingent variations in its beliefs—a key characteristic of intentional agents (Bratman et al. 1988).

- Once a new plan is adopted, it activates sequences of schemas in order to realize it (see the downward arrow in Fig. 2). This biases the agent’s behavior in a top-down manner; still, the system remains responsive to stimuli such as obstacles or novel events that are directly managed at the sensorimotor level. This division of labor between off-line plan specification and on-line adjustment allows agents to couple intentionality and responsiveness.

The Reasoner’s cycle has two main tasks: (1) to deliberate (select a goal and a plan) and (2) to set the activation of the modules. Both are realized through this cycle:

(1) Set the values of the FCM nodes according to the activity level of the corresponding modules, and their links.

---

1 Another, weaker form of commitment is implicitly realized by the system dynamics: a goal which is close to satisfaction inhibits the other goals, thanks to the feedback from plans succeeding and from conditions that are met.
(2) Run the FCM and set the nodes value. This value represents the contextual relevance of the corresponding modules.

(3) The most active Goal is selected (if over a threshold); if not already achieved, its status becomes Intended (new intended Goals replace old ones). Otherwise, another Goal has to be selected. A recurrent connection (hagin weight $\theta$) is set for the Intended Goal, which gains activation.

(4) The most active Plan for the intended Goal is selected (if over a threshold); its status becomes Adopted, and its termination conditions are set according to the intended goal. If there is an already adopted Plan, it is stopped only if its conditions conflict with those of the new adopted one. A recurrent connection is set for the Adopted Plan.

(5) If no Plans are possible for the intended Goal, its status becomes Waiting (and the recurrent connection is maintained); a new Goal has to be intended until the former becomes achievable (this is unlikely, since the evaluation of a Goal also depends on how suitable its plans are).

(6) If no goals or plans are over the aforementioned thresholds, said thresholds lower and the cycle restarts; otherwise, modules’ activity level are set according to the values of corresponding nodes in the FCM. As a consequence, the Reasoner resets the modules’ activity level only when a new Plan is adopted.

3.2.2. Beliefs, Conditions and Goals

For the sake of simplicity, we chose to use a uniform representational format for all declarative components: fuzzy logic (Kosko 1986, Zadeh 1965). All conditions (goal conditions, pre and post conditions of plans and actions, etc.) are special kinds of beliefs. For example, a Belief (“room 11 is far”) can be matched using fuzzy rules with the Precondition of a Plan (“room 11 is close”) to generate a graded truth value. Goal conditions also share this formalism; in this way they can be matched e.g. against postconditions in order to verify their satisfaction (e.g. the Goal “go to room 11” becomes more and more satisfied when the truth value of “room 11 is close” increases). There are separate policies for achieve and maintain goals. In Achievement goals (e.g., “reach room 11”), contextual relevance increases when the truth value of the goal postcondition increases. In Maintain goals (e.g., “stay close to room 11”), contextual relevance lowers when the truth value of the goal postcondition increases.

3.2.3. Plans

Plans are the main control structures in DiPRA. Plans are activated for satisfying an intended goal; once the plan is adopted, a subset of their postconditions is set as Goal. A Plan is basically an execution scheme, activating Actions and Goals from the ActionSet and subgoaling. This is their behavior:

- If the intended Goal is already achieved, the Plan stops immediately and no action is executed.
- If any plan precondition is false, the Plan ‘delegates’ its satisfaction to other modules by passing activation to them; subgoals activated in this way gain the status of Instrumental.
- If all preconditions are met, the Plan starts executing its sequence of actions, or activating subgoals; therefore, (sub)goals activate (sub)plans or actions, and so on. Goals activated in this way also gain the status of Instrumental.

---

2Goals, plans, actions and beliefs can also be more or less relevant in absolute. This is represented by the AbsRel value, which is also the value of a recurrent connection of the corresponding FCM node (not shown in Fig. 6).
Plans continue subgoal-ing and executing until all possible actions and subgoals fail. A failed plan returns control to the calling goal, which remains not satisfied and activates another plan. However, very often unsuccessful plans are stopped before exhausting all possibilities. If several conditions of a plan fail, the corresponding ‘branch’ in the FCM is doomed to weaken despite commitment, and other plans can replace it.

3.2.4. Actions

An Action is the minimal operation which is executable by plans. In the current DiPRA implementation, each action simply corresponds to one schema in the sensorimotor layer: plans activate sequences of actions by transferring activation to the corresponding schemas.

3.3. The sensorimotor layer

Like most schema-based systems (Arbib 1992), the sensorimotor layer includes several sets of sensorimotor schemas, each specialized for a basic behavior: navigation, treasure finding and grabbing, guards recognition and escape, and obstacle avoidance (see fig. 2, below the dotted line). For each behavior there are several schemas, each specialized for a sub-context, which cooperate and compete for control of the agent’s actuators. For example, schemas for navigation are specialized to know and operate in a portion of the environment.

The sensorimotor layer includes drives, as well, which play the role of basic motivational components of the architecture (Hull 1943) as opposed to more complex motivations such as goals, which belong to the intentional layer.

3.3.1. Schemas implementation

We designed schemas of two main kinds: perceptual schemas and motor schemas. Perceptual schemas (e.g., detect treasure and detect guard) control the vision of the agent by moving a camera (the only sensor). Motor schemas (e.g., catch treasure and escape guard) receive sensory input from related perceptual schemas (e.g., escape guard from detect guard) and control the movements of the agent by moving its wheels. Moreover, each perceptual schema sends sensory information to one (predefined) set of motor schemas. Typically several schemas are active, at different levels, in each situation. For example, if the agent is escaping from a guard, several schemas specialized for different ways to fulfill this task (e.g. escaping from slow or quick guards) can be active at once. Moreover, several schemas can be active for different reasons (e.g., a top-down activation by a plan and a bottom-up guard detection). Commands of all active schemas are fused, so that the actual agent’s behavior results from the sum of schemas’ (weighed) contextual pressures.

Fig. 2 (below the dotted line) shows the schemas model used in our experiments. Each schema is a complete sensorimotor unit: it includes specific (neural) circuits for processing stimuli and generating motor commands (inverse model) coupled with a predictive component for generating sensory predictions (forward model), see (Pezzulo and Calvi 2006, Tani 1996, Wolpert and Kawato 1998). For the sake of simplicity, in our implementation the inverse and forward modeling functionalities are integrated in the same neural network, a Jordan-type RNN (Jordan and Rumelhart 1992) illustrated in fig. 5 (right), which includes a sensorimotor loop and a context loop and permits both the control of action (at time $t$) and the prediction of its sensory effects (at time $t+1$). The sensory input node represents three nodes (encoding positions of object features in the three axis, $< x(t), y(t), z(t) >$) and the motor output node represents three nodes (encoding $< vx(t), vy(t), vz(t) >$,
i.e., the motor command for reaching \( <x(t+1), y(t+1), z(t+1)> \). See (Pezzulo 2008b, Pezzulo and Calvi 2006) for further details.

Figure 5. Schemas implementation. Each schema is implemented as a Jordan-type RNNs (Jordan and Rumelhart 1992) that realizes inverse and forward modeling and includes a context loop. See explanation in the text. It includes 3 sensory inputs S(t), 3 sensory outputs, 3 motor inputs M(t), 3 motor outputs, 2 context inputs C0(t) and C1(t), 2 context outputs, 10 hidden nodes.

3.3.1.1. Generation vs. Simulation. Schemas run in two modes: generation or simulation. Generation mode is the default. After preprocessing (features extraction) operated by hand-coded routines (see Pezzulo and Calvi 2006) the RNN receives the estimated position \( <x(t), y(t), z(t)> \) of the feature it is specialized to deal with (e.g., a red shape) with from the camera (in the case of perceptual schemas) or a perceptual schema (in the case of motor schemas), produces a motor command \( <vx(t), vy(t), vz(t)> \) and sends it to the camera (or wheels) controller. The RNN also receives an efference copy of the final motor command executed by the camera (or wheels) controller and generates the sensory prediction \( <xp(t+1), yp(t+1), zp(t+1)> \).

Any schema can run off-line in simulation mode, too, for predicting the long-term sensory consequences of its motor commands: this is a form of mental simulation of action (Jeannerod 2001). In this case its motor commands are inhibited (not sent to the actuators), but fed as sensory inputs to the forward model. It then produces new sensory predictions that are used by the inverse model for generating a new motor command ‘as if’ the agent actually sensed the predicted future. The loop between forward and inverse models allows the generation of long-term predictions for an arbitrary number of future steps, such as in imagery. The simulation mode is used in the experiments in sec. 4 for producing beliefs about possible future actions.

3.3.1.2. Schema Activity Level. Several schemas can be active at once. Each schema executes its operations asynchronously and with different speed, depending on its current activity level (calculated anew at the beginning of each cycle). More active schemas receive more up-to-date sensory information, and send motor commands with higher firing rate. Activity level is calculated as follows:

\[
rel + links + plan \text{ (normalized in } [0,1]) \text{ if this sum is larger than pool, otherwise to pool, where:}
\]

\[\]

\[^{1}\text{In the current implementation only navigation schemas can run in simulation. However, the same methodology has been used in related studies (Pezzulo 2008b) for the realization of more complex forms of imagery.}\]
• rel indicates how much the schema is expected to be successful in the current context; in our model (like in Wolpert and Kawato 1998) this depends on its prediction accuracy: position sensed and predicted (by the forward model) are compared, and rel is set to \((1 - || < x(t), y(t), z(t) > - < x_p(t), y_p(t), z_p(t) > ||)\).

• links is a contextual parameter that depends on the activity level of other schemas and drives (see below).

• plan is activation received by plan units (top-down influences);

• pool is the amount of resources currently available in the limited pool.

Each schema has a reliability value, which depends on its accuracy in prediction: schemas that predict better are considered to be ‘well attuned’ to the current situation and thus reliable. On the basis of reliability, the activity level of (the module corresponding to) each schema is assigned: position sensed and predicted (by the forward model) are compared, and rel is set to \((1 - || < x(t), y(t), z(t) > - < x_p(t), y_p(t), z_p(t) > ||)\). More active schemas send more frequently motor commands to the motor wheels and camera, and therefore influence both pragmatic (where to go) and epistemic one (which stimuli are attended to) activities. The schemas’ activity level can also be raised by drives, or by plan units belonging to the intentional layer.

Each schema has a threshold \(thr\) (set to 0.3 in our simulations). If \(act < thr\) the schema functions normally but its motor commands to the actuator are inhibited. (See Pezzulo and Calvi 2007b for a more detailed specification of the parameters.)

3.3.2. Implementation of drives

The sensorimotor layer includes also two internal, motivational states, fear and hunger, modeled as simple homeostatic variables whose satisfaction is a primary source of behavior (Hull 1943). Hunger is raised (by 0.1 every 10 cycles) by a ‘biological clock’, and is set to 0 when the treasure is grabbed. Fear is set to 1 when a guard is detected, and decreases (by 0.1 every 5 cycles) otherwise. This introduces a slight bias toward potentially dangerous events in the environment, which is reasonable in open-ended environments. However, as described above, fear and hunger develop associative links with schemas via Hebbian learning, so their values are updated depending on contextual conditions. Their main roles are steering behavior (by activating schemas for escaping guards and reaching the treasure respectively when the context is appropriate), and creating positive feedback and persistence (i.e., hysteresis). A Fuzzy Cognitive Map (FCM) (Kosko 1992), a hybrid neural network - fuzzy system, is used for calculating the drives’ activity level (in \([0,1]\)): fear and hunger are modeled as two nodes in the FCM and have mutually inhibitory links set to -0.6.

4. The Guards and Thieves Scenario

We tested DiPRA’s efficacy in two experiments in a guards-and-thieves scenario, with the DiPRA architecture playing the role of the thief. The environment is a 3D simulated house with 2 floors and 19 rooms (including 2 stairways) whose positions vary between -10000 and +10000 in the three axes. In the house are both a guard and a thief; see fig. 6. Guard and Thief have the same size and speed, and a limited range of vision (an angle of 90°). The use of the Irrlicht 3D simulation engine (irrlicht.sourceforge.net), with its realistic physics based on ODE (www.ode.org), ensures that agents embodiment (although simplified with respect to most robotic platforms) is appropriate for our objectives.
4.1. Experimental setup

In both experiments the guard \( G \) is modeled as a simple schema-based system with 2 drives, to protect the valuable \( (V) \) and to catch the thief \( (T) \) and a repertoire of 4 schemas (2 perceptual and 2 motor) for satisfying these drives. Drives and schemas are encoded by hand and there is no learning; overall, the capabilities of the guard are comparable to those of most bots (i.e., computer controlled players or opponents) in computer games.

We are instead interested in evaluating the capabilities of the DiPRA layered architecture introduced in this paper. For doing so, in Experiment 1 we compare the architecture of two thief types: the thief architecture \( T_I \), which is a full fledged DiPRA implementation (see fig. 2), and the simplest thief architecture \( T_S \), which has the same sensorimotor layer as \( T_I \) but a simplified intentional layer. In Experiment 2 we compare instead the thief \( T_I \) with alternative agent architectures.

4.1.1. Encoding of the Intentional Layer of the thief \( T_I \)

The thief \( (T_I) \) is the one shown in fig. 2. It is provided with an intentional layer, consisting of 2 goals \( (\text{collect valuable} (V) \text{ and } \text{escape guard} (G)) \) and 21 plans for realizing them. The initial values of the 2 goals of \( T_I \) are set by hand: \( \text{escape} G \) is 0.7 and \( \text{collect} V \) is 0.5, reflecting the fact that when both goals are achievable the former should have priority. The whole FCM of \( T_I \) (see examples in Figs. 2, 4, 8 and 9) was designed by hand, reflecting the fact that it already knows the map of the environment and in particular how to reach each room (we have empirically estimated that 21 plans, which can be merged and chained, are sufficient to ensure good navigation capabilities in the house). Associations between plans and schemas in the sensorimotor layer are set up by hand, too. Although machine learning techniques can be used for learning navigation plans autonomously (as in Pezzulo 2008b), this is outside the scope of this paper. See Appendix A for parameter specification.

4.1.2. Learning of the Sensorimotor Layer of the thief \( T_I \)

\( T_I \) also has a sensorimotor layer that includes a repertoire of 24 schemas (12 perceptual and 12 motor) for realizing actions planned in the intentional layer. These include navigation schemas for reaching all rooms, as well as for detecting and escaping the guard, detecting and catching the treasure, and avoiding obstacles (see fig. 2, below the dotted line). A learning phase is necessary for setting up schemas reliable enough in forward and inverse modelling, and for coupling schemas and drives.
4.1.2.1. Schemas learning. Perceptual and motor schemas for tracking and reaching (or avoiding) guards, treasure, and obstacles are learned with a supervised methodology after features extraction (a facility which is commonly available in the Irrlicht 3D simulation engine and permits to extract objects size, color, and shape). Specifically, each schema learns to predict the next sensory position \( < x_{p(t+1)}, y_{p(t+1)}, z_{p(t+1)} > \) of one feature based on its estimated position \( < x(t), y(t), z(t) > \) and the agent’s last motor command \( < vx(t), vy(t), vz(t) > \). To do so, during learning treasure and obstacles appear in fixed locations and guards follow predictable trajectories. Circular and oval trajectories having different amplitudes were used. All entities have quite distinguishable features (overlapping 20%) in order to facilitate learning. The RNN of each schema is trained with the BPTT algorithm Rumelhart et al. (1986). One example (of guard, treasure, or obstacle) was sampled every twelve, with a total of thirty-six; learning stopped when the error of at least one forward model (the Euclidean distance between the actual and predicted position in 3D, \( 0.1 \times 10^{-6} \)) was less than 0.0000001.

The methodology for learning perceptual and motor schemas for navigation is more complex, since it proved to be impossible to learn accurate schemas for navigating the whole house, or large portions of it. For this reason, we adopted a learning procedure that permits subgoals (i.e., parts of the map that can be learned successfully) to be learned individually. The procedure consists in two phases. In the first phase the agent navigates the entire map (without treasure and guards) at constant velocity with a wall-following predefined routine (encoded by hand) and a coupled perceptual-motor schema for obstacle avoidance, and places landmarks in the environment. For each 10 cycles of the wall-following routine a landmark is randomly placed in the vicinity of the agent. The landmarks are shown in fig. 7, left (note that in proximity of obstacles or turns the agent is slower and then it places more landmarks).

In the second phase the agent performs motor babbling for learning to navigate among landmarks, and its movements are selected or discarded on the basis of the simulated annealing (SA) algorithm Kirkpatrick et al. (1983). The agent is placed in a random landmark location (\( L_{start} \)). Another landmark (\( L_{goal} \)) is selected randomly with a predefined distance (\( 300 \pm v \) units, where \( v \) is a learned parameter). The task is now learning a schema for navigating from \( L_{start} \) to \( L_{goal} \) which allows accurate control and sensory prediction. The agent then begins its motor babbling. The next movement is selected by using the method described in Marsaglia (1972), which conducts the agent along one of the adjacent locations by performing a unitary movement in one of the two axes \( x \) or \( y \). The simulated annealing algorithm is used for evaluating if that movement gets the agent closer to \( L_{goal} \). If this is the case, the agent actually moves there. Otherwise, it executes

![Figure 7](image-url)
that movement with a probability of $e^{\delta E/T}$ where $\delta E/T$ is calculated as current goal location minus expected goal location, and $T$ is a function of the temperature parameter of simulated annealing (which decreases for each attempted movement).

If no movement is executed, the next execution cycle of simulated annealing starts. When the agent reaches $L_{\text{goal}}$, the whole ‘history’ of the movements is stored (a quadruple: position of the agent, desired position, movement executed, position reached). Fig. 7 (right) shows the movements executed while learning to navigate a portion of the environment.

The history is then used as the training set for the navigation schemas (RNNs) using the BPTT algorithm Rumelhart et al. (1986) (with the same method used for the other schemas). If a schema does not reach the desired level of accuracy in prediction it is discarded, the $\nu$ parameter is lowered (of 20 units) and the second phase restarts from the same $L_{\text{start}}$, but a new $L_{\text{goal}}$ is selected randomly. If learning succeeds, the schema is stored (note that stored schemas generalize the data set and permit navigation in a space that is wider than ‘from $L_{\text{start}}$ to $L_{\text{goal}}’’). Now a new random landmark in a fixed range (10 units) from $L_{\text{goal}}$ is selected as the new $L_{\text{start}}$, a new $L_{\text{goal}}$ is choose with the same criterion as before, and the process restarts. Learning ends when no new $L_{\text{start}}$ can be generated that lies outside any of the already learned schemas. In our simulations 16 schemas were learned.

4.1.2.2. Integration of schemas. The agent architecture now includes four sets of schemas learned independently: the challenge is integrating them. For this purpose the agent (which now includes all schemas) navigates again in the environment in which two guards dwell in fixed locations. All schemas are active but their motor commands are inhibited. The agent is instead controlled by the wall following routine used in the first learning phase. Schemas do not learn their internal models any more. Instead, energetic links among all the schemas and drives are learned with the following Hebb rule Hebb (1949): $\Delta W_{ji} = \eta a_i a_j$, where $\eta$ is the learning rate (set to 0.2) and $a_i$, $a_j$ represent the activations of two schemas, or a schema and a drive (normalization is applied at each cycle). This means that schemas and drives which are active in the same span of time develop associative links (e.g., fear with schemas for avoiding guards). The associative topology of the network implicitly encode possible trajectories or location/drive associations.

4.1.3. Description of the thief $TS$

The thief ($TS$) has the same sensorimotor layer as $TI$. Its intentional layer includes only the two aforementioned goals ($\text{collect } V$ and $\text{escape } G$), which are directly linked to schemas (more precisely, to the same schemas indicated in the corresponding plans of $TI$) and fuel them. Overall, $TS$ has a simplified intentional layer that cannot support practical reasoning.

4.2. Experiment 1

In Experiment 1 the performance of $TI$ and $TS$ are compared. The experiment serves to assess the effectiveness of the complete (layered) DiPRA architecture in real-word, dynamic scenarios (a hard challenge for most knowledge-based AI systems) as compared to a simpler, one-layer architecture. To make the comparison meaningful, we provided $TS$ and $TI$ with the same amount of computational resources in the energy pool. Since $TI$’s resources are split among the intentional and sensorimotor layers, while $TS$ has only the sensorimotor layer, the extra layer comes at the cost of less reactivity. In particular, in order to be advantageous the
intentional layer has to interact efficaciously with the sensorimotor one. Therefore, an advantage of the layered architecture TI will indirectly show that the method-ologies we adopted for obtaining knowledge and for activating schemas based on adopted plans (the two ways which the layers have to interact) are efficacious, too.

Table 1. Comparison between TI and TS (see text for explanation)

<table>
<thead>
<tr>
<th>Thief</th>
<th>V collected (%)</th>
<th>time (secs.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS (without intentions)</td>
<td>61%</td>
<td>154.3</td>
</tr>
<tr>
<td>TI (with intentions)</td>
<td>76%</td>
<td>78.1</td>
</tr>
</tbody>
</table>

We have compared the performance of two thieves TI and TS, operating individually in 100 simulations. The task consists in collecting V without being captured by G. The thief, the guard and the valuable appeared randomly in the house, and the simulation ended with either the valuable collected by the thief, or the thief captured by the guard. As illustrated in Tab 1, TI performs significantly better than TS both with respect to (average) percentage of collected valuables and (average) time for successfully achieving the task ($p < 0.001$ in both cases with analysis of variance, ANOVA). All this happens despite the fact that reasoning in the intentional level wastes computational resources (the exact amount varies during the experiments, depending on the number of nodes in the FCM). This indicates that TI’s knowledge is sufficiently good to provide it better adaptivity with respect to its opponent TS.

4.3. Experiment 2

In order to provide a better measure of DiPRA’s effectiveness, we have also compared it with other popular architectures. Four implementations of the Thief were tested: (1) TI (described previously); (2) the $A^*$ algorithm (Hart et al. 1968) (which has full knowledge of the environment, including the location of V and Gs, plans the shortest path to it but and replans when something changes in the environment, e.g. a door closes); (3) a classic BDI (based on Rao and Georgeff 1995), with the same goals, plans and beliefs of DiPRA. Finally, a baseline (random system) is compared to the other architectures.

Percentage of success (having V without being captured by the Guard) was measured in 100 runs: analysis of variance (ANOVA) shows that TI (81%) performs significantly better than the other strategies ($p < 0.0001$ in all cases): Baseline (12%), $A^*$ (56%) and BDI (43%). This experiment further illustrates that DiPRA is able to exploit effectively and flexibly its intentional layer, and to select appropriate goals and plans based on grounded knowledge that is produced on-line by situated action and its prediction.

5. Discussion of the main features of DiPRA

After the DiPRA specification and the above discussed experimental results, in this section we explain in more detail its main features and design principles sketched in sec. 1: the division of labour between off-line planning and on-line specification of action, the grounding of beliefs in sensorimotor interaction and anticipation, the use of bounded resources and knowledge, and the realization of deliberation and means-ends reasoning as intertwined processes.
5.1. Integration of levels of control

In DiPRA there is a division of labour between off-line planning and on-line specification of action. This means that intentional action is first planned and then realized by the same mechanisms subserving automatic situated action: that is, by a prediction-based competition between sensorimotor schemas. Moreover, when environmental conditions change, belief revision processes determine a variety of behavioral responses, which range from revising the priority of schemas, to replanning and intention reconsideration. The choice to change action, plan or goal determines completely different responses to the same situation, enhancing DiPRA’s behavioral flexibility. In addition, taking intentional action makes novel beliefs available in the intentional layer. Overall, the loop between DiPRA’s actions and beliefs, which are mutually constraining, provides a seamless integration among its two layers of control.

![Diagram of the intentional layer after replanning.](image)

Figure 8. The intentional layer after replanning. Like in fig. 2, the goal is to reach room 11, but a novel plan is selected which permits traversing (in sequence) rooms 19, 18, 17, 16, 15, 13 (see fig. 6 for the house map).

The description of some episodes which occurred in Experiment 1 (TI vs. TS) can help clarify this point. Suppose that the thief TI is in room 1 and, due to contingent factors such as the absence of the guard, intends to have V, as in Fig. 4, case1. Shortly after, a plan is generated for the intention: the reasoner collects the relevant plans and beliefs, as shown in the FCM above the dotted line in Fig. 2. The resulting intentional structure is tree-like, involving the plan to find V, the (sub)goal to go to room 11, the plan to pass through rooms 7-...-11. In order to support this means-ends structure two beliefs are added on as plan preconditions: the expectation that V is in room 11 and the expectation that the path to room 7 is free. In turn, this activates corresponding processes of monitoring of the activity level of sensorimotor schemas with the aim to set and continuously update the values of such beliefs. Suppose now that, while going to room 7, the detect guard schema succeeds in its predictions and, as an effect, the thief discovers that the latter expectation was false (the corresponding node in the FCM is removed), and a new belief guard in sight is ‘popped in’. As illustrated in Fig 8, a replanning occurs: a new plan, pass rooms 19-...-11 is generated for the same intention, go to room 11, since invalidation of a plan’s precondition still lets the thief committed to the same intention. Once in room 11, the thief may, however, discover that even the former expectation (V is in room 11) is false, but see (from a balcony permitting a glimpse of room 4) that V is in room 4. In that case, a failed expectation and new evidence lead to drastic belief revision in the FCM. As a consequence, there is intention reconsideration, too: as illustrated in Fig. 9 a new goal, go to room 4, is selected. It is worth noting that intention reconsideration can occur at different levels: it can modify high-level
goals, as shown in Fig 4 (case 1 vs. case 2), or sub-goals, as in this case. These sample situations illustrate that relevant and situated knowledge, when acquired at the right time by the agent (as an effect of its reasoning and action), lead to flexible goal-directed behavior controlled by the intentional layer. While drives are related to immediate needs and dictate here-and-now actions (related to immediate stimuli and necessities), goals are based not only on evidences, but on beliefs and expectations and are thus much more versatile. In particular, the possibility to predict permits the building of complex plans, which take into consideration future conditions (such as doors expected to be open or closed). Another very important advantage is acting (and regulating action) in view of future, and not only present, opportunities and affordances, and to negotiate present and future opportunities for actions beyond the possibilities of pure sensorimotor systems. One example is the possibility to put attention to events which are relative to future parts of the plan. For example, when following a planned path the agent can control if any of the doors is open (if they are in sight), not only the next door to enter, since any closed door invalidates the whole plan.

Although quicker in its sensorimotor interaction, TS is less efficacious in solving this task. It is unable to plan and it can only exploit the more immediate affordances in its environment, such as the presence of a guard or the treasure. Moreover, it is memoryless and thus it risks repeating the same behavior several times even if it is maladaptive (e.g., attempting the same path 7-[…]-11 even if there is a guard patrolling it, since once the guard exists the visual field it is no longer considered. Finally, TS lacks commitment to its goals, so when its environment changes quickly, it shows oscillating behavior between them.

5.1.1. Epistemic processes

As remarked in sec. 1, one complementary aspect of layer integration is the management of epistemic dynamics, which include perceptual and attentional processes, as well as the ‘extraction’ of knowledge and beliefs from sensory states. In DiPRA perceptual and attentional processes are managed by perceptual schemas, which, like motor schemas, can be activated by current stimuli or plans. Since perceptual schemas are coupled with motor schemas, the functional integration of action planning and execution automatically translates to an integrated attentional strategy. As a consequence, perceptual schemas relative to ongoing and planned actions are able to direct attention to information relative to the current action or the planned one, respectively, and their selection depends on the same mechanisms of action selection. For this reason, no extra mechanism for governing shifts in attention is needed.
5.1.2. Grounding beliefs in sensorimotor interaction and anticipation

Another important epistemic process is the retrieval from the environment of information that is useful for practical reasoning, and in particular for deciding future actions, replanning, and intention reconsideration. To be used in deliberation and means-ends reasoning, such information has to be encoded in declarative format: as explicit beliefs. In DiPRA beliefs are produced by monitoring sensorimotor schemas either involved in situated action, or re-enacted in mental simulations. This is an abductive process (Peirce 1897 / 1940) in that beliefs, like hypotheses, are tested when relative to the current (or to-be-planned) course of action. Such verification is based on sensory prediction. For example, the belief the treasure is at reaching distance can be grounded in a high activity level of the grab treasure schema, which in turn is determined by its success in (sensory) prediction (e.g., the ability to predict specific patterns of sensory inputs which are associated to guards). The rationale is that an high activity level of a schema is a sign that it is adequate to the current context, or that its preconditions and expectations are true; for example, that a treasure is expected to be reachable. Such information can thus be used for adding or revising beliefs in the FCM. Another example: the expectation the room in front of me is free can be produced by running ‘in simulation’ the schema go forward (i.e., by chaining several short term predictions to produce long-term ones, and then ‘virtually exploring’ the room) and monitoring its expected success or failure.

One important aspect of our methodology is that explicit beliefs have double nature. First, unlike sensory information, they are persistent epistemic structures decoupled from any actual situated action, and can trigger different goals in different contextual situations. Second, although they are manipulated independently of situated action, they remain constrained by physical interaction between the agent and the environment, since they depend on the actual or expected success of schemas; this guarantees their grounding and intentionality. Another important aspect is that objects and events are conceptualized in terms of possibilities of action, that is different from a mere perceptual understanding or categorization (Pezzulo 2008a). Schemas, indeed, generate predictions about action outcomes, such as ‘if I go left I will reach the treasure’. This happens both in their normal functioning and when they are re-enacted ‘in simulation’ to predict future courses of action and to generate beliefs. In turn, the action-oriented nature of schemas makes them suitable for taking goal-directed action, and the action-oriented nature of beliefs makes them suitable for supporting practical reasoning.

Recently the prediction-based approach to representation grounding has begun to be used for designing robotic systems that aim to bridge the gap between sensorimotor interaction and higher order cognitive skills such as decision making and language use. A particularly interesting design methodology is offered by Roy (2005). He describes a schema-based robotic system in which perceptual features are grounded into sensory information: for example, ‘red’ is grounded in some (expected) values of the robot’s sensors. More complex attributes are grounded thanks to (actual and potential) actions: for example, the meaning of ‘sponge’ is grounded in terms of what actions can be used to operate on it. More precisely, it is the set of expected consequences of one’s own actions with a sponge (e.g. the anticipated softness) which constitutes the grounding of the word. Finally, concepts for objects which are for example reachable or graspable are grounded by schemas that regulate actual behavior and, at the same time, encode predictions of the consequences of (actual or possible) interaction. Based on such grounded knowledge, robots exhibit sophisticated communication and planning capabilities (Mavridis 2006, Roy 2005).
Other research groups have been exploring the grounding of concepts in sensorimotor interaction. For example, Hoffmann (2007) uses internal simulation of possible trajectories for grounding concepts related to navigation. In this model, distance from obstacles is grounded and estimated by running simulations until they encounter the obstacle. Dead-ends are recognized through simulated obstacle avoidance, while passages are understood in terms of successfully terminated simulations of navigation. Other experiments have been conducted that show how to exploit re-enactment of motor programs and mental simulations for mental state inference (Oztop et al. 2005), imitation (Demiris and Khadhouri 2005), and planning goal-directed action (Pezzulo 2008b, Tani 1996, Ziemke et al. 2005). The DiPRA system is conceptually similar to all the models mentioned above. However, in these systems mental simulations produce implicit knowledge, while in DiPRA schemas’ re-enactment gives rise to (grounded) beliefs that are explicitly represented in a declarative format in the intentional layer. Based on such explicit representations, flexible forms of internal manipulation can be realized, such as deliberation and means-ends analysis, without violating situatedness and thus ensuring layer integration.

5.1.3. Relevance of information

We have discussed how beliefs are grounded in sensorimotor action and anticipation. It is equally important to ensure that such beliefs are relevant to the agent’s current and to-be-planned actions. In DiPRA this happens thanks to a combination of bottom-up and top-down constraints on knowledge extraction.

Knowledge about the current situation is added incrementally as long as the agent continues to reason and act, which guarantees its relevance to the agent’s current action. In addition to this bottom-up constraint on knowledge relevance, there is a top-down, goal directed constraint. Indeed, goals with more resources can re-enact more related schemas in simulation for the sake of checking their conditions; this means that the system implicitly explores first the most promising directions suggested by means-ends analysis. The rationale is that knowledge related to the goals and plans (e.g. about conditions and actions) becomes more and more relevant (and is added to the FCM), as long as they continue to have a high activity level and are thus considered promising. Plans with stronger ‘branches’ in the FCM (i.e., those whose subplans and actions have true conditions) are privileged, because top-level plans gain more activation from them. This is a self-reinforcing process: as long as the agent continues acting and reasoning, new knowledge about the plan is added that can further support it. On the contrary, if conditions begin to fail, the whole ‘branch’ loses activation. It is worth noting that the two constraints, bottom-up and top-down, interact. Reasoning in the FCM leads agents to select some courses of action and not others (and then to activate some schemas and not others), so there is actually a loop among new knowledge added, new action selected, etc. This loop produces a bias toward adding knowledge that is both relevant for the agent’s current goals (and in fact some branches of the FCM are longer than others), and situated. Note that this kind of loop is typical of active perception systems (Ballard 1991), too, the difference being that here internal modeling and explicit representations are used.

5.1.4. Production vs. retrieval of information

It is worth noting that our methodology for belief production and revision, which consists of monitoring and re-enacting sensorimotor schemas, is an alternative to the classic view of the ‘storage and retrieval’ of information, popular in cognitive science and basic theories of memory, and is instead compatible with embodied cognitive theories (Barsalou 1999, Glenberg 1997). In DiPRA part of the knowledge
is stored in the FCM structure and can be retrieved. Another part, however, is produced during sensorimotor interaction or with mental simulations. The rationale is that anticipation and mental simulations provide access to tacit knowledge inside internal forward models (see Pezzulo submitted). Such knowledge is typically used for the sake of action control; it has a procedural and not a declarative format, and thus cannot be simply ‘retrieved’ from memory. For this reason, the monitoring of schemas or their re-enactment in simulation is necessary in making it available outside the original sensorimotor processes and in transforming it from a procedural to a declarative format. This view is in accordance with recent simulative views of cognition, which are investigating to what extent tacit knowledge about one’s own, or others, is accessible (Knoblich and Flach 2003), and how content is retrieved from memory on-the-fly by means of such embodied simulations (Barsalou 1999). Although our experiments can be considered preliminary, if this methodology proves to be effective it will become of extreme interest for computer scientists designing knowledge-based systems, since it is highly context-dependent and thus potentially less prone to the frame problem than the classical ‘storage and retrieval’ approach.

5.2. Use of bounded (computational and knowledge) resources

DiPRA’s intentional layer has two forms of boundednesses. Since few schemas can run at once, the number of ‘beliefs’ that can be produced is limited, too—this is the first kind of boundedness in our system. As a side-effect, the system’s ‘epistemic horizon’ is limited to the current sensorimotor state and (a few) ‘simulations’ of future states. The second form of boundedness depends on the fact that the Reasoner has limited resources, too (whose upper bound can be set by the designer). These resources put a limit on how much ‘imagination’ can be used during reasoning, and how many schemas can be re-enacted in simulation to produce expectations—all of which provides full support for flexible reasoning about the agent’s goals (proximal or distal).

It is worth noting that, due to these two forms of boundedness, the amount of computational resources distributed among the layers varies depending on the current context—this is another fundamental aspect of how DiPRA tackles the issue of balancing levels of control. For example, in situations that require urgent behavioral responses (e.g. the presence of a guard), schemas in the sensorimotor layer tend to tap most resources due to the high activity level of the drives that fuel them. Many more resources are instead used by the intentional layer when there is nothing urgent to deal with, and planning is needed. Overall, not only are DiPRA’s actions determined in a context-sensitive way, but so are its level of control (intentional or automatic) and the amount of resources it uses in deliberating or acting.

In order to be effective, the bounded resources allocation mechanism should provide more resources to those modules which are more important, based on the agent’s internal context (goals and drives) and external context and stimuli. How are resources allocated? The mechanism of bounded resource allocation is based on two factors: desirability and achievability of outcomes. The first aspect, desirability, is managed in a simple way: goals and drives, which are motivational components, are able to transfer resources to plans or actions that can satisfy them (goal-plan links are hand coded in the current implementations, while drive-schema links are learned). The second aspect, achievability, is based on a prediction-based mechanism in both layers. The better a schema’s predictions, the more resources it gains. The same happens to means-ends chains based on reliable beliefs, which
in turn are based on schemas sensory predictions. This mechanism is based on the idea of prediction-based selection of context currently popular in motor control studies (Wolpert et al. 1998, Wolpert and Ghahramani 2004), here extended to layered architectures.

5.3. Deliberation and means-ends analysis as intertwined processes

In most practical reasoning systems such as BDI (Rao and Georgeff 1995) deliberation and goal selection precede means-ends reasoning and the adoption of plans to achieve intended goals. However, this mechanism has two limitations. First, knowing that some goals are more achievable than others—something discovered through means-ends analysis—certainly influences goal selection. Second, during means-ends analysis, novel outcomes and side-effects of goal-directed actions are ‘discovered’ and evaluated that can bias deliberation. For these reasons it is better to consider deliberation and means-ends analysis as intertwined processes that potentially influence each other. This happens naturally in DiPRA’s reasoner. As discussed above, for the sake of means-ends analysis, beliefs are added to the FCM that are relevant for the agent’s intended goal(s). This happens since the intentional layer monitors the sensorimotor layer and can also trigger information-seeking strategies. For example, a plan needing to check its preconditions can trigger attention shifts and mental simulations, which in turn cause beliefs to be added to, or removed from, the FCM. However, it is worth noting that not only are plans more or less supported by novel beliefs, but this process can provide feedback on deliberation, too, as long as novel outcomes that support or unsupport goals are considered in the means-ends chain; see, for example, the above mentioned process of intention reconsideration occurring as a consequence of belief revision. This mechanism is inspired by recent evidence of co-specification of means and ends in the brain, which produces a competitive race between multiple goal-plan ensembles (Cisek 2007, Cisek and Kalaska 2005). It is important to remark that the usual boundedness principles are in play, so not all possible goals/plans are considered—only the more promising.

6. Conclusions and Future Work

We have presented a layered agent architecture in which intentional and sensorimotor layers are seamlessly integrated and interact in two ways:

- Courses of action planned in the intentional layer are realized by schemas at the sensorimotor layer. This is obtained by activating appropriate sensorimotor schemas, whose expected results satisfy plan conditions and ultimately the agent’s intention. At the same time, current plans trigger epistemic actions and channel the gathering of relevant information, which can then be used to refine practical reasoning.

- Schemas’ activity produces grounded beliefs used for practical reasoning. The value of beliefs depends on activity level of corresponding schemas, which in turn is determined by the accuracy of their predictions and simulations.

This loop is regulated by a mechanism of bounded resource allocation, which operates at both layers based on the desirability of outcomes and prediction accuracy. Our experiments illustrate that DiPRA’s strategy for integrating practical reasoning and situated action makes it able to deal with a multiplicity of conflicting goals and the demands of an open-ended environment. Consistent with the ideas
of Norman and Shallice (1986), DiPRA can steer and control intentional action, although of course its capabilities are quite far from humans’ higher-level executive functions. In particular, DiPRA has a remarkable behavioral flexibility in the face of changing environmental conditions: it is able to replan when novel opportunities arise for achieving intended goals, and to revise its intentions when these are no longer achievable. These two capabilities are rarely implemented in robot or agent architectures operating in complex environments.

As a designer’s note, it is worth mentioning that due to its modular structure and its mechanisms for allocating resources based on contextual relevance, DiPRA makes it easy to integrate novel schemas, plans, and goals without hindering existing ones. At the same time, the current DiPRA implementation has several limitations that we intend to tackle in our future work. First, in the experiments presented here, agent embodiment is simplified, and the sensorimotor layer operates on preprocessed information. However, conceptually similar schema architectures have been successfully applied to more complex and realistic robotic scenarios, but at the price of simplifying learning (Roy 2005) or reducing the number of schemas (Tani 1996). We still lack a sound learning methodology that permits the agent to autonomously acquire sets of predictors and controllers, in order to operate in open-ended environments, especially when the agent has multiple drives and goals. Second, goals and plans are designed by hand, which hinders the system’s scalability. Unfortunately, although in some cases automatic solutions can replace hand-coding (e.g., see Pezzulo 2008b on plan generation based on mental simulation), we still lack a sound methodology for autonomously generating hierarchies in interoperating schemas, plans and goals. Recent theoretical and computational advances in hierarchical learning architectures (Botvinick et al. 2008, Haruno et al. 2003, Yamashita and Tani 2008) are promising attempts to tackle these problems. The extent to which these systems can be integrated with practical reasoning capabilities, such as those implemented in DiPRA, is an important direction for further investigation.

Another limitation of DiPRA is that it is only able to represent knowledge of ‘tangible’ objects and situations. Although we have emphasized here a sensorimotor and anticipatory origin and grounding of beliefs, this does not mean that there are not other possible ways of acquiring knowledge for natural and artificial systems. Not all our knowledge derives from—or is about—our sensorimotor interaction, real or simulated (see Pezzulo and Castelfranchi 2007, Sloman and Chappell 2005). For example, our knowledge can depend on memory of past events (the door is believed to be close because before it was found closed), on other beliefs (e.g., via inference), or can be formed thanks to (linguistic or not linguistic) communication with other agents (the door is believed to be close because I was told it is close). Moreover, there are some forms of knowledge, including abstract concepts and non perceivable entities, that have a significant degree of detachment from sensorimotor activity. Although some conceptual solutions to this problem are being developed in the context of simulative theories of cognition (Barsalou 1999, Grush 2004, Pezzulo and Castelfranchi 2007), how to ground all these forms of knowledge remains an open challenge. However, it is worth noting that one example of alternative forms of grounding, which that can be defined ‘structural’, ‘tethered’, or ‘internal’ (Minsky 2006, Sloman and Chappell 2005), is already present in DiPRA: inside the reasoner, some beliefs are grounded on other beliefs instead of on schemas’ activity. Modules can send activation to each other, as in the case of goals fueling plans that can satisfy them, or beliefs that sustain a goal or a plan, or that imply another belief. Beliefs that receive activation by other modules are actually grounded in internal, coherence-based dynamics and not on sensorimotor interaction. Although
this is not described in our experiments, it is easy to extend this approach by using deduction or other forms of inference to produce new beliefs inside the intentional level. How to systematically extend an agent’s knowledge by means of such ‘structural’ or ‘internal’ grounding is an interesting direction for future work.

6.0.0.1. Acknowledgments. This work was supported by the European Community, project HUMANOBS: Humanoids That Learn Socio-Communicative Skills Through Observation (FP7-231453).

Appendix A. Formal description of the intentional layer

Let \( S \) be a set of worldstates, \( P^+ \) a set of atoms, and \( \pi : P^+ \times S \to [0..1] \) a function assigning a truth value to each atom in each worldstate. \( P \) is a set of atoms and negated atoms where \( \pi(p, s) = 1 - \pi(\neg p, s) \). \( L \) is a propositional language over \( P \) and the logical connectives \( \land \) and \( \lor \), where: \( \pi(p \land q, s) := \pi(p, s) \otimes \pi(q, s) \) and \( \otimes \) is any continuous triangular norm (e.g. \( \min(p, q), pq \)); \( \pi(p \lor q, s) := \pi(p, s) \oplus \pi(q, s) \) and \( \oplus \) is any continuous triangular conorm (e.g. \( \max(p, q), x + y - xy \)) (see Saffiotti et al. 1995). The intentional layer of DiPRA is a tuple \((\Psi, \Gamma, \Pi, \Phi, \text{Bel}, \Omega)\), where:

- \( \Psi \) is the reasoner, a tuple \((\text{FCM}, \text{Body})\). \( \text{FCM} \) is a Fuzzy Cognitive Map (Kosko 1986), a representation of the state and the relations between all the modules; \( \text{Body} \) is the procedural body, whose main task is to assign the status intended to a goal and adopted to a plan.
- \( \Gamma \) is the set of goals, tuples \((\text{Type}, \text{Status}, G\text{Cond}, \text{AbsRel}, \text{ConRel})\). \text{Type} is the type of goal: Achieve or Maintain; \text{Status} is the current status of the goal: Intended, Instrumental, Waiting or Not Intended; \text{GCond} \in L \) is the (graded) satisfaction condition of the goal; \text{AbsRel} is the absolute relevance; \text{ConRel} is the contextual relevance.
- \( \Pi \) is the set of plans, tuples \((\text{Status}, S\text{Cond}, E\text{Cond}, P\text{Cond}, \text{ActionSet}, \text{Body}, \text{Goals}, \text{Results}, \text{AbsRel}, P\text{CondRel}, \text{ConRel})\). \text{Status} is the current status of the plan: Adopted or Not Adopted; \text{SCond} is the set of start conditions \( sc \in L \) which are checked at the beginning of plan execution and must be true to start it; \text{ECond} is the set of enduring conditions \( ec \in L \) which are checked continuously during plan execution; if an enduring condition becomes false, the plan is stopped; \text{PCond} is the set of beliefs \( \beta \in L \) which are expected to be true after the plan (but not all of them have to be intended); \text{ActionSet} is the set of actions \( \phi \) or (sub)goal \( \gamma \) activated by the plan. Actions and goals are chained inside \text{ActionSet} by logical connectives \( \lambda \); \text{Body} is the behavior which is executed; it normally consists of activating actions and (sub)goals in the \text{ActionSet}; \text{Goals} is the set of goals \( \gamma \) that make the plan satisfied; they are the subset of \text{PCond} which are intended (typically when a plan is selected one of its \text{PCond} becomes the Goal, i.e., the main reason for activating the plan); \text{Results} is the set of the final plan results \( pr \in L \), corresponding to the \text{GCond} of the Goals at the end of the plan; \text{AbsRel} is the absolute reliability value of the Plan, i.e. how reliably it succeeds; \text{PCondRel} is the set of the reliability values \( ar \in L \) of the plan with respect to its \text{PCond}s, i.e. how reliably it produces its \text{PCond}s; \text{ConRel} is the contextual relevance.
- \( \Phi \) is the set of actions, tuples \((S\text{Cond}, P\text{Cond}, \text{Body}, \text{Goals}, \text{Results}, \text{AbsRel}, P\text{CondRel}, \text{ConRel})\). \text{SCond} is the set of start conditions \( sc \in L \) which are checked at the beginning of action execution and must be true to start it; \text{PCond} is the set of beliefs \( \beta \in L \) which are expected to be true after the action (but
not all of them have to be intended); **Body** is the behavior which is executed once the action is executed; **Goals** is the set of goals $\gamma$ that make the action satisfied; they are the subset of PCond which are intended (actually the reasons for activating the action); **Results** is the set of the final action results $ar \in L$, corresponding to the GCond of the Goals at the end of the action; **AbsRel** is the absolute reliability value of the Action, i.e. how reliably it succeeds; **PCondRel** is the set of the reliability values $ar \in L$ of the action with respect to its PConds, i.e. how reliably it produces its PConds; **ConRel** is the contextual relevance.

- **Bel** is the set of epistemic states, i.e. beliefs $\beta \in L$. All the conditions (GCond, SCond, PCond, ECond) are kinds of beliefs. Bel are tuples ($\beta, AbsRel, ConRel$). $\beta \in L$ is the value of the belief or condition; **AbsRel** is the absolute relevance; **ConRel** is the contextual relevance.

- $\Omega$ is a set of **parameters** used to control the energetic dynamics of the modules: $\sigma$ is the activation of the reasoner; $\zeta$ is the threshold for Intending a goal; $\eta$ is the threshold for Adopting a plan; $\theta$ is the commitment level of intended goals; $\kappa$ is the commitment level of adopted plans; $\vartheta$ is the total amount of resources available to the whole system.

### A.1. Choice of parameters

In the experiments detailed above, parameters were set as follows: $\sigma = 3; \zeta = 0.8; \eta = 0.8; \theta = 0.3; \kappa = 0.3; \vartheta = 7$. The choice of parameters is the result of empirical estimation; however, similar results were obtained by systematically varying parameters up to 20% of their values, and this indicates a certain robustness1.

### A.2. Link types

In the FCM there are six kinds of (weighted) links: (1) **Satisfaction** A goal links plans and actions whose PCond satisfy its GCond; in this way, activation is spread from intended goals to plans which realize them. (2) **Predecessor** A plan links (sub)goals whose GCond realize its SCond or ECond; in this way, if a plan has a missing PCond or ECond it can “subgoal”. (3) **Support** A belief links goals, plans or actions which correspond to their GCond, SCond or ECond; this is a way to represent contextual conditions: goals, plans and actions which are “well attuned” with the context are preferred. (4) **Feedback** a plan or action feedbacks on goals; this special case of support permits to select goals having good paths to action (5) **Inhibition** Goals and plans with conflicting GConds or PConds, and plans realizing the same GCond have inhibitory links. In this way it is possible for a goal or plan (especially if intended or adopted) to inhibit competitors. (6) **Contrast** Beliefs have inhibitory links with plans and goals having conflicting GConds, PConds, SConds and EConds: this is a kind of “reality check”.

### References


---

1See (Pezzulo and Calvi 2007a) for a more general discussion on how local excitation (based on prediction accuracy and Hebbian links) and global inhibition (based on bounded resources) of modules guarantees what is called ‘cooperative computation’ and an overall system robustness.

REFERENCES


URL citeeseer.ist.psu.edu/brooks91intelligence.html


URL citeeseer.ist.psu.edu/kirkpatrick83optimization.html


REFERENCES


