Proceedings ACS (2019) 518–536

The Double Metareasoning Cycle in the CARINA Metacognitive Architecture

Manuel F. Caro MANUELCARO@CORREO.UNICORDOBA.EDU.CO
Dalia P. Madera-Doval DMADERADOVAL@CORREO.UNICORDOBA.EDU.CO
Departamento de Informática Educativa, Universidad de Córdoba, Montería, Colombia
Michael T. Cox MICHAEL.COX@WRIGHT.EDU
Wright State Research Institute, Wright State University, Dayton, OH 45435 USA
Ron Sun DR.RON.SUN@GMAIL.COM
Rensselaer Polytechnic Institute, Troy, NY 12180 USA
Darsana P. Josyula DARSANA@CS.UMD.EDU
Department of Computer Science, Bowie State University, Bowie, MD 20715 USA
Catriona M. Kennedy CATM.KENNEDY@GMAIL.COM
School of Computer Science, University of Birmingham, UK
Rakefet Ackerman ACKERMAN@IE.TECHNION.AC.IL
Industrial Engineering and Management, Technion, Haifa, Israel

Abstract

Metacognitive mechanisms allow cognitive agents to monitor and control their reasoning processes and events that occur in memory. Existing cognitive architectures present simple metacognitive cycles in which they monitor the reasoning that is executed at the object level and do not have much in the way of specific mechanisms to monitor and control the memory tasks that are required in the reasoning process. The main contribution of this article is the presentation of a double metareasoning cycle implemented in the metacognitive architecture CARINA. The results obtained show that a double metareasoning cycle can be implemented to simultaneously monitor and control the reasoning processes as well as memory events of a cognitive agent.

Keywords: Metareasoning, cognitive architecture, metacognitive architecture, perception action cycle, artificial intelligence, computational metacognition, cognitive agent.

1. Introduction

We encounter situations when we cannot recollect some information from the past; for instance, not being able to recollect the names or faces of someone's school mates. When such situations occur, we use different strategies to cope with the memory failure. For instance, we may dig out an old album or call up friends to help with recollecting a name or face. The ability to notice that a memory failure has occurred is important for agents to function effectively. The strategies that one needs to employ to discover new knowledge may be different from what one may use to recover

from memory failure. These strategies may be different from how one handles failures in one's plans to achieve one's goals. For an artificial agent, a memory failure in short term memory may call for a slower deep search through long term memory while a failure to make adequate progress towards its goals may mandate re-planning or re-prioritization of its goals.

A cognitive processing cycle manages the flow of information that takes place between an organism and the environment in which it is immersed (Fuster, 2004). The cognitive cycle is also called the perception-action cycle in cognitive science and artificial intelligence (AI). This cycle may be viewed as the fundamental decision logic of the brain and the mind, serving as the "atoms" of cognition of which higher-level cognitive processes are composed.

The cognitive systems' community has designed a series of cognitive architectures over the years, which specify the underlying infrastructure for intelligent systems (Langley, Laird & Rogers, 2009) and enable simulation and exploration of the cognitive cycle in humans. A cognitive architecture may be defined as a mechanistic (computational) psychological theory about the functioning of the mind, which is composed of processes that produce thoughts and behaviors (Sun, 2018). Cognitive architectures are implemented as computer languages and simulations, which constitute a fixed infrastructure that can be used to create cognitive models of specific tasks (e.g., driving a car) (Forstmann & Wagenmakers, 2015).

In a cognitive architecture, the perception-action cycle (also called the reasoning cycle or the cognitive cycle) forms the basis on which stable, robust and adaptable behavior of the system occurs. A cognitive agent is an intelligent agent whose blueprint is a cognitive architecture. A cognitive agent can contain a set of cognitive mechanisms that describes its internal and external behavior. Cognitive agents are becoming increasingly complex; the selection of actions frequently requires sophisticated reasoning using sensory data and an internal model of the environment. Similarly, cognitive agents process large amounts of often-unstructured information and perform complex searches in memory (such as knowledge organized in the form of semantic networks). Reasoning and memory failures can affect the performance of the agent. To be robust, cognitive agents need mechanisms to detect anomalies and failures that occur in their own cognitive processes. Detection of reasoning failures and analysis of anomalies in information retrieval can enhance robustness, fault-tolerance, and self-repair, which in turn can improve performance. A metacognitive cycle can supplement the cognitive perception-action cycle to detect such reasoning and memory anomalies.

Computational metacognition is an extension of AI and cognitive systems research that represents and models the capacity of intelligent systems to monitor and control their own cognitive processes as opposed to their overt behavior (Anderson, Oates, Chong & Perlis, 2006; Cox, 2005; Cox & Raja, 2011). The goal of computational metacognition is to increase the autonomy and knowledge that an intelligent agent has about their own learning and reasoning process. Much of the work on metacognition (e.g., the metacognitive loop; Schmill et al., 2011) uses a metacognitive cycle similar to the perception-action cycle for introspective monitoring and control of the agent's reasoning processes. This level of information processing is called the meta-level in contrast to direct information processing directed toward solving a cognitive challenge (e.g., calculations) which are at the object level (Cox & Raja, 2011). Now, as an agent gathers more information from its environment, or as the environment changes, the current knowledge in the agent's memory may become inconsistent. Therefore, a need exists for introspective monitoring and meta-level control of the memory events in addition to the reasoning events of an agent.

This article presents the metareasoning processes that occur at the meta-level of the CARINA computational metacognitive architecture. CARINA implements a double cycle of metareasoning as opposed to a single cycle found in other computational architectures. We claim that two distinct metareasoning cycles (one for memory and another for reasoning) can be used as improved approach of single metacognitive cycle, as done before. This claim constitutes a significant differentiating factor that allows CARINA to have broader control, encompassing both the reasoning process and the memory operations independently. CARINA implements a double metacognitive cycle, because the cognitive functions that take place in the object level for addressing reasoning challenges often merit the use of memory functions (e.g., information and strategy retrieval), which must be monitored by the meta-level in addition to monitoring reasoning functions. The double meta-reasoning cycle operates in parallel and the CARINA meta-level monitors both the reasoning process and the memory processes that take place at the cognitive agent's object-level. The main contribution of this article is in describing in detail how the mechanisms of introspective monitoring and meta-level control can be applied to both memory and reasoning processes and compare this approach with similar systems .

This article is organized as follows. Section 2 provides a brief summary of related research, and then Section 3 "Metacognitive loop in CARINA architecture" presents a general overview about the structure of this cognitive architecture. Section 4 "Validation and Comparison with other Models" describes the validation of the proposed architecture through the implementation of a cognitive agent and then compares CARINA with similar architectures. Finally, Section 5 presents the conclusions of this research.

2. Related Work

In cognitive psychology, metacognition refers to "quality assurance" of a person regarding his or her own cognitive processing, like learning, searching information, or solving a problem. It includes metacognitive monitoring—self assessment of chance of success, before, during, and after performing each task item, and metacognitive control—the decisions taken based on the self-monitoring (Nelson & Narens, 1990; see Fiedler, Ackerman, & Scarampi, 2019, for an up-to-date review). With particular relevance for the present study is the metareasoning framework, which provides insights into the ways people monitor and control their reasoning and problem-solving processes (see Ackerman & Thompson, 2017). Understanding how people monitor their knowledge and make decisions during work with software tools, may help designing better user interfaces (Ackerman, Parush, Nassar, & Shtub, 2016). Finally,, metamemory refers to the metacognitive processes relevant to memory performance, such as judgements of learning (Dunlosky, Mueller & Thiede, 2016; Dunlosky & Thiede, 2013).

In the cognitive systems literature, various models of metacognition are used by artificial agents (Caro, Josyula, Cox & Jiménez, 2014), and several authors have studied the metacognitive cycle in cognitive architectures. For example, the *Metacognitive Integrated Dual-Cycle Architecture (MIDCA)* (Cox et al., 2016) is a cognitive architecture that integrates metacognitive theory (Anderson & Perlis, 2005; Schmill et al., 2011) and introspective multistrategy learning theory (Cox & Ram, 1999). The MIDCA architecture depends on " perception-action" cycles both at the object-level (cognitive) and at the meta-level (metacognitive). This work proposes an integral theory of cognition and metacognition, which can instantiate different domains of problem solving

and planning. Other research efforts consider metacognitive processes but without an explicit metacognitive cycle (e.g., Anderson & Fincham, 2014).

Cox and Raja (2011) proposed that using a self-model to perform the metacognitive processes in an intelligent agent was necessary for effective decisions. Here the metacognitive mechanisms of meta-level introspective monitoring and control focus on multi-strategic reasoning. In this model, a dual cycle of reasoning is established. Its first cycle is a perception-action cycle of reasoning. The intelligent agent in this cycle receives perceptions of the environment, makes decisions (reasoning) and acts by modifying the environment (Raja & Lesser, 2007). However, the second cycle of the model operates on input that the meta-level has regarding the object-level. That is, the meta-level makes decisions (metareasoning) about the information that comes from the object-level. The model is also extensible to the distributed case for a multi-agent system (see Raja, Alexander, Lesser, & Krainin, 2011; Kennedy, 2011).

The *Metacognitive Loop (MCL)* (Schmill et al., 2011) uses a note-assess-guide cycle to note anomalies, assess failures and suggest responses to deal with the failures that an agent may encounter. MCL operates on an ontology of generic indications, failures and responses. The instantiation of a cognitive agent based on MCL connects the ontology of indications to agent-specific expectation violations and the ontology of responses to agent-specific actions. In this sense, MCL based agents follow a uniform processing for both meta-level and object-level failures.

Emotion Machine One (EM-ONE) is a complex cognitive system based on Minsky's ideas in his book *The Emotion Machine* (Minsky, 2006) that performs commonsense reasoning. The metamemory strategy of EM-ONE is based on a case-based memory and reasoning system. In EM-ONE, there are mental critics for answering problems in the world and other mental critics for answering the problems in the EM-ONE system itself (Caro et al., 2014).

The architectures mentioned above present simple metacognitive cycles which monitor the reasoning that is executed at the object level, while the specific mechanisms for monitoring and controlling the memory tasks involved in the reasoning process are overlooked. Based on the above considerations, the research problem for the CARINA architecture is to describe the interactions that take place in the metacognitive cycle that implements the mechanisms of introspective monitoring and metacognitive control over both the reasoning processes as well as the memory functions that take place in the object level of a cognitive agent when addressing a reasoning challenge.

3. The Metacognitive Cycle of the CARINA Architecture

The meta reasoning cycle of CARINA focuses on the following two basic processes: (a) detection of reasoning failures; and (b) detection of anomalies in the events that occur in its own memory system. Figure 1 represents the metacognitive cycle in CARINA. The inputs for self-regulation of reasoning are the computational data generated by the reasoning task and the output consists of recommendations, which may vary according to the reasoning task. While for metamemory, the inputs are the memory events that occur in the long-term memory (LTM) and the outputs are the recommendations that may vary according to the memory events.

Self-regulation in the present work is focused on the metareasoning process that allows choosing the best strategy to correct a reasoning failure, while metamemory is centered on the reasoning process that allows adaptation to anomalies related to retrieving information from LTM.

The following subsections describe the mechanisms of introspective monitoring and metacognitive control.

3.1 Introspective Monitoring

Introspective monitoring is a metareasoning mechanism implemented at the meta-level in CARINA. Using the common terms of cognitive science, the notion of ‘‘mechanism” involves both representations as well as processes operating on them (Sun, 2009). Introspective monitoring provides functions to identify reasoning failures at the object-level. The main objective of introspective monitoring is to collect enough information to make effective decisions for meta-level control (Caro, Gomez, & Giraldo, 2017). In this way, the monitoring process is performed based on the information gathered at the meta-level from the object-level.

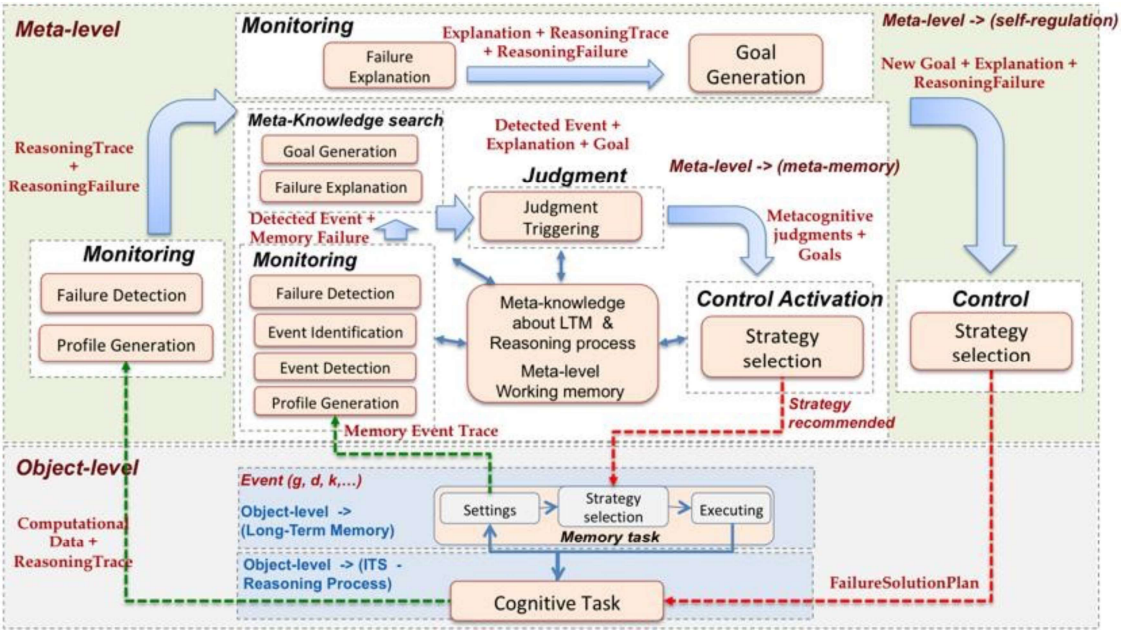

Figure 1. Functional view of double metacognitive cycle in CARINA including introspective monitoring and meta-level control. The inner loop monitors and controls memory; whereas, the outer loop monitors and controls reasoning.

3.1.1 Metareasoning monitoring

The main cognitive functions performed in the metareasoning monitoring process are directed at self-regulation of the reasoning process: `FailureDetection`, `FailureExplanation` and `GoalGeneration`. `FailureDetection` is a metacognitive function that allows the detection of failures in the reasoning processes that occur at the object-level. `FailureExplanation` is a metacognitive function that allows the generation of explanations for the reasoning failures identified in reasoning processes and performed at the object-level. The `GoalGeneration` metacognitive function allows the generation of new goals in order to deal with reasoning failures at the object-level.

Cognitive tasks generate computational data during their execution. `ProfileGeneration` processes the computational data and generates a metacognitive profile of the `CognitiveTask` that is in execution. Each `CognitiveTask` at the object-level has a `PerformanceProfile` associated at the meta-level. The profile allows the meta-level to be informed in real time about the state of the reasoning processes that take place at the object level.

The function of the Sensor is to monitor the profiles of cognitive tasks to detect anomalies, which can generate reasoning failures. `FailureDetection` reads the properties of a `Sensor`. `FailureExplanation` generates an explanation or cause of the reasoning failure, using as inputs, the evaluation of the failure and the trace of reasoning. `GoalGeneration` uses the Explanation as input to produce new goals to solve the detected error. `FailureSolutionPlan` is a plan that is generated based on the new goal to solve the reasoning failure.

3.1.2 *Metamemory monitoring*

The metamemory monitoring includes mechanisms for detecting events in memory and performing deep search processes on the meta-level knowledge about the object-level.

The main cognitive functions performed in the metamemory process are: `ProfileGeneration`, `EventDetection`, `EventIdentification`, `FailureDetection`, `Failure-Explanation`, `JudgmentTriggering` and `GoalGeneration`. A `MemoryTask` generates computational data when it is running. `ProfileGeneration` reads the computational data to generate a profile for the `MemoryTask` at the meta-level.

In CARINA, the processes operating on the memory such as encode, retrieval, and storage of information are considered as `MemoryEvent`. Sensors monitor instances of `MemoryEvent` to detect anomalies between expectations and observations about the performance of memory tasks. `FailureDetection` evaluates the anomalies and identifies possible `ReasoningFailures`. The `FailureExplanation` task generates an Explanation of the possible cause of the `ReasoningFailure`. `JudgmentTriggering` reads the explanations and triggers a `MetamemoryJudgment` about the `ReasoningFailure`. For example, if the `ReasoningFailure` has something to do with data that cannot be retrieved from memory, then `MetamemoryJudgment` can report that the system knows there is no sufficient information for the search.

3.2 **Meta-Level Control**

The goal of meta-level control is to improve the quality of CARINA's decisions by spending effort to decide what and how much reasoning to do as opposed to what actions to do.

3.2.1 *Metareasoning control*

The main function of the metareasoning control mechanism is to recommend to the object-level the best computational strategy to resolve a reasoning failure; in this way the meta-level control improves the quality of decisions made by the object-level. The meta-level control decides whether or not to invoke a task, which task to invoke, and how much resource to invest in the reasoning process (Dannenhauer et al., 2014). `ControlActivation` and `StrategySelection` are the main control functions in CARINA. When a reasoning failure is detected then the meta-level control mechanism is activated. The implementation of the failure solution plan is the main action started by `ControlActivation`. Once a `ReasoningFailure` is detected and explained by the meta-level, then this metacognitive task assesses the available strategies to be selected and the most appropriate one to address the reasoning failure at the object-level.

3.2.2 *Metamemory control*

The metamemory control mechanism includes processes for the recommendation of search strategies on memory. The main control functions in CARINA are: `StrategySelection` and `PlanExecution`. In metamemory, `StrategySelection` uses search task constraints and metamemory judgments as additional inputs. Additional inputs in the metacognitive control are inherent to memory functions, for example, the meta-level using a `MetamemoryJudgment` may: (i) assess whether or not the information is being stored; and (ii) consider making a deeper search for information. `PlanExecution` maintains the same structure as the self-regulation functions. `SearchStrategy` is a strategy of searching for information that may be used by a search task.

4. Validation and Comparison with Other Models

This section provides a brief description of an experiment performed with CARINA to illustrate the benefit of the double metareasoning cycle. It then compares the CARINA approach to other architectures that include a metacognition component.

4.1 Validation

The proposed double metareasoning cycle was validated in the CARINA version 3-beta (Caro et al., 2019), which was implemented in JavaScript 6. The CARINA-based cognitive agent developed for this test is named *CAT-SDG (Cognitive Agent Translator for Sustainable Development Goals)*. A set of 17 Sustainable Development Goals (SDGs) is part of United Nations 2030 agenda (United Nations, 2019). The SDGs address the global challenges, including those related to poverty, inequality, environmental degradation, prosperity, and peace and justice. It is important to achieve each Goal by 2030. Figure 2 shows a sample display screen of the CAT-SDG cognitive agent that translates from natural language to the SDG language. In this case, the response of a user to the question "What are the three main problems that overwhelm your community?" is presented.

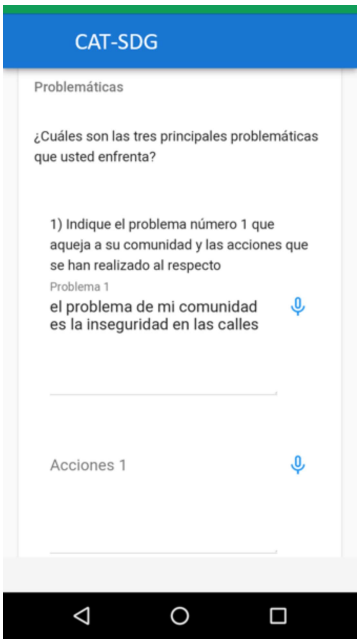

Figure 2. The screenshot of CAT-SDG cognitive agent shows one of the questions asked to the users about the objectives of sustainable development. The question shown is "What are the three main problems that overwhelm your community?"

CAT-SDG aims to translate guided conversations in natural language into the language of the SDGs. The agent takes as input the Spanish text of the participants' response in guided conversations about the perception of the people about the status of the Sustainable Development Goals in the regions of Colombia.

The test consisted of processing the responses of 15 subjects selected at random. The subjects were voluntary students of the Faculty of Education and Human Sciences of the University of Córdoba, with a range of ages between 18 and 25. The processing time of each response on average was 357.778 milliseconds. For the test, the memory of CARINA had a semantic network formed around 300 concepts related to the objectives of sustainable development. Figure 3 shows the output of the agent's console, where evaluations and judgments of introspective monitoring mechanisms are rendered for display.

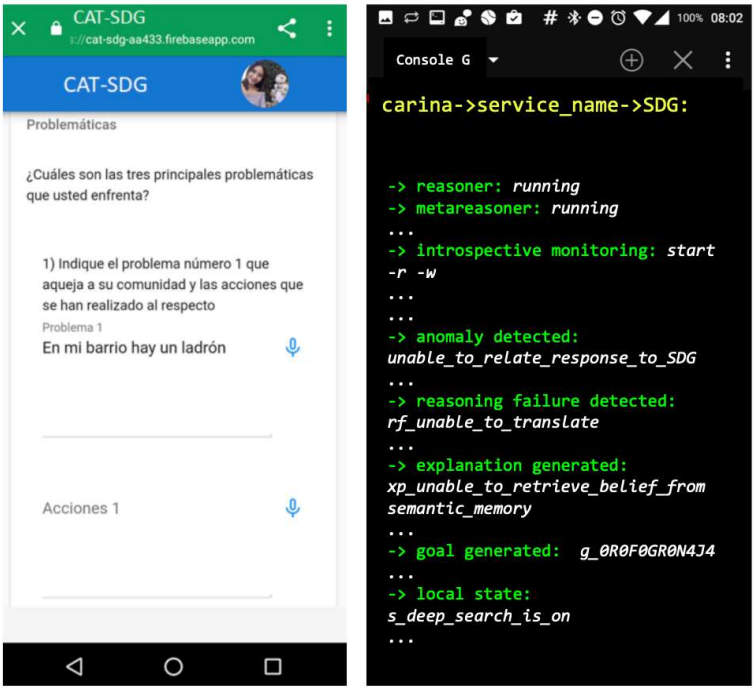

Figure 3. User response that failed to be translated to SDG and the CARINA console output with a meta-reasoning cycle trace.

The meta-level of the agent verifies if there are reasoning failures or anomalies in reasoning and memory processes. The `ProfileGeneration` function reads the computational data and the reasoning trace to generate a cognitive profile that is kept up to date at all times. In formula (1) we can see the profile of the function at the meta level. Table 1 present an example of a profile in CAT-SDG.

Profile λ consists of the set of values related to the processing and performance of the cognitive function.

$\lambda = \{ID, B, E, S, C, IP, OP, T\}$ Formula(1)

with:

ID is the identifier of the cognitive function.
B is the time stamp of when the cognitive function was started.
E is the time stamp of when the cognitive function is finished.
S is the state of the cognitive function, $s \in S$ and $S=\{active, inactive\}$
C is the priority level for focus attention $c \in C$ and $C=\{low, medium, high\}$.
IP is the set of parameters used as input of the cognitive function.
OP is the output of the cognitive function.
T is the reasoning trace of the cognitive function

Table 1. Profile of a cognitive function at the meta level

id	B	E	S	C	IP	OP	T
…	…	…	…	…	…	…	…
043BxyZ4	1562592403346	156259240383	active	high	04XgG847	04rTW473	04OfR483
…	…	…	…	…	…	…	…

The meta level uses the `Sensor` to read the profile and the `FailureDetection` function compares the `Sensor` observations with the performance expectations about the reasoning process. Figure 4 describes the data structures used to generate a *reasoning failure* cognitive element.

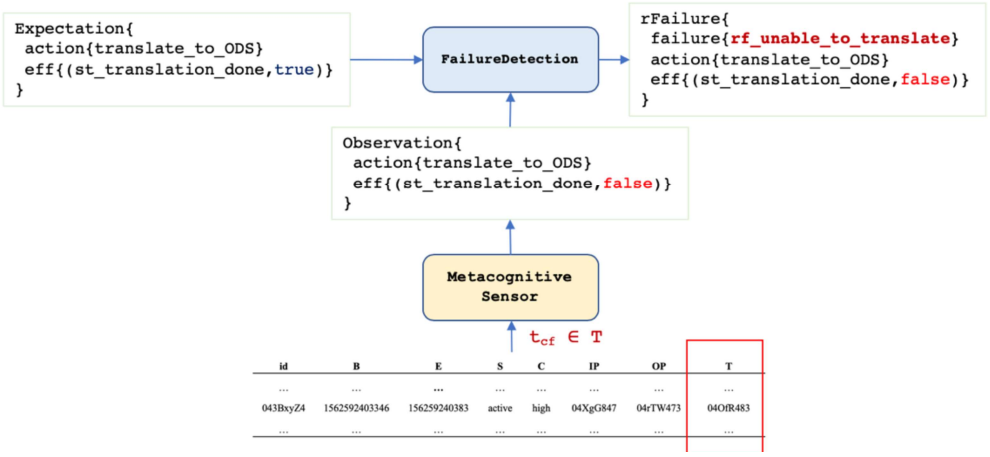

Figure 4. Sensor in metareasoning cycle

Figure 5 shows a partial view of the possible fails that can occur at object level. The set of reasoning failures and memory failures are called `cog_failure.`

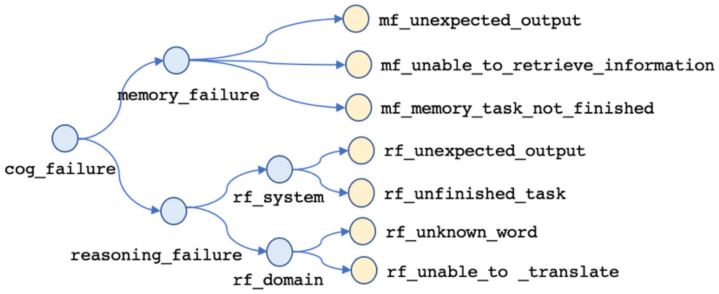

Figure 5. Partial view of cog-failure including reasoning failures and memory failures

In the test, the double metareasoning cycle found one reasoning failure and one memory failure. Figure 6 partially shows the trace of the meta-reasoning cycle. `FailureExplanation` generates an explanation or cause of the reasoning failure, using as inputs, the evaluation of the failure and the trace of reasoning. In case there are no failures, the agent makes the translation and generates the output with respect to the Sustainable Development Goals.

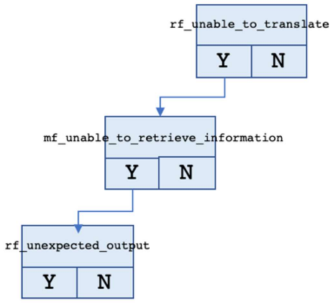

Figure 6. Partial view of metacognitive explanations in the form of causal tree.

The reasoning failure was caused by a memory failure. This is because the keyword "*ratero*" (i.e., thief in English) in the user's response had no associations with any ODS in CARINA's semantic memory. The `GoalGeneration` metacognitive function generate a new goal in order to deal with reasoning failure at the object-level.

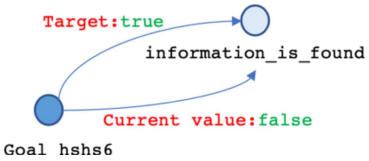

Figure 7. Partial view of goal ontology with the goal that satisfy the explanation

Finally, the metamemory control recommended a deep search, which was carried out by reviewing the synonyms of the word "*ratero*". Then a synonym for the word that was related to SDG 16 (i.e., peace, justice and strong institutions) was found. This fact solved the reasoning failure, and the discourse of the subject was translated successfully into the SDG language. Figure 8 shows the CARINA console, once the translation has been done.

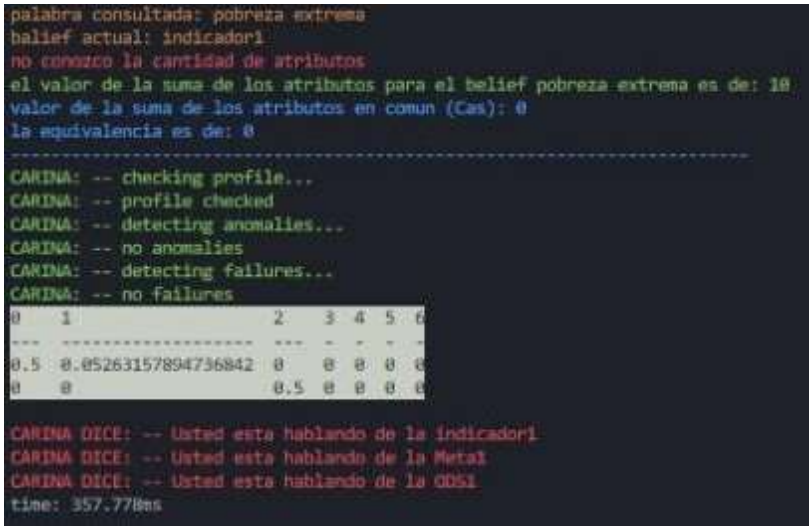

Figure 8. The console of the CAT-SDG agent with evaluation and judgement output.

4.2 **Comparison**

With regard to the support of the main metacognition components, i.e., metamemory and self-regulation, it can be seen that most of the architectures do not provide support for both. Table 2 summarizes each system's approach to monitoring and controlling memory and cognitive activity. Monitoring and controlling memory are listed as metamemory; whereas, monitoring and controlling cognition is termed self-regulation.

Table 2. Functional view of the double metacognitive cycle in CARINA including introspective monitoring and meta-level control.

Model	Metamemory	Self-regulation
Meta–AQUA (Cox & Ram, 1999)	Memory awareness	Detects explanation failures and learns not to repeat them.
Clarion (Sun, 2016, 2018; Sun et al., 2006)	Memory monitoring and metacognitive reasoning on that basis	A metacognitive subsystem monitors and regulates (sets parameters for) other subsystems.
MCL, Alfred (Anderson, Oates, Chong, & Perlis, 2006; Schmill et al., 2011; Josyula, 2005)	Basic mechanisms of short-term memory	Anomaly detection - monitoring and control
DMF (Kennedy & Sloman, 2003)	Distributed memory system	Simple anomaly detection and self-repair

MIDCA (Cox et al., 2016)	No monitoring or control.	Introspective monitoring and meta-level control to change object-level goals.

4.2.1 Meta-AQUA

The Meta-AQUA system (Cox & Ram, 1999) is a multistrategy learning system that improves its ability to understand and explain characters and events in various input stories. When it encounters characters that perform unexpected actions, it attempts to retrieve from its case-based memory an explanation pattern that can resolve the interpretation anomaly. However, explanations can be missing from memory or the indexes used to retrieve explanations might not match the cues used to attempt retrieval (Cox, 1994). Like the cognitive cycle that retrieves explanation patterns to explain story events, the meta-level cycle retrieves meta-explanations to explain reasoning failures in the story understanding task (e.g., explanation failure). These meta-level structures link the symptoms of failure to the causes of such failures and contain learning goals to guide the learning of new explanations or to correct existing ones.

In relation to metamemory, Table 1 indicates that Meta-AQUA has an awareness of its memory component but does not have a specialized cycle to manage its memory activity separately. Instead retrieval is treated as a basic cognitive process along with other reasoning, and traces of all such activities are bound together for introspective monitoring. Meta-level control (i.e., self-regulation) is in the form of learning that modifies the explanations and their storage indexes in its casebase (i.e., memory), hence improving future performance. Furthermore, the object-level cycle includes perception in the form of the interpretation of story input, but it does not include action execution. Note however that in a previous experiment, Meta-AQUA was combined with an action planner to demonstrate how the system might be extended in the future (Cox, 2007).

4.2.2 Clarion

The Clarion cognitive architecture is divided into a number of subsystems, including, in particular, the metacognitive subsystem (Sun, 2016). The metacognitive subsystem carries out a variety of metacognitive functions (as broadly defined) within Clarion: setting goals, filtering information, determining reasoning methods, determining learning methods, monitoring ongoing processes and interrupt or control them as needed, setting other essential parameters for the other subsystems, and so on (Sun, 2016, 2018). Some detailed simulations of human metacognition have been carried out. For example, in Sun et al. (2006), meta reasoning was simulated through capturing data of human reasoning in which lack of knowledge was used to infer conclusions that could not be obtained otherwise. A metacognitive monitoring buffer was involved as the basis of such meta reasoning. However, this simulation did not have complex control of reasoning or memory per se as in the present work, which adds to the overall sophistication of the cognitive agent.

In addition, it should be noted that in Clarion, the metacognitive subsystem is closely tied to the motivational subsystem. Motivation within the motivational subsystem provides the foundation for metacognitive control and regulation, while at the same time metacognition regulates motivation as well (e.g., as in emotional regulation; Sun et al., 2016). In Clarion, most of the metacognitive functions within the metacognitive subsystem are carried out on the basis of motivation, that is, on the basis of the needs and motives at each moment of decision making or reasoning (Sun, 2016).

4.2.3 MCL

In MCL, aspects referring to the metamemory strategies that can be used to learn from the detected failures are omitted (Schmill et al., 2007). Moreover, MCL's basic mechanisms of short-term memory are matched with long-term memory in the meta-level. McNany, et al. (2013) analyzed the benefits of metacognition in the dialogue agent Alfred (Josyula 2005), who acts as a mediator between a human and a task-oriented domain to examine the expected pause time between the utterances of a conversation using MCL inside the intelligent agent Alfred uses interleaved metacognition wherein MCL runs in sync with the cognitive processing and thus avoids the problem of meta reasoning blocking the cognitive processing. Alfred relies on expectations to detect anomalies and reasoning failures using MCL. Alfred has a limited mechanism to deal with anomalies in memory that do not manifest as expectation violations. The underlying Alma/Carne (Purang et al., 2001) reasoning engine detects direct contradictions that may occur in its current memory and trigger a memory event to distrust them. The contradiction detection mechanism, the expectation violation detection mechanism and the perception action cycle all happen in sync and hence it is not easy to study the interactions that happen between the different processes.

4.2.4 DMF

The *Distributed Metacognition Framework (DMF)* represents metacognition as a distributed system (Kennedy, 2010) that involves various metacognitive agents that act together. These agents can also "discuss" with each other (Kennedy, 2011). A reliable autonomous system needs a sufficient agreement of the agents on what type of metacognitive control is required. A metacognitive agent can be a complete agent with its own reasoning at the object level. A meta-level process analyzes one or more traces of reasoning generated by an object-level or another meta-level. A simplified version was implemented in Kennedy and Sloman (2003) as a multiagent simulation, where each agent has an object level cycle (sequence "sense-decide-act") followed by a meta-level cycle. The meta-level processing does not involve memory events. Self-regulation is a simple process of "self-repair" and does not include reasoning about the causes of failure.

4.2.5 MIDCA

The *Metacognitive Integrated Dual-Cycle Architecture (MIDCA)* (Cox et al., 2016; Cox, Oates, & Perlis, 2011) integrates both comprehension and problem-solving processes at the object-level with a cycle of monitoring and control at the meta-level. MIDCA aims to provide intelligent agents with greater autonomy to solve problems. Goals formulation is an essential process in MIDCA and takes place both at the object-level and at the meta-level. At the meta-level, the reasoning cycle generates goals that change the object-level goals. In the same way, the metacognitive "perceptual" components monitor in an introspective way the object-level processes and changes in the mental state of the agent. They do this by recording a declarative trace of the object-level activities and then reasoning about the trace structure. This is similar to the introspective monitoring mechanism of CARINA (Florez, 2019; Florez, Gomez, & Caro, 2018).

However MIDCA does not reason about memory functions in any way at this time. In fact, the current implementation has an extremely simple memory consisting mainly of a series of state variables. Future research intends to add a more sophisticated memory such as Meta-AQUA's. We experimented with adding Meta-AQUA as a component to MIDCA (see Paisner, Cox, Maynord,

& Perlis, 2014), but given that MIDCA is in Python and Meta-AQUA Lisp, the sharing of memory elements is extremely convoluted. A longer-term solution remains to be created.

4.3 Discussion

The analyzed cognitive architectures present a well-defined component that supports the self-regulation functions within the computational architecture. In MIDCA, the meta-level can act as an executive function similar to CLARION. CLARION and MCL have a better development of metacognitive processes than the rest of architectures. The results obtained in the validation allow affirming that two different metareasoning cycles (one for memory and the other for reasoning) allow better performance in cognitive systems than the combination of both in a single metacognitive model. The double cycle of meta reasoning allowed to solve a problem of reasoning at the level of object-level that had origin in a memory failure. Had metacognitive control not been activated, CARINA would have been unable to solve the reasoning failure in the translation.

The mechanisms of the double metareasoning cycle lead to a better understanding of a complex topic in itself (i.e., computational metacognition), because having a single cycle combines functionally different processes, whereas the double metareasoning cycles separate the processes into functionally independent mechanisms. In addition, this concept facilitates a modular approach to building effective software and, therefore, applied cognitive agents.

In psychology, metacognition plays an important role in monitoring and control of a cognitive task, for example, by providing a "feeling of rightness" or "feeling of error" (Ackerman & Thompson, 2017) and by deciding on strategies to resolve uncertainty (such as memory searching or asking for help). Notably, so far, in cognitive psychology the metamemory and the metareasoning frameworks were accounted as “sister” sub-domains of metacognitive research (see Fiedler et al., 2019). The interplay between metacognitive processes involved in reasoning and in memory during the performance of a single task, as suggested in this computational work, has not been considered for understanding human behavior. Thus, this work carries the potential to feedback theoretical ideas to psychological research.

Metacognition can interact with emotion (e.g. Hudlicka, 2005) and can also help to regulate emotion (e.g. Sun et al. 2016). For emotion regulation, meta-level monitoring can detect poor performance on a cognitive task, such as repeated distraction or increased perception of difficulty caused by emotion or lack of motivation. Current work on emotion regulation (Kennedy, 2018) is developing an architecture H-Meta, which is partly based on H-CogAff (Sloman et al, 2005) and uses the working definition of emotion for H-CogAff, namely, an "interruption or modulation" of a current deliberative process. H-Meta also builds on earlier work on metacognition as a distributed system (Kennedy, 2011), which involves diverse metacognitive specialists. Such specialists may also "argue" with each other. A reliable autonomous system would need sufficient agreement from specialists on what kind of meta-level control (if any) is required. In human cognitive modelling, however, metacognitive specialists may compete with each other (causing indecision).

The current focus of H-Meta is to use Gross's model of emotion regulation (Gross & Thompson, 2007) and cognitive reappraisal as a regulation strategy. According to Gross, the reapprasial process is defined as "changing how we think about a situation in order to decrease its emotional impact". This involves monitoring and control of memory. The H-Meta approach to emotion regulation does

not specify metacognitive cycles in detail and only has a very basic metamemory. Therefore, it can benefit from the current research on CARINA and comparable architectures.

5. Conclusions

In this work a double cycle of metareasoning has been described. The double metacognitive cycle has been implemented in the metacognitive architecture CARINA, this architecture allows cognitive agents to monitor and control their reasoning processes and events that occur in memory. The metacognitive mechanisms of introspective monitoring and meta-level control are implemented in metareasoning cycles. The main contribution of this work is the integration of introspective monitoring and metacognitive control of the memory processes that are carried out during the meta-association cycle in a cognitive agent. The process of implementing the double metacognitive cycle was done using JavaScript 6 in CARINA version 3.

In this sense, a cognitive cycle was developed by implementing the double metacognitive cycle using the translation of natural language into the language of sustainable development goals. The results of the test of execution of cognitive agent showed that the processes of monitoring and control of reasoning were executed in parallel with the processes of monitoring and control of memory events invoked by the cognitive functions of the agent. The results obtained show that a double metacognitive cycle can be implemented to simultaneously monitor and control the reasoning processes and the events that occur in the memory of a cognitive agent.

As a continuation of this work there are several lines of research that remain open and in which the application of metacognitive CARINA architecture in productive problems that involve novel situations where the performance of double metacognitive CARINA cycles is required.

On possible future direction is taking into consideration of a cognitive agent's needs and goals at each moment in deciding what metacognitive functions to perform or how to perform them. For example, how deep reasoning may be determined on the basis of current task demands and cost-benefit considerations.

Acknowledgements

This research was funded in part by the project COL / 86815 – CONV. 647/15 of the Ministry of Information Technologies and Communications of Colombia (MinTIC) and Internal Call Sustainability of Research Groups of the University of Córdoba. Also, the material is partially based upon work supported by AFOSR grant #FA2386-17-1-4063, by ONR grant #N00014-18-1-2009, and by DARPA contract number N6600118C4039. Finally, we thank Danielle Brown for comments and suggestions on an earlier draft of this paper.

References

Ackerman, R., Parush, A., Nassar, F., & Shtub, A. (2016). Metacognition and system usability: Incorporating metacognitive research paradigm into usability testing. *Computers in Human Behavior, 54*,101-113.

Ackerman, R., & Thompson, V. (2017). Meta-reasoning: Monitoring and control of thinking and reasoning. *Trends in Cognitive Sciences, 21*(8), 607-617.

Anderson, J. R., & Fincham, J. M. (2014). Extending problem-solving procedures through reflection. *Cognitive Psychology, 74*, 1-34.

Anderson, M. L., Oates, T., Chong, W., & Perlis, D. (2006). The metacognitive loop I: Enhancing reinforcement learning with metacognitive monitoring and control for improved perturbation tolerance. *Journal of Experimental and Theoretical Artificial Intelligence, 18*(3), 387-411.

Anderson, M. L., & Perlis, D. (2005). Logic, self-awareness and self-improvement: The metacognitive loop and the problem of brittleness. *Journal of Logic and Computation, 1*5(1), 21-40.

Caro, M. F., Gómez, A. A., & Giraldo, J. C. (2017, July). Algorithmic knowledge profiles for introspective monitoring in artificial cognitive agents. In *2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)* (pp. 475-481). IEEE.

Caro, M. F., Josyula, D. P., Cox, M. T., & Jiménez, J. A. (2014). Design and validation of a metamodel for metacognition support in artificial intelligent systems. *Biologically Inspired Cognitive Architecture, 9*, 82–104.

Caro, M. F., Josyula, D. P., Madera, D. P., Kennedy, C. M., & Gómez, A. A. (2019). The CARINA metacognitive architecture. *IJCINI*, Volume 13, Issue 4, Article 4. 260119-053119

Cox, M. T. (2007). Perpetual self-aware cognitive agents. *AI Magazine, 28*(1), 32-45.

Cox, M. T. (1994). Machines that forget: Learning from retrieval failure of mis-indexed explanations. In A. Ram & K. Eiselt (Eds.), *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* (pp. 225-230). Hillsdale, NJ: Lawrence Erlbaum Associates.

Cox, M. T. (2005). Metacognition in computation: A selected research review. *Artificial Intelligence. 169*,(2), 104–141.

Cox, M. T., Alavi, Z., Dannenhauer, D., Eyorokon, V., Munoz-Avila, H., & Perlis, D. (2016). MIDCA: A metacognitive, integrated dual-cycle architecture for self-regulated autonomy. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Vol. 5 (pp. 3712-3718). Palo Alto, CA: AAAI Press.

Cox, M. T., Oates, T., & Perlis, D. (2011). Toward an integrated metacognitive architecture. In P. Langley (Ed.), *Advances in Cognitive Systems: Papers from the 2011 AAAI Fall Symposium* (pp. 74-81). Technical Report FS-11-01. Menlo Park, CA: AAAI Press.

Cox, M. T., & Raja, A. (2011). Metareasoning: An introduction. In M. T. Cox & A. Raja (Eds.) *Metareasoning: Thinking about thinking* (pp. 3-14). Cambridge, MA: MIT Press.

Cox, M. T., & Ram, A. (1999). Introspective multistrategy learning: On the construction of learning strategies. *Artificial Intelligence, 112*, 1-55.

Dannenhauer, D., Cox, M. T., Gupta, S., Paisner, M., & Perlis, D. (2014). Toward meta-level control of autonomous agents. *Procedia Computer Science*, *41*, 226-232.

Dunlosky, J., Mueller, M. L., & Thiede, K. W. (2016). Methodology for investigating human metamemory: Problems and pitfalls. In J. Dunlosky & S. K. Tauber (Eds). *Handbook of Metamemory*. (pp. 23-37). New York: Oxford University Press.

Dunlosky, J., & Thiede, K. W. (2013). Metamemory. In D. Reisberg (Ed.) *Oxford Library of Psychology*. (pp. 283-298). New York: Oxford University Press.

Fiedler, K., Ackerman, R., & Scarampi, C. (2019). Metacognition: Monitoring and controlling one's own knowledge, reasoning and decisions. In R. J. Sternberg & J. Funke (Eds.). *Introduction to the Psychology of Human Thought* (pp. 89-111). Heidelberg: Heidelberg University Publishing.

Florez, M. A. (2019). *Reasoning traces of the cognitive function perception of the metacognitive architecture CARINA*. Undergraduate thesis, Cordoba University, Department of Educational Informatics, Monteria.

Florez, M. A., Gomez, A., & Caro, M. (2018). Formal Representation of Introspective Reasoning Trace of a Cognitive Function in CARINA. In *Proceedings of the 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)* (pp. 620-628). IEEE.

Forstmann, B. U., & Wagenmakers, E. J. (2015). *An introduction to model-based cognitive neuroscience*. (pp. 1–354). Berlin: Springer.

Fuster, J. M. (2004). Upper processing stages of the perception-action cycle. *Trends in Cognitive Science, 8*(4), 143–145.

Gross, J. & Thompson, R. (2007). Emotion regulation: Conceptual foundations. In: Gross, J. (Eds.) *Handbook of emotion regulation* (pp. 3-24). New York, NY: The Guilford Press.

Hudlicka, E. (2005). Modeling interaction between metacognition and emotion in a cognitive architecture. In *AAAI Spring Symposium: Metacognition in Computation*. Tech. Rep. No. SS-05-04, 55-61. Menlo Park, CA: AAAI.

Josyula, D. (2005). *A unified theory of acting and agency for a universal interfacing agent*. Doctoral dissertation, University of Maryland, Department of Computer Science, College Park.

Kennedy, C. M. (2018). Computational modelling of metacognition in emotion regulation. *Emotion and Computing workshop at the 41st German Conference on Artificial Intelligence* (KI 2018), Berlin, September 24-28.

Kennedy, C. M. (2011). Distributed metamanagement for self-protection and self-explanation. In M. T. Cox & A. Raja (Eds.) *Metareasoning: Thinking about thinking* (pp 233–347). Cambridge, MA: MIT Press.

Kennedy, C. M. (2010). Decentralised metacognition in context-aware autonomic systems: some key challenges. In *Workshops at the Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Kennedy, C. M., & Sloman, A. (2003). Autonomous recovery from hostile code insertion using distributed reflection. *Cognitive Systems Research*, *4*(2), 89-117.

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research, 10*, 141-160.

McNany, E., Josyula, D., Cox, M., Paisner, M., & Perlis, D. (2013). Metacognitive guidance in a dialog agent. In *Proceedings of the Fifth International Conference on Advanced Cognitive Technologies and Applications* (pp. 137-140).

Minsky, M. (2006). *The emotion machine: Commonsense thinking, artificial intelligence, and the future of the human mind*. New York: Simon & Schuster.

Nelson, T. O., & Narens, L. (1990). Metamemory: A theoretical framework and new findings. *The Psychology of Learning and Motivation, 26*, 125-141.

Paisner, M., Cox, M. T., Maynord, M., & Perlis, D. (2014). Goal-driven autonomy for cognitive systems. In *Proceedings of the 36th Annual Conference of the Cognitive Science Society* (pp. 2085–2090). Austin, TX: Cognitive Science Society.

Purang, K. (2001). Alma/Carne: implementation of a time-situated meta-reasoner, In *Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence*. (pp. 103-110). IEEE.

Raja, A., Alexander, G., Lesser, V., & Krainin, M. (2011). Coordinating agents' metalevel control. In M. T. Cox & A. Raja (Eds.) *Metareasoning: Thinking about thinking* (pp 201-215). Cambridge, MA: MIT Press.

Raja, A., & Lesser, V. (2007). A framework for meta-level control in multi-agent systems. *Autonomous Agents and Multi-Agent Systems, 15*(2), 147-196.

Schmill, M. D., Anderson, M. L., Fults, S., Josyula, D., Oates, T., Perlis, D., Shahri, H., Wilson, S. & Wright, D. (2011). The Metacognitive loop and reasoning about anomalies. In M. T. Cox & A. Raja (Eds.) *Metareasoning: Thinking about thinking* (pp. 183–198). Cambridge, MA: MIT Press.

Schmill, M. D., Josyula, D., Anderson, M. L., Wilson, S., Oates, T., Perlis, D., & Fults, S. (2007). Ontologies for reasoning about failures in AI systems. In *Proceedings from the Workshop on Metareasoning in Agent Based Systems at the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems*.

Singh, P. (2005). *EM-ONE: An architecture for reflective commonsense thinking.* Doctoral dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge.

Sloman, A., Chrisley, R., & Scheutz, M. (2005). The architectural basis of affective states and processes. In: Fellous, J.-M., Arbib, M.A. (eds.) *Who needs emotions?* New York: Oxford University Press.

Sun, R. (2018). Why is a computational framework for motivational and metacognitive control needed? *Journal of Experimental and Theoretical Artificial Intelligence*, *30*(1), 13-37.

Sun, R. (2016). *Anatomy of the Mind: Exploring Psychological Mechanisms and Processes with the Clarion Cognitive Architecture*. Oxford University Press, New York.

Sun, R. (2009). Theoretical status of computational cognitive modeling. *Cognitive Systems Research*, *10*(2), 124-140.

Sun, R., Wilson, N., & Lynch, M. (2016). Emotion: A unified mechanistic interpretation from a cognitive architecture. *Cognitive Computation*, *8*(1), 1-14.

Sun, R., Zhang, X., & Mathews, R. (2006). Modeling meta-cognition in a cognitive architecture. *Cognitive Systems Research*, *7*(4), 327-338.

United Nations (2019). *Sustainable Development Goals.* New York: The United Nations. https://www.un.org/sustainabledevelopment/sustainable-development-goals/. Visited, May 24 2019.