Biologically Inspired Cognitive Architectures (2013) **5**, 1–9

Available at www.sciencedirect.com

SciVerse ScienceDirect

journal homepage: www.elsevier.com/locate/bica

RESEARCH ARTICLE

A meta-cognitive architecture for planning in uncertain environments

Vincenzo Cannella, Antonio Chella, Roberto Pirrone *

Dept. of Chemical, Management, Computer, and Mechanical Engineering (DICGIM), University of Palermo, Italy

Received 30 May 2013; received in revised form 1 June 2013; accepted 1 June 2013

KEYWORDS
Cognitive and meta-cognitive artificial agents;
Planning;
Uncertainty;
Markov Decision Processes

Abstract
The behavior of an artificial agent performing in a natural environment is influenced by many different pressures and needs coming from both external world and internal factors, which sometimes drive the agent to reach conflicting goals. At the same time, the interaction between an artificial agent and the environment is deeply affected by uncertainty due to the imprecision in the description of the world, and the unpredictability of the effects of the agent's actions. Such an agent needs meta-cognition in terms of both self-awareness and control. Self-awareness is related to the internal conditions that may possibly influence the completion of the task, while control is oriented to performing actions that maintain the internal model of the world and the perceptions aligned. In this work, a general meta-cognitive architecture is presented, which is aimed at overcoming these problems. The proposed architecture describes an artificial agent, which is capable to combine cognition and meta-cognition to solve problems in an uncertain world, while reconciling opposing requirements and goals. While executing a plan, such an agent reflects upon its actions and how they can be affected by its internal conditions, and starts a new goal setting process to cope with unforeseen changes. The work defines the concept of ''believability'' as a generic uncertain quantity, the operators to manage believability, and provides the reader with the u-MDP that is a novel MDP able to deal with uncertain quantities expressed as possibility, probability, and fuzziness. A couple u-MDPs are used to implement the agent's cognitive and meta-cognitive module. The last one is used to perceive both the physical resources of the agent's embodiment and the actions performed

* Corresponding author.
E-mail addresses: vincenzo.cannella26@unipa.it (V. Cannella), antonio.chella@unipa.it (A. Chella), roberto.pirrone@unipa.it (R. Pirrone).

http://dx.doi.org/10.1016/j.bica.2013.06.001

by the cognitive module in order to issue goal setting and re-planning actions.

1. Introduction

The effects of the actions performed by an artificial agent like an autonomous robot, interacting with a natural environment cannot be predicted exactly, due to both internal and external factors. Autonomous robots operating outdoor, and engaged in rescue or equipment repairing tasks are a good example of such a scenario.

Natural environments are prone to sudden changes of the operating conditions (i.e. weather conditions, earthquakes, accidents, and so on). Moreover, the robot may experience imprecise perceptions from the environment itself. Finally, the internal representation of the world may be not well defined; so the agent plans its actions erroneously.

On the other side of the agent's boundary, the internal status of the robot can influence its performance severely. The agent may experience partial failures or a decrease in power supply; so new goals arise that are aimed at repairing malfunctions, and can be in conflict with the task at hand.

A robot aimed at coping with all the issues mentioned above, has to be *self-aware* to plan its behavior properly. Moreover, the robot has *control* because it performs actions, which maintain its uncertain model of the world and its uncertain perceptions aligned; as a consequence the agent can *judge* if it is performing properly. In a few words, such a robot must exhibit meta-cognitive abilities while planning in uncertain environments.

Classical robotics deals with uncertainty in both perception and the model of the world, while neglecting the rest of the problem. A classical solution is offered by the Probability Theory. The concept of ''uncertainty'' has been investigated over the years, and new definitions have been formulated to capture other aspects of the meaning. Possibility Theory and Fuzzy Logic represent some of the result of such studies (Zadeh, 1999). On the contrary, meta-cognition in artificial agents is a crucial topic in the BICA literature (Chella, Cossentino, Gaglio, & Seidita, 2012; Samsonovich, 2012). This work presents a novel architecture that reconciles planning in uncertain conditions and meta-cognition expressed in terms of self-awareness and control on the coherence between the model of the world and the outer environment.

Some of the authors investigated the problems related to the interaction of an artificial agent with uncertain environment. In particular natural language interaction between humans and the agent was investigated. The problem arose when designing TutorJ (Pirrone, Cannella, & Russo, 2008) an architectural framework for building ITSs that are able to support a student in the learning process by supplying learning material customized to her cognitive needs, skills, and goals. The architecture is inspired to the Human Information Processor Model (HIPM) (Todorovski, Bridewell, Shiran, & Langley, 2005) where perceptual, cognitive and sensor-motor modules can be devised. Understanding natural language is an uncertain process, which makes not sure the meaning of the user's sentences. At first, the problem was faced up by adopting a planner agent that is able to manage uncertainty expressed through probability: the Partially Observable Markov Decision Process (POMDP) (Cannella & Pirrone, 2009). The agent's actions are a wide range of communication acts, aimed at coping with the learner's cognitive processes.

Next, the authors moved to modeling meta-cognition in such an agent. The artificial tutor mentioned above has to reflect on questions like ''how well I understood the user?'' or ''how well the user understood my sentence?'' in order to refine its next dialogue move. The agent has to access to its internal state, and to change it through proper actions. In a few words it needs meta-cognition. We used the classical definition of meta-cognition as *cognition about cognition* (Metcalfe, Shimamura, Metcalfe, & Shimamura, 1994) so the original POMDP is analyzed by the agent itself, whose internal reasoner was modeled by another POMDP that is a meta-cognitive module inserted in a two-level structure (Cannella, Pipitone, Russo, & Pirrone, 2010). The second POMDP has its own perceptions both from the external world (the dialogue flow) and from the actions issued by the cognitive POMDP. Cognitive actions are the actions in the previous version of the system. Meta-cognitive actions are communication acts aimed both to evaluate self regulated skills in the learner and to stimulate her to reflect on her meta-cognitive state in order to pursue a self-regulated learning.

In subsequent works (Cannella, Pirrone, & Chella, 2012) the authors investigated a unified management of uncertainty in Markov Decision Processes (MDPs), and presented a planner model able to manage different kinds of uncertainty together, expressed as probability, possibility, and fuzzy logics. We called this model ''uncertainty based MDP'' (u-MDP). The present work is a synthesis of the research activity described above. We present a meta-cognitive architecture for mobile robots engaged in tasks in natural environments. Such an architecture is based on a couple of u-MDPs where the cognitive MDP is devoted to deal explicitly with the external environment, while the meta-cognitive one governs the meta-cognitive abilities of the robots in terms of self-awareness and control. Choosing an implementation based on MDP instead of POMDP is not a limitation. In the outlined scenario, the autonomous robot can be supposed to work in a ''observable'' even if uncertain world. We're not dealing here with issues related to understandability of percepts. Moreover, we're currently investigating the extension of the presented architecture to the use of POMDP.

The rest of the paper is arranged as follows. Section 2 provides the reader with some theoretical background about the research on unified models of uncertainty. Section 3 presents our model for dealing with uncertainty and details the u-MDP. Section 4 presents the meta-cognitive architecture. Finally, in Section 5 conclusions are reported and future work is outlined.

2. Theoretical background

This section reports a discussion about modeling uncertainty in the literature. Uncertainty was first defined in terms of the Probability Theory, but such a concept has been widened over the years to include also possibility. In what follows, the relation between probability and possibility will be deepened first, and then the notion of epistemic belief will be faced.

2.1. Relation between possibility and probability

Many researchers have investigated relationships between possibility and probability, formulating different solutions. In the section, a brief introduction of these formulations will be supplied. The original theoretical background used to define possibility was fuzzy logic, which originated the Possibility Theory. This term (Zadeh, 1999) was used for the first time by Zadeh to express the intrinsic fuzziness of natural languages as well as uncertainty information. In this way the *possibility* was related to fuzzy sets. In his seminal paper, Zadeh stated the *possibility—probability consistency principle*, according to which a high (low) probability implies a high (low) possibility. The Zadeh's possibility—probability consistency principle affirms that if possibilities $\Pi = (\pi_1, \ldots, \pi_n)$ and probabilities $P = (p_1, \ldots, p_n)$ are assigned to a same event expressed by a variable X, the degree of consistency γ of these values is given by:

$$\gamma = \sum_{i=1}^{n} \pi_i * p_i \qquad (1)$$

The following theorem holds:

Theorem 1. *maximizing the degree of consistency implies the following conditions*

$$Pos(A) < 1 \Rightarrow Nec(A) = 0 \qquad (2)$$

$$Nec(A) > 0 \Rightarrow Pos(A) = 1 \qquad (3)$$

Dubois and Prade (2003), Dubois and Prade (1983), Dubois, Prade, and Sandri (1993), Dubois, Prade, and Smets (2001), Dubois, Foulloy, Mauris, and Prade (2004) defined a transformation from probabilities to possibilities and vice versa, which is based on the so-called *probability possibility consistency* and *preference preservation* principles. These principles can be synthetically expressed by

$$p(u_i) > p(u_j) \Longleftrightarrow \pi(u_i) > \pi(u_j) \qquad (4)$$

This transformation satisfies consistency and preference but it is not the only possible one, as shown in (Yamada, 2001).

Klir (1995), Klir and Folger (1988), Novák (2003) proposed a transformation approach based on the principle of uncertainty and information invariance, and also derived by Jumarie (1994) and developed by Wonneberger (1994). Yamada (2001) has proposed three different transformations inspired to the so-called Evidence Theory (or Dempster—Shafer Theory).

2.2. Epistemic belief

Zadeh stated that the association of an uncertain quantity to a fuzzy set induces a possibility distribution for this quantity. This distribution represents the information related to the values assumed by this quantity. After this first formulation, the so-called Evidence Theory or Dempster—Shafer Theory that is the theory dealing with ''belief functions'', was proposed to express uncertainty as a generalization of the Bayesian theory of subjective probability. Evidence Theory combines empirical evidences to build a coherent picture of reality (Glenn & Princeton, 1976). The Spohn's theory of epistemic beliefs (EBs) (Giang & Shenoy, 2000) is known also as *kappa calculus*, and is considered as a qualitative counterpart of Bayesian probability theory. This calculus has been designed to represent and reason with plain human beliefs, and to describe a formalism for describing both plain EBs and procedures aimed at revising beliefs when new information is obtained. An epistemic state is represented by the so-called *disbelief function*, which can be revised through ad hoc rules when new information is obtained. A disbelief function is very similar to a probability distribution function, and it is specified by the values it takes for each possible subset of the variable's space. After the introduction of possibility, agents dealing with uncertainty have been modeled accordingly. In general, the new models were possibilistic or fuzzy extensions of their probabilistic counterparts. Examples of fuzzy or possibilistic (PO)MDPs can be found in (Pardo & Fuente, 2008) and in (Sabbadin, Fargier, & Lang, 1998) where an extended MDP manages the different uncertainty models in a unified framework, and plans are computed using a reformulation of the backward induction algorithm.

3. A unified approach to manage uncertainty

In the past, many researchers remarked that all the uncertainty descriptions share many features, and several unifying approaches have been proposed. Authors proposed the *believability* (Cannella et al., 2012), which is defined as the generic uncertain quantity u. Such a value can be unified with either probability, possibility, or EB. Moreover, it can be referred to as a fuzzy value $a \in A$. Believability can be joined to a believability distribution $u(a)$, defined similarly as a probability/possibility distribution. Some definitions are reported in the following.

Definition 1. The conditional believability distribution $u(b|a)$ (Gwét, 1997) for all fuzzy values $a \in A$ and $b \in B$ is defined as the extent to which $b \in B$ appears to be certain provided that a is true.

Definition 2. a and b are independent if and only if $u(b|a) = u(b)$.

Definition 3. The joint believability distribution $u(a, b)$ for all fuzzy values $a \in A$ and $b \in B$ is the extent to which it is certain that the element $a \in A$ and $b \in B$ are both true.

These functions are manipulated through a set of *abstract* operators, which can be unified to whatever uncertainty definition. In the following, such operators are detailed, and some useful definition are given to detail the model.

Definition 4. Given a set of believability values $U = \{u\}$, the believability accumulation operator, $\oplus_b : U \times U \Rightarrow U$ sums the effects of new information to the previous knowledge.

Definition 5. Given a set of believability values $U = \{u\}$, the believability combination operator, $\odot_b : U \times U \Rightarrow U$ represents how new pieces of informations interact together.

Given a set of fuzzy values $A = \{a_1, \cdots, a_n\}$, we introduce a compact form of $\oplus_b$:

$$\underset{a \in A}{\oplus_b} u(a) = u(a_1) \oplus_b \cdots \oplus_b u(a_n) \tag{5}$$

Given the joint believability distribution $u(a,b)$, the marginal believability distribution over A is defined by

$$u_1(a) = \underset{b \in B}{\oplus_b} u(a,b) \forall a \in A \tag{6}$$

At the same time, the marginal believability distribution over B is given by

$$u_2(b) = \underset{a \in A}{\oplus_b} u(a,b) \forall b \in B \tag{7}$$

To solve a MDP based on believability, we must be able to manage the uncertainty of the expected reward from a policy. Reward can be managed in a similar way as believability.

Definition 6. Given the reward set R, the gain accumulation operator $\oplus_g : R \times R \Rightarrow R$, sums the effects of new gain to the previous one.

The expected reward over all next states $s \in S$ can be expressed as a weighted combination of their believabilities $u(s)$.

The following operator is introduced to manage reward:

Definition 7. Given the set of possible reward values $R = \{r\}$, and the believability set $U = \{u\}$, the gain combination operator $\odot_g : R \times U \Rightarrow R$ combines the reward of a state with its believability.

Finally, the expected reward ER in a state s is defined as:

$$ER = \underset{s \in S}{\oplus_g} [\odot_g (r(s,a), u(s))] \tag{8}$$

where S is the state space and $r(s,a)$ is the reward obtained if the agent performs the action a when being in the state s.

Table 1 Operators defined for different kinds of uncertainties: probability (first row), possibility (second row), and Spohn's epistemic belief (third row).

$\oplus_b$	$\odot_b$	$\oplus_g$	$\odot_g$
+	*	+	*
max	*min*	*min*	$max(1 - believability, gain)$
min	+	+	+

In (Dubois & Prade, 1995) the operators defined above have been identified for each kind of uncertainty i.e. probability, possibility, and epistemic beliefs. Table 1 shows the most common ones.

Mixed operators could allow to combine two or more kinds of uncertainty together. In particular, possibility has been deeply investigated in the past, exploiting the relation that exist between possibility, probability, and fuzzy logic. Many approaches and criteria to convert probability into possibility and vice versa have been defined (Dubois & Prade, 1983; Jumarie, 1994; Klir & Parviz, 1992; Wonneberger, 1994; Yamada, 2001; Zadeh, 1999).

3.1. The uncertainty based MDP

The definitions provided in the previous section enable the design of a general MDP that is able to deal with believabilities in a wide sense. We called this model *uncertainty based MDP* (u-MDP).

Definition 8. A u-MDP is defined as a tuple $\{S, A, B_T, r, \oplus_b, \odot_b, \oplus_g, \odot_g\}$ where:

- S represents a finite set of states
- A represents a finite set of actions
- $B_T : S \times A \times S \Rightarrow \Pi(S)$ is the state transition function where $B_T(s'|s,a)$ is the conditional believability of moving to state s' when the action a has been executed in the state s.
- $r : S \times A \Rightarrow R$ is the reward function and $r(s,a)$ is the expected reward for taking an action a when the agent is in the state s.
- $\oplus_b, \odot_b, \oplus_g, \odot_g$: the operators used to manage the uncertainty and reward.

Each state $s \in S$ is described by an array of F_s fuzzy features $A_1, A_2, \ldots, A_{F_s}$, whose membership functions are $\mu_i(A_i)$. Computing believabilities in each state relies on the following definitions.

Definition 9. The uncertain belief state $Ub = [u(s_1), \ldots, u(s_{|S|})]^T$ is the believability distribution on the state space $S = \{s_1, \ldots s_{|S|}\}$.

Definition 10. $Ub_a = [u_a(s_1), \ldots, u_a(s_{|S|})]^T$ represents the believability distribution obtained after performing the action a. In particular $u_a(s_i)$ is the believability of reaching state s_i after performing the action a.

At each step, the uncertain belief state is updated using the equation:

$$Ub_a = \underset{s \in S}{\oplus_b} [\odot_b (B_T(s'|s,a), Ub)] \quad \forall s' \in S \tag{9}$$

Planning is based on the reward function. After having performed an action, the agent updates the amount of the total gain by adding the new gain to the previous one. The backward induction algorithm is used to compute the plan. The algorithm is described in the Procedure 1.

Procedure 1. Calculate an optimal policy

```
Input: $S, B_T, Ub, \oplus_b, \odot_b, \oplus_g, \odot_g$
pol = []; {the policy is an array of actions}
for all$s \in S$ do
    $V^*_{N+1}(s) = 0$; {initialize the Bellman value function for each state}
end for
for $t = N \rightarrow 1$do
    for all $s \in S$ do
        $V^*_t(s) = \max_{a\in A} \oplus_g [r(s,a), \underset{s'\in S}{\oplus_g} [\odot_g(u_a(s), V^*_{t+1}(s'))]]$; {at each step, the partial reward is maximized through the Bellman equation computed over all the states and actions, while belief state is updated through Eq. 9}
        $a^*_t(s) = \underset{a\in A}{\arg\max} \oplus_g [r(s,a), \underset{s'\in S}{\oplus_g} [\odot_g(u_a(s), V^*_{t+1}(s'))]]$; {at each step, the best action $a^*_t(s)$ in the state $s$ maximizes the partial reward.}
    end for
    pol.append($a^*_t(s)$);
end for
Output: pol, $V^*$
```

4. The proposed architecture

In this section the detailed description of the meta-cognitive architecture based on a couple of u-MDPs is reported. We'll deep the processes related to planning and goal setting at first, putting into evidence how meta-cognition arises when the robot selects its goal. Then the design issues will be discussed.

4.1. Planning and goal setting

For the purpose of keeping the planning process as general as possible, we can assume that the whole state of the robot (i.e. both internal and external aspects) is arranged in facets. Each facet is a collection of fuzzy features, so a vectorial description of the whole state is possible for u-MDP computing purposes that is the array of all the features independently of the facet they belong to. A state as a whole is described by the values taken by all facets. The goal state is reached when some specific state's facets attain the desired value. Moreover, different facets can be related to one another.

A good model for the state is a graph. Each facet is represented by a node. Arcs reflect the constraints between the state's facets along with their degree of uncertainty. Both probabilistic and possibilistic uncertainty is allowed in the presented model. Each arc is directed according: the destination facet of the arc is precondition for the source one. Acyclic graphs are used to avoid loops. We called such a model the *knowledge network* of the agent. As mentioned above, a goal can be made by some nodes of the graph only. The agent is not always interested to all the aspects of the state. During the plan execution contrasting goals can arise, which are made by either different collections of nodes or different values in the features of the same node. In this case, arcs represent the constraints of the overall state on the goal configuration.

Each time the agent wants to reach a specific goal, it has to make true each single facet of the goal. This task can be executed step by step. After having processed a facet, the agent goes on with the subsequent connected ones. Planning in the knowledge network consists in moving from a state configuration (that is the whole arrangement of the knowledge network at a given time instant) to the next one using Procedure 1. In the following the process is detailed with a toy example.

The whole process is divided into two main parts. In the first one, the system explores the graph to assess which parts of the graph do not match with the goal state. In the second part, the system executes a series of actions aimed at reaching the goal by making the unmatched features true.

Fig. 1 shows a little fragment of the state described as a graph. The image shows six nodes. Three nodes contain only one feature, respectively $\langle A\rangle, \langle F\rangle$, and $\langle L\rangle$. We'll call them ''simple'' nodes. Three ''composite'' nodes are present, which contain respectively the $\langle B,K\rangle, \langle K,C\rangle$, and $\langle B,C,K\rangle$ groups of features. Each generic feature f can be investigated through a test T_f, and executing a test can have a cost C_f. Test and corresponding costs are depicted in the figure near the features. The total cost of a node is equal to the sum of the costs of the tests for each feature in the node. As already explained, arcs are directed in a way that a node at the beginning of an arc (start-node) depends on the node at the end of he same arc (end-node). An uncertainty degree is associated to each arc. This value represents how much the dependency is uncertain. The start-node can be reached by the agent only when the end-node has been just reached. When the end-node is a composite one, the start-node depends on all its features.

In our example, the agent must reach the goal consisting in making true the node $\langle A\rangle$ (see Fig. 2). This node depends on B, K, and C but in different ways. $\langle A\rangle$ depends at the same time on $\langle B,K\rangle, \langle K,C\rangle$, and $\langle B,C,K\rangle$. This is stated by the arcs starting from the node.

The agents chooses to try the node $\langle K,C\rangle$. The agent executes the test T_K, with cost C_K (see Fig. 3).

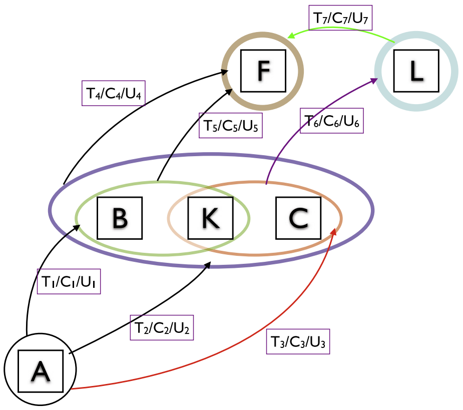

Fig. 1 A fragment of the knowledge network of the model.

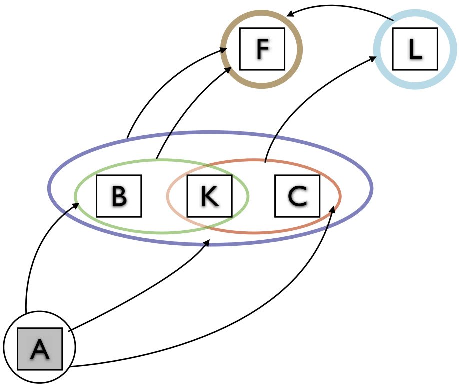

Fig. 2 The system tries to make *A* true.

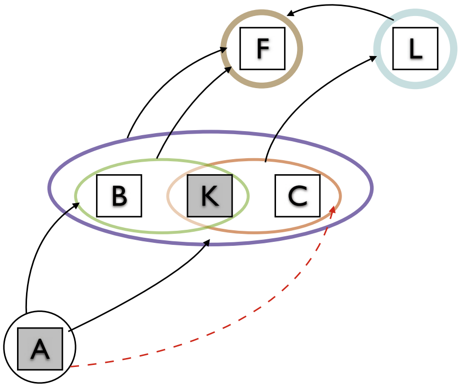

Fig. 3 *K* tested. Confirmation found.

Then the test T_C is executed. The obtained results support $\langle A \rangle$ sufficiently (see Fig. 4).

Now the agent explores another path, and tries the node $\langle B, K \rangle$. Since *K* has just been tested, the system tests *B*, which in turn does not support the goal (see Fig. 5). The whole node $\langle B, K \rangle$ depends on $\langle F \rangle$. If it will come out that *F* is true, the agent will perform an action for modifying *B*.

Finally, the agent tests *F*, which results true (see Fig. 6). The second part of the process starts. The agent acts to modify *B*, and eventually *A* to reach the goal. The result is the combination of a backward planning process, and a corresponding forward acting process.

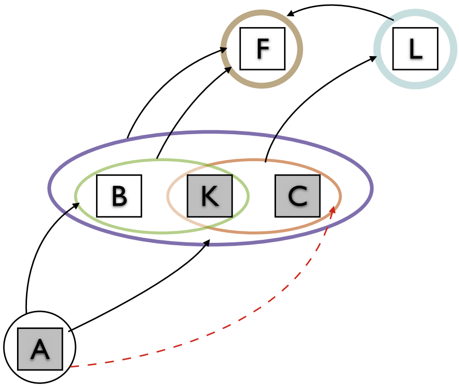

Fig. 4 *C* tested. Confirmation found.

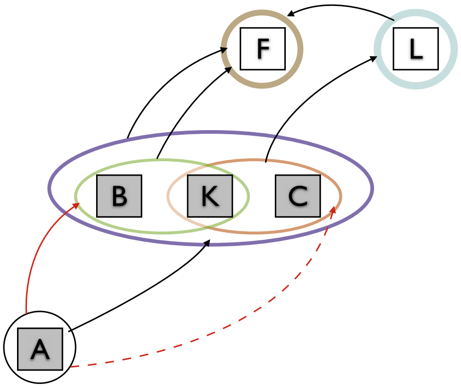

Fig. 5 *B* tested. Partial confirmation found. Now the $\langle B, K \rangle$ node has to be confirmed as a whole.

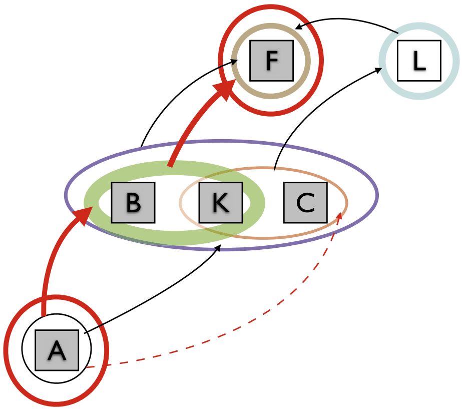

Fig. 6 *F* tested. Final confirmation found.

If we move to goal setting, the knowledge network comes out to be a useful model again. In this respect, the knowledge network can be regarded as a graph $G = \{SG, DL\}$, where $SG = \{sg_1, sg_2, \ldots, sg_n\}$ is the set of all the sub-goals the agent can reach (i.e. the nodes), while *DL* is the set of arcs that link nodes and represent prerequisite conditions. Either positive or negative prerequisites can be devised, so arcs will be labeled as either ''positive'' or ''negative''. When a positive arc links a node A to a node B, B must be made true before for making true A. On the contrary, a negative arc implies that B must be false for making true A.

A generic goal g is a set of sub-goals $g \subset SG$ because reaching a goal implies making true more than one facet at a time. Two goals g_1 and g_2 are distinct if $g_1 \cap g_2 = \emptyset$ otherwise the two goals are partially superimposed. Two goals can be incompatible. Fig. 7 shows four nodes linked in the graph. Node A is true if node B is true, and node C is false. On the other hand, if the agent wants to make the node D true, C should be false. As a consequence, A and D represent conflicting goals, and the agent has to choose between them. Goal setting in our architecture relies on manipulating goals with respect to the labels of their sub-goals.

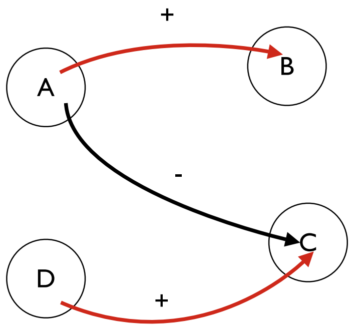

Fig. 7 Sub-goals can be in conflict with each other.

More formally, each goal can be modeled as an array of n labels each of them being associated with a node in the graph (i.e. a sub-goal). Each label can have one of the three values: 0, +, $-$. Label + means that the related sub-goal supports the final goal, label $-$ has the opposite meaning, and label 0 means that the sub-goal does not influence the goal. Let us consider the example shown in Fig. 7. In this case the goal associated to A can be described by the array [+, +, $-$, 0], while B is described by [0, 0, +, +].

Now let us introduce a comparison operator comp$(\cdot,\cdot)$ that analyzes two goals element by element, and is defined by the following Table 2

The When one of the values is equal to 0, the element does not affect the goal, so the result of the comparison is not important (*NI*). In the other cases, if the compared values are equal, the goals are not conflicting (*OK*). On the contrary, conflicting goals cannot be reached together (*KO*). In the proposed example $\mathrm{comp}(\mathrm{A},\mathrm{B}) = [NI, NI, KO, NI]$. In this case, the presence of the value *KO* means that the two goals are conflicting.

The following definitions are given:

Definition 11. A group of goals *GG* is a set of not mutually conflicting goals.

A goal g_i, can belong to more then one set at the same time $GG_k^{(i)}, k = 1, 2, \ldots$. When the agent makes a goal true, it makes also partially true all the goals g_j such that $g_j \in GG_k^{(i)}, j \neq i, k = 1, 2, \ldots$.

Definition 12. Given a set of goals G and a single goal $g_i, MGG^{(i)}$ is the maximum groups of goals associated to g_i that is the set of all goals $g_j \in G$ not conflicting with both g_i and each other.

The $MGG^{(i)}$ can contain one or more sets $GG_k^{(i)}$ because g_i could belong to more group of goals at the same time, while some elements belonging to distinct $GG_k^{(i)}$ are mutually conflicting.

Table 2 definition of the comparison operator.

comp	0	+	$-$
0	*NI*	*NI*	*NI*
+	*NI*	*OK*	*KO*
$-$	*NI*	*KO*	*OK*

Definition 13. Given a set of goals G, a no-conflict partition *NCP* is the set of all possible maximum groups of goals generated for all the goals belonging to G that is $NCP = \{MGG^{(i)}\} \quad \forall g_i \in G$.

Goal setting proceeds as follows. The agent keeps a priority queue for satisfying the goals because each goal has a proper relevance value, which derives from the nature of task at hand. Goals in the queue are arranged in the $MGG^{(i)}$ sets as obtained when building the *NCP*. Each $MGG^{(i)}$ is ranked according to the maximum value of the relevance of its goals. The agent executes the more important goal of the first $MGG^{(i)}$ in the queue. By reaching this goal, the agent reaches partially the other goals of the group.

Goal setting can be regarded as an action at the meta-cognitive level of the architecture. Actually, meta-cognition in the agent implies changing the relevance of some of the goals and a consequent re-ranking of the $MGG^{(i)}$ sets.

4.2. Design of the architecture

The structure of the agent is made by two u-MDP layers: the ''cognitive u-MDP'', and the ''meta-cognitive u-MDP'' that is laid upon the previous one. Fig. 8 shows the conceptual design of this architecture.

The cognitive u-MDP can be regarded as a single instance of the theoretical model presented in Section 3.1. At this level, perceptions are obtained from the external environment, while the state describe the agent's model of the world and its modifications by means of the actions performed on it.

The meta-cognitive u-MDP has its own nodes for perceptions, states, and actions. Its state reflects the meta-cognitive state of the agent, and results from its own perceptions. In this case, perceptions can be related to self-awareness of the internal conditions of the robot or to the need of controlling the alignment between the internal model of the world and external perception. As a consequence, the agent needs to assess the state of the evolution of the plan being executed. For this reason while some perception nodes of the metacognitive u-MDP are connected to the environment, the other ones are connected to the outputs and the states of the cognitive u-MDP. In particular, the output nodes of a u-MDP return the next actions of the

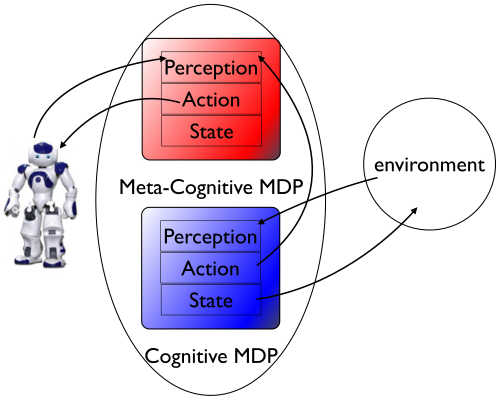

Fig. 8 The cognitive and the meta-cognitive u-MDPs.

agent. The cognitive u-MDP includes new action nodes to pass new perceptual stimuli to the meta-cognitive u-MDP.

Perceptions at the meta-cognitive level can condition the ranking of the agent's goals. Changing the order of the goals' priorities and forcing the re-computation of the cognitive MDP are possible actions of the metacognitive u-MDP. At the same time, some simple actions to manage the internal state of the agent can be devoted to the cognitive u-MDP. In fact, the cognitive u-MDP is devoted to managing the interactions between the agent and the external environment. Actions like the re-arranging memories to get more free space can be managed at the cognitive level. Similarly, when component in the robot works no longer, and it has redundant copies in the system, the metacognitive u-MDP can autonomously switch from the faulty element to the functioning one. In this sense, the agent reflects upon itself, and acts on itself too.

From an implementation perspective, the distinction between the cognitive and the meta-cognitive modules is a mere conceptualization. The two u-MDP have a very tight coupling. The meta-cognitive u-MDP has been designed to perceive not only the mental state of the agent represented by its decisions but its physical parts too:

- The charge of cells (in percentage terms);
- Performance of actuators (a binary variable representing if the actuator is working or not);
- CPU usage (in percentage terms);
- RAM usage (in percentage terms);
- Storage usage (in percentage terms).

At the same time, even if not directly perceived, the agent has a model of the actuators' wear, which modeled as a failure rate described by a negative exponential distribution.

The previous considerations point out that the two layers share many perceptual nodes, so a mere two-stage implementation would result in duplicate computations. As a consequence, cognitive and meta-cognitive nodes have been connected to each other: they belong to a unique graph, and work in a unique u-MDP. Even if they have different semantic and functional roles, they share the same mathematical nature. Obviously transitions between nodes pertaining to different part of the state (i.e. cognitive and meta-cognitive state nodes) are not allowed and the corresponding values for B_T are forced to 0.

5. Conclusions and future work

A novel meta-cognitive architecture for autonomous robots acting in uncertain environments has been presented. In this context, uncertainty is to be considered in relation to both sudden changes in the environment that are not encompassed by the agent's world model and imprecise perceptions coming from sensors. Another source of uncertainty is related to the internal conditions of the robot that may experience malfunctioning and/or diminishing availability of its resources. Such problems, in turn may generate new goals which are partially or totally conflicting with the task at hand. Meta-cognitive abilities are needed in this respect as regards both self-awareness of the internal condition of the robot and control of the actions to maintain the internal model of the world and the external perceptions aligned.

The presented solution derives deals with both these aspects that is uncertainty and meta-cognition. The architecture is able to deal with different uncertainty sources deriving from either inadequate perception of the environment or imprecise actuators independently of their models: probability, possibility, and fuzzy logic. The architecture relies on the u-MDP that is an extended version of the classical MDP and can be instantiated seamlessly using all different kinds of transition uncertainties together.

A two-stage architecture has been developed, which is based on a cognitive u-MDP devoted to the classical perception-action cycle with the external world, and a meta-cognitive u-MDP which perceives the internal conditions of the robot as well as the state of the cognitive module, and acts on goal setting and re-planning. The two u-MDPs have been implemented in a unique graph for increasing performances.

In the proposed formulation the agent accesses the environment without errors. Future work will be devoted to overcome this limitation, and to propose POMDP based version of the architecture to cope with the uncertainty related to perception of the physical environment.

References

Cannella, V., Pipitone, A., Russo, G., & Pirrone, R. (2010) A framework to induce self-regulation through a metacognitive tutor. In *2010 AAAI fall symposium series*.

Cannella, V., & Pirrone, R. (2009). Managing conversation uncertainty in tutorj. In *Proc. of the AAAI 2009 fall symposium in cognitive and metacognitive educational systems (FS-09-02)* (pp. 52—57). Arlington VA, USA: AAAI Press.

Cannella, Vincenzo, Pirrone, Roberto, & Chella, Antonio (2012). Comprehensive uncertainty management in MDPs. In *BICA*. In Antonio Chella, Roberto Pirrone, Rosario Sorbello, & Kamilla R. Johannsdottir (Eds.). *Advances in intelligent systems and computing* (Vol. 196, pp. 89—94). Springer.

Chella, A., Cossentino, M., Gaglio, S., & Seidita, V. (2012). A general theoretical framework for designing cognitive architectures: Hybrid and meta-level architectures for BICA. *Biologically Inspired Cognitive Architectures, 2*, 100—108.

Dubois, D., & Prade, H. (May 2003). Possibility theory and its applications: A retrospective and prospective view. In *The 12th IEEE international conference on fuzzy systems, 2003. FUZZ '03* (Vol. 1, pp. 5—11).

Dubois, Didier, Foulloy, Laurent, Mauris, Gilles, & Prade, Henri (2004). Probability-possibility transformations, triangular fuzzy sets and probabilistic inequalities. *Reliable Computing, 10*.

Dubois, Didier, & Prade, Henri (1983). Unfair coins and necessity measures: Towards a possibilistic interpretation of histograms. *Fuzzy Sets Systems, 10*, 15—20.

Dubois, Didier, & Prade, Henri (1995). Possibility theory as a basis for qualitative decision theory. *Proceedings of the 14th international joint conference on Artificial intelligence* (Vol. 2, pp. 1924—1930). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Dubois, Didier, Prade, Henri, & Sandri, Sandra (1993). On possibility/probability transformations. In *Proceedings of fourth IFSA conference* (pp. 103—112). Kluwer Academic Publ.

Dubois, Didier, Prade, Henri, & Smets, Philippe (2001). New semantics for quantitative possibility theory. In *Proceedings of the 6th European conference on symbolic and quantitative* *approaches to reasoning with uncertainty, ECSQARU '01* (pp. 410—421). London, UK: Springer-Verlag.

Giang, P. H., & Shenoy, P. P. (2000). A qualitative linear utility theory for Spohn's theory of epistemic beliefs. In *Uncertainty in artificial intelligence: Proceedings of the sixteenth conference* (pp. 220—229). San Francisco, CA: Morgan Kaufmann Publishers.

Gwét, Henri (1997). Normalized conditional possibility distributions and informational connection between fuzzy variables. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 5*(2), 177—198.

Jumarie, Guy (1994). Possibility—probability transformation: A new result via information theory of deterministic functions. *Kybernetes: The International Journal of Systems & Cybernetics, 23*(5), 56—59.

Klir, George J., & Parviz, Behzad (1992). Probability—possibility transformations: A comparison. *International Journal of General Systems, 21*(3), 291—310.

Klir, G. J. (1995). Principles of uncertainty: What are they? Why do we need them? *Fuzzy Sets and Systems, 74*(1), 15—31.

Klir, G. J., & Folger, T. A. (1988). *Fuzzy sets, uncertainty, and information*. Prentice Hall.

Metcalfe, J. A., Shimamura, A. P., Metcalfe, J. A., & Shimamura, A. P. . *Metacognition: Knowing about knowing* (Vol. 34). MIT Press.

Novák, Vilém. (2003). *Fuzzy sets: An overview of fundamentals, applications, and personal views George Klir*. Beijing: Beijing Normal University Press, 358 pp. ISBN 7-303-05294-I, Fuzzy Sets and Systems, 133(2):263—265,2003.

Pardo, María José, & Fuente, David de la (2008). Design of a fuzzy finite capacity queuing model based on the degree of customer satisfaction: Analysis and fuzzy optimization. *Fuzzy Sets and Systems, 159*(24), 3313—3332.

Pirrone, R., Cannella, V., Russo, G. (2008). Awareness mechanisms for an intelligent tutoring system. Samsonovich, Alexei V. (Ed.), AAAI Technical, Report FS-08-04.

Sabbadin, Régis, Fargier, Hélène, & Lang, Jérôme (1998). Towards qualitative approaches to multi-stage decision making. *International Journal of Approximate Reasoning, 19*(3—4), 441—471.

Samsonovich, Alexei V. (2012). On a roadmap for the BICA challenge. *Biologically Inspired Cognitive Architectures, 1*, 100—107.

Shafer, Glenn (1976). *A mathematical theory of evidence*. Princeton: Princeton University Press.

Todorovski, Ljupco, Bridewell, Will, Shiran, Oren, & Langley, Pat (2005). Inducing hierarchical process models in dynamic domains. In *Proceedings of the twentieth national conference on artificial intelligence* (pp. 892—897). AAAI Press.

Wonneberger, Siegfried (1994). Generalization of an invertible mapping between probability and possibility. *Fuzzy Sets Systems, 64*, 229—240.

Yamada, K. (July 2001). Probability-possibility transformation based on evidence theory. In *Joint 9th IFSA World Congress and 20th NAFIPS International Conference, 2001* (Vol. 1, pp. 70—75).

Zadeh, L. A. (1999). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems, 100*(Supplement 1(0)), 9—34.