DYNA
ISSN: 0012-7353
Universidad Nacional de Colombia

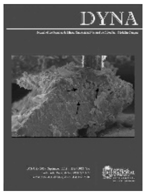
Madera-Doval, Dalia P.; Caro-Piñeres, Manuel F.; Gómez-Salgado, Adán A.; Cardozo-Soto, Ana M.; Jiménez-Builes, Jovani A.

Design of metacognitive expectations of cognitive functions through ontological representations

DYNA, vol. 85, no. 206, 2018, July-September, pp. 194-201
Universidad Nacional de Colombia

DOI: https://doi.org/10.15446/dyna.v85n206.71687

UNIVERSIDAD
NACIONAL
DE COLOMBIA
SEDE MEDELLÍN
FACULTAD DE MINAS



DYNA
http://dyna.medellin.unal.edu.co/

Design of metacognitive expectations of cognitive functions through ontological representations

Dalia P. Madera-Doval [a], Manuel F. Caro-Piñeres [a], Adán A. Gómez-Salgado [a], Ana M. Cardozo-Soto [a]
& Jovani A. Jiménez-Builes [b]

[a] *Departamento de Informática Educativa, Universidad de Córdoba, Montería, Colombia. dmaderadoval@correo.unicordoba.edu.co, manuelcaro@correo.unicordoba.edu.co, aagomez@correo.unicordoba.edu.co, acardozosoto30@correo.unicordoba.edu.co*
[b] *Facultad de Minas, Universidad Nacional de Colombia, Medellín, Colombia. jajimen1@unal.edu.co*

Received: April 13th, 2018. Received in revised form: August 28th, 2018. Accepted: September 7th, 2018.

Abstract
This paper presents a model of metacognitive expectations about the running time of cognitive functions in the metacognitive architecture CARINA. A formal and ontological representation is created that establishes the relationship between the process of observing a fact in the self-model and a belief stored in the semantic memory of the system. The cognitive ontology evidenced tracing and interchange information process among different kind of memories, such as: sensorial memory, semantic memory, procedural memory, prospective memory and working memory. The experiment carried out demonstrated the functionality of the model where expectations were generated for each observation and could be compared with the observed values in real time. Another type of result was the conceptual advance of an expectation, the formal mathematical representation, the design of the ontology and the model as a mechanism of implementation in CARINA architecture.

Keywords: artificial intelligence; expectations; computational metacognition; cognitive architecture CARINA; cognitive agent.

Diseño de expectativas metacognitivas de funciones cognitivas a través de representaciones ontológicas

Resumen
Este artículo presenta un modelo de expectativas metacognitivas sobre el tiempo de funcionamiento de las funciones cognitivas en la arquitectura metacognitiva CARINA. Se crea una representación formal y ontológica que establece la relación entre el proceso de observar un hecho en el auto-modelo y una creencia almacenada en la memoria semántica del sistema. La ontología cognitiva evidenció el proceso de búsqueda e intercambio de información entre diferentes tipos de recuerdos, tales como: memoria sensorial, memoria semántica, memoria de procedimientos, memoria prospectiva y memoria de trabajo. El experimento realizado demostró la funcionalidad del modelo en donde se generó expectativas para cada observación y podía compararlas con los valores observados en tiempo real. Otro tipo de resultado fue avance conceptual de una expectativa, la representación matemática formal, el diseño de la ontología y el modelo como un mecanismo de implementación en la arquitectura CARINA.

Palabras clave: inteligencia artificial; expectativas; metacognición computacional; arquitectura cognitiva; agente cognitivo.

1. Introduction

Humans and other agents are able not only to coordinate their actions with their current sensorimotor state, but also plan and act in view of the future. They can propose to achieve distal goals using anticipatory mechanisms named expectations [1].

A crucial feature of an Intelligent Agent is being proactive and not just reactive, i.e., they have a capacity to use anticipatory mechanisms [2]. This ability allows them to deal with the future by mental representations or specific forms of learning [3]. The use of this anticipation determines the purposive character of the behavior of the Intelligent Agent for its own internal explicit goal representation and

How to cite: Madera-Doval, D.P., Caro-Piñeres, M.F., Gómez-Salgado, A.A., Cardozo-Soto, A.M. and Jiménez-Builes, J.A., Design of metacognitive expectations of cognitive functions through ontological representations. DYNA, 85(206), pp. 194-201, September, 2018.

Revista DYNA, 85(206), pp. 194-201, September, 2018, ISSN 0012-7353
DOI: https://doi.org/10.15446/dyna.v85n206.71687

allows self-modification of its own mental representation for virtually ‘exploring’ possible actions, events, and results: “what will/would happen if…?" [4]. The metacognitive ability may be to explain errors in a cognitive task or it may be to select between cognitive “algorithms” to perform reasoning.

An Intelligent System with metacognitive abilities makes explicit its components, capabilities, actions, percepts, and internal state information in a structure called a self-model. It uses this information to declare expectations about how its activities will affect perceptual and state information as well as monitoring the operation of its actions and its sensory feedback against its expectations, waiting for violations to occur [5]. The execution time, or running time, is the time of continued system operation required in a given environment in order to execute a program or algorithm [6].

Several authors have addressed the generation of expectations regarding the execution time of programs. Duda [6] defines the elapsed time *t*, as the time required to execute a program correctly in a given computing system, where the expectation of the elapsed time will be derived as functions of the execution time *t*. The expectation depends on: i) frequency and duration of failures, (ii) workload of the system and (iii) minimum execution time *t*. Sun and Peterson [7] propose a formal model where the estimation of the system execution time is calculated from the expectation of the maximum execution time of all processors in the computing system.

In the described context, our motivation is to provide a model of metacognitive expectations that represents a hypothesis or projection of current knowledge about running time of a cognitive function that is executed at the object-level. Expectations enable the reasoner to be prepared for the future, the reasoner can thus generate contingency plans to avoid anticipated failures [8]. The identified problem is related to the modeling of metacognitive expectations based on an observation process that uses information obtained from the self-model of a cognitive agent when a cognitive function is executed at its object-level.

This paper is organized as follows: Section 2 "Materials and Methods" describes the formal representation of metacognitive expectation based on observation and presents a general panorama about the structure of the CARINA cognitive architecture. Section 3 "Model of expectations in CARINA" presents the process of triggering a metacognitive expectation in the CARINA architecture. Section 4 "Computational implementation of metacognitive expectations" describes the internal representation of a metacognitive expectation; and some components of the memory system that use the ontology. Section 5 “Validation, result and discussion” presents empirical results within a metacognitive architecture demonstrating the ability to generate metacognitive expectations; and the final section presents the conclusions of this research.

2. Materials and methods

2.1. Formal definition of expectations

In this section we describe the formal representation of metacognitive expectation using elements of denotational mathematics proposed by Wang [9].

An expectation represents a hypothesis or projection of current knowledge [10]. A metacognitive expectation with respect to the execution time of a cognitive function that is executed at the object-level has an observation of a metacognitive sensor (o), an attribute of the observation that represents the execution time (ϱ), the expected value (ν) and the set of rules that generates the expectation (P). Formally a metacognitive expectation (η) about observation (o) is a 4-tuple, i.e.:

$$\eta \triangleq < o, \varrho, \nu, P \ > \tag{1}$$

Where:

η is an expectation about an observation o.

o represents an observation from a sensor ς, with $o \triangleq < \psi, \tau, \upsilon >$, where:

ψ is the instance of a profile of a cognitive function that is running in the object level.

τ is the current execution time of cognitive function.

υ is the unit in which the execution time is measured in milliseconds.

ϱ is an attribute of the observation o, with $o \prec \psi \triangleq A_o \supset A_\psi$. A_o represents the set of attributes of the observation o from the sensor ς. A_ψ represents the set of attributes of profile ψ of a cognitive function χ.

ν is expected value for the field ρ; the field is retrieved from semantic memory. With $o = \beta \triangleq A_o = A_\beta$, where β is a belief in the semantic memory and ϱ is a common field for o and β.

P is a set of rules that retrieve information from different types of memory in order to generate an expectation.

2.2. CARINA cognitive architecture

CARINA is a meta-cognitive architecture for artificial intelligent agents. CARINA is derived from the MISM Metacognitive Metamodel [11]. CARINA integrates self-regulation and meta-memory with support for the meta-cognitive mechanisms of introspective monitoring and meta-level control. In this sense CARINA assumes a functional approach to philosophy of mind, according to Piccinini, Fodor, and Scheutz [12-14]. According to Machamer, Darden, and Craver [15], the term ‘‘mechanism” includes entities and activities involving entities (i.e., including both static and dynamic aspects). The entities in CARINA are called “cognitive elements”.

In accordance with MISM, CARINA is composed of three types of cognitive elements: structural elements, functional elements, and basic elements. Structural elements are containers into which the functional and basic elements are embedded. The main structural element is the “Cognitive level”. The functional elements are tasks that enable reasoning and decision-making. Basic elements consist of the set of elements that participate and interact in reasoning and meta-reasoning processes. The main functional elements in CARINA are reasoning task and meta-reasoning task. Reasoning tasks (RT) are actions that enable the processing (transformation, reduction, elaboration, storage and retrieval) of information by applying knowledge and decision making in order to meet the objectives of the system. A meta-reasoning task (MT) may be used to explain errors in some

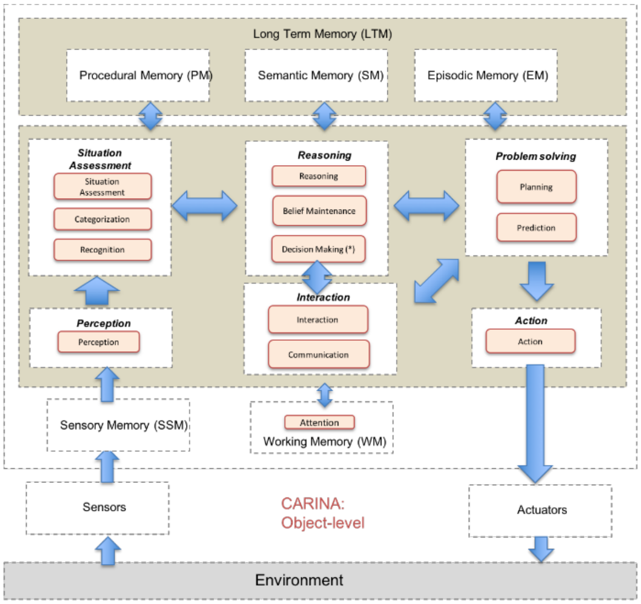


Figure 1. CARINA: Cognitive architecture for artificial intelligent agents.
Source: Image Generated by Authors

reasoning task or it may be used to select between "cognitive algorithms" to perform the reasoning [16].

CARINA ≜ (OL, ML)
Where OL ≜ (PS, SAS, RS, IS, PSS, AS, MoW)
With PS ≜ (perception)
SAS ≜ (situation assessment, categorization, recognition)
RS ≜ (reasoning, belief maintenance, decision making)
IS ≜ (interaction, communication)
PSS ≜(planning, prediction)
AS≜ (action)
MoW

In CARINA, object-level has stages which are sets of cognitive functions (CF): Perception Stage (Perception CF), Situation Assessment Stage (Situation Assessment CF, Categorization CF, and Recognition CF), Reasoning Stage (Reasoning CF, Belief Maintenance CF, and Decision Making CF), Problem Solving Stage (Planning CF and Prediction CF), Interaction Stage (Interaction CF and Communication CF) and Action Stage (Action CF) as shown in Fig. 1. The meta-level contains a dynamic model of the object-level.

The meta-level includes the components, knowledge and mechanisms necessary for a system to monitor and control its own learning and reasoning processes. The meta-level of CARINA has two types of metacognition:

ML ≜ (SR, M)
SR ≜ (M, C)
M≜(Identification,Detection, Explanation, Goal Generation)
C ≜ (Strategy Selection)
MM≜ (MS, MKSS, JS, CS)
MS ≜ (Identification, Detection)
MKSS ≜ (Goal Generation, Explanation)
JS ≜ (Judgment)
CS ≜ (Strategy Selection)

i. Self-regulation: this type of metacognition monitors and controls the reasoning process that takes place at the object-level. Self-regulation contains two stages, which are sets of meta-cognitive functions (MCF): Monitoring Stage (Identification MCF, Detection MCF, Explanation MCF, and Goal Generation MCF) and Control Stage (Strategy Selection MCF).

ii. Meta-memory: is a component of meta-cognition [17]. The meta-memory in CARINA is a mechanism that allows adaptation to change in the constraints of a memory task [18].This meta-memory mechanism contains four stages which are sets of metacognitive functions (MCF): Monitoring Stage (Identification MCF and Detection MCF), Meta-Knowledge Search Stage (Goal Generation MCF and Explanation MCF), Judgment Stage (Judgment MCF) and Control Stage (Strategy Selection MCF).

3. Model of expectations in CARINA

Introspective monitoring is a meta-reasoning 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 cognitive mechanisms and processes operating on them [19]. Introspective monitoring includes mechanisms for detecting reasoning failures at the object-level. The main purpose of monitoring is to provide enough information to make effective decisions in the meta-level control [20]. The monitoring process is done through information feedback that is gathered at the meta-level from the object level, see Fig. 2. Thus, each cognitive task executed in the object level has a performance profile that is continuously updated in the meta-level. The performance profile is used to evaluate the results of each reasoning task.

CARINA is conformed by two cognitive levels called object level and meta-level i.e.

CARINA ≜ (OL, ML)

The object level (OL) contains the working memory where the self-model is located.

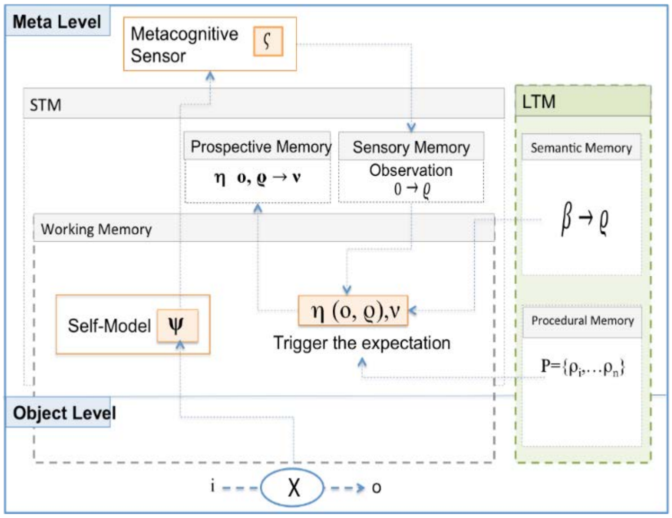


Figure 2. Process of triggering metacognitive expectations in CARINA.
Source: Image Generated by Authors

According to Caro, Gomez, and Giraldo [20], when the cognitive function (χ) is executed at the object level with an input (IP), output (OP) and start time (B), a profile is automatically generated for the cognitive function in the self-model with the fields.

$\psi \triangleq (ID, B, E, S, C, IP, OP)$, 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∈S and S={active, inactive}

C is the priority level for focus attention c∈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.

The self-model contains the profile (ψ) of the cognitive function (χ). The meta-level includes the necessary mechanisms for a system to monitor its own learning and reasoning processes. In the meta-level of CARINA there is a metacognitive sensor (ς). A metacognitive sensor is an element of the meta-level. The function of metacognitive sensor is to monitor the profile of the cognitive function; this sensor has a sensory memory that stores the observation (o), taking into account that:

$$o \prec \psi \triangleq A_O \supset A\_\psi \qquad (2)$$

In formula (2), we can appreciate the fields of profile of cognitive function expressed as: $\mathrm{A}_\psi \triangleq (\mathrm{ID}, \mathrm{B})$ and the fields that are observed $\mathrm{A}_\mathrm{O} \triangleq (\mathrm{ID}, \mathrm{B}, \mathrm{E}, \mathrm{S}, \mathrm{C}, \mathrm{IP}, \mathrm{OP})$.

Whereas in semantic memory a field (ρ) is created, which consists of the average time of execution of all cognitive functions.

$$\varrho_e = \frac{\sum_{i=1}^{n} \beta_i \cdot \varrho_\nu}{n} \; With \;\; o = \beta \triangleq A_o = A_\beta \qquad (3)$$

The formula (3) is based in [10] who affirms that expectations may emanate from a mental process such as an arithmetic check of a mathematics computation [10], For this, formula (3) represents the average of the execution time where (β) represents a belief retrieved from the semantic memory, (o) is a fact of the observation that is generated in the sensorial memory taking into account $\mathrm{A}_\mathrm{o} = \mathrm{A}_\beta$. In this case, they are equivalent and the belief (β) is extracted from the semantic memory and all the values of A_β are compared with the values of A_o. In the working memory the expectation is generated with the observation field (ρ) The expected value of the semantic memory is generated with the average time and the unit of time, which causes an expectation. This is based in [21] which establishes that specific mental process that forms an expectation may be a memory process that retrieves an expectation from memory.

To verify the expectation (η) refer a set of rules (P), which are in the procedural memory. There are two types of rules to trigger and one [10] to validate; the rule we use is the process of triggering the expectation.

Prospective memory is a particular type of short-term memory, understood as the anticipatory capacity for the future, where it is expected to develop at a given time and the expected value (ν). If the field of observation (o) does not match, then a violation of expectation will be determined..

4. Computational implementation of metacognitive expectations

In this section we describe the internal representation of a metacognitive expectation and some components of memory system using ontology. Ontology is an explicit specification of knowledge level of a conceptualization, which can be affected by a particular domain or task, for which it has been created [22].

According to Gómes, Benjamins [23], Stojanovic, Maedche, Motik and Stojanovic [24]. Ontologies are comprised of several components:

- A set of concepts (also known as classes) and a hierarchy among concepts.
- A set of relationships between concepts.
- A set of properties, which are a special case of relationship.
- A set of axioms, rules that are always valid.

Table 1.
Piece of code: Structure of the cognitive ontology.

1	`ObservationField ⊑Field`
2	`ObservationCFIP ⊑ObservationField`
3	`ObservationCFP ⊑ObservationField`
4	`ObservationTime ⊑ObservationField`
5	`ObservationUnitOfTime ⊑ ObservationField`
6	`ObservationField ⊑Field`
7	`Field ⊑Thing`
8	`MetacognitiveSensor`
9	`MetacognitiveSensor ⊑Thing`
10	`CognitiveFunction ⊑Belief`
11	`CognitiveFunctionInstanceProfile ⊑ Belief`
12	`CognitiveFunctionProfile ⊑Belief`
13	`ComputationalStrategy ⊑Belief`
14	`Number ⊑Belief`
15	`UnitOfTime ⊑Belief`
16	`CognitiveFunction ⊑Belief`
17	`Belief ⊑SemanticMemoryUnit`
18	`SemanticMeroryUnit ⊑Thing`
19	`Observation ⊑Fact`
20	`Fact ⊑Thing`

Source: Author's own elaboration

4.1. Metacognitive representation in ontology

Our ontology generates a metacognitive expectation of time with respect to the execution of a cognitive function in the metacognitive architecture. The implementation of the ontology was done using Protégé, version 5.2.0. This tool allows good level of details and its interface is user-friendly. WOL-DL (Web Ontology Language – Description Logic) was the formal language used for representation, World Wide Web Consortium (W3C) recommends WOL-DL. In this context the metacognitive problem is noted as ρ. ρ represents how to generate a metacognitive expectation with respect to the execution time of a cognitive function that is executed at the object level. Metacognitive expectation is represented as a hierarchical structure in ontology. First, the taxonomy of terms was created with the main concepts (classes): SemanticMemoryUnit. Creating subclasses Beliefs and Facts from derived concepts refined this hierarchy. Next, the fields of the Observation were defined. Malviya, Mishra, and Sahu define Object Properties as relationships between two instances of classes, whereas data properties describe relationships between instances and data values [25].

According to the structure of the ontology the classes depend on a super class Thing, the classes contain individuals and the object properties. The structure of the ontology is shown in Table 2 which contains an example of object and data properties of the cognitive ontology.

In the object properties, the domain and the range are created.

Table 2.
Piece of code: Object and Data Properties

1	hasCFIP(CogitiveFunctionInstanceProfile), (CognitiveFunction)
2	hasCFIP(CogitiveFunctionInstanceProfile), (CognitiveFunctionProfile)
3	hasCFIP(CogitiveFunctionInstanceProfile), (ComputacionalStrategy)
4	hasCFIPRunningTime(CogitiveFunctionInstanceProfile),(Number)
5	hasCFIPUnitOfTime(CogitiveFunctionInstanceProfile), (UnitOfTime)
6	hasCFP(CogitiveFunctionProfile),(CogitiveFunctionInstanceProfile)
7	hasObservation(Observation),(ObservationCFIP)
8	hasObservation(Observation),(ObservationCFP)
9	hasObservation(Observation),(ObservationTime)
10	hasObservationUnitOfTime(Observation),(ObservationUnitOfTime)
11	triggerRunningTimeExpectation(Observation),(ObservationTime)

Source: Author's own elaboration

Table 3.
Piece of code: ID Data Properties

1	DataProperty(hasIDConcept)

Source: Author's own elaboration

Table 4.
Piece of code: Range and Domain of the data properties

1	hasIDConcept(SemanticMemoryUnit),(xsd:string)

Source: Author's own elaboration

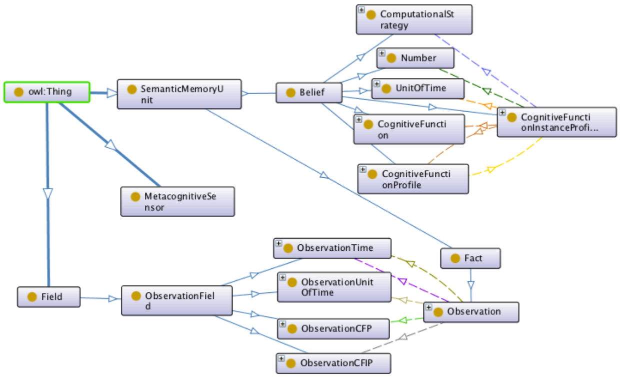


Figure 3. Network of concepts
Source: Image Generated by Authors

The data property has `IDConcept` which defines the identifier of the semantic memory unit. In the same way, the properties pertaining to each concept were represented as is shown in Table 4.

Ontology Instantiation is implemented as a simulation of the process of triggering expectations in CARINA. The next step is the use of the ontology as a simulation of triggering expectations at meta-level. This work explores the role of different types of memories in the triggering of metacognitive expectations in CARINA. We use production rules for representing the procedural knowledge needed to generate a metacognitive expectation. Table 5 shows a rule named R1. R1 gets the current observation from the sensory memory and generates an expectation for the field running time.

The sensorial memory contains the observation flow from the metacognitive sensor. In the ontology an observation has a relation (`hasObservationCFP`) that points to the profile of a cognitive function, and based on the observation, it looks for the beliefs (β) in the semantic memory to verify and extract the field of cognitive function profile (OCFP). Since the fact field of the observation must be equal to the beliefs (β), then the fields in the ontology should not be equal. One way to identify it is to make a datum of the `hasIDConcept` property of script type when the value is the same.

As stated by Gordon, Hobbs and Cox the reasoner calculates some expected outcome and compares it with the actual outcome that constitutes the feedback [26]. The observation field (OCFP) and the field (ϱ) in the belief (BCFP) are extracted and the two script data are matched to obtain the execution time.

Table 5.
Piece of code: Rule R1. R1 gets the current observation from the sensory memory and generate an expectation.

```
<Rule1001>
Observation(?O)∧
hasObservationCFP(?O, ?OCFP) ∧
Belief(?B) ∧
hasCFIPCognitiveFunctionProfile(?B,?BCFP) ∧
hasIDConcept(?OCFP, ?IDO) ∧
hasIDConcept(?BCFP, ?IDB) ∧
equal(?IDO,  IDB) ∧
hasCFIPRunningTime(?B, ?RT)
→triggerRunningTimeExpectation(?O,?RT)
```

Source: Author's own elaboration

`hasCFIPRunningTime` is the average of the sum of i=1 to n of the field (ϱ) in equation (3), the field (ϱ) of the belief about in formula (3), the field (ϱ) of the belief is automatically extracted in a new relationship called `triggerRunningTimeExpectation.`

4.1. *Validation, result and discussion*

An experiment was conducted to validate the performance of the model and its ability to generate expectations regarding the execution time of cognitive functions that are executed in the object-level of a cognitive agent based on CARINA. The syntactic analysis of a sentence was the cognitive task of the real world used in the experiment. In this experiment a cognitive agent identified the different grammatical categories in which the words of the language are grouped. These categories include the word's meaning, its grammatical function and the way in which it is structured, combined and modified.

The Left-corner is the algorithm implemented in the cognitive agent to decide if the chain of input symbols is a sentence in a given language. The algorithm determines the syntactic structure of the sentence according to a grammar [27].

We collect data from 200 executions of the cognitive task in CARINA, varying the available resources such as the size of the content of the semantic memory with respect to the domain of the cognitive task. Figure 4 shows the results of CARINA in terms of the expectations created with respect to the observations of the profile of cognitive functions that were executed in each cognitive cycle at object-level. The x-axis represents the 12 cognitive functions that were executed in CARINA to perform the cognitive task. The y-axis represents the execution time in milliseconds.

The blue bars represent the run-time expectation of 12 cognitive functions that were executed in CARINA to perform the cognitive task. CARINA calculates the run-time values using equation (3). The blue lines represent the observed runtime values at a time t_i. This is a clear example of the functionality of the model, because it generates expectations for each observation and is able to compare them against the observed values in real time.

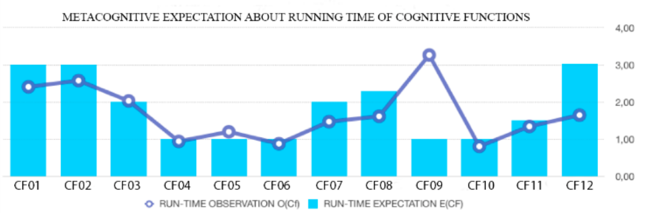


Figure 4. Metacognitive Expectations about Running Time of Cognitive Functions
Source: Image Generated by Authors

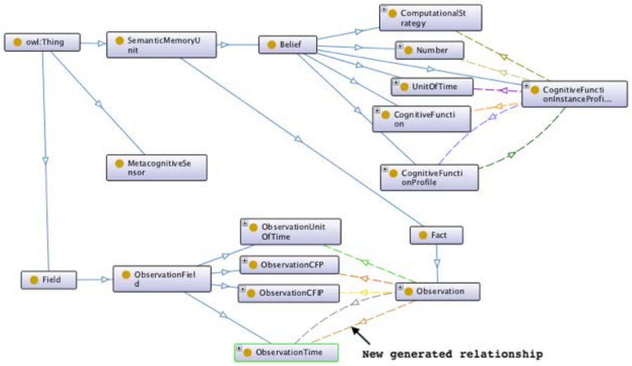


Figure 5. New generated relationship
Source: Image Generated by Authors

The scope of this investigation only applies to the generation of execution time expectations. In this sense, the model tests demonstrate the ability to generate expectations and the possibility of detecting anomalies in the expected execution times for cognitive functions. An interesting aspect for future investigations is to analyze the reasoning trace of the function that presents the anomaly, with the goal to determine the cause and possible corrective actions.

Another type of result obtained in this research was in terms of the conceptual advance of an expectation, the formal mathematical representation, the design of ontology and the model as an implementation mechanism. The cognitive ontology for reasoning about expectations in CARINA architecture facilitates the visualization of the expectations generated from the observations of the profiles of cognitive functions, as shown in Figure 5.

In Figure 5, the new relationship in ontology represents an expectation generated by the CARINA meta-level in the process of introspective monitoring. This result is promising because it will serve as a basis for the implementation of predictive capabilities of introspective monitoring in Carina's meta-level.

5. Conclusions

In this paper we have proposed an ontology based on an approach to automatically generate Metacognitive expectation about observations of the self-model. The ontology is based on [28], In this way, formal model represents the average of the execution time of a cognitive function executed at the object-level, using elements of the denotational mathematics proposed by Wang [9]. In this formal model, the beliefs (β) (stored in semantic memory), and the observation process of facts (o) (stored in sensorial memory), are compared, expecting that $A_o = A_\beta$. In the working memory, the expectation is generated with the observation field (ρ), the expected value of the semantic memory (average running time of cognitive function) and the unit of time of execution.

Cognitive Ontology allowed us to develop a reasoning test according to a specific cognitive task in the CARINA Metacognitive Architecture. In this test, we applied rules that represent the necessary procedural knowledge to trigger the metacognitive expectations.

Also, the cognitive ontology resulted in the tracing and interchange of information process among different kind of memories, such as: sensorial memory, semantic memory, procedural memory, prospective memory and working memory. The above reaffirms what is proposed by Cox “a specific mental process that forms an expectation may be a memory process that retrieves an expectation from memory” [29]. This work provides advances in the implementation of metacognitive mechanisms such as introspective monitoring in CARINA architecture.

References

[1] Pezzulo, G., Coordinating with the future: the anticipatory nature of representation. Minds and Machines, 18(2), pp. 179-225. 2008. DOI: 10.1007/s11023-008-9095-5

[2] Castelfranchi, C. and Lorini, E., Cognitive anatomy and functions of expectations 1. Artificial Intelligence, (October). 2001.

[3] Castelfranchi, C. and Lorini, E., Cognitive anatomy and functions of expectations. In: IJCAI03 Workshop on Cognitive Modeling of Agents and Multi-Agent Interactions, 2003, pp. 9-11. DOI: 10.1109/IAT.2006.2

[4] Castelfranchi, C., For a systematic theory of expectations. BVAI, 3704, 2007, pp. 258-276. DOI: 10.1007/s11229-006-9156

[5] Haidarian, H., Dinalankara, W., Fults, S., Wilson, S., Perlis, D., Schmill, M.D. and Anderson, M.L., The metacognitive loop : an architecture for building robust intelligent systems. AAAI Fall Symposium, (II), 2010, pp. 33-39.

[6] Sevcik, C., The Effects Andrzej Duda. Information Processing Letters, 16, pp. 221-229,1983.

[7] Sun, J. and Peterson, G.D., An effective execution time approximation method for parallel computing. IEEE Transactions on Parallel and Distributed Systems (TPDS), 23(11), pp. 2024-2032, 2012. DOI: 10.1109/TPDS.2012.21

[8] Fox, S. and Leake, D., Introspective reasoning for index refinement in case-based reasoning. Journal of Experimental & Theoretical Artificial Intelligence, 13(1), pp. 63-88, 2001. DOI: 10.1080/09528130010029794

[9] Wang, Y., On contemporary denotational mathematics for computational intelligence. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5150 LNCS, 2008, pp. 6-29. DOI: 10.1007/978-3-540-87563-5_2

[10] Cox, M.T. and Ashwin, R., Introspective multistrategy learning: on the construction of learning strategies. Artificial Intelligence, 112(1), pp. 1-55. 1999. DOI: 10.1016/S0004-3702(99)00047-8

[11] Caro, M., Josyula, D., Cox, M. and Jiménez, J., Design and validation of a metamodel for metacognition support in artificial intelligent systems. Biologically Inspired Cognitive Architectures, 9, pp. 82-104, 2014. DOI: 10.1016/j.bica.2014.07.002

[12] Piccinini, G., The mind as neural software? Understanding functionalism, computationalism, and computational functionalism. Philosophy and Phenomenological Research, 81(2), pp. 269-311, 2010. DOI: 10.1111/j.1933-1592.2010.00356.x

[13] Fodor, J.A., The language of thought. New York (Vol. 4). 1975. DOI: 10.1016/0093-934X(77)90028-1

[14] Scheutz, M., Computational versus causal complexity. Minds and Machines, 11(4), pp. 543-566, 2001. DOI: 10.1023/A:1011855915651

[15] Machamer, P., Darden, L. and Craver, C.F., Thinking about mechanisms. Philosophy of Science, 67(1), pp. 1-25, 2000. DOI: 10.1086/392759

[16] Cox, M. and Raja, A., Metareasoning: an introduction. In: Cox and Raja (Ed.), In Metareasoning: Thinking about thinking, 2012, pp. 1-23, Cambridge. MA: MIT. DOI: 10.7551/mitpress/9780262014809.003.0001

[17] Cox, M., Oates, T. and Perlis, D., Toward an integrated metacognitive infrastructure. 2011 AAAI Fall Symposium, 2011, pp. 74-81. DOI: 10.1016/j.procs.2014.11.107

[18] Caro, M., Diaz, D. and Josyula, D., MetaThink: a MOF-based metacognitive modeling tool. Ieeexplore.Ieee.Org, pp. 351-358, 2013. DOI: 10.1109/ICCI-CC.2016.7862059

[19] Sun, R., Theoretical status of computational cognitive modeling. Cognitive Systems Research, 10(2), pp. 124-140, 2009. DOI: 10.1016/j.cogsys.2008.07.002

[20] Caro, M., Gomez, A. and Giraldo, J., Algorithmic knowledge profiles for introspective monitoring in artificial cognitive agents. In: 2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), 2017, pp. 475-481. DOI: 10.1109/ICCI-CC.2017.8109792

[21] Cox, M., Introspective multistrategy learning: constructing a learning strategy under reasoning failure. Doctoral dissertation - Cox - 1996.

[22] Gruber, T.R., A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2), pp. 199-220, 1993. DOI: 10.1006/knac.1993.1008

[23] Gómez, A. and Benjamins, R., Overview of knowledge sharing and reuse components: ontologies and problem-solving methods. In: IJCAI-99 Workshop on Ontologies and Problem-Solving Method (KRR5), 1999, pp. 1-15. DOI: 10.1.1.39.249.

[24] Stojanovic, L., Maedche, A., Motik, B. and Stojanovic, N., User-driven ontology evolution management. In: Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, 2002, pp. 133-140. DOI:10.1145/637411.637413

[25] Malviya, N., Mishra, N. and Sahu, S., Developing university ontology using protégé OWL tool: process and reasoning. International Journal of Scientific & Engineering Research, 2(9), pp. 1-8, 2011.

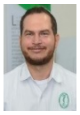
[26] Gordon, A., Hobbs, J. and Cox, M., Anthropomorphic self-models for metareasoning agents. Metareasoning: Thinking about Thinking, 129-135. 2011. DOI: 10.7551/mitpress/9780262014809.003.0019

[27] La Serna, N., Un analizador sintático eficiente para gramáticas del español, Rev. investig. sist. inform., 1(1), pp. 19-26, 2004.

[28] Zhang, Z.J., Shi, T.Y. and Yu, Z.H., A method for building geographical conditions ontology attribute library: taking Tianjin as an example. Applied Mechanics and Materials, 156, pp. 411-414, 2013. DOI: 10.4028/www.scientific.net/AMM.411-414.156

[29] Cox, M., Introspective multistrategy learning: constructing a learning strategy under reasoning failure, Thesis Dr. of Philosophy in Computer, Science Georgia Institute of Technology Introspective Mu. 1996b.

D.P. Madera-Doval, received a BSc. in Computer Science and Audiovisual Media (2017) from the Universidad de Córdoba, Colombia. She is currently a junior researcher in the Computer Research and Cognitive Computing Group at the Faculty of Education of the Universidad de Córdoba, Colombia. His research interests include: cognitive informatics, cognitive computing and artificial intelligence in education.
ORCID: 0000-0003-3451-9110



M.F. Caro-Piñeres, received a PhD. in Engineering - Systems and Information (2015) at the Universidad Nacional de Colombia - Medellín. Currently, he is an associate professor in the Department of Educational Computing and coordinator of the Computer Science and Cognitive Computing Research Group at the Faculty of Education of the Universidad de Córdoba, Colombia. His research interests include: cognitive informatics, cognitive computing and artificial intelligence in education.
ORCID: 0000-0001-6594-2187

A.A. Gómez-Salgado, received MSc. in Technology of Information Applied to Education (2012) in the Universidad Pedagógica Nacional, Esp. in Informatic and Multimedia (2006) in the Fundación Universitaria los Libertadores, Lic. in Educational Informatic and Audiovisual Media (2001). Currently, he is associate professor in the Department of Educational Computing at the Faculty of Education of the Universidad de Córdoba, Colombia. His research interests include: artificial intelligence - educational software development, knowledge representation, cognitive informatic, cognitive computing.
ORCID: 0000-0002-7676-8492

A.M. Cardozo-Soto, received a BSc. in Computer Science and Audiovisual Media (2017) from the Universidad de Córdoba, Colombia. His research interests include: cognitive informatics, cognitive computing and artificial intelligence in education.
ORCID: 0000-0002-0915-6015

J.A. Jiménez-Builes, is a full time professor, at the Facultad de Minas, Universidad Nacional de Colombia, Medellín, Colombia. He holds a BSc. degreee in Computer Teaching Universidad de Medellin, Colombia and received an MSc. in System Engineering, and a PhD. Engineering from Universidad Nacional de Colombia, Medellín, Colombia. His research areas include: artificial intelligence in education, knowledge management, E-learning and T-learning.
ORCID: 0000-0001-7598-7696