DEVELOPMENT OF COGNITIVE MODELS FOR THE METACOGNITIVE ARCHITECTURE CARINA

ALBA JUDITH JERÓNIMO MONTIEL

DIRECTOR:

MSc. ADÁN ALBERTO GÓMEZ SALGADO

UNIVERSIDAD DE CÓRDOBA

FACULTAD DE EDUCACIÓN Y CIENCIAS HUMANAS

DEPARTAMENTO DE INFORMÁTICA EDUCATIVA

LICENCIATURA EN INFORMÁTICA Y MEDIOS AUDIOVISUALES

MONTERÍA

2020

DEVELOPMENT OF COGNITIVE MODELS FOR THE METACOGNITIVE ARCHITECTURE CARINA.

ALBA JUDITH JERÓNIMO MONTIEL

Research Project in fulfilment of the requirements for the undergraduate degree of Licenciada en Informática y Medios Audiovisuales

DIRECTOR:

MSc. ADÁN ALBERTO GÓMEZ SALGADO

UNIVERSIDAD DE CÓRDOBA

FACULTAD DE EDUCACIÓN Y CIENCIAS HUMANAS

DEPARTAMENTO DE INFORMÁTICA EDUCATIVA

LICENCIATURA EN INFORMÁTICA Y MEDIOS AUDIOVISUALES

MONTERÍA

2020

Dedicated to the memory of my mother, and the promise of my two sisters... Paty y Tato.

ACKNOWLEDGMENTS

Thank you, God, for giving me a family that always gave me support and encouragement to never give up and continue. To my grandfather Cesar, who has trusted me since I started college, and helped me with his relentless character to achieve my dreams. To my grandmother Leonor for her infinite love and to my two sisters Paty and Tato for their jokes and tenderness when moments of discouragement appeared. I want to thank them for all their understanding, support and love.

The research group: Cognitive Informatics and Cognitive Computing, now called EduTlan, provided me intellectual background for this work when we get together in the team workplace and the workshops developed in the undergraduate program. Professor Manuel Caro, for his extensive experience and knowledge in this research issue.

I would like to thank Heider Zapa for his immensurable support in the development of the functional prototype and for the time we spent in the team workplace and also by remote working discussing questions on this aspect development.

Most importantly, I would like to thank my advisor, Professor Adan Gomez, for believing in me since the first semester when I started the university by joining me to his research team and for never leaving me alone in all this process. He showed me how to research, to be more productive and to question myself in the right and difficult ways to solve problems. Infinite thanks for showing me that science is accessible to everyone, and that women in computer science play an important role as transformers of society through knowledge networking. God bless you today and ever.

ABSTRACT

Cognitive modeling is a methodology of cognitive sciences that allows the simulation of human cognitive processes in a variety forms, commonly in a computational and mathematical way. The cognitive modeling aims at understanding cognition basis by designing cognitive models based on mathematical or computational processes, mechanisms and representations. A cognitive model is a verbal-conceptual computational and mathematical description of some mental processes, whose main purpose is to understand and/or predict human or animal behavior. Cognitive models developed for a cognitive architecture are characterized by being executables and producing a set of specific behaviors. CARINA is a metacognitive architecture to create artificial intelligent agents derived from *Metacognitive Metamodel MISM*. CARINA is a metacognitive architecture structured by two cognitive levels called *object-level* and *meta-level*. The *object-level* has the model of the world to solve problems. The *meta-level* represents the reasoning of an artificial intelligent agent.

Furthermore, the *meta-level* has the components, the knowledge and the mechanisms for an intelligent system to monitor and control its own learning and reasoning processes. The main objective of this research project is to develop cognitive models as knowledge acquisition mechanisms for the metacognitive architecture CARINA, through the following specific objectives: i) *to represent formal, semantic and computationally cognitive models for the CARINA metacognitive architecture*, ii) *to build a functional prototype of a framework for the creation of cognitive models in the metacognitive architecture CARINA* and iii) *to create cognitive models in several knowledge domains using CARINA based intelligent systems*. The methodology used for this research project was part of the research methods (R+D) used in computer science, called modeling, structured by five steps: i) *Formal representation*, ii) *Semantic representation*, iii) *Computational representation of a cognitive model*, iv) *Creation of*

a functional prototype for build cognitive models and v) *Prototype testing and maintenance*. The developed research project allows simplifying the developing intelligent agents process and the easiness to enable any programmer to uses CARINA to solve cognitive tasks, focusing only on descriptions of cognition and relationships with algorithms and programs based on computer science and technology, using a functional prototype (MetaThink version 2.0). As a result, an open standard file format, simplifying the complexities of detailed descriptions of cognitive mechanisms of brain functioning was created.

RESUMEN

El modelado cognitivo es una metodología de las ciencias cognitivas que permite la simulación de procesos cognitivos humanos en diversas formas, comúnmente de manera computacional y matemática. El modelado cognitivo pretende comprender los fundamentos de la cognición mediante el diseño de modelos cognitivos basados en procesos matemáticos o computacionales, mecanismos y representaciones. Un modelo cognitivo es una descripción verbal-conceptual, computacional y matemática de algunos procesos mentales, cuyo objetivo principal es comprender y / o predecir el comportamiento humano o animal. Los modelos cognitivos desarrollados para una arquitectura cognitiva se caracterizan por ser ejecutables y producir un conjunto de comportamientos específicos. CARINA es una arquitectura metacognitiva para la creación de agentes inteligentes artificiales, derivados del Metamodelo Metacognitivo MISM. CARINA es una arquitectura metacognitiva estructurada por dos niveles cognitivos llamados *nivel-objeto* y *meta-nivel*. El *nivel-objeto* tiene el modelo del mundo para resolver problemas. El *meta-nivel* representa el razonamiento de un agente inteligente artificial.

Además, el *meta-nivel* tiene los componentes, el conocimiento y los mecanismos para que un sistema inteligente monitoree y controle sus propios procesos de aprendizaje y razonamiento. El objetivo principal de este proyecto de investigación es desarrollar modelos cognitivos como mecanismos de adquisición de conocimiento para la arquitectura metacognitiva CARINA, a través de los siguientes objetivos específicos: i) *representar modelos cognitivos formales, semánticos y computacionalmente para la arquitectura metacognitiva CARINA*, ii) *construir un prototipo funcional de un marco para la creación de modelos cognitivos en la arquitectura metacognitiva CARINA* y iii) *crear modelos cognitivos en varios dominios del conocimiento utilizando sistemas inteligentes basados en CARINA*. La metodología utilizada para el desarrollo del proyecto de investigación fue parte de los métodos de investigación (I + D) utilizados en

informática, llamada modelado, estructurada en cinco pasos: i) *Representación formal*, ii) *Representación semántica*, iii) *Representación computacional de un modelo cognitivo*, iv) *Creación de un prototipo funcional para construir modelos cognitivos* y v) *Prueba y mantenimiento del prototipo*. El proyecto de investigación desarrollado permite simplificar el proceso de desarrollo de agentes inteligentes y permite que cualquier programador use CARINA para resolver tareas cognitivas, centrándose solo en descripciones de cognición y relaciones con algoritmos y programas, basados en ciencia y tecnología informática, utilizando un prototipo funcional (MetaThink versión 2.0), que crea un formato de archivo estándar abierto, simplificando las complejidades de las descripciones detalladas de los mecanismos cognitivos del funcionamiento del cerebro.

LIST OF CONTENTS

1. Chapter I Introduction 16

1.1. Motivation 19

1.2. Thesis Project 20

1.2.1. Research Project 20

1.2.2. Research Problem 20

1.3. Research Question 22

1.4. Objectives 23

1.4.1. General Objective 23

1.4.2. Specific Objectives 23

1.5. Methodology 23

1.6. Document Organization 25

2. Chapter II Theoretical Background 27

3. Chapter III Theoretical Framework 51

3.1. Cognitive Modeling 51

3.2. Cognitive Models 53

3.3. Cognitive Architectures 56

3.4. Metacognitive Architectures 57

3.5. Knowledge Representation 59

3.6. Denotational Mathematics 60

4. Chapter IV The Metacognitive Architecture CARINA 61

5. Chapter V Cognitive Models for the Metacognitive Architecture CARINA 66

5.1. Formal Representation of Cognitive Models in CARINA 66

5.1.1. Comparison with other Cognitive Architectures 73

5.1.2. Similarities 74

5.1.3. Differences 74

5.2. Semantic Representation of Cognitive Models in CARINA 75

5.2.1. Semantic Knowledge Representation of a Cognitive Model in CARINA 76

5.2.2. Formal Specification of Semantic Memory Units (SMU) in CARINA 78

5.3. Computational Representation of Cognitive Models for the CARINA Metacognitive Architecture 80

5.4. MetaThink Version 2.0 83

5.4.1. MetaThink Version 2.0 Validation 88

5.5. Illustrative Examples of Cognitive Models in CARINA 93

6. Chapter VI Conclusions 105

6.1. Recommendations 106

7. Chapter VII References 108

LIST OF TABLES

Table 1 Format to synthesize the cognitive task description when the information source comes from experts. 31

Table 2 Specification of a Problem in CARINA... 69

Table 3 Mental States in CARINA 70

Table 4 Goals in CARINA's motivational system 71

Table 5 Production Rules in CARINA.. 72

Table 6 Basic structure of a Belief in CARINA.. 73

Table 7 Specification of a Problem in ACT-R.. 74

Table 8 Beliefs network related to the Semantic Memory Units (SMU).................................... 78

Table 9 Basic structure of a Belief in CARINA.. 80

Table 10 Format to synthesize the cognitive task description when the information source comes from users.. 94

Table 11 Pre-processing stage.. 95

Table 12 Processing stage: production system and decision tree.. 96

Table 13 Description of the cognitive task in natural language.. 96

Table 14 Cognitive model of the TOH algorithm based on NGOMS - L.................................... 98

Table 15 Inventory of Goals, Mental States and Actions of the TOH algorithm 99

LIST OF FIGURES

Figure 1 Research methodology 25

Figure 2 "Input Processing" to complete the elaboration of factoid-Wh questions..................... 33

Figure 3 Cognitive model for the factoid-Wh questions generation in natural language 33

Figure 4 Cognitive model for the factoid-Wh questions generation in GOMS (NGOMS-L) 37

Figure 5 Inventory of Goals, Mental States and Operators of cognitive model for the factoid-Wh questions generation in GOMS (NGOMS-L) 39

Figure 6 Inventory of Mental States of cognitive model for the factoid-Wh questions generation in GOMS (NGOMS-L) 40

Figure 7 Inventory of Operators of cognitive model for the factoid-Wh questions generation in GOMS (NGOMS-L) 41

Figure 8 Main elements in M++ 42

Figure 9 Mental States, Actions and Goals of cognitive model for the factoid-Wh questions generation in M++ 43

Figure 10 Representation of the NGOMS-L model in M++.. 44

Figure 11 Mental State in cognitive model for factoid-Wh questions....................................... 45

Figure 12 Goals in cognitive model for factoid-Wh questions.. 46

Figure 13 Production Rules in cognitive model for factoid-Wh questions................................. 47

Figure 14 Conclusion of Production Rules in cognitive model for factoid-Wh questions. 48

Figure 15 Login and registration in the Toolkit agent. 49

Figure 16 Interface to create the factoid questions. 49

Figure 17 The Memory System in CARINA 62

Figure 18 Semantic representation of a cognitive model in CARINA 77

Figure 19 Computational representation in JSON format in CARINA 81

Figure 20 MetaThink version 2.0 registration page ... 84

Figure 21 MetaThink version 2.0 login page ... 85

Figure 22 Start menu page MetaThink version 2.0 ... 85

Figure 23 Create a new cognitive model page MetaThink version 2.0 ... 86

Figure 24 Elements page to create a cognitive model MetaThink version 2.0 ... 86

Figure 25 MetaThink version 2.0 cognitive model creation page ... 87

Figure 26 MetaThink version 2.0 saved cognitive model page ... 87

Figure 27 Detail page of the JSON file cognitive model MetaThink version 2.0 ... 88

Figure 28 MetaThink V2.0 functionality category ... 89

Figure 29 MetaThink V2.0 reliability category ... 90

Figure 30 MetaThink V2.0 usability category ... 90

Figure 31 MetaThink V2.0 efficiency category ... 91

Figure 32 MetaThink V2.0 maintainability category ... 92

Figure 33 MetaThink V2.0 portability category ... 92

Figure 34 Specification in M ++ of the cognitive model in NOGOMS-L of the TOH algorithm in CARINA ... 100

Figure 35 Pre-conditional and post-conditional actions in M++ for the TOH algorithm in CARINA ... 101

Figure 36 Representation of the NGOMS-L model in M++ ... 102

Figure 37 Runnable cognitive model in CARINA of TOH algorithm ... 103

Figure 38 Executable cognitive model for the Towers of Hanoi algorithm in the metacognitive architecture CARINA ... 104

APPENDIXES

Appendix A Jerónimo, A. J., Caro, M. F., & Gómez, A. A. (2018). Formal Specification of Cognitive Models in CARINA. In 2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC) (pp. 614-619). IEEE.................................... 118

Appendix B 17th IEEE International Conference on Cognitive Informatics & Cognitive Computing in Berkeley, California. Conference paper certificate. 119

Appendix C II Workshop 2018: Metacognition Seminar in Monteria – Colombia. Certificate of participation 120

Appendix D Chapter proposal approved... 121

Appendix E Update on Encyclopedia of Information Science and Technology, 5th ed. (New tentative release date)... 122

Appendix F Completed and requested evaluations as well as reviewer statistics in IGI GLOBAL 123

Appendix G Completed and requested evaluations. Verbal Protocol which was used as an instrument for research in Master's studies at the Universidad de Medellín, Colombia 123

Appendix H Metathink V2.0 Assessment Instrument. Jensen, R., de Moraes Lopes, M. H. B., Silveira, P. S. P., Ortega, N. R. S., & others. (2012). The development and evaluation of software to verify diagnostic accuracy. Revista Da Escola de Enfermagem Da USP 124

Appendix I Flórez, Y. P., Jerónimo, A. J., Castillo, M. E., & Gómez, A. A. (2019, March). User-Based Cognitive Model in NGOMS-L for the Towers of Hanoi Algorithm in the Metacognitive Architecture CARINA. In The ICAETT. DOI: https://doi.org/10.1007/978-3-030-32022-5_44. 126

Appendix J . Certificate of participation in: 1st International Conference on Advances in Emerging Trends and Technologies ICAETT 2019 127

Appendix K Certificate of participation in: “Primer congreso Nacional de Semilleros de Investigación y Emprendimiento” 128

Appendix L V Encuentro Interinstitucional de Semilleros de Investigación 2020. Mayo 12 y 13 129

Appendix M Research project sponsored by the Universidad de Córdoba 131

Chapter I

Introduction

Cognitive modeling is a methodology of the cognitive sciences that allows the simulation of human cognitive processes in various forms, commonly in a computational and mathematical way (Sun, 2008a). The cognitive modeling aims understanding the fundaments of cognition by designing cognitive models based on mathematical or computational processes, mechanisms and representations (Sun, 2008a; Caro, Josyula, Madera, Kennedy, & Gómez, 2019 & Flórez, Jerónimo, Castillo, & Gómez, 2019).

A cognitive model is a verbal-conceptual, computational and mathematical description of some mental processes, whose main purpose is understanding and/or predicting human or animal behavior (Sun, 2008b; Flórez et al., 2019 & Lieder & Griffiths, 2019). Cognitive models represent mental elements from a theoretically and empirically perspective of processes included in a cognitive task (Lieder & Griffiths, 2019; Caro, Josvula, Gomez, & Kennedy, 2018; Jerónimo, Caro, & Gómez, 2018 & Cox, Oates, & Perlis, 2011).

Cognitive architectures can be used to represent a set of cognitive models in a variety of intelligent systems (Caro et al., 2018). Cognitive models developed for a cognitive architecture are characterized by to being executables and to produce a set of specific behaviors (Flórez et al., 2019). The use of cognitive architectures to represent cognitive models allows the integration of rational principles, the application of optimization principles and the understanding of neural representations (Lieder & Griffiths, 2019). However, the use of cognitive architectures is not a necessary requirement to represent cognitive models. Cognitive models can be designed creating optimal algorithms that outline complex real-world problems (Lieder & Griffiths, 2019) through assumptions, principles of rationality and adaptation made by researchers (Lieder & Griffiths, 2019).

CARINA is a metacognitive architecture for the creation of artificial intelligent agents, derived from the Metacognitive Metamodel MISM (Caro et al., 2018 & Caro, Josyula, Jiménez, Kennedy, & Cox, 2015). CARINA is structured by two cognitive levels called the *object-level* which involves the model that an artificial intelligent agent has for reasoning about the world/environment to solve problems and the *meta-level* integrates the elements, knowledge and processes to the development of monitoring and control of its own learning and reasoning mechanisms (Caro et al., 2019).

A metacognitive architecture according to Caro et al., (2019); Cox et al., (2011); Caro et al., (2018); Cox, (2005) & Flórez et al., (2019) is a framework for the modeling of mechanisms that an intelligent agent integrates introspectively monitoring and meta-level control of its own reasoning process.

With the use of metacognitive architectures it is possible to design structural and functional elements in order to give capabilities of introspective monitoring and meta-level control to intelligent systems (Caro et al., 2019). Cognitive modeling is used to study each one of the complex processes involved in intelligent agents, as offers specific descriptions of cognitive mechanisms using algorithms and programs based on cognitive computing (Caro & Jiménez, 2014).

Different authors have created cognitive and metacognitive architectures and have investigated how intelligent systems can acquire knowledge through different specific mechanisms. ACT-R is a cognitive architecture structured by a set of independent modules acting around a central procedure module. The sub-modules are classified into *perception* (visual, aural), *control* (vocal, manual), *memory* (declarative) and *state* (problem and control). The interaction between modules is through small buffers. The procedure module has rules to make cognitive actions stored in the buffers (Borst & Anderson, 2015).

CLARION is a cognitive architecture that integrates a set of subsystems for various psychological functionalities with a dual specification in each subsystem. The subsystems are focused in *the action-centered subsystem* (the ACS) to control external (physical) or internal (mental) actions, *the non-action-centered subsystem* (the NACS) to store declarative knowledge, *the motivational subsystem* (MS) to indicate if a result in action, perception and cognition is satisfactory or not, and *the metacognitive subsystem* (MCS) to monitor, direct and modify other subsystems (Sun & Helie, 2015).

MIDCA is a metacognitive architecture structured in two cycles of *action-perception* at a cognitive (*object-level*) and metacognitive (*meta-level*). The outputs of the cycles are based on the *intention*, *planning* and *action execution*. The inputs are based on the *perception*, *interpretation* and *goal evaluation*. A cycle selects a goal to achieve, the agent establishes a plan to achieve the goal with a series of actions. The agent identifies changes in the actions based on the plans and evaluates the goal. In the *object-level* the cycle activates the goals that change the environment. In the *meta-level* and cycle it activates the goals that change the *object-level* and it monitors introspectively the mental processes and the changes of state at a cognitive level (Paisner et al., 2014).

CARINA is a metacognitive architecture structured by two cognitive levels called *object-level* and *meta-level*. The *object-level* has the model of the world to solve problems. The *meta-level* represents the reasoning of an artificial intelligent agent. Furthermore, the *meta-level* has the components, knowledge and mechanisms for an intelligent system to monitor and control its own learning and reasoning processes.

In the context described, the main objective of this thesis is to create in the CARINA metacognitive architecture mechanisms of knowledge acquisition called, cognitive models. With the aim of simplifying the process of developing intelligent agents and allowing any programmer to use CARINA to solve cognitive tasks. It will only on descriptions of cognition and relationships with algorithms and programs, based on computer science and technology simplifying the complexities of detailed descriptions of cognitive mechanisms of brain functioning.

1.1. Motivation

This research focuses on the construction of a knowledge acquisition structure for the CARINA metacognitive architecture, called, cognitive models. The motivation of this project is the creation of a standardized structure that allows the CARINA metacognitive architecture to obtain knowledge through of the formalizing of problems and the automatic production of plans as product of its reasoning process. Thus, any cognitive designer and developer will can solve real world problems using CARINA. Given the complexity of the human mind, it is necessary to create well-structured theories based on processes. These theories help to understand cognitive processes, by specifying processes in detail as algorithmic specificity, that is, detailed steps, exactly specified and carefully thought out, organized in precise but flexible sequences, allowing conceptual clarity and precision (Sun, 2008a).

In this sense, computational models are executable in a cognitive or metacognitive architecture. Cognitive and metacognitive architectures facilitate the creation of artificial systems capable of showing intelligent behavior in a general environment through a specific analogy with the constitutive and developmental functioning and mechanisms underlying human cognition (Lieto et al., 2018).

1.2. Thesis Project

1.2.1. Research Project

This research focuses on the development of cognitive models for the metacognitive architecture CARINA. Below, the problem, the questions and the objectives which conduct this research are in detail presented.

1.2.2. Research Problem

A cognitive architecture allows studying essential structures, mechanism and process of human mind (Sun, 2009). A metacognitive architecture also studies cognition, specifies definition of structural and functional elements of metacognition (Caro et al., 2019). In this sense, with a metacognitive architecture it is possible the creation of mechanisms for an intelligent systems using introspective monitoring and meta-level control of its own reasoning process (Caro et al., 2019 & Paisner et al., 2014).

In Artificial Intelligence, according to Russell & Norvig, (2002) an intelligent agent must have an *environment*, *perceptions*, *sensors* and *actuators*. In this sense, Caro et al., (2018) propose that an intelligent agent based on CARINA metacognitive architecture, must have: *object-level* (i.e., environment), *meta-level* (i.e., mechanisms, knowledge and components necessary for self-monitoring of its own knowledge acquisition process), which will allow detecting reasoning failures and at the same time create computational strategies to solve them. For this reason, it is necessary to create knowledge acquisition mechanisms in intelligent systems. The acquisition of knowledge in an Intelligent Tutoring System refers to the level of knowledge and skills of adapting the material to individual needs (Hatzilygeroudis & Prentzas, 2006).

To achieve the decision making, the domain of knowledge and to adapt the information towards a user, techniques of the Artificial Intelligence are used. Thus, the fundamental purpose of an intelligent tutoring system is to represent the related knowledge and how the reasoning for decision making is achieved, through the use of knowledge acquisition schemes that will change according to the knowledge domain approach. In this sense, according to Hatzilygeroudis & Prentzas, (2006) & Hatzilygeroudis & Prentzas, (2004) the requirements that an intelligent system must have for acquisition of knowledge are: *construction phase*, *operation phase* and *maintenance phase*. There are three types of users, such as: (i) *domain experts,* (ii) *knowledge engineers*, and (iii) *students*. And each type of user has requirements for knowledge schemes. The domain experts offer information about problems, how to deal with them and practices obtained from their experience, also the system must acquire knowledge from external sources (ease of acquisition). The end user (student), refers to time efficiency. Finally, the system requirements that refer to the types of knowledge, which are represented in schemes such as i) *Structural Knowledge* (Semantic nets/frames, description logics), ii) *Relational Knowledge* (Semantic nets/frames, belief networks, description logics), iii) *Uncertain Knowledge* (belief network, iv) *Vague Knowledge* (fuzzy rules, neurofuzzy representations) and v) *Heuristic Knowledge* (symbolic rules, fuzzy rules, neurules).

Likewise, cognitive and metacognitive architectures have been created with mechanisms to acquire knowledge. For example, according to Sun & Helie, (2015) in the cognitive architecture CLARION, knowledge is based in computational cognitive models, using one of the four subsystems, for the analysis process that underlies human mind, expressed in a computer program way: *the non-action-centered subsystem* (the NACS) to store declarative knowledge.

In cognitive architecture ACTR, the knowledge acquisition is based on “chunks” which are simply collections of *key-value* pairs (Anderson, 1996). Which is in one of the four modules of the architecture known as memory (declarative). In other hand the metacognitive architecture MIDCA according to Paisner et al., (2014) uses case-based knowledge representations implemented as frames tied together by explanation-patterns that represent general causal structures.

Acquiring knowledge in a metacognitive architecture is important since this will allow to create processes of reasoning, monitoring and introspection, in addition to facilitating elements to the developers to create a variety of intelligent systems (Caro et al., 2019). For this reason, it is necessary to create a mechanism of knowledge acquisition in CARINA. This knowledge is expressed into cognitive models. The cognitive models, are inspired by a functional approach to human mind philosophy, so that the cognitive systems derived from CARINA, will execute cognitive models of different cognitive tasks that solve cognitive and metacognitive problems.

1.3. Research Question

The research question is presented according to the context described in the research problem:

RQ: How to develop cognitive models as knowledge acquisition mechanisms for the CARINA metacognitive architecture?

The research problem is described in a set of questions which fulfill the function of decomposing the main problem into problems of less complexity.

SRQ1: How to represent formal, semantic and computationally cognitive models for the metacognitive architecture CARINA?

SRQ2: How to build a functional prototype of a framework for the creation of cognitive models in the metacognitive architecture CARINA?

SRQ3: How to create cognitive models in several knowledge domains using CARINA based intelligent systems?

1.4. Objectives

1.4.1. General Objective

To develop cognitive models as knowledge acquisition mechanisms for the CARINA metacognitive architecture.

1.4.2. Specific Objectives

- To represent formal, semantic and computationally cognitive models for the CARINA metacognitive architecture.
- To build a functional prototype of a framework for the creation of cognitive models in the CARINA metacognitive architecture.
- To create cognitive models in several knowledge domains using CARINA based intelligent systems

1.5. Methodology

This research is part of the research methods (R+D) used in computer science, called modeling (Barchini, 2005). According to Barchini, (2005), modelling allows the study and the analysis of phenomena related to information, when designing, developing and when solving problems guided theoretically or empirically. In this research, a type of modeling will be developed which uses complex process-based theories to explain intricate details of the human mind, this type of modeling is called cognitive modeling (Sun, 2008a).

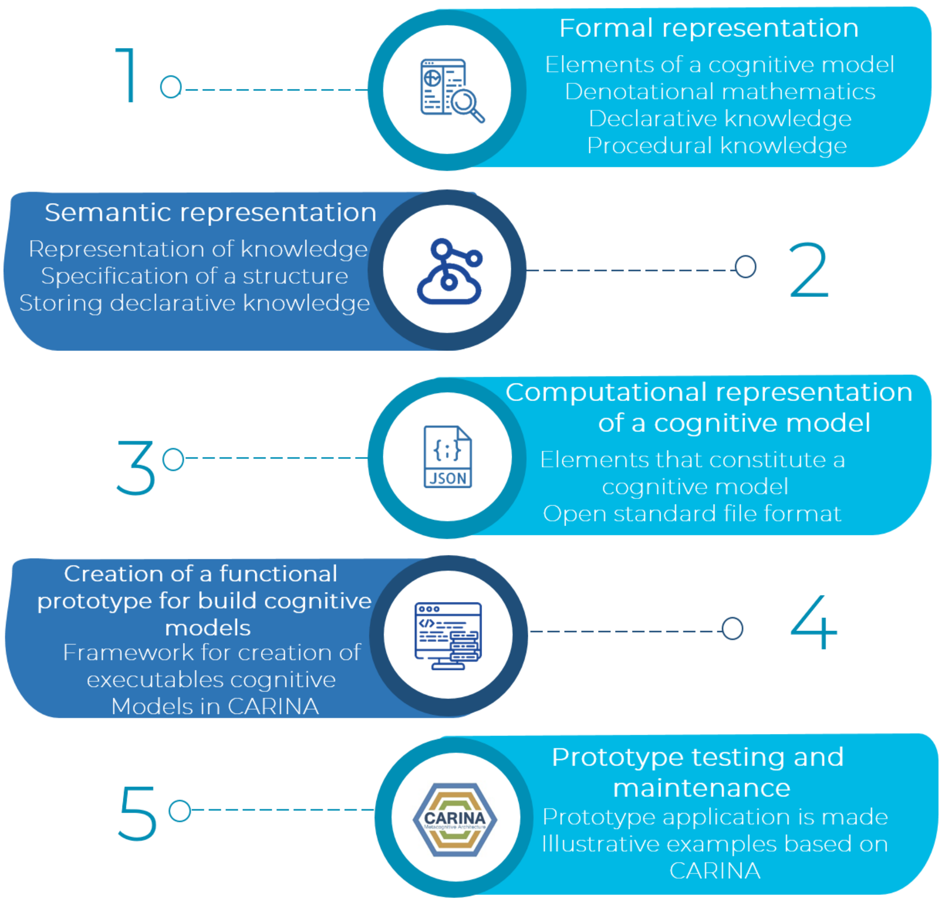
This research is divided into five phases, which are described below:

1. *Formal representation:* at this stage the elements of a cognitive model are defined, through the use of denotational mathematics (Wang, 2008a), specifying declarative knowledge and procedural knowledge of a cognitive task.
2. *Semantic representation:* in this stage a semantic representation of knowledge is detailed, for cognitive models in CARINA, through the specification of a structure that facilitates the storing of declarative knowledge.
3. *Computational representation of a cognitive model:* in this stage, the elements that constitute a cognitive model will be described using an open standard file format for data interchange to be executed in CARINA.
4. Creation of a functional prototype for build cognitive models: in this stage a functional prototype of a framework for the creation of executable cognitive models in CARINA is showed.
5. *Prototype testing and maintenance:* in this phase the prototype application is made, through illustrative examples that allow the execution of cognitive models in several knowledge domains, using intelligent systems, based on CARINA.

Different methodological phases that compose the research are shown in Figure 1.

Figure 1

Research methodology



1.6. Document Organization

This thesis project is structured as follows: Chapter "Introduction" provides an overview of the project, presenting the motivation, thesis project, the research project, the research problem, the problem question, the objectives, the methodology and contributions. The Chapter "Theoretical background" describes an overview in the research areas covered in the thesis project. The Chapter "Theoretical Framework" describes the theoretical and most important aspects of the categories that are part of this research. Chapter “The Metacognitive Architecture CARINA” describes a general overview about the structure of metacognitive architecture, in which are created cognitive models. The chapter “Cognitive Models for the Metacognitive

Architecture CARINA" presents a description of Formal Representation of Cognitive Models in CARINA, Formal and Semantic Representation of a Cognitive Model in the Metacognitive Architecture CARINA, Computational representation of cognitive models for the CARINA metacognitive architecture and the creation of a functional prototype by the elaboration of cognitive models in a visual way, called: MetaThink version 2.0. Then, chapter "Illustrative Examples of Cognitive Models in CARINA" shows a set of examples developed using CARINA. Finally, the conclusions are describing the results from the cognitive model created in a formal, semantic and computational way. The results of validation of MetaThink version 2.0, recommendations and future works in the generation of cognitive models.

Chapter II

Theoretical Background

This theoretical background is constituted by three main categories: Cognitive Modeling, Metacognitive Architectures and Metacognitive Architecture CARINA. Different studies according to these categories are presented below:

Lebiere, (1999) in his thesis: “The dynamics of cognition: An ACT-R model of cognitive arithmetic”, facilitates a structured process model of the process of adding using a general-purpose cognitive modeling architecture (ACT-R). This model produces minimal assumptions of the elements which belong to this process, using the architecture’s Bayesian learning mechanism to derive the desired results from the statistical structure of the task. The behavior of this model is examined through separated simulations of each main result, a single simulation of a lifetime of arithmetic learning, a formal analysis of the model’s dynamic and an empirical variation of the simulation’s parameters. The thesis provides a unifying of the cognitive arithmetic principal results. Through its parameter analysis, it offers some practical lessons for arithmetic teaching. The constrains of a simulation of arithmetic learning also expose the underlaying assumptions of ACT-R’s associative learning mechanism. Lebiere, (1999), realizes the basic representation of a simple arithmetic problem (the addition), expressed in the form of "chunks", accompanied by a production rule for its retrieval during the resolution of the arithmetic problem stored in the long-term memory and thus, achieved a source of mathematical knowledge.

For this research, was taken the specification of numbers and arithmetic facts in additional problems, using backup computation as a strategy of empirical phenomena in the field of cognitive arithmetic in which users (children and adults), solve an arithmetic problem and this strategy allows the development of the elements of a cognitive model in a declarative way.

Furthermore, the model proposed by Lebiere, (1999) in his thesis will allow the comparison of the models in ACT-R and CARINA through an illustrative example (the addition of two numbers) in the domain of cognitive arithmetic, where the compared cognitive model will become a source of arithmetic knowledge stored in CARINA's long-term memory.

In: “Using Cognitive Models to Understand Multimodal Processes: The Case for Speech and Gesture Production” proposed by Kopp & Bergmann, (2017) discuss how computational cognitive models can be useful for the field of multimodal and multisensory interaction. This chapter describes a cognitive model as a deeper level of study in terms of processes and mechanisms that underlie a certain behavior. The focus of discussion on one case of natural multimodal behavior that has been extensively researched, the use of spontaneous speech and gesture in dialogue. The steps by the discussion are: i) reviewing speech and gesture as a pervasive case of natural multimodal behavior, ii) motivate its relevance for practical multimodal interfaces, virtual characters, or social robotics iii) discuss existing theoretical and computational models of the cognitive underpinnings and iv) elaborate on one particular cognitive model of speech gesture production that explains the role of mental representation and memory processes up to a degree that does afford computational simulation under varying conditions. To finally, demonstrate how cognitive modeling can be used to gain a better understanding of multimodal production processes and to inform the design of multimodal interactive systems.

This research provides the theoretical assumptions that are part of cognitive modeling or a cognitive model which captures structural and functional properties generally assumed from the human mind. Thus, the definitions of a cognitive model in the field of cognitive modeling are used for the purpose of comprehension and prediction in fields of study such as cognitive psychology, cognitive science, and artificial intelligence.

Paisner et al., (2014) in: “Goal-Driven Autonomy for Cognitive Systems” present an approach to autonomy in autonomous agents that seeks to maximize robustness rather than optimality on a specific task involved in complex and dynamic environments. This approach is called: goal-driven autonomy in a cognitive architecture. Goal-Driven Autonomy (GDA) is a notion that gives full independence to autonomous agents, rather than common anomaly-detection, the agent explores for problems in the context of its current goals and mission. Rather than general assessment of an entire world state, the agent should abductively explain the causal factors increasing the problem. Given an explanation, a GDA agent can produce a (may be new) goal that resolves the problem (e.g., by removing its supporting conditions). In these terms, GDA includes recognizing possibly new problems, explaining what origins the problems and generating goals to resolve them. The model it is presents within the MIDCA cognitive architecture and show that under certain conditions this model outperforms a less flexible approach to handle unexpected events. Thus, they examine the distinction between such approaches to intelligent reasoning and behavior in a metacognitive architecture called MIDCA using an implemented instantiation of the GDA model, called XPLAIN.

XPLAIN relies on general domain knowledge, a case library of prior plan schemas and a set of general explanation patterns that are used to characterize useful explanations involving that background knowledge. These knowledge structures are stored in a (currently) separate memory sub-system and communicated through standard socket connections to the rest of MIDCA. In this sense, the contribution of this work is the synergy between the use of data-driven techniques in anomaly detection, neural networks, and machine learning, as well as a predicate logic state representation and techniques for explanation generation and planning that rely on high level formalisms. Thus, the integration of these approaches is one of the most promising opportunities in modern Artificial Intelligence (AI), and one of the central focuses of MIDCA.

The description given by Paisner et al., (2014), provided in this thesis definitions of the elements of an goal, the characteristics and functionalities within a cognitive model that will be designed and implemented in the CARINA metacognitive architecture.

According to Caro et al., (2018) in their research: “Introduction to the CARINA Metacognitive Architecture” proposed a metacognitive architecture known as CARINA, which is a metacognitive architecture for the development of artificial intelligent agents. CARINA is derived from the MISM Metacognitive Metamodel, and constitutes self-regulation and metamemory with support for the metacognitive mechanisms of introspective monitoring and meta-level control. This research project is based on the metacognitive architecture CARINA to create and execute the cognitive models that will be runnable in the any cognitive agent created by cognitive designers.

Olier et al., (2018) proposed: “Cognitive Modeling Process in Metacognitive Architecture CARINA” which is a cognitive modeling methodology for the elaboration of cognitive models in the metacognitive architecture CARINA structured by seven steps as: i) *Selection of cognitive task*, ii) *Obtaining information for describing the cognitive, task*, iii) *Description of cognitive task in natural language*, iv) *Description of cognitive task in GOMS*, v) *Codification of cognitive model from GOMS to M++ language*, vi) *Execution of runnable cognitive model in CARINA* and vii) *Testing and Maintenance of Cognitive Model.* All these steps must be completed to be developed for designing, creating and executing cognitive models in this metacognitive architecture. Through an illustrative example, Olier et al., (2018) detail the syntactic analysis process of sentences. The main objective of this research was the specification of a cognitive modeling process for the cognitive modeler to use detailing and accurately replication when it requires the solution of problems using cognitive agents based on CARINA.

The present research project uses this methodology of cognitive modeling, for the construction of the methodological phases in the development of cognitive models which will executed on the metacognitive architecture CARINA.

Following the steps proposed by Olier et al., (2018) to create an executable cognitive model in CARINA, a cognitive model based on experts was created by López et al., (2018) for the representation in M++ of the Cognitive Model for the generation of Factoid-WH questions. The cognitive model is presented below:

1) *Selection of Cognitive Task:*

   The cognitive task that was selected was the development of a cognitive model for the creation of Factoid- Wh question in English as a foreign language.

2) *Obtaining Information for describing the cognitive task:*

The information presented in the cognitive task was acquired using experts and some documentary sources as sources of information (see Table 1).

Table 1

Format to synthesize the cognitive task description when the information source comes from experts.

	Experts	**X**
Knowledge Area	Cognitive Computing and Applied Linguistics	
Number of Experts	1 MSc. in Technology of Information Applied to Education	

	2 BSc. in English
Synthesis of Cognitive Task Description	Cognitive modeling is a methodology derived from cognitive sciences, which purpose of generating theories, expressed in terms of computer programs. The aims of cognitive modeling are: (a) describe (b) predict, (c) and prescribe human behavior using computational models of cognitive processes called Cognitive Models.

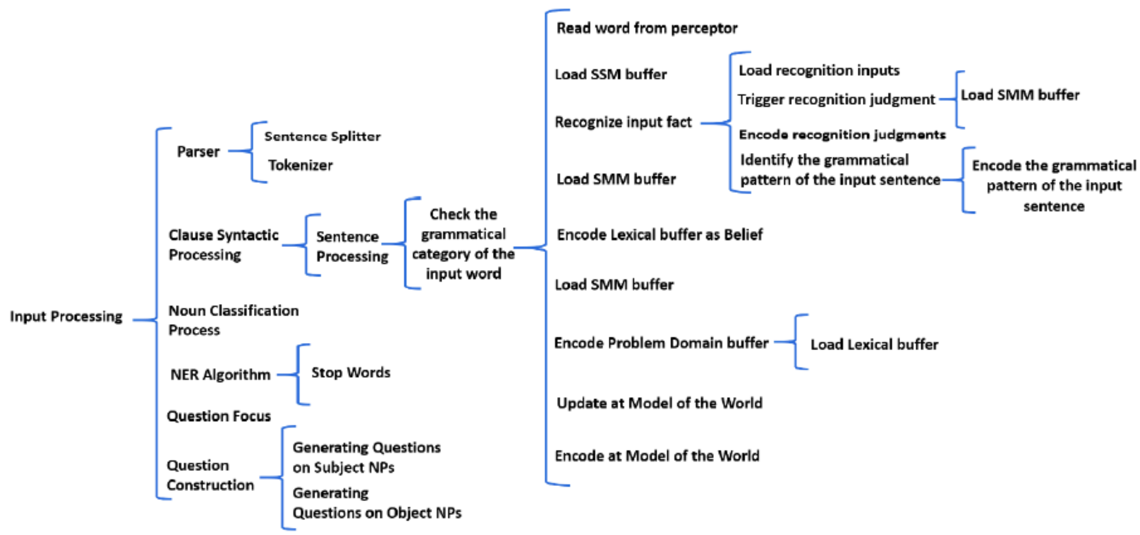
Note. The table was adapted from: López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica, 18(2),* 11-20. DOI: https://doi.org/10.25044/25392190.967.

3) *Description of Cognitive Task in Natural Language:*

The cognitive model for Factoid-Wh questions is constituted by elements such as: *Goals*, *Actions*, *Mental States* and *Production Rules*. This cognitive model has a central Goal denominated: "*Input Processing*" which is structured by different *sub-goals* that allow to complete the elaboration of Factoid-WH questions detailly. According to Rus et al., (2012), the *Goals* and *sub-goals* are showed below, where the question generation is expressed as a three-step process: *Content Selection*, *Selection of Question type* and *Question Construction*. In Figure 2, are offered the *goals* and *sub-goals* that should be achieved out for the elaboration of the questions Factoid-WH question.

Figure 2

"Input Processing" to complete the elaboration of factoid-Wh questions



Note. Goals and Sub-goals that Should be Achieved out for the Elaboration of the Questions Factoid-WH Question in CARINA. Figure was adapted from: López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica, 18(2)*, 11-20. DOI: https://doi.org/10.25044/25392190.967.

Based on the main Goal and the Sub-goals showed in Figure 2, the cognitive model for the Factoid-WH question is represented in natural language below (see Figure 3):

Figure 3

Cognitive model for the factoid-Wh questions generation in natural language

```
The input is gotten and a Parsing process is developed:
  a. The input is structured into sentences, thus word by word is read
     using a Parsing process called “Left Corner” and then the end of
     each sentence is detected by identifying a question mark, an
     exclamation point, "/ n", end point or the end of the text.
  b. Words are separated from other textual elements such as
```

Figure 3

parentheses, brackets, question marks, exclamation marks, numbers, digits, etc. Then, each of the remaining elements becomes Tokens. A token is each element that structures a sentence.

2 Each sentence is word by word syntactically processed, verifying the grammatical category of each detected word:

a. The word of the perceptor is read.
b. The word as an input fact is encoded.
c. The coded input fact is stored in the buffer of the sensory memory (SSM Buffer).
d. The input fact is copied from the SSM buffer to BCPU.Input
e. The input is loaded from the BCPU. Input (Buffer SSM)
f. A Pattern τ of the Pattern Set of the Short-Term Memory is loaded

Figure 3

g. The Belief β of the Semantic Memory (SMM) is retrieved using BCPU.Input as a cue.
h. The Belief β into the SMM Buffer is copied.
i. A recognition judgment is triggered ϑ only if the SMM Buffer is empty
j. A new Belief β is encoded with the recognition judgment ϑ
k. The Belief ϑ into BCPU.Pattern is copied

3 The lexical buffer is loaded

a. The input data is loaded from the BCPU.Input (SSM)
b. The input fact is copied to <Word Node> Buffer / Field in MoW

Figure 3

4 The buffer of the problem domain is encoded

5 The Belief β de Buffer / Campo in MoW is retrieved

Figure 3

```
6   The Belief β is copied in STM Lexical in MoW

7   The word node is updated in the MoW
       a. The Belief β of <Word Node> Buffer / Field in MoW is retrieved
8   Word Node> is encoded in MoW
       a. The Belief β of the Lexical Buffer is retrieved
       b. The Belief β is loaded from Buffer / Field in MoW
       c. The Belief β is copied in Lexical STM in MoW
       d. The Belief β is copied to <Word Node> Buffer / Field in MoW
9   The classification of nouns is processed
       a. Search for nouns in the structure of the sentence
       b. Appropriate, not appropriate nouns and adverbs are labeled in the
          structure of the sentence.
```

Figure 3

```
10  Recognized Algorithm of Nominated Entities is executed
       a. The output of the classification of nouns is obtained
       b. Who / Whom is used for PERSON or ORGANIZATION or Where for LOCATION
          # If <Proper Noun> = true the algorithm is activated
       c. If the <Proper Noun> is false the algorithm is activated
       d. The question word How is used is an adverb.
11  Connector words are chosen
       a. It is obtained from the output of the noun's classification process
       b. The connecting words of the sentence are eliminated
       c. Use of Who / Whom for PERSON or ORGANIZATION or Where for LOCATION
          or What in case THERE IS NO ENTITY
12  The question is focused
```

Figure 3

```
13  The question is generated
```

Figure 3

```
14  If the question Factoid WH-Question is subject NP is attached to the main
    verb of the sentence and is identified: NP = n1> (S = n2> ROOT) & $ ++ VP
    = n3 (4)
15  If it is Factoid question WH-Question is NP objects are attached to the
    front of the sentence and the NP object is identified: NP = n1! >> NP >>
    (VP> (S = n2> ROOT))
```

Note. Figure was adapted from: López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica, 18(2)*, 11-20. DOI: https://doi.org/10.25044/25392190.967.

4) *Cognitive Task in GOMS (NGOMS-L)*

In order to continue describing the analysis of the cognitive task for the construction of the cognitive model, it is necessary to use methodologies that specify a set of *Objectives* (*Goals*) and *sub-steps* (*steps*) allowing greater ease when performing the analysis of cognitive tasks and turn them into cognitive models.

One methodology that allows this process is *Goals*, *Operators* and *Methods* (GOMS). As a variation of *GOMS*, for this cognitive task, was implemented *NGOMS-L*, according to Kieras, (1999) can be defined as a natural language notation to represent GOMS models and a generate a process to build them. In addition, its function is to give predictions of the operator's sequence, execution time and time to learn the methods (John & Kieras, 1996).

In this sense, a *GOMS* model is structured by methods with the purpose to achieve *Goals*, which are constituted by *Operators* (are specific steps apply by a user performs in a specific execution time). If a Goal can be accomplished by more than one method, the S*election Rules* are used to establish the appropriate *Method* (see Figure 4).

Figure 4

Cognitive model for the factoid-Wh questions generation in GOMS (NGOMS-L)

Method for goal γ_{300}: Input Processing
Step 1. (α^c_{101}) Accomplish goal:γ_{301} *# Parser*
Step 2. (α^c_{102}) Accomplish goal:γ_{304} *# Clause Syntactic Processing*
Step 3. (α^c_{103}) Accomplish goal:γ_{310}*#Noun Classification Process*
Step 4. (α^c_{104}) Accomplish goal:γ_{306} *# NER Algorithm*
Step 5. (α^c_{105}) Accomplish goal:γ_{305} *# Question Focus*
Step 6. (α^c_{106}) Accomplish goal:γ_{307}*#Question Generation*
Step n. (α^c_{107}) Return with goal accomplished.

Method for goal γ_{301}: Parser
Step 1. (α^c_{108}) Accomplish goal:γ_{302} *# Sentence Splitter*
Step 2. (α^c_{109}) Accomplish goal:γ_{303} *# Tokenizer*
Step n. (α^c_{107}) Return with goal accomplished.

Method for goal γ_{302}: Sentence Splitter
Step 1. (α^c_{110}) Read word by word using left corner parsing
Step 2. (α^c_{111})Detecting ending of sentences *# detecting a question mark, an exclamation mark, /n, full stop or the end of text.*
Step n. (α^c_{107}) Return with goal accomplished

Step 3. (α^c_{107}) Return with goal accomplished.

Method for goal γ_{103}: Load SSM buffer
Step 1. (α^c_{125}) Copy input fact ϕ into SSM buffer at BCPU. Input
Step 2. (α^c_{107}) Return with goal accomplished.

Method for goal γ_{123}: Encode recognition judgments
Step 4. (α^p_{126})Encode a new belief with judgment ϑ
Step 5. (α^p_{127})Copy belief ϑ into BCPU. Pattern
Step n. (α^c_{107}) Return with goal accomplished.

Method for goal γ_{124}: Recognize input fact
Step 1. (α^c_{128}) Accomplish goal:γ_{122}
Step 2. (α^c_{129}) Accomplish goal:γ_{125}
Step 3. (α^c_{130}) Accomplish goal:γ_{123}
Step n. (α^c_{107}) Return with goal accomplished.

Method for goal γ_{125}: Trigger recognition judgment
Step 1. (α^c_{131}) Accomplish goal:γ_{104}
Step 2. (α^c_{132}) Trigger a judgment ϑ *# if rule#103 <SMM buffer is empty> is fired*
Step n. (α^c_{107}) Return with goal accomplished.

Method for goal γ_{303}: Tokenizer
Step 1. (α^c_{112}) Separation of words from other textual elements *# parenthesis, brackets, question marks, exclamation marks, currency marks, numbers, digits included in words, etc.*
Step 2. (α^c_{113}) Becomes all this element in tokens *# we call tokens to each element that composes a clause*
Step n. (α^c_{107}) Return with goal accomplished

Method for goal γ_{100}: Sentence Processing
Step 1. (α^c_{114}) Accomplish goal: γ_{101}
Step 2. (α^c_{107}) Return with goal accomplished.

Method for goal γ_{101}: Check the grammatical category of the input word
Step 1. (α^c_{115}) Accomplish goal:γ_{102} *# cf Perception*
Step 2. (α^c_{116}) Accomplish goal:γ_{103} *# cf Perception*
Step 3. (α^c_{117}) Accomplish goal:γ_{124} *# cf Recognition*
Step 4. (α^c_{118}) Accomplish goal:γ_{104} *# cf Categorization*
Step 5. (α^c_{119}) Accomplish goal:γ_{106}
Step 6. (α^c_{120}) Accomplish goal:γ_{107}*# if rule#101 <π:inicio =null> is fired*
Step 7. (α^c_{121}) Accomplish goal:γ_{108}*# if rule#101 <π:inicio $\neq$null> is fired*
Step 8. (α^c_{122}) Accomplish goal:γ_{109} *# if rule#102 <MoW: word_node $\neq$null> is fired*
Step n. (α^c_{107}) Return with goal accomplished.

Method for goal γ_{122}: Load recognition inputs
Step 1. (α^c_{133}) Load input fact ϕ from BCPU.Input (SSM buffer)
Step 2. (α^c_{134}) Load Pattern τ from Pattern (Short Term Memory)
Step 3. (α^c_{107}) Return with goal accomplished.

Method for goal γ_{104}: Load SMM buffer
Step 1. (α^c_{135}) Retrieve belief β from SMM using BCPU.Input ϕ: *word* as cue
Step 2. (α^c_{136}) Copy belief βinto SMM buffer
Step 3. (α^c_{107}) Return with goal accomplished.

Method for goal γ_{105}: Load Lexical buffer
Step 1. (α^c_{137}) Load input fact ϕ from BCPU.Input (SSM)
Step 2. (α^c_{138}) Copy input fact ϕ into <Word Node> Buffer/Field at MoW
Step 3. (α^c_{107}) Return with goal accomplished.

Method for goal γ_{106}: Encode Lexical buffer as Belief
Step 1. (α^c_{139} Accomplish goal:γ_{105}
Step 2. (α^c_{140}) Load belief β from SMM Buffer
Step 3. (α^c_{141}) ALTER ϕ: Update the **ID** of input fact

Method for goal γ_{102}: Read word from perceptor <text_perceptor_1>
Step 1. (α^p_{123}) Encode word as input fact ϕ
Step 2. (α^c_{124}) Save encoded fact ϕ into SSM

uniqueID) #encode (WordNode)
Step 4. (α^c_{142}) ALTER ϕ: Add field <next> to ϕ
Step 5. (α^c_{143})ALTER ϕ: Update ϕ:*next* with <null>

Step 6. (α^{c}_{144}) ALTER ϕ: Update ϕ:*typeSMU* with β:*tipo*
Step n. (α^{c}_{107}) Return with goal accomplished.

Method for goal γ_{107}: Encode Problem Domain buffer <updating π: *inicio*>
Step 1. (α^{c}_{145}) Retrieve belief ϕ from <Word Node> Buffer/Field at MoW
Step 2. (α^{c}_{146}) ALTER π: Update π:*inicio* with ϕ
Step 3. (α^{c}_{147}) Copy belief π into Lexical STM at MoW
Step n. (α^{c}_{107}) Return with goal accomplished.

Method for goal γ_{108}: Update <Word Node> at Model of the World (MoW)
Step 1. (α^{c}_{148}) Retrieve belief ϕ from <Word Node> Buffer/Field at MoW
Step n. (α^{c}_{107}) Return with goal accomplished.

Method for goal γ_{109}: Encode <Word Node> at MoW
Step 1. (α^{c}_{149}) Retrieve belief ϕ from Lexical Buffer ϕ: *next is null* as cue

Step 2. (α^{c}_{150}) Load belief ϖ from <Word Node> Buffer/Field at MoW
Step 3. (α^{c}_{151}) ALTER ϕ: Update ϕ: *next* with ϖ
Step 4. (α^{c}_{152}) Copy belief ϖ into Lexical STM at MoW
Step 5. (α^{c}_{153}) Copy belief ϕ into <Word Node> Buffer/Field at MoW
Step n. (α^{c}_{107}) Return with goal accomplished.

Method for goal γ_{120}: Identify the grammatical pattern of the input sentence
Step 1. (α^{c}_{154}) Accomplish goal:γ_{121}
Step n. (α^{c}_{107}) Return with goal accomplished.

Method for goal γ_{121}: Encode the grammatical pattern of the input sentence
Step 1. (α^{c}_{155}) Encode grammatical pattern as fact μ
Step n. (α^{c}_{107}) Return with goal accomplished.

Method for goal γ_{304}: Clause Syntactic Processing
Step 1. (α^{c}_{156}) Accomplish goal:γ_{100} # *Sentence Processing*
Step 2. (α^{c}_{157}) Link Tokens # *Sentence Structure*
Step n. (α^{c}_{107}) Return with goal accomplished

Method for goal γ_{310}: Noun Classification Process
Step 1. (α^{c}_{158}) Searching Process of Nouns in the Sentence Structure

Step 2. (α^{c}_{159}) Labeling Process of Proper Nouns, No Proper Nouns and adverbs in the Sentence Structure
Step n. (α^{c}_{107}) Return with goal accomplished

Method for goal γ_{306}: NER Algorithm
Step 1. (α^{c}_{160}) Getting the output of Noun Classification Process
Step 2. (α^{c}_{161}) Using *Who/Whom* for PERSON or ORGANIZATION or *Where* for LOCATION # *If rule#104 <Proper_Noun = true> is fired*
Step 3. (α^{c}_{162}) Accomplish goal:γ_{311} # *Stop Words -If rule#104 <Proper_Noun = false> is fired*
Step 4. (α^{c}_{163}) Using the question word *How* # *If rule#104 <Adverb = true> is fired*
Step n. (α^{c}_{107}) Return with goal accomplished

Method for goal γ_{311}: Stop Words
Step 1. (α^{c}_{164}) Getting the output of Noun Classification Process
Step 1. (α^{c}_{165}) Remove stop words from the sentence
Step 2. (α^{c}_{166}) Using Who/Whom for PERSON or ORGANIZATION or Where for LOCATION or What for NO ENTITY
Step n. (α^{c}_{107}) Return with goal accomplished

Method for goal γ_{305}: Question Focus
Step 1. (α^{c}_{167}) Identify Question Word
Step 2. (α^{c}_{168}) Selection of Question Focus
Step n. (α^{c}_{107}) Return with goal accomplished

Method for goal γ_{307}: Question Generation
Step 1. (α^{c}_{169}) Accomplish goal:γ_{308}
Step 1. (α^{c}_{170}) Accomplish goal:γ_{309}
Step n. (α^{c}_{107}) Return with goal accomplished

Method for goal γ_{308}: Generating Questions on Subject NPs
Step 2. (α^{c}_{171}) The factoid WH-Question is attached to the main verb of the sentence.
Step 3. (α^{c}_{172}) The subject NP is identified # *NP = n1> (S = n2> ROOT) & $ ++ VP = n3 (4)*
Step n. (α^{c}_{107}) Return with goal accomplished

Method for goal γ_{309}: Generating Questions on Object NPs
Step 2. (α^{c}_{173}) The Factoid WH-Question word is attached to the front of the sentence.
Step 3. (α^{c}_{174}) The object NP is identified # *NP=n1!>> NP >> (VP > (S=n2 > ROOT))*

Step n. (α^{c}_{107}) Return with goal accomplished

Note. Figure was adapted from: López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica, 18(2)*, 11-20. DOI: https://doi.org/10.25044/25392190.967.

4.1) Inventory Goals, Mental States and Operators

Inventories of Goals, Mental States and Operators are presented (see Figure 5), as well as the inventory of Mental States (see Figure 6) and inventory of Operators of Cognitive Model for the Factoid-WH Questions Generation in GOMS (NGOMS-L) (see Figure 7):

Figure 5

Inventory of Goals, Mental States and Operators of cognitive model for the factoid-Wh questions generation in GOMS (NGOMS-L)

Goals from NGOMSL	
γ_{300}: Input Processing	γ_{108}: Update <Word Node> at Model of the World (MoW)
γ_{301}: Parser	γ_{109}: Encode <Word Node> at MoW
γ_{302}: Sentence Splitter	γ_{120}: Identify the grammatical pattern of the input sentence
γ_{303}: Tokenizer	γ_{121}: Encode the grammatical pattern of the input sentence
γ_{100}: Sentence Processing	γ_{304}: Clause Syntactic Processing
γ_{101}: Check the grammatical category of the input word	γ_{310}: Noun Classification Process
γ_{102}: Read word from perceptor <text_perceptor_1>	γ_{306}: NER Algorithm
γ_{103}: Load SSM buffer	γ_{311}: Stop Words
γ_{123}: Encode recognition judgments	γ_{305}: Question Focus
γ_{124}: Recognize input fact	γ_{307}: Question Generation
γ_{125}: Trigger recognition judgment	γ_{308}: Generating Questions on Subject NPs
γ_{122}: Load recognition inputs	γ_{309}: Generating Questions on Object NPs
γ_{104}: Load SMM buffer	
γ_{105}: Load Lexical buffer	
γ_{106}: Encode Lexical buffer as Belief	
γ_{107}: Encode Problem Domain buffer <updating π: *Start*>	

Note. Figure was adapted from: López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica, 18(2)*, 11-20. DOI: https://doi.org/10.25044/25392190.967.

Figure 6

Inventory of Mental States of cognitive model for the factoid-Wh questions generation in GOMS (NGOMS-L)

Inventory of Mental States

Mental States	**Comments**
σ_{101}: input_is_processed	γ_{300}: Input Processing
σ_{102}: parser_is_done	γ_{301}: Parser
σ_{103}: sentence_is_splitted	γ_{302}: Sentence Splitter
σ_{104}: tokenizer_is_loaded	γ_{303}: Tokenizer
σ_{105}: sentence_is_processed	γ_{100}: Sentence Processing
σ_{106}: grammatical_gategory_of _the_input_word_is_checked	γ_{101}: Check the grammatical category of the input word
σ_{107}: word_from_perceptor_is_read	γ_{102}: Read word from perceptor <text_perceptor_1>
σ_{108}: SSM_buffer_is_loaded	γ_{103}: Load SSM buffer
σ_{109}: recognition_judgments_is_encoded	γ_{123}: Encode recognition judgments
σ_{110}: input_fact_is_recognized	γ_{124}: Recognize input fact
σ_{111}: recognition_judment_is_triggered	γ_{125}: Trigger recognition judgment
σ_{112}: recognition_inputs_is_loaded	γ_{122}: Load recognition inputs
σ_{113}: SMM_buffer_loaded	γ_{104}: Load SMM buffer
σ_{114}: lexical_buffered_is_loaded	γ_{105}: Load Lexical buffer
σ_{115}: lexical_buffer_as_belief_is_encoded	γ_{106}: Encode Lexical buffer as Belief
σ_{116}: problem_domain_buffer_is_encoded	γ_{107}: Encode Problem Domain buffer <updating π: *inicio*>
σ_{117}: word_node_at_model_of_the_world_is_updated	γ_{108}: Update <Word Node> at Model of the World (MoW)
σ_{118}: word_node_at_model_of_the_world_is_encoded	γ_{109}: Encode <Word Node> at MoW
σ_{119}: grammatical_pattern_of_the_input_senteced_is_identified	γ_{120}: Identify the grammatical pattern of the input sentence
σ_{120}: grammatical_pattern_of_the_input_senteced_is_encoded	γ_{121}: Encode the grammatical pattern of the input sentence
σ_{121}: clause_is_processed _syntactically	γ_{304}: Clause Syntactic Processing
σ_{122}: noun_classification_process_is_done	γ_{310}: Noun Classification Process
σ_{123}: NER_algorith_is_done	γ_{306}: NER Algorithm
σ_{124}: stopo_words_are_done	γ_{311}: Stop Words
σ_{125}: question_focus_is_done	γ_{305}: Question Focus
σ_{126}: question_generation_is_done	γ_{307}: Question Generation
σ_{127}: questions_on_subject_NPs_are_generated	γ_{308}: Generating Questions on Subject NPs
σ_{128}: questions_on_object_NPs_generated	γ_{309}: Generating Questions on Object NPs

Note. Figure was adapted from: López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica, 18(2)*, 11-20. DOI: https://doi.org/10.25044/25392190.967.

Figure 7

Inventory of Operators of cognitive model for the factoid-Wh questions generation in GOMS (NGOMS-L)

Inventory of operators

The types of operators are shown with a notation which is presented below:
α^{p}_{id}: perceptual act
α^{m}_{id}: motor act
α^{c}_{id}: cognitive act

α^{c}_{101}: Accomplish Goal <accomplishGoal(aGoal)>
α^{c}_{102}: Return Goal Accomplished <returnGoalAccomplished(aGoal)>
α^{p}_{103}: Encode Fact <encodeSMU(inputData)>
α^{c}_{104}: Save Fact Into SSM <saveSMUtoSSM(aSMU)>
α^{c}_{105}: Copy Fact Into BCPU.input <copySMUtoBCPUinput(aSMU)>
α^{c}_{105}: Copy Fact Into BCPU.pattern <copySMUtoBCPUpattern(aSMU)>
α^{c}_{106}: Retrieve Belief From SMM using BCPU.Input <retrieveBeliefFromSMM (cue)>
α^{c}_{107}: Copy Belief Into SMM Buffer <copyBeliefIntoSMMBuffer (belief)>
α^{c}_{108}: Retrieve Fact from SSM <retrieveFactFromSSM (inputFact)>
α^{c}_{109}: Copy Fact Into Lexical Buffer <copyFactIntoLexicalBuffer(inputFact)>
α^{c}_{110}: ALTER Refactory Fact ID <refactorySMU (inputFact, id)>
α^{c}_{112}: ALTER Update Fact Add Field <updateSMU (inputFact, newField, value)>
α^{c}_{113}: Retrieve Belief From Lexical Buffer <retrieveBeliefFromLexicalBuffer (belief)>
α^{c}_{114}: ALTER Update Problem < updateProblem (problem)>
α^{c}_{115}: Copy Belief Into Lexical STM at MoW <copyBeliefIntoLexicalSTMatMoW (belief)>
α^{c}_{116}: Copy Input Fact Into Word Node at MoW <copyInputFactIntoWordNodeatMoW (belief)>
α^{c}_{117}: Retrieve Belief From Word Node at MoW <retrieveBeliefFromWordNodeatMoW (belief)>
α^{c}_{122}: Load fact from BCPU.Input <loadFactFromBCPUInput (inputFact)>
α^{c}_{123}: Load Pattern τ from Pattern Buffer (Sort Term Memory) <loadPatternFromPatternBuffer ()>
α^{c}_{124}: Encode a new belief with judgment ϑ <encodeaNewBeliefWithJudgment (judgment)>
α^{c}_{125}: Copy belief ϑ into Pattern Buffer at BCPU.Pattern <copyBeliefIntoPatternBufferAtBCPUPattern (belief)>
α^{c}_{126}: Trigger a judgment ϑ judgment # <triggeraJudgment (judment)>
α^{c}_{128}: Load belief ϖ from <Word Node> Buffer/Field at MoW # <loadBeliefFromWordNodeatMoW (belief)>
α^{c}_{129}: ALTER φ Update φ :next with ϖ# <updateFact (aNext, aValue)>
α^{c}_{130}: Copy Input Fact φ into <Word Node> at MoW <copyInputFactIntoWordNode (inputFact)>
α^{c}_{131}: Load belief β from SMM Buffer. # <loadBeliefFromSMMBuffer (belief)> #
α^{c}_{132}: Read word by word using left corner parsing < readWordByWordUsingLeftCornerParsing (-)>
α^{c}_{133}: Detect end of sentences <detectEndOfSentences ()>
α^{c}_{134}: Separate the words from other textual elements <separateTheWordsFromOtherTextualElements (-)>
α^{c}_{135}: Becomes all this element in tokens< Becomes all this element in tokens (-)>
α^{c}_{136}: Encode grammatical pattern as fact μ< EncodeGrammaticalPatternAsFact (-)>
α^{c}_{137}: Link Tokens < linkTokens (-)>

α^{c}_{138}: Searching Process of Nouns in the Sentence Structure < SearchingProcessOfNounsInTheSentenceStructure(-)>
α^{c}_{139}: Labeling Process of Proper Nouns, No Proper Nouns and adverbs in the Sentence Structure < LabelingProcessOfProperNouns, NoProperNounsAndAdverbsInTheSentenceStructure (-)>
α^{c}_{140}: Getting the output of Noun Classification Process < GettingTheOutputOfNounClassificationProcess(-)>
α^{c}_{141}: Using Who/Whom for PERSON or ORGANIZATION or Where for LOCATION < UsingWho/WhomForPersonOrOrganizationOrWhereForLocation(-)>
α^{c}_{142}: Using the question word How < UsingTheQuestionWordHow (-)>
α^{c}_{143}: Getting the output of Noun Classification Process< GettingTheOutputOfNounClassificationProcess(-)>
α^{c}_{144}: Remove stop words from the sentence <RemoveStopWordsFromTheSentence (-)>
α^{c}_{145}: Using Who/Whom for PERSON or ORGANIZATION or Where for LOCATION or What for NO ENTITY <UsingWho/WhomForPersonOrOrganizationOrWhereForLocationOrWhatForNoEntity (-)>
α^{c}_{146}: Identify Question Word < IdentifyQuestionWord(-)>
α^{c}_{147}: Selection of Question Focus < SelectionOfQuestionFocus (-)>
α^{c}_{148}: The factoid WH-Question is attached to the main verb of the sentence< TheFactoidWh-QuestionIsAttachedToTheMainVerbOfTheSentence (-)>

α^{c}_{149}: The subject NP is identified < TheSubjectNpIsIdentified (-)>
α^{c}_{150}: The Factoid WH-Question word is attached to the front of the sentence < TheFactoidWh-QuestionWordIsAttachedToTheFrontOfTheSentence (-)>
α^{c}_{151}: The object NP is identified < theobjectNPisidentified(-)>

Note. Figure was adapted from: López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica, 18(2)*, 11-20. DOI: https://doi.org/10.25044/25392190.967.

5) Cognitive model from GOMS to M++ language:

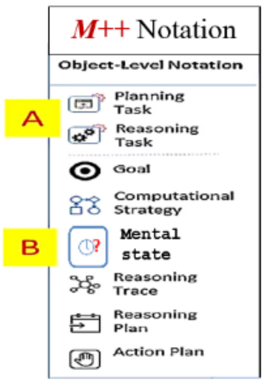
The next step shows the representation of the cognitive model in the notation language called: M ++

According to Caro et al., (2015), M++ is a domain-specific visual language (DSVL) for metacognitive level modelling in intelligent systems. In M++, abstract syntax is detailed with MOF-based metamodels and concrete syntax is shown using a mapping of abstract syntax elements to visual elaborations.

According to Caro et al., (2015) the central elements of the M++ language are visually specified models. In Figure 8, specifies the icons used to represent *object-level* tasks.

Figure 8

Main elements in M++

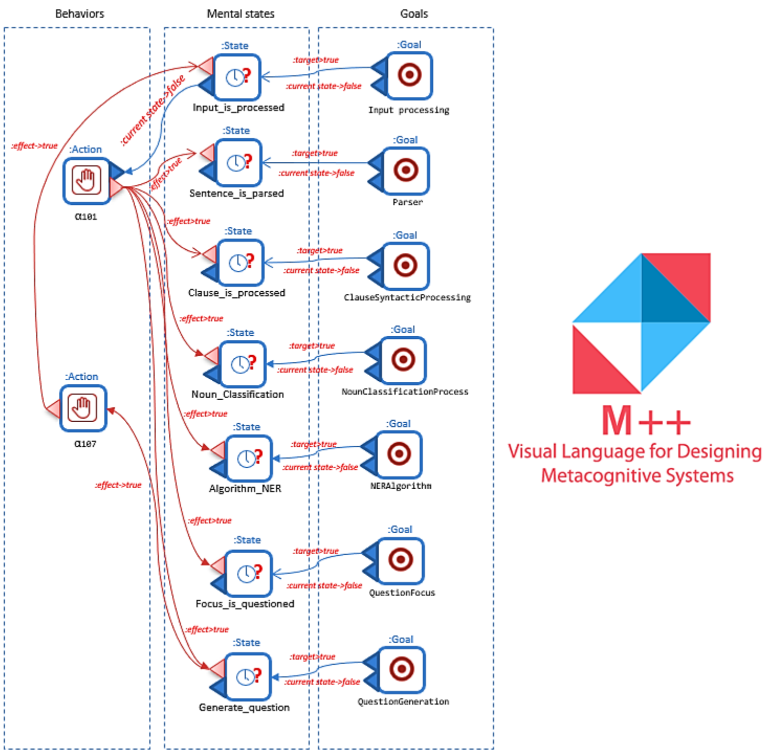


Note. Figure was adapted from: Caro, Manuel F, Josyula, D. P., Jiménez, J. A., Kennedy, C. M., & Cox, M. T. (2015). A domain-specific visual language for modeling metacognition in intelligent systems. *Biologically Inspired Cognitive Architectures*, 13, 75–90. DOI: http://dx.doi.org/10.1016/j.bica.2015.06.004.

The question generation process is shown in M++ as follows: (see Figure 9).

Figure 9

Mental States, Actions and Goals of cognitive model for the factoid-Wh questions generation in M++



Note. Figure was adapted from: López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica, 18(2)*, 11-20. DOI: https://doi.org/10.25044/25392190.967.

According to López et al., (2018) in her thesis proposes that model of the environment in CARINA is represented in the working memory using the *Mental States* and *Actions* where each Mental State is related to an Action. Thus, a cognitive model represented in M++ denotes in its center the Mental States associated with the Actions that are located in the left part of the figure, which change each mental state. In this sense, the actions have post-conditions that are affected by the mental states after performing an action, modifying their value from false to true. In addition, actions have pre-conditions that determine if the mental states to be executed have been achieved, these conditions are: i) the current state of the mental state and the goal, ii) and iii) the target state that verifies if the desired condition was achieved or not.

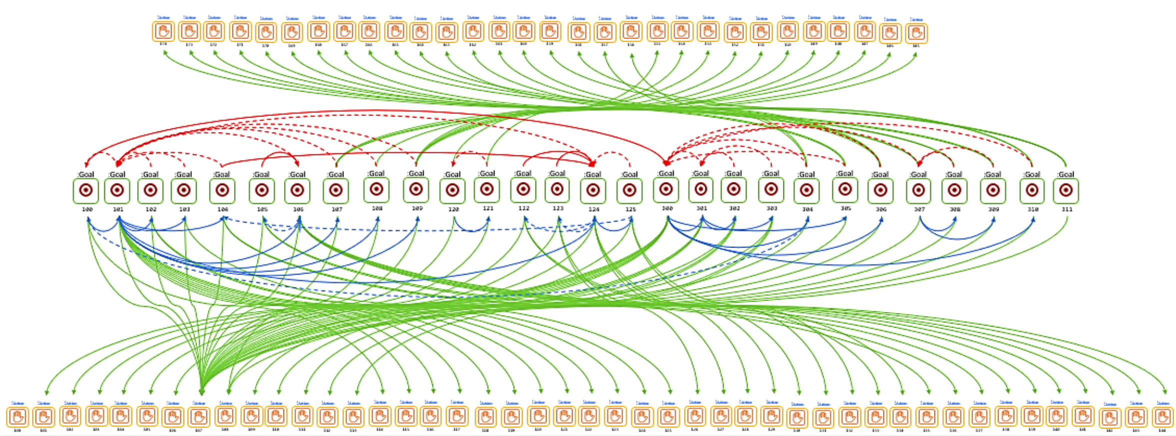
Figure 9, only presents actions that accomplish the function of completing the mental states and returning to the goal if the condition is accomplished. The goals are on the right side of the model and these indicate the mental states.

The Goals are achieved when the mental state to which it is related is completed. As well, the reasoning process of CARINA's object-level searches changes a problem from a set of initial states to a set of final states.

The following figure (see figure 10), presents all the Goals and Actions of the model expressed in M++.

Figure 10

Representation of the NGOMS-L model in M++



Note. Figure was adapted from: López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica, 18(2)*, 11-20. DOI: https://doi.org/10.25044/25392190.967.

6) Runnable Cognitive Model in Carina

The cognitive model for the Factoid-WH questions was created through an executable code in an open standard format file, called JSON, the code fragments are described in detail:

The mental states indicate to CARINA how to accomplish a specific task. For this, all the mental states which are part of the cognitive model contain an identifier for the system, a name, a type and an identifier of the cognitive model. Thus, the cognitive model starts with the mental states, which are the goals to be modified from a false state to a true state. The mental states then become the tasks that the cognitive model contains to accomplish the problem.

The cognitive model for the Factoid-WH questions was created through an executable code in an open standard file format, called JSON, the code fragments are described in detail (see Figure 11):

Figure 11

Mental State in cognitive model for factoid-Wh questions

```
[
  {
    "_id": "5becc375fa3c95fda65b7572",
    "name": "Translate User Response to SDG",
    "type": "CMEC",
    "ius": "5becc3751ff39189340f3274",
    "cm": "5becc3755bbc4254e5c674d8",
    "mentalStates": [
      {
        "name": "stimulus_from_environment_is_read",
        "state": false
      },
      {
        "name": "Translate User Response to SDG",
        "state": false
      },
```

Note. Figure was adapted from: López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica, 18(2),* 11-20. DOI: https://doi.org/10.25044/25392190.967.

Goals are required to achieve and change every mental state. In the goals are the reference, the mental state, the current state in which it is (false or true) and the description (see Figure 12).

Figure 12

Goals in cognitive model for factoid-Wh questions

```
{
  "goal": {
    "_id": "5becc3758cdc20aea06fbd62",
    "name": "Translate User Response to SDG",
    "currentValue": false,
    "targetValue": true,
    "description": "Translate User Response to SDG"
  }
```

Note. Figure was adapted from: López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica, 18(2),* 11-20. DOI: https://doi.org/10.25044/25392190.967.

The rules of production present a condition that must be met in order for the conclusions to be executed. It is necessary to specify in the condition the following aspects: i) the cognitive model with which the problem is solved and ii) the objective that is affected at the same time (see Figure 13).

Figure 13

Production Rules in cognitive model for factoid-Wh questions

```
],
"productionRules": [
  {
    "rules": {
      "_id": "5becc37552accf1704e02dfa",
      "condition": {
        "reasonState": {
          "cm": "5becc3755bbc4254e5c674d8",
          "goal": "5becc3758cdc20aea06fbd61"
        }
      },
```

Note. Figure was adapted from: López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica, 18(2)*, 11-20. DOI: https://doi.org/10.25044/25392190.967.

The conclusions have the actions, which are constituted by: a name, a module (indicates the origin of the function), a function identifier (indicates the action accomplished), the function identifier also specifies when the action is accomplished. Finally, all the actions and all the rules associated with this mental state are executed. Thus, when all the mental states are true, the problem is solved (see Figure 14).

Figure 14

Conclusion of Production Rules in cognitive model for factoid-Wh questions.


```
"conclusion": [
  {
    "action": "accomplishGoal",
    "module": "carinaModules",
    "idFS": "5becc3754579f6751b5c4341",
    "params":{
      "goal":"5becc3758cdc20aea06fbd62"
    },
    "accomplish": false
  },
  {
    "action": "returnWithGoalAccomplished",
    "module": "carinaModules",
    "idFS": "5becc3754579f6751b5c4341",
    "params":{
      "goal":"5becc3758cdc20aea06fbd62"
    },
    "accomplish": false
  }
]
```

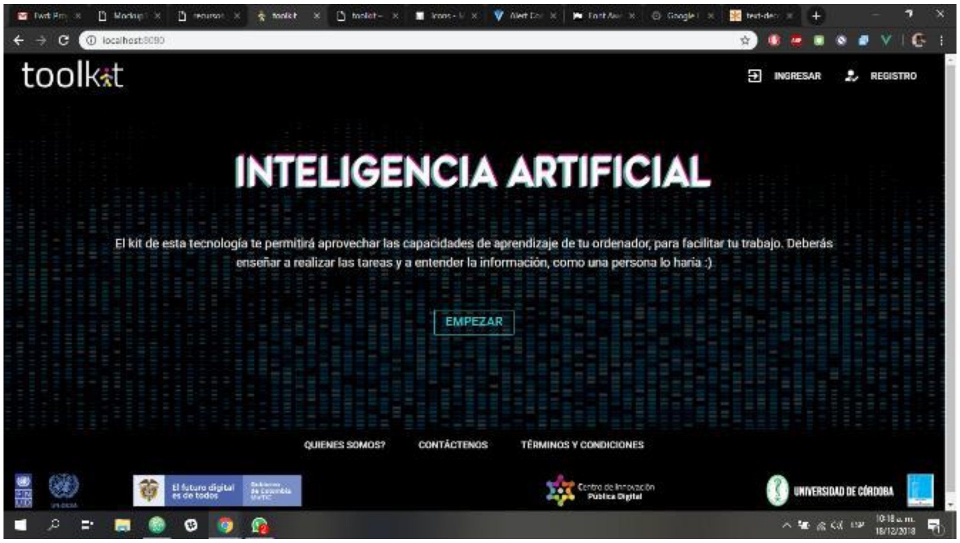

Note. Figure was adapted from: López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica, 18(2)*, 11-20. DOI: https://doi.org/10.25044/25392190.967.

7) *Testing and Maintenance of Cognitive Model*

The cognitive model developed for the elaboration of Factoid-WH questions was tested with a cognitive agent that answers the Factoid questions in Spanish. The results of the cognitive agent (TOOLKIT) are shown below. TOOLKIT is an agent created with Artificial Intelligence to answer factoid questions in a specific domain of knowledge (see Figures 15-16).

Figure 15

Login and registration in the Toolkit agent.



Note. Figure was adapted from: López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica, 18(2)*, 11-20. DOI: https://doi.org/10.25044/25392190.967.

Figure 16

Interface to create the factoid questions.

Note. Figure was adapted from: López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica, 18(2)*, 11-20. DOI: https://doi.org/10.25044/25392190.967.

Chapter III

Theoretical Framework

This chapter present the theoretical aspects of the analyzes categories that are part of this research project:

3.1. Cognitive Modeling

Cognitive modeling generates conceptually well-defined theories for a variety of purposes (Sun, 2009). Cognitive modeling research began in the 1960s with the symbolic model studies proposed by Simon y Newell (Newell & Simon, 2007). Simon and Newell intended to present extensive models capable of processing large data for using in Artificial Intelligence and simulating human cognitive processes (Sun, 2008a).

However, these proposals offered a limited view of human cognitive processes as the models could not be compared with human data (Sun, 2008a). In this sense, authors have proposed different cognitive symbolic models, based on different complex data structures that store knowledge such as scripts (Kronenfeld, 1978) or frames (Minsky, 1974).

Sun, (2008a); Borst & Anderson, (2015) & Olier, Gómez, & Caro, (2018) have presented computational cognitive models based on studies symbolic model. Olier, Gómez, & Caro, (2018), present a methodology for computational cognitive modeling which provides detailed descriptions of mechanisms and processes of cognition that underlies human behavior. The methodology of the modelling uses structure by seven steps, which are: i) *Selection of cognitive task*, ii) *Obtaining information for describing the cognitive task* iii) *Description of cognitive task in natural language*, iv) *Description of cognitive task in GOMS*, v) *Codification of cognitive model from GOMS to M++ language*, vi) *Execution of runnable cognitive model in CARINA* and vii) *Testing and Maintenance of Cognitive Model*. In addition, this cognitive modeling methodology is used in CARINA for the development of cognitive models. Thus, cognitive modeling is a research methodology derived from the cognitive sciences, which results in well-formulated theories defined in terms of computer programs (Strube, 2000). For to being a methodology, cognitive modeling is an instrument used to understand the process that underlies behavior (including, perception, emotion, motivation, etc.), comprehensible for studying cognition (Johns et al., 2018 & Prezenski et al., 2017).

Furthermore, cognitive modeling makes a specification of cognition in terms of algorithms and programs (Sun, 2009), using the computer sciences, to computationally represent some cognitive functions and, to create as well, computer models that can run on cognitive or metacognitive architectures (Olier et al., 2018). Cognitive modeling uses both for theoretical purposes, as a method to extend formal relationships of the human mind, and as a problem-solving tool applied to various domains of knowledge (Johns et al., 2018).

For example, Olier et al., (2018) propose the use of cognitive modeling in the domain of syntactic sentence analysis, to create computational cognitive models, which are to be implemented in an intelligent system. For this purpose, the elements required for the identification of the grammatical structures of a sentence must be specified, as well as the cognitive functions involved in the system (e.g., perception, recognition, categorization, and action).

Similarly, Pew & Mavor, (1998) propose the use of cognitive modeling in organizational human behavior domains, through the specification of the following steps: i) *to Develop task analysis and structure*, ii) *Establish model purposes*, iii) *Support focused modeling efforts*, iv) *Employ interdisciplinary teams*, v) *Benchmark*, vi) *Promote interoperability*, and vii) *Employ substantial resources and thus development cognitive models* in areas such as: attention, multitasking, memory and learning, human decision making, situation awareness, planning, behavior moderators, among others. That will allow more realism, understanding and application in intelligent cognitive systems for building and applying a large number of models of units that underlying distributed Artificial Intelligence, Psychology, Sociology and Organizational Sciences, among others (Pew & Mavor, 1998).

Thus, cognitive modeling can be applied in educational environments specifying computational models of child development describing the developmental processes (Sun, 2009). Through the use of neural networks, it is possible to see how evolution limits the emergence of a brain function during individual development (developmental psychology). Using neural networks and computational cognitive models it is possible to see the difference between development and learning, since the mechanisms of learning are different from a representation of knowledge in a given domain.

3.2. Cognitive Models

Cognitive models aim to answer how human beings act, through which psychological mechanisms, and through which knowledge processes and structures (Sun, 2008a). That is, cognitive models are studied to understand various aspects of cognition, attention and multitasking, judgment and choice in decision-making and skill acquisition in dynamic situations (Gonzalez & Gonzalez, 2002).

The application of cognitive models is evident in learning, memory, individual thinking, social interaction, and even intellectual skills (Sun, 2008a). According to Sun, (2008a) in cognitive sciences, models can be conceptual, computational or mathematical verbal. Computational models allow the detailing processes through algorithmic descriptions. Mathematical models allow to detail the relationships between variables through mathematical equations. Verbal conceptual models detail entities, relationships and processes in natural languages known as informal. Thus, the paradigms for the development of cognitive models are classified as: symbolic, hybrid, connectionist or dynamical models (Polk & Seifert, 2002). According to Sun, (2008a) these type of models can be defined as: Symbolic models are defined as representation of knowledge of a variety complex data capable of processing a large number of information, but limited for comparison with human data. Hybrid models are defined as the combination of neural networks and symbolic models, which can be used to model a large number of cognitive phenomena through diverse and expressive representations, and Connectionist or Dynamical Models which are defined as various models created in developmental psychology that address explanations of flexible behavior, real-time performance, adaptive behavior, broad knowledge base, dynamic behavior, knowledge integration, natural language, learning, development, evolution, and brain realization.

In this terms, this thesis presents a theoretical specification of computational cognitive models based on cognitive and metacognitive architectures (Olier et al., 2018 & Sun, 2008a). The reason for using computational cognitive models is because, according to Sun (2008a) allow flexibility and expressiveness since they offer several modeling techniques and methodologies and also allow applying cognitive theories (Barchini, 2005 & Sun, 2008a).

Computational cognitive models are developed to study different aspects of cognition, attention, multitasking, judgment and choice in decision-making and skill acquisition in dynamic situations, thus, a cognitive model is a simplified and detailed description of cognitive processes with the purpose of understanding or predicting a certain behavior (Kopp & Bergmann, 2017).

Cognitive models are characterized by being developed based on a cognitive architecture representing structural and functional elements commonly inspired by the human mind (Olier et al., 2018). However, this does not imply that cognitive models are only limited to their development through cognitive architectures. In fact, a variety of cognitive models can be created providing a detailed notion of cognitive or mental processes in computational terms that in turn allow for simulation-based testing and evaluation (Lieder & Griffiths, 2019).

Cognitive models can be applied in a variety of knowledge domains from individual cognitive tasks to specific behavioral predictions (Kopp & Bergmann, 2017).

For example, Lebiere, (1999) proposes a model for analysis of cognitive arithmetic elements, which allow study how certain child and adults resolve arithmetic problems (addition, subtraction, multiplication and division). For this example, was create an addition's cognitive model in the ACT-R cognitive architecture, presenting problems as "chunks" (number and operator) and production rules (set of steps to accomplish the cognitive task stored in the ACTR's long-term memory) was created.

Similarly, in the educational field, cognitive models are used to understand the organization and processing of people's information, which are identified through well-structured problem-solving proposals, for example, Jerónimo et al., (2017) use verbal protocols as a method for the application analysis of the metacognitive strategies in high school students, through the Towers of Hanoi. Thus, this will make possible to know what are the processes underlying an individual's mind, to analyze them and then write them down in computational terms, which will notably facilitate human cognition by providing thinking tools, and thus create a variety of intelligent cognitive systems (Kopp & Bergmann, 2017). In these terms a relationship is created between cognitive psychology, cognitive science and artificial intelligence (Sun, 2008b).

3.3. Cognitive Architectures

The design and development of Cognitive Architectures is a wide and active area of research in Cognitive Science, Artificial Intelligence and the areas of Computational Neuroscience, Cognitive Robotics, and Computational Cognitive Systems (Lieto et al., 2018). Cognitive architectures have been developed with the following purposes: i) to capture, the mechanisms of human cognition, underlying the functions of reasoning, control, learning, memory, adaptivity, perception and action (at the computational level), ii) to develop cognitive capabilities through ontogeny over extended periods of time and iii) to reach human level intelligence (General Artificial Intelligence) creating artificial artifacts (Lieto et al., 2018). In the literature there are cognitive architectures as ACT-R, CLARION, MIDCA, SOAR etc., which have developed agents based on such infrastructures and have been widely tested in several cognitive tasks involving reasoning, learning, perception, action execution, selective attention, recognition etc., (Anderson, 1996 & Sun & Naveh, 2004).

Cognitive Architectures refers both abstract models of cognition, in natural and artificial agents, and the software instantiations as well as models which are then used in the field of Artificial Intelligence. The main function of Cognitive Architectures in Artificial Intelligence is that one of allowing the realization of artificial systems able to showing intelligent behavior in a general setting through a comprehensive analogy with the constitutive and developmental functioning and mechanisms underlying human cognition (Lieto et al., 2018). In this terms, a cognitive architecture is a control framework that explains psychological aspects in animals and humans based on scientific theories (Lieder & Griffiths, 2019 & Ritter et al., 2019).

According to Sun, (2009) a cognitive architecture is necessary because it provides a comprehensive framework for broad analysis across multiple domains and various cognitive functions. In addition, theories may be inspired by available scientific data, (i.e. psychological or biological data), philosophical data (thoughts and arguments), or computationally oriented hypotheses. Thus, a cognitive architecture offers scaffolding structures and symbolizes fundamental theories (Sun, 2008a) .

The cognitive models used in a cognitive architecture allow interpreting learning data, through the interaction of cognitive processes, which is more efficient, than studying each process in a disarticulated way, since it is more specific and detailed, predictive and facilitates a more rigorous evaluation (Kopp & Bergmann, 2017). For example, data from a variety of task domains such as: artificial grammar learning tasks, process control tasks, serial reaction time tasks (Proctor & Capaldi, 2012), as well as some complex task domains as Towers of Hanoi (Sun, 2008a), thus enabling for a theoretical integration and explanation of the cognitive and metacognitive processes involved.

3.4. Metacognitive Architectures

Due to progressively complex Artificial Intelligence, agents that execute decisions based on multiple variables the metacognitive architectures are developed (Cox et al., 2011).

A metacognitive architecture offers a specific framework for detailed modeling of mechanisms for an Artificial Intelligence agent's high-level reasoning about itself, by specifying essential structures, divisions of modules, relations among modules, and a variety of other essential aspects (Caro et al., 2019). Metacognitive architectures are different from cognitive architectures because the agent itself is the referent of the cognitive processing but sharing the formalisms for representing knowledge, memories for storing this domain content, and processes that utilize and acquire the knowledge (Caro et al., 2019).

In this sense, with a metacognitive architecture it is possible the creation of artificial intelligent systems (Caro et al., 2019). In this sense, with a metacognitive architecture it is possible to create in intelligent systems capacities to be autonomous and be able to adjust dynamically, without any or with a limited human intervention, identifying anomalies, analyzing alternatives for self-adaptation or generating new goals that allow to completing different cognitive tasks applied in several knowledge domains (Paisner et al., 2014; Caro et al., 2019 & Gerasimou et al., 2019).

Currently, with CARINA it is possible to create intelligent tutoring systems, for example, Caro et al., (2019) in their article propose FUNPRO as an intelligent tutoring system with the aims of making detection and recovery of reasoning failures. The real-world task that was modeled was the generation of an instructional plan of the instructional design domain, where the personalization of a course lesson for a student is achieved, for this the instructional designer must select the activities and learning resources according to the student's learning style.

According to Sun, (2009) tutorial systems have been developed in the ACT-R cognitive architecture, these systems were based on the analysis of production rules that were required to complete coursework competency the mathematics and computers domains. The modeling process allowed the interpretation of the student's behavior and, also the management of the student's interactions in the tutorial. Therefore, such tutoring systems are based on the validity of the cognitive model and the validity of the attributions that the model tracking process makes about student learning. Tutoring systems have been used to provide instruction to over 100,000 students so far. Thus, demonstrating the practical utility of computational cognitive modeling both for the study of the processes underlying human mind and for the creation of intelligent cognitive systems based on their own monitoring processes.

3.5. Knowledge Representation

The knowledge representation in intelligent systems refers to the process of articulating, structuring, and critically evaluating a model of some domain (Ford & Bradshaw, 1993). In the knowledge acquisition, the knowledge engineer and expert collaborate in constructing an explicit model of problem solving in a specific domain. This external model is largely based on the expert's internal mental "model" of the domain. The knowledge engineer's role involves developing important tools and methods to support experts in their labors to express, elaborate, and improve their models of the domain (Ford & Bradshaw, 1993).

Modeling is particularly purposive, this means to be involved in modeling is necessarily to be engaged in using the model (in some particular setting) for particular reasons that determine together what should be modeled, how to model it, and what can be ignored (Ford & Bradshaw, 1993).

Research efforts began in1990s as advances in implementation mechanisms were in the 1980s. Results of knowledge-acquisition research and practice have already been felt in areas as collection of papers found in this issue. It is that knowledge acquisition is a modeling process, not merely an exercise in "expertise transfer" or "knowledge extraction (Aamodt, 1995). Thus, in the literature have developed various studies that apply Semantic Knowledge Representation in Artificial Intelligent Systems. For example Peters & Shrobe,(2003), present a semantic network with the purpose to represent knowledge by constructing intelligent spaces to encapsulate rooms, users, groups, roles and other type of information, as an important design tool. Using the semantic network structures to save meetings in the order they occur, connecting together the principal meeting topics with others contributors and attendees. The semantic networks allow facility in the addition, the change of information can be done in a simple way. The network searching language allows in this way a user can define a segment of the network they are looking for, and to obtain receive personalized notifications.

According to Miller, (1995) other use of the knowledge modeling tool is WordNet, defined as a resource to detect essential characteristics of lexical and semantic connections, which can be implemented in Machine Learning experiments. Thus, Semantic Knowledge Representation in Artificial Intelligent Systems is developed both in symbolic and intelligent robotic intelligence control system architectures and in educational systems for the creation of production systems, semantic networks, automatic learning and sub-symbolic processing to perform real-time control (Avery et al., 2006).

3.6. Denotational Mathematics

Denotational mathematic is a category of expressive mathematical structure to belong concept algebra (Wang, 2008a). Denotational mathematics is a category of expressive mathematical elements that shares with high-level mathematical entities numbers and sets, such as abstract objects, concepts, knowledge, behavioral information, complex relations, processes, intelligence, and systems (Wang, 2008b). In denotational mathematics, a concept is formally modeled as an abstract and dynamic mathematical structure that captures relations, objects and attributes (Wang, 2008a). Thus, the specifications in using denotational mathematics of process vinculated to the concept algebra, system algebra, and real-rime process algebra, which can be implemented in computational intelligence, software engineering, cognitive informatics and knowledge engineering. through examples in domains of iterative and recursive systems architectures and various behaviors, because it is necessary to formally define and operate software and instructional behaviors in terms of operational logic, timing, and memory manipulation (Wang, 2008b).

Chapter IV

The Metacognitive Architecture CARINA

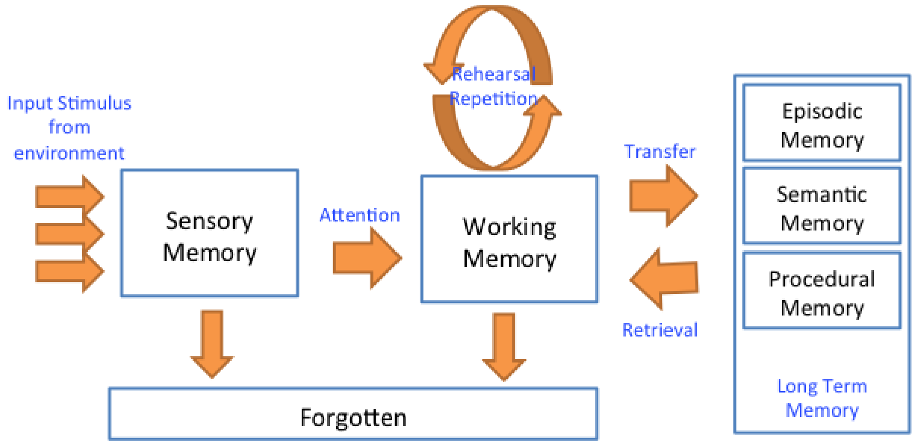
CARINA is a metacognitive architecture for the creation of intelligent artificial agents, derived from the Metacognitive Metamodel MISM. With CARINA, it is possible to create cognitive systems that solve real world tasks. For the creation of these tasks, a mechanism of knowledge acquisition must be specified, where CARINA has all the elements that intervene in the problem to give it a solution. Thanks to metacognition, the level of autonomy of intelligent systems has increased (Caro et al., 2019). However, the design of systems with metacognitive abilities is a difficult task due to the number of processes and theories involved.

CARINA adopts a functional approach to the philosophy of mind, incorporating self-regulation and metamemory based on meta-cognitive mechanisms of introspective monitoring and *meta-level* control (Caro et al., 2018). According to Bechtel, (2012) the mechanisms are entities and activities in which static and dynamic aspects are involved, in this sense the entities in CARINA are known as "*cognitive elements*" (Caro et al., 2018). According to Olier et al., (2018) CARINA is constituted by three types of cognitive elements: *structural elements*, *functional elements* and *basic elements*. The structural elements incorporate the functional and basic elements, such as the cognitive level. Functional elements are tasks used to reason and make decisions. The basic elements are constituted by the processes of reasoning and meta-reasoning that interact with each other. As the main functional elements of CARINA are the task of reasoning and the task of meta-reasoning. Reasoning tasks (RT) are actions that facilitate the processing (transformation, reduction, elaboration, storage and retrieval) of information using knowledge and applying decisions to achieve the objectives of the system. A meta-reason task can perform two functions: explain failures in a reasoning task or select from "cognitive algorithms" the appropriate one to execute the reasoning (Caro et al., 2018).

The Memory System in CARINA (see Figure 17), is constitute by Sensory Memory, Working Memory and Long-Term Memory (Atkinson & Shiffrin, 1968 & Glass, 2008).

Figure 17

The memory system in CARINA



Note: Figure was adapted from: Caro, et al. (2018). Introduction to the CARINA metacognitive architecture [figure]. Available: https://ieeexplore.ieee.org/abstract/document/8482051.

According to the previous figure (Figure 17), the Sensory Memory is a transitory buffer which temporarily keeps information that has not been directly saved (Scheutz, 2001) for to be used later (Caro et al., 2018).

CARINA's *Working Memory* is integrated by various "*Basic Cognitive Processing Units*" (BCPU) as well as a selective attention subsystem. Thus, a *BCPU* is defined as a <*buffer*> that includes the information that interacts between the different cognitive processes that intervene in a *cognitive loop* in CARINA (Caro et al., 2018). In addition, *working memory* is a memory space used to temporarily reserve information in the process of executing a set of cognitive tasks, for example: *perception*, *reasoning*, *planning*, etc. (Schmid et al., 2011& Sun, 2009).

According to Sun, (2007) *long-term memory* allows information to be stored for a long period of time. Thus, in CARINA, *long-term memory* collects the saved information in a semantic way. In this metacognitive architecture, *declarative memory* is a subset of *long-term memory* and is classified into *episodic memory* and *semantic memory*. In these terms, *procedural memory* is part of *long-term memory*, but is not considered declarative. Below, are the main elements of these types of memory in CARINA.

First, *Episodic Memory* has detailed cases in the form of events using a *Case Based Reasoning* (CBR) system. An event is defined as sensory-perceptual information shown in the form of perception processes, motor commands and internal data structures (Murray, 1985 & Tulving & et al., 1972) made on the basis of recent intelligent agent experience (Sun et al., 2006). Thus, the *episodic memory* facilitates the knowledge stored in the semantic memory (Turing, 1950 & Unsworth, 2010).

Second, *semantic memory* saves the knowledge achieved from the intelligent agent's world (VanPatten & Williams, 2014). The main characteristic of semantic knowledge is that it is not contextualized in time and space. Additionally, CARINA's *semantic memory* presents: simulation of the activation process of the frontal and temporal cortexes, implementation through ontologies in the form of "*mental thesaurus*" using the approaches of Tulving et al.,, (1972) & Van Patten & Williams, (2014).

Third, *procedural memory* stores the actions and behaviors of the intelligent agent, with the purpose of classifying the sequences, categories, rules and routes used in cognitive processes. In this sense, procedural memory is a non-declarative memory, which displays unconscious learning processes and keeps a specific set of information about "*how to do it*".

CARINA's memory allows the use of a hybrid model based on rules to support sequences that keep a record of motor and behavioral skills of the intelligent agent.

In CARINA, the process of execution of a cognitive model, begins at the moment of loading it into the attention system, achieving each of the cognitive functions that are developed at the *object- level*. When the cognitive model has achieved all the Goals that have planned without reasoning failures, the cognitive model is saved in CARINA's semantic memory as a belief.

CARINA symbolizes the problems to be solved using the Mental States. A Mental State can be defined as a representation that is capable of elaborating a plan of execution of tasks to achieve a goal. The Mental State acts according to environmental events (Isern et al., 2008). Thus, these mental states are stored in its working memory structure called "*world model*". With the purpose of achieving the Mental States, CARINA develops a set of Goals stored in its motivation system.

Goals are defined as objectives that the intelligent system must complete in order to complete a task or process (Caro et al., 2014). Goals allow the accomplishment of the Mental States which are in the *working memory* and thus change them, using a plan made up of actions created in its procedural memory.

Actions are a class of situations (seen in an intuitive way), which are derived from the activity of some agent or agents in the fulfilment of some Goal (Georgeff, 1988). Additionally, a Production Rule is a statement of the programming logic that specifies the execution of one or more actions when a condition is achieved (Boley et al., 2010). Thus, Production Rules are constitute by Procedural Knowledge in CARINA (Jerónimo et al., 2018). This research presents the elements that structure a cognitive model in CARINA, in a formal, semantic and computational way to be implemented in CARINA.

Chapter V

Cognitive Models for the Metacognitive Architecture CARINA

The methodological phases used in this research for the development of cognitive models are presented. In this section the formal, semantic and computationally cognitive models for the CARINA metacognitive architecture is represented bellow:

5.1. Formal Representation of Cognitive Models in CARINA

This section presents a formal characterization of a cognitive model in CARINA. A cognitive model has declarative knowledge, which implies facts, and procedural knowledge, as well, which implies rules of reasoning. Furthermore, a cognitive model also facilitates modeling about how to reason, for this it is also necessary to model control over when to reason about what (Muller & Heuvelink, 2008). This representation was made using denotational mathematics proposed by (Wang, 2008b).

A cognitive model (CM) in the Metacognitive Architecture CARINA has:

$$CM \triangleq (P, G, S, MS, PK, SK) \quad (1)$$

Where:

$\boldsymbol{p}$ is the problem to be solved using CARINA.

$\boldsymbol{g} \in \boldsymbol{G}$, where $\boldsymbol{g}$ is a goal. A goal is an objective pursued by the system. The formulations of goals therefore refer to the properties that are intended to be ensured; they are optional as opposed to indicative statements, and are delimited by the subject matter, (Van Lamsweerde, 2001). In this sense, goals are objectives that drive a task or process (Caro et al., 2014).

$\boldsymbol{s} \in \boldsymbol{S}$ where $\boldsymbol{s}$ is a Sensor. A sensor has the role of monitoring the profiles of cognitive tasks with the purpose of identifying irregularities that may generates reasoning failures produced by the cognitive task (Caro et al., 2014).

$\boldsymbol{m} \in \boldsymbol{MS}$, where $\boldsymbol{m}$ is a mental state. Mental state can be defined as variables Booleans (that may be true or false). In this sense, a mental state is a state of mind which an agent could be found.

$\boldsymbol{PK}$, represent the Procedural Knowledge system requires to perform the cognitive task. Production rules structure the Procedural Knowledge in CARINA.

Let:

$\boldsymbol{r} \in \boldsymbol{R},\ \boldsymbol{r}$ is a production rule, with

$\boldsymbol{r}\ \triangleq (condition, conclusion)$ (2)

With:

$condition\ \triangleq (\boldsymbol{AS}, \boldsymbol{C})$(3)

With If

$\boldsymbol{AS}$ is a set of variables that compose the values used to active the rule $(\boldsymbol{r})$, i.e., a rule $(\boldsymbol{r})$ is actives if there is a complete correspond with $\boldsymbol{AS}$ variables

$\boldsymbol{c}\ \in \boldsymbol{C}$ and $\boldsymbol{c}\ \neq \boldsymbol{C}$ and $\boldsymbol{c}\ \neq \boldsymbol{AS}$

$\boldsymbol{c}$ denotes a specific constraint made by cognitive designer.

A constraint $(\boldsymbol{c})$ is a specific condition to achieve, in some cases, conditions could be empty $(\boldsymbol{c}) = \{\}$

$\boldsymbol{C}$ are the conclusions. Conclusions are the actions that underlies when the rule $(\boldsymbol{r})$ is active

In this sense $\boldsymbol{a}\ \in \boldsymbol{A}$, where $\boldsymbol{a}$ is an action.

$\boldsymbol{SK}$ is a Semantic Knowledge. The Semantic Knowledge is the required knowledge to achieve a cognitive task. Semantic Knowledge is a set of beliefs saved in the CARINA’s Semantic Memory which can be recovered during the reasoning process.

Let:

$\boldsymbol{f} \in \mathbf{F}$ with $\boldsymbol{f}$ is a Field.

Where:

$\boldsymbol{FD} \subset \boldsymbol{F} \wedge \boldsymbol{FB} \subset \boldsymbol{F}$

And **F** is a set of characteristics of a $\boldsymbol{SMU}$

$\boldsymbol{SMU}$ is the Semantic Memory Unit, and is the bases of semantic memory, which is composed by:

$\boldsymbol{FD}$ are basic set of data type specified as string or integer.

$\boldsymbol{FB}$ are fields that select the Beliefs which are in the semantic memory.

And $\boldsymbol{\beta}$ are the Beliefs. The beliefs are specific epistemic atoms to denotes declarative knowledge (Pezzulo & Calvi, 2004). Rao & Georgeff, (1991) specify the Beliefs receive and store information created by the environment. According to Caro et al., (2014), beliefs in CARINA are the elements of declarative knowledge, i.e., information inspired on facts or notions that are saved. In this context beliefs compose the minimum unit that constitute semantic memory in CARINA.

Below, a detailed description of the main elements of a CARINA's Cognitive Model is presented through of an illustrative example of a simple real-world cognitive task. This real-world task comprises the cognitive processes and mental representation of numbers and arithmetic facts of addition problems. This cognitive task is based on the basic theories of computing proposed by Lebiere, (1999) which describe strategies of empirical phenomena in the domain of cognitive arithmetic in which children, and some adults, resolve the response to an arithmetic operation. According to this author, a specific arithmetic problem, (for example, the addition of 3 + 4) allow to children to select and to count (possibly 4, 5, 6, and 7) and thus, give the answer. In this way, in CARINA, the cognitive model that model this cognitive task specifies the problem to be solved, in this case (the addition of two numbers). This problem will be structured by three elements: $addendum\_1$, $addendum$ 2 and sum. In this sense, these necessary elements to solve the problem are provided to CARINA through sensors, in which the information is obtained from long-term memory (see Table 2).

Table 2

Specification of a Problem in CARINA

```
"problems": {
"type": "addition-single-col-pro",
"addendum_1": "empty",
"addendum_2": "empty",
"sum": "empty"
 }
```

When CARINA reads a problem, a problem space arises (related to the cognitive task to be solved). Thus, for each problem space, CARINA incorporates a set of mental states in order to obtain updated information on the current state of the information required to solve the problem in process (see Table 3).

Table 3

Mental States in CARINA

```
"mentalStates": {
"problem_is_initialized”: false,
"addendum_1_is_read": false,
"addendum_2_is_read": false,
"sum_is_calculated": false,
"sum_fact_is_saved": false,
"sum_is_displayed”: false,
"problem_is_done": false
      }
```

CARINA can produce goals through the mental states in the problem space. That is, each goal is used to achieve changes in one or some mental states. Then, a Goal shows: i) the current value of the current mental state and ii) the expected value after the execution of the goal method.

The following two code fragments show an example of the knowledge structure that stores a Goal in CARINA. Given a Goal for the addition in this example, goal $g701$ purposes to change the $addendum\_1\_is\_read$ mental state. When the goal is achieved the internal state changes from $False$ to $True$. In CARINA, the set of goals that the system generates within the current problem space is in the $Motivational\ System$ (see Table 4).

Table 4

Goals in CARINA's motivational system

```
“g701”: {
"mental_state": "addendum_1_is_read",
"current_value": false,
"target_value": true
}
```

CARINA's procedural memory is structured by a set of production rules and an inventory of available cognitive functions constituting a production rule. In the following code fragment production rules are showed (see Table 5):

Table 5

Production Rules in CARINA


```
"rule702": {
"condition": {
"attention_system": {
"problem": "p2",
"mental_state": "addendum_2_is_read",
"goal": "g702",
"sensor": "sensor1"
},
"constraint": {}
},
"conclusion": {
"action1": {
"name": "readNumberFromPerceptor",
"complete": false
},
"action3”: {
"name": "nextGoal",
"arg": {
"goal": "g703"
},
"complete": false
}
}
}
```

Line 1 has the name of the rule, defined as a unique identifier. The condition section of the rule encoded from lines 2 to 8. The condition has the variables of the attention system that are necessary for the rule to be active in the context of the current problem.

The conclusion section of the rule groups lines 11 to 23. This section describes the order in which the basic cognitive functions of CARINA must be executed. Lines 12 and 16 designate the calls to the two functions $readNumberFromPerceptor$ and $nextGoal$.

Additionally, cognitive model has semantic knowledge in the form of Beliefs. In the cognitive model, the Beliefs requires for CARINA to solve problems in a knowledge domain are specified. In the following fragment of coding, the basic structure of a belief that represents the $number1$ in the problem of the sum of two numbers is described (see Table 6).

Table 6

Basic structure of a Belief in CARINA

```
"one": {
    "typeSMU ": "number",
    "has": {
        "type": "integer",
        "value": "1",
    }
}
```

5.1.1. Comparison with other Cognitive Architectures

This section describes the similarities and differences between the implementation of cognitive models in CARINA and ACT-R. ACT-R were taken for comparing by its popularity and many published examples in cognitive models. Thus, the comparison made, used the example propose by Lebiere, (1999) in his thesis.

5.1.2. Similarities

In ACT-R, declarative knowledge is specified by using elements, known as “chunks”, which are simply collections of “$key - value\ pairs$”. In CARINA, beliefs are used in the similarity of chunks. The difference is the internal structure of each one, since the beliefs of CARINA have two sections with two relationships: “$IS - A$” and “HAS”.

The procedural knowledge in CARINA and ACT-R is specified as production rules, although they do not have the same structure, they function similarly.

5.1.3. Differences

The specification of a problem in CARINA has a particular specific structure, i.e., in ACT-R the problem is expressed using the condition of one or more production rules, as shown in the example below (see Table 7).

Table 7

Specification of a Problem in ACT-R

```
(P initialize-addition
=goal>
ISA      add
arg1 =num1
arg2 =num2
sum    nil
==>
```

CARINA additionally uses production rules that can have specialized algorithms in a more detailed form which are activated through the production rules.

This “Formal Specification of Cognitive Models in CARINA” was published in: 2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC) (pp. 614-619). In addition, the results of the short paper were presented at the 17th IEEE International Conference on Cognitive Informatics & Cognitive Computing in Berkeley, California and II Workshop 2018: Metacognition Seminar in Monteria - Colombia. (see Appendixes A, B & C).

5.2. Semantic Representation of Cognitive Models in CARINA

This section presents a semantic representation of a cognitive model in CARINA.

The representation of semantic knowledge is defined as a way to model and specify knowledge using tools that represent notions or concepts (Ghasemzadeh, 2010) as well as formal symbols in a collection of propositions. The representation of semantic knowledge is the area of Artificial Intelligence that aims to study how knowledge can be represented symbolically and in an automated way through reasoning programs (Levesque, 1986).

In this sense, the Semantic Representation of Knowledge is a method of modeling and specifying knowledge that uses tools to represent elements such as: formal notions, concepts and symbols and a set of propositions (Ghasemzadeh, 2010).

Thus, there are several ways of specifying semantic knowledge such as rules, tables, frames (Tanaka et al., 1995), trees decision and paradigms, (Kwasnik, 1999), ontologies (Madera-Doval et al., 2018), metadata (Matta et al., 1998), agents (O’Leary, 1998), semantic networks (Peters & Shrobe, 2003), chunks (Anderson, 1996), Graphics (Arevalillo-Herráez et al., 2013) and neural networks (Zhou et al., 2016) among others.

Authors have proposed research to represent semantic knowledge through the use of cognitive architectures. For example, ACT-R represents semantic knowledge using units called "chunks" (Anderson, 1996). SOAR is based on the ACT-R form of knowledge representation, but each "chunk" groups a set of pieces of environmental information into a single unit (Gobet et al., 2001). CLARION uses several representations according to the type of knowledge used, which means that the explicit objective knowledge is symbolic, while implicit procedural knowledge is sub-symbolic (Kotseruba et al., 2016).

5.2.1. Semantic Knowledge Representation of a Cognitive Model in CARINA

The metacognitive architecture CARINA executes cognitive models in its *object-level* through its working memory. Thus, cognitive models must be updated each time a new cognitive task is specified. Currently, CARINA has not mechanisms that allow save these cognitive models it in its semantic memory. In these terms, it is necessary to semantically represent a cognitive model with the purpose of retrieving them anytime when is necessary. Thus, these cognitive models will can be executed after without directing supervision of the cognitive designer.

In this chapter, the semantic representation of a cognitive model implemented in the Metacognitive architecture CARINA is presented. A symbolical representation facilitates the manipulation of any computational structure (Lebiere, 1999). This semantic representation uses beliefs for storing cognitive models in the Semantic Memory of CARINA. Facilitating, the future design of learning processes of the architecture.

The semantic representation of a cognitive model represents behaviors of each element of the model. In the Figure 18, the semantic representation increases the representation the aspects which structured a cognitive model in CARINA such as:

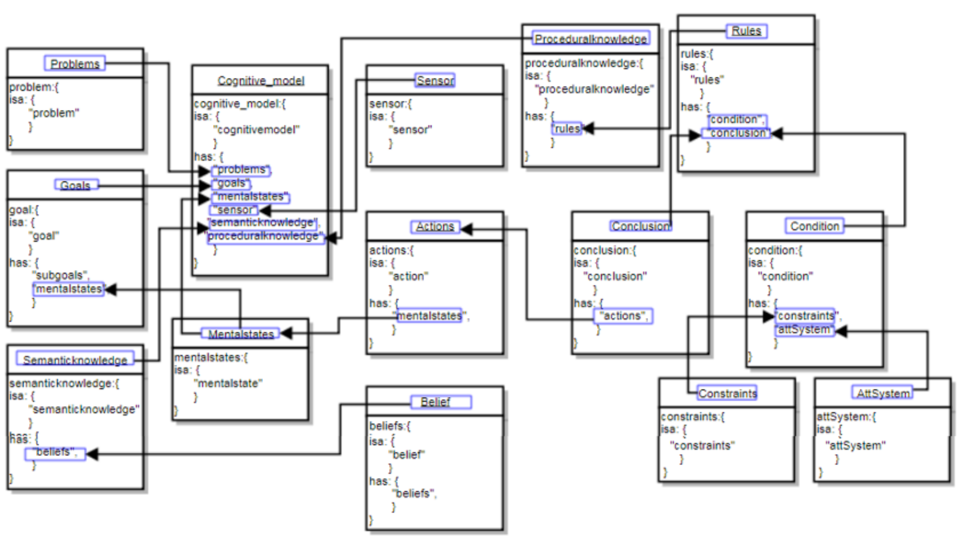
Problem which contains a *Mental States* generated on the space problem.

A *Goal* contains details both the current value and the expected value of *Mental States* to which it is associated.

Additionally, a *Goal* has *sub-goals* that allow its achievement. Also, CARINA executes a set of *Rules* according to *Goals* that it has and also executes some conditions that will allow achieving a series of *Actions* for the accomplishment of Mental *States* placed in its *Attention System*. Thus, CARINA is constituted by a *Metacognitive Sensor* that permanently monitors *reasoning traces* in CARINA *self-model*, which contains cognitive functions profiles that have been runnable in its *object-level*. In this sense the reasoning traces are composed by *Goals*, *Mental States* and *Actions* of the cognitive task that solves the identified problem about *Attention System*.

Figure 18

Semantic representation of a cognitive model in CARINA



Note. Figure was adapted from: Barrera, M.P., Jerónimo, A, J., Caro, M. F. & Gómez, A.A. (2020). Semantic and Formal Representation of a Cognitive Model of Metacognitive Architecture CARINA [figure]. In process of publication.

5.2.2. Formal Specification of Semantic Memory Units (SMU) in CARINA

Cognitive agents must have the capability to represent knowledge, learn and reason which continue as research challenges in the area of agent world (Nwana & Ndumu, 1999).Thus, all the knowledge got by CARINA is saved in the semantic memory as beliefs. In this terms, the nodes detected in the beliefs network related to the semantic memory units (SMU), and connections are linked with the relations between these units (Shi et al., 1997). Table 8 (see Table 8) presents the network of beliefs related to Semantic Memory Units (SMU).

With,

$\boldsymbol{\beta} =< \boldsymbol{ISA}, \boldsymbol{HAS} >$ (4)

Where,

Table 8

Beliefs network related to the Semantic Memory Units (SMU)

β ∈ β and β is a set of SMU,
With:
ISA is a set of fields, where f ∈ IAS ^ f ∈ β ^ ISA ≠ {}
HAS is a set of fields, where f ∈ HAS ^ f ∈ β ^ f is a basic data type

In formula (4) $\boldsymbol{\beta}$ is a $\boldsymbol{SMU}$ which can be defined as the minimum unit of information included in the semantic memory of a CARINA-based cognitive agent. Shi et al., (1997), propose that a Semantic unit are components that are removed from knowledge after it is "computed'. The set of fields **C** indicate relationships of a hierarchical type between the $\boldsymbol{SMU}$. A $\boldsymbol{SMU}$ may contain one or several $\boldsymbol{ISA}$ fields; in addition, it is necessary the identification the context of each $\boldsymbol{SMU}$ for the errors of the semantic relations within the declarative memory. The set of $\boldsymbol{HAS}$ fields may specify addition relationships (i.e. is-part-of links) or characterize field relationships (qualities and characteristics of a concept).

Beliefs can be defined according to Pezzulo & Calvi, (2004) as explicit epistemic atoms for specification of declarative knowledge.

CARINA, denotates the basic units of representation of knowledge in form of beliefs. In this sense, the beliefs are elements of declarative knowledge about facts, concepts or notions from the environment and from same the cognitive agent (Apt et al., 1988). Thus, beliefs represent the minimum unit of semantic memory.

In this sense, each Belief is structured into two essential parts: i) $\boldsymbol{ISA}$, which indicates the category where belongs to this belief (and that actives another belief that is saved in the semantic memory) and ii) $\boldsymbol{HAS}$ which indicates the other fields that describe a belief in particular.

The next fragment of coding the basic structure of a belief that represents the $number\_1$ in the problem of the addition of two is described (see Table 9).

Table 9

Basic structure of a Belief in CARINA

```
"one": {
    "isa": {
    "number"
    }
    "has": {
    "type": "integer",
    "value":"1",
    }
    }
```

This "Semantic and Formal Representation of a Cognitive Model in the CARINA Metacognitive Architecture" was accepted as a chapter of a book for publication in the Encyclopedia of Information Science and Technology, 5th edition of IGI GLOBAL, and is in process of publication (see Appendixes D and E). In addition, submitting the article to IGI GLOBAL allowed me to be a reviewer of two book chapters before their publication in IGI GLOBAL (see Appendix F). I also evaluated a verbal protocol which was used as an instrument for research in Master's studies at the University of Medellin, Colombia (see Appendix G).

5.3. Computational Representation of Cognitive Models for the CARINA Metacognitive Architecture.

CARINA uses files with open standard data interchange format to receive information about world. For this reason, the computational representation of cognitive models in CARINA is presented using this type of files. Below, this computational representation in JSON format is described (see Figure 19).

Figure 19

Computational representation in JSON format in CARINA

```json
{
  "name": " ",
  "type": " ",
  "problems": {
      "type": " "
  },
  "mentalStates": {
  },
  "goals": {
    " ":{
      "mental_state":"empty",
      "current_value":false,
      "target_value":true
    }
  },
  "productionRules": {
    "rule": {
      "condition": {
        "attention_system": {
          "problem": " ",
          "mental_state": "empty",
          "goal": " ",
          "state": "empty",
          "sensor": "empty"
        },
        "constraint": {}
      },
      "conclusion": {
        "action1": {
          "name": "accomplishGoal",
          "complete": false
        }
      }
    }
  },
  "beliefs":{
    " ": {
      "typeSMU":" ",
      "have":{
        "type": " ",
        "value": 0
      }
    }
  }
}
```

A JSON file which represents an executable cognitive model that will be able to be read by CARINA's working memory is structured by the following elements:

`name`: it is the name assigned to the knowledge domain to be resolved in CARINA

`type`: it is the category used by the cognitive designer to classify the cognitive model will be created.

`problem`: are the general elements necessary to resolve the problem, which is structured by: a `“type” ()` and the different variables where calculations will be saved, and are initialized in “`empty`”

`mental States:` Mental state can be defined as variables Booleans (that may be true or false). In this sense, a mental state is a state of mind which an agent could be found, which will be initialized in “`false`”

`goals`: A goal is an objective pursued by the system. The formulations of goals therefore refer to the properties that are intended to be ensured, i.e., goals are objectives that drive a task or process, which are constitute by: “`mental_state_name`” (mental state’s name) which will be initialized in “`empty`”, second, `"current_value":false`, defined as the value which represents the current state of the mental state, it is initialized by default in “`false`”; and a `"target_value":true`, defines as the value which must be achieved by the system after to execute the cognitive model and generally is initialized by default in “true”.

`production rules:` Production rules structure the Procedural Knowledge in CARINA, which are constitute by: `"rule_name"` (rule’s name), which contains: `{ "condition", "attention_system", "problem": “problem_name" , "mental_state": "empty"` (initialized in “empty”), `"goal", "state": "empty"` (initialized in “empty”), `"sensor": "empty"` (initialized in “empty”), `"constraint":` (constraint could be stay in empty), `"conclusion":` which contains:

“`action1"`, `"name_action"` (action’s name), `"complete":` (is the state of action, initialized in “false”)

`beliefs:` Beliefs in CARINA are the elements of Declarative Knowledge, i.e., information inspired on facts or notions that are saved. In this context beliefs compose the minimum unit that constitute Semantic Memory in CARINA. Which contain: elements' name and `"typeSMU"`: (knowledge's type, stored in semantic memory) and has `"type"`, which is the variables' name and `"value"`, which change if is a different special character or element for save the beliefs.

5.4. MetaThink Version 2.0

In this stage, a framework functional prototype for the creation of executable cognitive models in CARINA is showed.

Cognitive models are used to specify each of the cognitive and metacognitive concepts that CARINA uses to solve problems. Currently, cognitive models are developed manually and it turns out to be a time-consuming task for cognitive designers who, besides understanding the program logic, must also understand the elements that integrate a given cognitive function of the human mind and express it in computational terms to be executed. For this reason, MetaThink is created with the fundamental objective of making rapid and exploratory prototypes of metacognitive systems using this tool. In this sense, it is also necessary to create a software that allows the development of cognitive models, which has been called: MetaThink version 2.0.

MetaThink version 2.0 is a scientific software for the creation of cognitive models in a visual way to be executed in the CARINA metacognitive architecture. A cognitive model in CARINA is a computational description of the elements that integrate a cognitive task.

In these terms, the software aims at turning the user design into a graphic way what was previously done manually. This allows the user to easily create cognitive models. For the design of a cognitive model, the elements that structure a cognitive model must be taken into account previously, which are: *Goals*, *Mental States*, *Actions* and *Objects*.

When the software is initialized, the work window will be displayed. Thus, the creation of a cognitive model will be an action of drag and drop the elements to the assigned space, in this sense, the user can add in the elements necessary to create the new cognitive model.

When this process is completed, the cognitive model is saved and validated in CARINA and is automatically executed. Then, the user has the facility to transform the file into a JSON format. The user can create "n" number of cognitive models as long as the domain permits and does not exist, in order to make the creation of cognitive models mechanical The views (see Figures 20-27) of MetaThink V2.0 and link to access: https://yes-lake.now.sh/login

Figure 20

MetaThink version 2.0 registration page

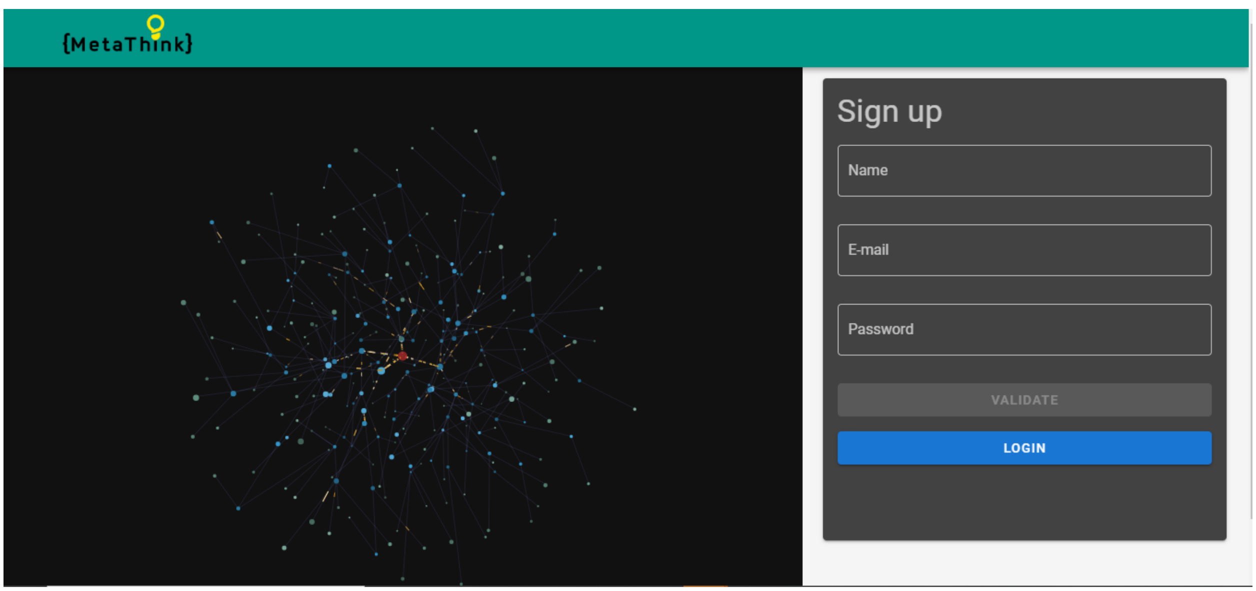


Figure 21

MetaThink version 2.0 login page

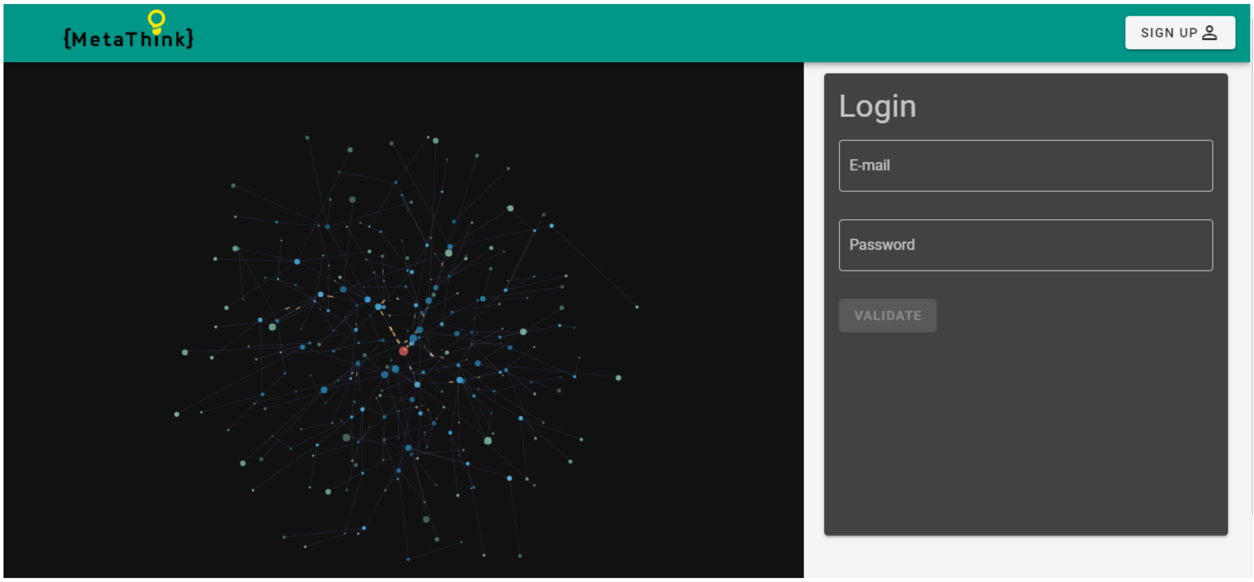


Figure 22

Start menu page MetaThink version 2.0

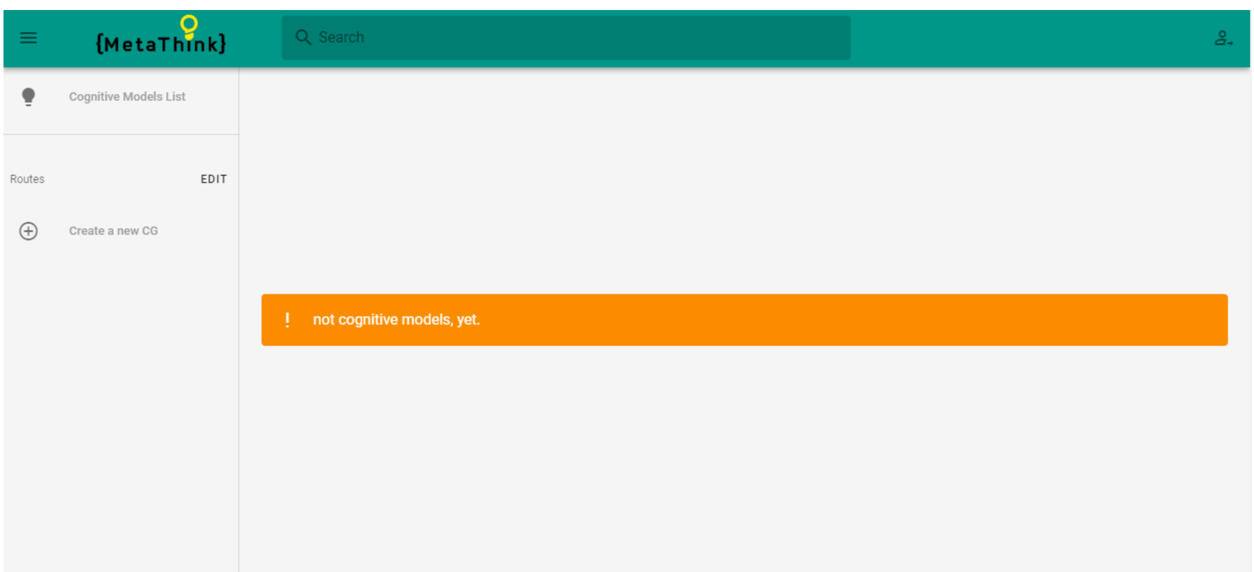


Figure 23

Create a new cognitive model page MetaThink version 2.0

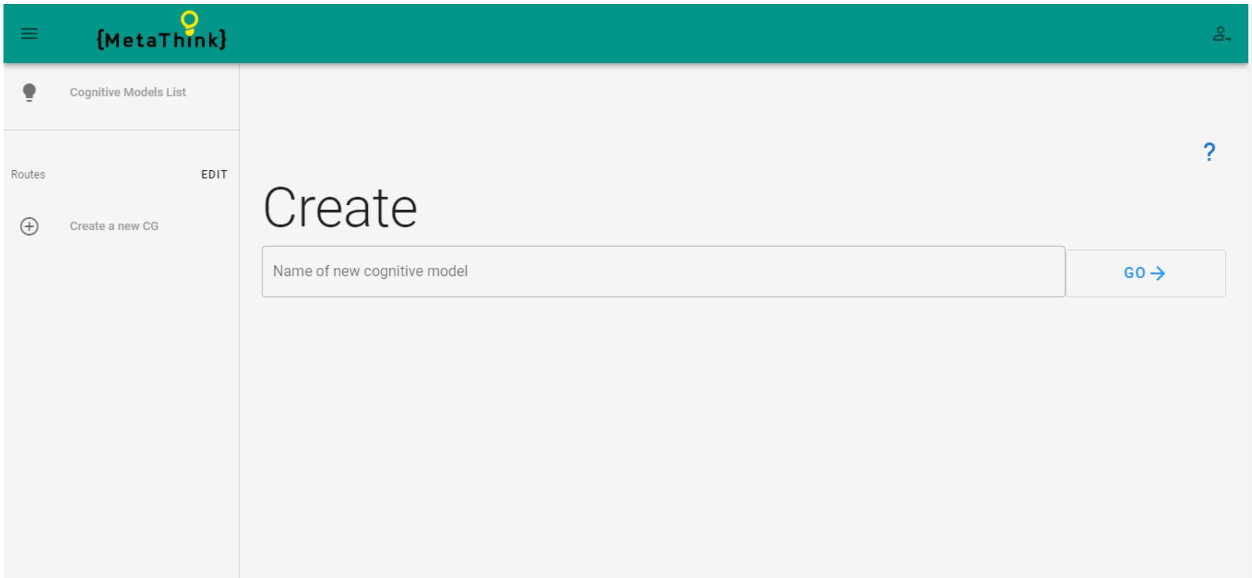


Figure 24

Elements page to create a cognitive model MetaThink version 2.0

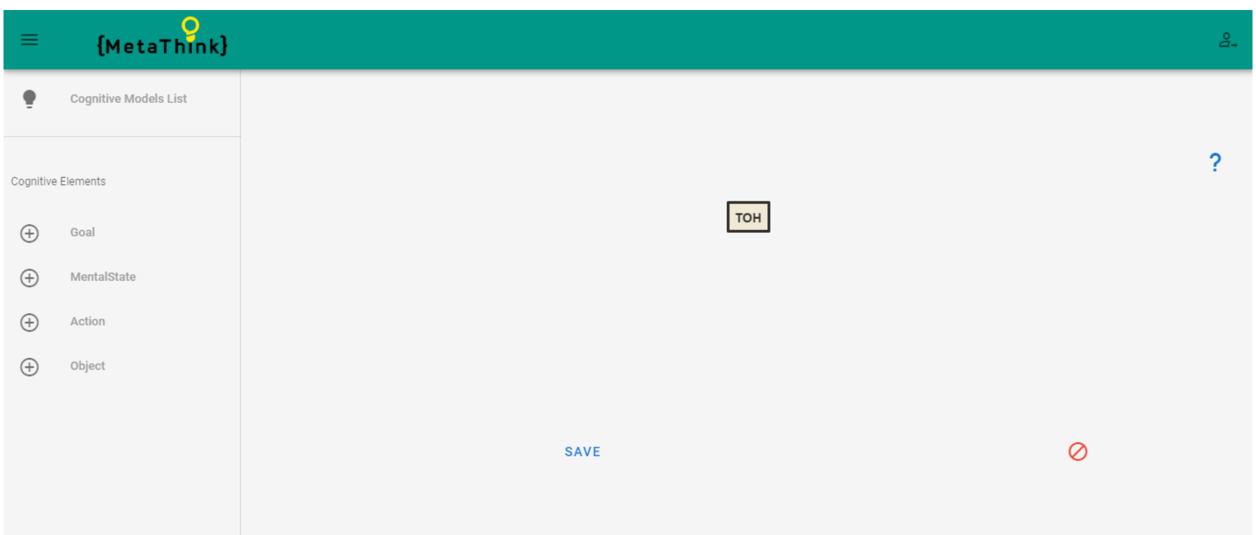


Figure 25

MetaThink version 2.0 cognitive model creation page

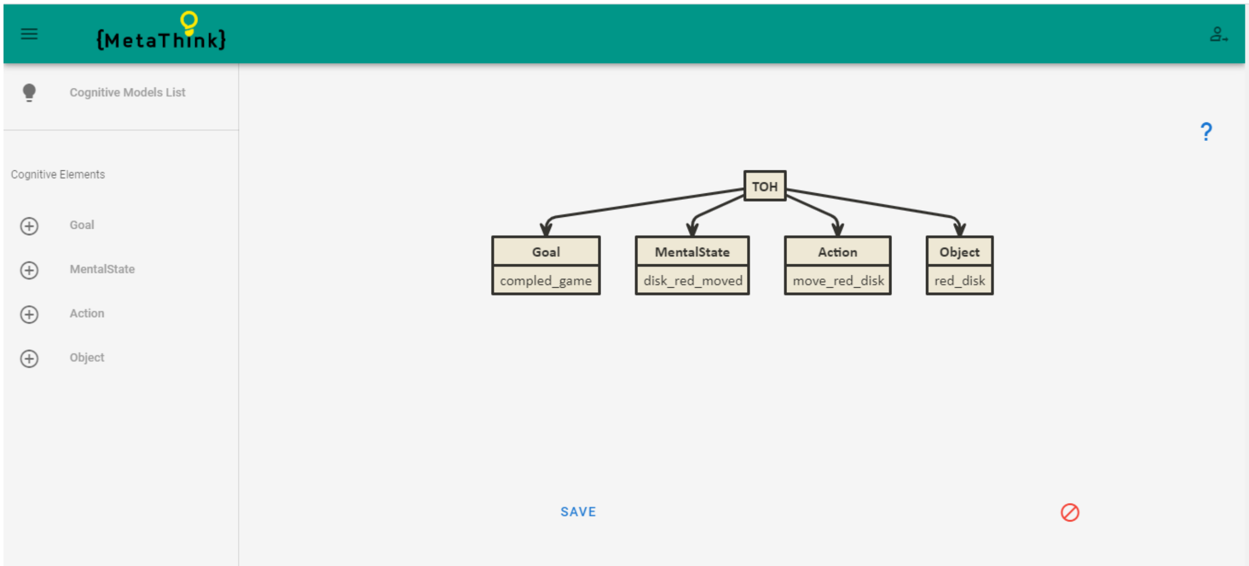


Figure 26

MetaThink version 2.0 saved cognitive model page

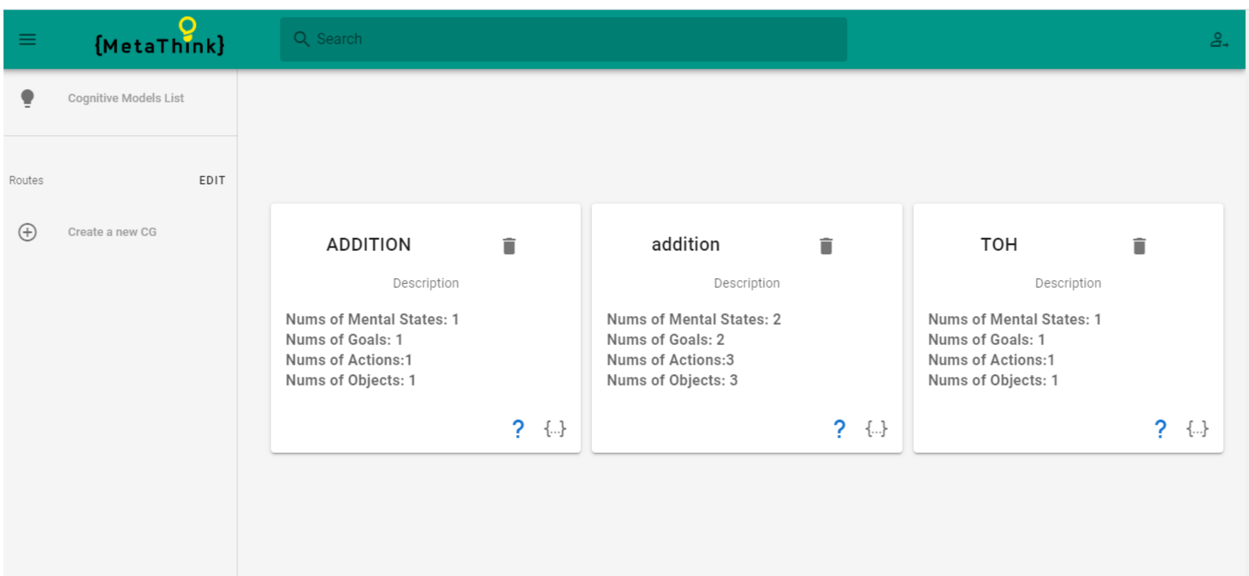


Figure 27

Detail page of the JSON file cognitive model MetaThink version 2.0

5.4.1. MetaThink Version 2.0 Validation

The functional prototype was evaluated through of an assessment instrument proposed by Jensen et al., (2012), used to evaluate software taking into account the following criteria of technical of *quality* and *usability*, through the categories of *functionality*, *reliability*, *usability*, *efficiency*, *maintainability* and *portability* (see Appendix H).

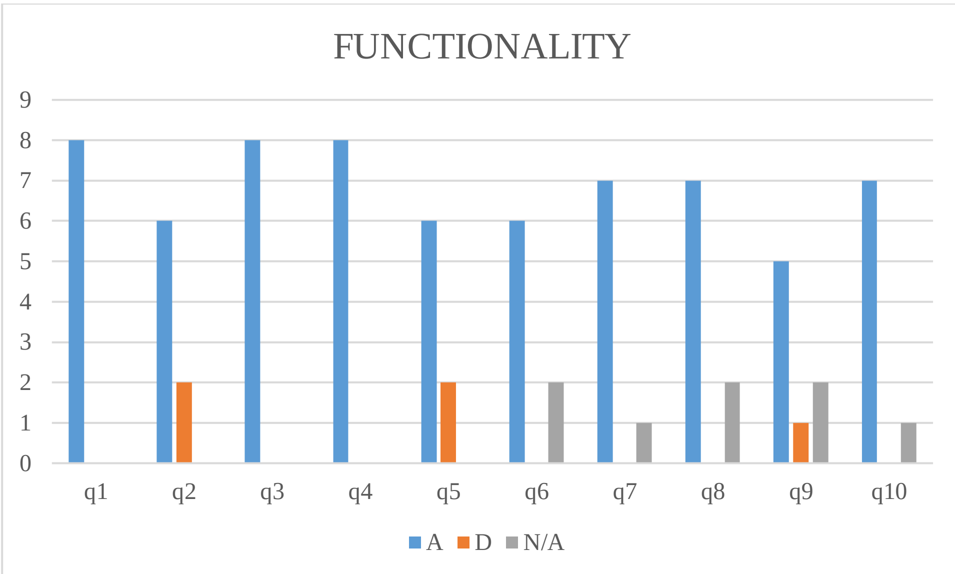
In this case, eight experts in computer science and systems development were involved. The experts were provided with access to the software register and general guidance on the evaluation process. The expert assessment was conducted individually and there was no interference from the researchers. Also, the experts were provided with a user manual containing information on how the software was developed, detailed specifications of each item under evaluation and instructions for the evaluation process.

The results of the evaluation are shown below:

In the *functionality* category, 85% of the experts agreed with the criteria of adequacy, accuracy, interoperability, conformity and secure access, however 15% disagreed and considered it necessary to improve the criteria of adequacy, accuracy and conformity (see Figure 28).

Figure 28

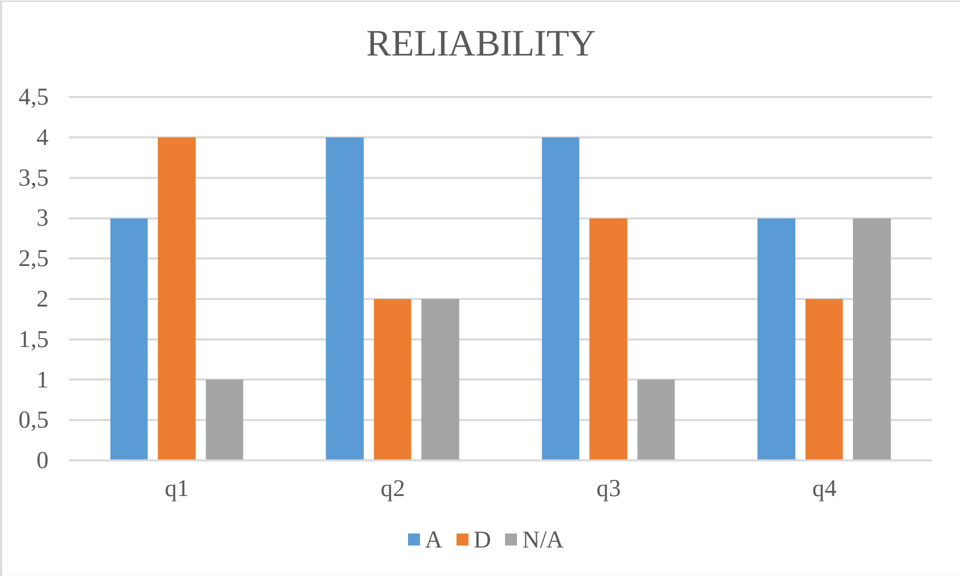
MetaThink v2.0 functionality category



In the *reliability* category 43,75% of the experts agreed the tolerance to failures and recoverability, however the 56, 25% disagreed or consider it necessary to improve the criteria for maturity, tolerance to failures and recoverability (see Figure 29):

Figure 29

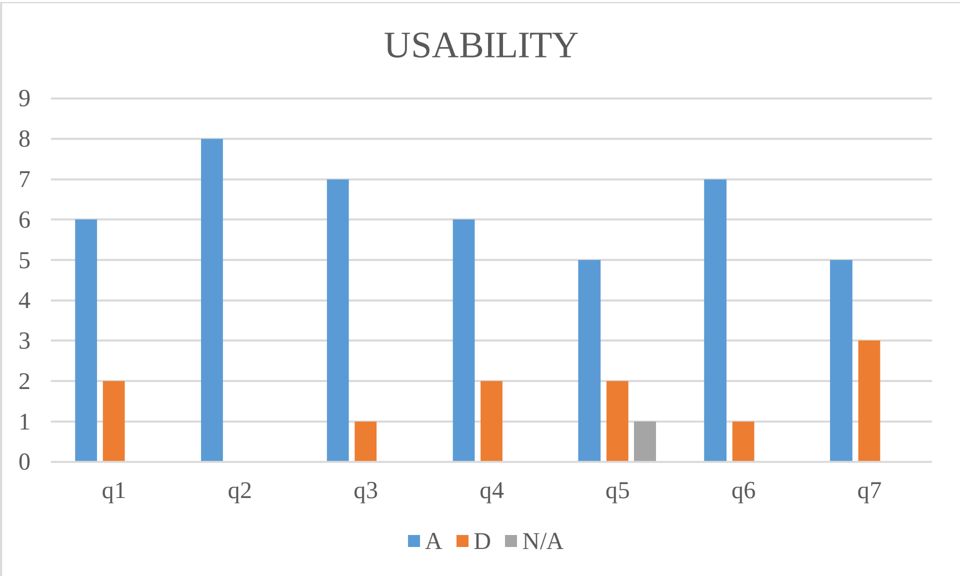
MetaThink v2.0 reliability category



In *usability* category 78.57% of experts agreed with the criteria of intelligibility, learnability and operability, however 21.42% disagreed or considered them not applying (see figure 30):

Figure 30

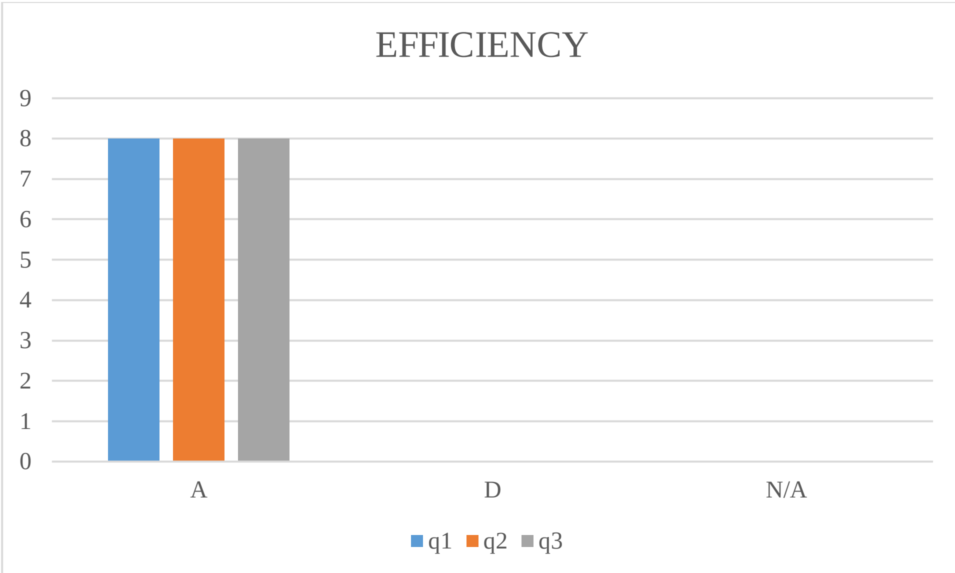
MetaThink v2.0 usability category



In *efficiency* category a 100% of the experts agreed with the criteria of time and resources (see figure 31):

Figure 31

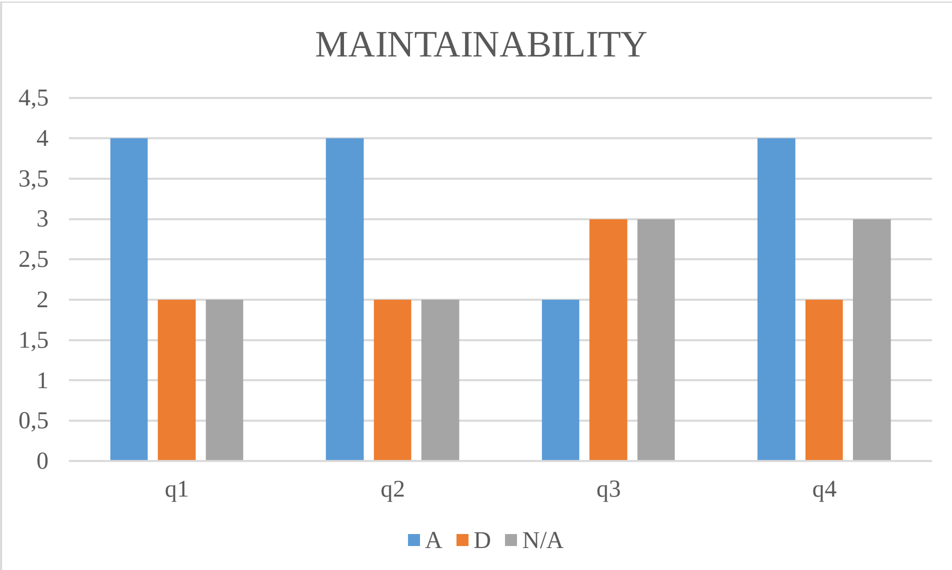
MetaThink v2.0 efficiency category



In *maintainability* category a 43,75% of experts agreed with the criteria of analyzability, modifiability, stability and testability. However, 59,375% of the experts stated that they disagree with the previous criteria or that the criteria do not apply (see figure 32):

Figure 32

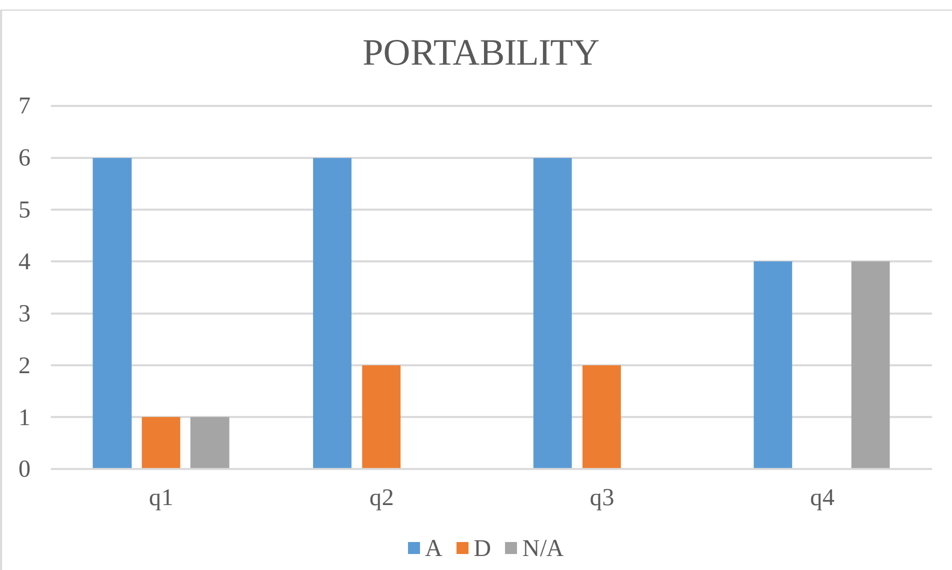
MetaThink v2.0 maintainability category



Finally, in *portability* category 68,75 % of the experts agreed with the criteria of adaptability, capacity to be installed, conformity and capacity to be replace. However, 31,25 % considered that they disagreed with the criteria or that they did not apply (see Figure 33).

Figure 33

MetaThink v2.0 portability category



Thus, according to the order of above results, it can be demonstrate that functional prototype implemented has *efficiency*, *functionality* and *usability*, as a result, usefulness to development cognitive models in agile and visual way is obtained. However, the categories of *portability*, *reliability*, and *maintainability* must be improved.

Recently this project was presented at a national research event organized by Fundación Universitaria Horizonte – Bogotá (see Appendix L).

5.5. Illustrative Examples of Cognitive Models in CARINA

In this section the examples of cognitive models in several knowledge domains using CARINA based intelligent systems are created.

The first example of cognitive model in different domains in this thesis, was created by López et al., (2018) for the representation in M++ of the Cognitive Model for the Generation of Factoid-WH questions. The cognitive model was presented in the chapter: Theoretical Background.

Other example, following the before steps proposed by Olier et al., (2018) a user-based cognitive model, created by Flórez et al., (2019) for the representation in M++ of the Cognitive Model for the Towers of Hanoi algorithm in the metacognitive architecture CARINA, is presented below:

1) *Selection of Cognitive Task*

The cognitive task that was selected was the development of a cognitive model for the Towers of Hanoi Algorithm in the metacognitive architecture CARINA.

2) *Obtaining Information for Describing the Cognitive Task*

The information presented in the cognitive task was acquired using users and some documentary sources as sources of information (see Table 10).

Table 10

Format to synthesize the cognitive task description when the information source comes from users.

Users	
Description of User Type	4 students' High School in the programming course
Description of Cognitive Task to be Developed by User	User-based cognitive models are computational representations where the subject solves a specific cognitive task, with the purpose of analyzing user behavior and making understandings and predictions using observations as sources of information, through well-structured problems such as the Towers of Hanoi. Thus, user-based cognitive models are used to predict user behavior, obtain knowledge or improve existing computational models. To develop a cognitive model, it is necessary to study the structure of the cognitive task. The analysis process is then performed with the Goals, Operators, Methods and Selection Rules (GOMS) technique, which is a specification of the knowledge that a system requires to achieve a cognitive task and NGOMS-L is a natural language notation structured to represent the GOMS models.

Note. This table was adapted from: Flórez, Y. P., Jerónimo, A. J., Castillo, M. E., & Gómez, A. A. (2019, March). User-Based Cognitive Model in NGOMS-L for the Towers of Hanoi Algorithm in the Metacognitive Architecture CARINA. In *The International Conference on Advances in Emerging Trends and Technologies* (pp. 473-484). Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-32022-5_44.

3) Description of Cognitive Task in Natural Language

The first stage (see Table 11) in which the computational cognitive model was developed starting with the subjects verbalization recordings and the transcription of the recorded material taking note of voice and repetitions.

Table 11

Pre-processing stage

1.Pre-processing stage		
Subject 1	E 46 I am going to write ... eh ... the orange wheel goes to the "A".	E: Each expression … pauses or silences
Subject 2	E12... I am thinking, I am thinking, I am thinking.	
Subject 3	E1 I move the green piece to stake "B"	
Subject 4	E2...I'm doin..., I'm writing the instruction...move the stake...the green hoop to stake "B"...	

Note. This table was adapted from: Flórez, Y. P., Jerónimo, A. J., Castillo, M. E., & Gómez, A. A. (2019, March). User-Based Cognitive Model in NGOMS-L for the Towers of Hanoi Algorithm in the Metacognitive Architecture CARINA. In *The International Conference on Advances in Emerging Trends and Technologies* (pp. 473-484). Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-32022-5_44.

The second stage is called: "processing stage" in the application of the TOH, it was observed that three of the four subjects achieved to resolve the problem in approximately one hour, with an approximate of 80 movements. Subject 1 solved the problem with 122 movements. Subject 2 solved the problem with 60 movements and subject 3 solved the problem with 67 movements. And subject 4 quit the problem with 173 movements (see Table 12).

Table 12

Processing stage: production system and decision tree

Nº. Steps	Initial state A{G,Y, O, R, P}, B{}, C{}	Final state A{}, B{}, C{ G,Y, O, R, P }	Operators
Subject 1: 122 (Successful)	A{G,Y, O, R, P}, B{}, C{}	A{}, B{}, C{ G,Y, O, R, P }	Move (G,C)
Subject 2: 60 (Successful)	A{G,Y, O, R, P}, B{}, C{}	A{}, B{}, C{ G,Y, O, R, P }	Move (G,C)
Subject 3: 67 (Successful)	A{G,Y, O, R, P}, B{}, C{}	A{}, B{}, C{ G,Y, O, R, P }	Move (G,C)
Subject 4: 173 (Unsuccessful)	A{Y,O}, B{G,R,P},C{}	A{O}, B{G, R, P}, C{Y}	Move{Y,C}

Note. This table was adapted from: Flórez, Y. P., Jerónimo, A. J., Castillo, M. E., & Gómez, A. A. (2019, March). User-Based Cognitive Model in NGOMS-L for the Towers of Hanoi Algorithm in the Metacognitive Architecture CARINA. In *The International Conference on Advances in Emerging Trends and Technologies* (pp. 473-484). Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-32022-5_44.

The third stage is: "classification of subjects", in the application, it was observed that in performing the cognitive task three of the subjects understood the problem (i.e., per-forming appropriate mental actions) and the rules of the problem. A subject also left before finish the problem by explaining that he did not understand the problem (see Table 13).

Table 13

Description of the cognitive task in natural language

Successful subject	Unsuccessful Subject
Move green disk to peg B	Move green disk to peg B
Move yellow disk to peg C	Move yellow disk to peg C
Move green disk to peg C	Move green disk to peg A
Move orange disk to peg B	Move yellow disk to peg B

Note. This table was adapted from: Flórez, Y. P., Jerónimo, A. J., Castillo, M. E., & Gómez, A. A. (2019, March). User-Based Cognitive Model in NGOMS-L for the Towers of Hanoi Algorithm in the Metacognitive Architecture CARINA. In *The International Conference on Advances in Emerging Trends and Technologies* (pp. 473-484). Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-32022-5_44.

4) Cognitive Task in GOMS (NGOMS-L)

The fourth stage to build a computational cognitive model, one must start by analyzing the cognitive task (Wong et al., 2010). For this, methodologies are required that allow specifying a set of goals and sub-goals defined as the steps to be executed to solve the cognitive task that will be described computationally. Thus, the method commonly used for the analysis of user-based cognitive problems is called: NGOMS-L, which is defined as a natural language notation for the specification of GOMS models and a series of steps to elaborate them (Kieras, 1999). GOMS is an acronym for: Objectives, Operators, Methods and Rules of Selection. Operators are particular steps created by users given time. If a goal is not achieved by more than one method, selection rules are used to determine the appropriate method (John & Kieras, 1996). Thus, Table 4 shows the cognitive model of the TOH algorithm based on NGOMS-L (see Table 14).

Table 14

Cognitive model of the TOH algorithm based on NGOMS - L

NGOMS-L notation

Method for goal γ_{700}: Complete the game
Step 1. (α^{ς}_{700}) Accomplish goal:γ_{701}
Step 2. (α^{ς}_{701}) Accomplish goal:γ_{702}
Step n. (α^{ς}_{702}) Return with goal accomplished.

Method for goal γ_{701}: Subject moves the green disk to peg B.
Step 1. (α^{ς}_{703}) Choose green disk
Step 2. (α^{ς}_{704}) Select peg
Step 3. (α^{ς}_{705}) Put green disk in the selected peg
Step n. (α^{ς}_{702}) Return with goal accomplished.

Method for goal γ_{702}: Subject moves the yellow disk to peg C.
Step 1. (α^{ς}_{706}) Choose yellow disk
Step 2. (α^{ς}_{707}) Select peg
Step 3. (α^{ς}_{708}) Put yellow disk in the selected peg
Step n. (α^{ς}_{702}) Return with goal accomplished.

Method for goal γ_{703}: Subject moves the green disk to peg C.
Step 1. (α^{ς}_{709}) Choose green disk
Step 2. (α^{ς}_{710}) Select peg
Step 3. (α^{ς}_{711}) Put green disk in the selected peg
Step n. (α^{ς}_{702}) Return with goal accomplished.

Method for goal γ_{704}: Subject moves the orange disk to peg B.
Step 1. (α^{ς}_{712}) Choose orange disk
Step 2. (α^{ς}_{713}) Select peg
Step 3. (α^{ς}_{714}) Put orange disk in the selected peg
Step 4. (α^{ς}_{715}) Accomplish goal:γ_{701}
Step n. (α^{ς}_{702}) Return with goal accomplished.

Method for goal γ_{705}: Subject moves the yellow disk to peg A.
Step 1. (α^{ς}_{716}) Choose yellow disk
Step 2. (α^{ς}_{717}) Select peg
Step 3. (α^{ς}_{718}) Put yellow disk in the selected peg
Step 4. (α^{ς}_{719}) Accomplish goal:γ_{703}
Step n. (α^{ς}_{702}) Return with goal accomplished.

Method for goal γ_{706}: Subject moves the yellow disk to peg B.
Step 1. (α^{ς}_{720}) Choose yellow disk
Step 2. (α^{ς}_{721}) Select column
Step 3. (α^{ς}_{722}) Put yellow disk in the selected peg
Step 4. (α^{ς}_{723}) Accomplish goal:γ_{701}
Step n. (α^{ς}_{702}) Return with goal accomplished.

Method for goal γ_{707}: Subject moves the red disk to peg C.
Step 1. (α^{ς}_{724}) Choose red disk
Step 2. (α^{ς}_{725}) Select peg
Step 3. (α^{ς}_{726}) Put red disk in the selected peg
Step n. (α^{ς}_{702}) Return with goal accomplished.

Method for goal γ_{708}: Subject moves the green disk to peg A.
Step 1. (α^{ς}_{727}) Choose green disk
Step 2. (α^{ς}_{728}) Select peg
Step 3. (α^{ς}_{729}) Put green disk in the selected peg
Step 4. (α^{ς}_{730}) Accomplish goal:γ_{702}
Step 5. (α^{ς}_{731}) Accomplish goal:γ_{701}
Step 6. (α^{ς}_{732}) Accomplish goal:γ_{703}
Step n. (α^{ς}_{702}) Return with goal accomplished.

Method for goal γ_{709}: Subject moves the Orange disk to peg A.
Step 1. (α^{ς}_{733}) Choose orange disk
Step 2. (α^{ς}_{734}) Select peg
Step 3. (α^{ς}_{735}) Put orange disk in the selected peg.
Step 4. (α^{ς}_{736}) Accomplish goal:γ_{701}
Step 5. (α^{ς}_{737}) Accomplish goal:γ_{705}
Step 6. (α^{ς}_{738}) Accomplish goal:γ_{703}
Step 7. (α^{ς}_{739}) Accomplish goal:γ_{706}
Step 8. (α^{ς}_{740}) Accomplish goal:γ_{701}
Step n. (α^{ς}_{702}) Return with goal accomplished.

Method for goal γ_{710}: Subject moves the Orange disk to pet C.
Step 1. (α^{ς}_{741}) Choose Orange disk
Step 2. (α^{ς}_{742}) Select pet
Step 3. (α^{ς}_{743}) Put orange disk in the selected pet
Step 4. (α^{ς}_{744}) Accomplish goal:γ_{701}
Step 5. (α^{ς}_{745}) Accomplish goal:γ_{706}
Step 6. (α^{ς}_{746}) Accomplish goal:γ_{703}
Step n. (α^{ς}_{702}) Return with goal accomplished.

Method for goal γ_{711}: Subject moves the purple disk to pet B.
Step 1. (α^{ς}_{747}) Choose purple disk
Step 2. (α^{ς}_{748}) Select pet
Step 3. (α^{ς}_{749}) Put purple disk in the selected pet.
Step 4. (α^{ς}_{750}) Accomplish goal:γ_{701}
Step 5. (α^{ς}_{751}) Accomplish goal:γ_{705}
Step 6. (α^{ς}_{752}) Accomplish goal:γ_{708}
Step 7. (α^{ς}_{753}) Accomplish goal:γ_{704}
Step 8. (α^{ς}_{754}) Accomplish goal:γ_{703}
Step 9. (α^{ς}_{755}) Accomplish goal:γ_{706}
Step 10. (α^{ς}_{756}) Accomplish goal:γ_{701}
Step n. (α^{ς}_{702}) Return with goal accomplished.

Method for goal γ_{712}: Subject moves the red disk to pet A.
Step 1. (α^{ς}_{757}) Choose red disk
Step 2. (α^{ς}_{758}) Select pet
Step 3. (α^{ς}_{759}) Put red disk in the selected pet
Step 4. (α^{ς}_{760}) Accomplish goal:γ_{703}
Step 5. (α^{ς}_{761}) Accomplish goal:γ_{705}
Step 6. (α^{ς}_{762}) Accomplish goal:γ_{708}
Step 7. (α^{ς}_{763}) Accomplish goal:γ_{710}
Step 8. (α^{ς}_{764}) Accomplish goal:γ_{703}
Step 9. (α^{ς}_{765}) Accomplish goal:γ_{706}
Step 10. (α^{ς}_{766}) Accomplish goal:γ_{701}
Step 11. (α^{ς}_{767}) Accomplish goal:γ_{709}
Step 12. (α^{ς}_{768}) Accomplish goal:γ_{703}
Step 13. (α^{ς}_{769}) Accomplish goal:γ_{705}
Step 14. (α^{ς}_{770}) Accomplish goal:γ_{708}
Step n. (α^{ς}_{702}) Return with goal accomplished.

Method for goal γ_{713}: Subject moves the purple disk to pet C.
Step 1. (α^{ς}_{771}) Choose purple disk
Step 2. (α^{ς}_{772}) Select pet
Step 3. (α^{ς}_{773}) Put purple disk in the selected pet.
Step 4. (α^{ς}_{774}) Accomplish goal:γ_{701}
Step 5. (α^{ς}_{775}) Accomplish goal:γ_{702}
Step 6. (α^{ς}_{776}) Accomplish goal:γ_{703}
Step 7. (α^{ς}_{777}) Accomplish goal:γ_{704}
Step 8. (α^{ς}_{778}) Accomplish goal:γ_{708}
Step 9. (α^{ς}_{779}) Accomplish goal:γ_{706}
Step 10. (α^{ς}_{780}) Accomplish goal:γ_{701}
Step 11. (α^{ς}_{781}) Accomplish goal:γ_{707}
Step 12. (α^{ς}_{782}) Accomplish goal:γ_{703}
Step 13. (α^{ς}_{783}) Accomplish goal:γ_{705}
Step 14. (α^{ς}_{784}) Accomplish goal:γ_{708}
Step 15. (α^{ς}_{785}) Accomplish goal:γ_{710}
Step 16. (α^{ς}_{786}) Accomplish goal:γ_{701}
Step 17. (α^{ς}_{787}) Accomplish goal:γ_{702}
Step 18. (α^{ς}_{788}) Accomplish goal:γ_{703}
Step n. (α^{ς}_{702}) Return with goal accomplished.

Note. This table was adapted from: Flórez, Y. P., Jerónimo, A. J., Castillo, M. E., & Gómez, A. A. (2019, March). User-Based Cognitive Model in NGOMS-L for the Towers of Hanoi Algorithm in the Metacognitive Architecture CARINA. In *The International Conference on Advances in Emerging Trends and Technologies* (pp. 473-484). Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-32022-5_44.

The five step of the cognitive task of the TOH algorithm is represented in NGOMS-L describing the Goals, Actions, and Mental States with the respective inventories. A fragment of the cognitive model in NGOMS-L is shown in the Table 15 (see Table 15).

Table 15

Inventory of Goals, Mental States and Actions of the TOH algorithm

Goals Inventory	Mental States Inventory	Actions Inventory
Goalγ_{701}: Subject moves green disk to peg B.	**Mental Stateσ_{703} : Green disk is moved to peg C.**	α^{c}_{700}) Accomplish goal γ_{700}
Goalγ_{702}: Subject moves yellow disk to peg C.	**Mental Stateσ_{704} : Orange disk is moved to peg B.**	α^{c}_{702}: Return with goal accomplished.
Goalγ_{703}: Subject moves green disk to peg C.	**Mental Stateσ_{705} : Yellow disk is moved to peg A.**	α^{c}_{703}: Choose green disk
Goalγ_{704}: Subject moves orange disk to peg B.	**Mental Stateσ_{706} : Yellow disk is moved to peg B.**	α^{c}_{704}: Select peg
Goalγ_{705}: Subject moves yellow disk to peg A.	**Mental Stateσ_{707} : Red disk is moved to peg C.**	α^{c}_{705}: Put green disk in selected peg
Goalγ_{706}: Subject moves yellow disk to peg B.	**Mental Stateσ_{708} : Green disk is moved to peg A.**	α^{c}_{706}: Choose yellow disk
Goalγ_{707}: Subject moves red disk to peg C.	**Mental Stateσ_{709} : Orange disk is moved to peg A.**	α^{c}_{708}: Put yellow disk in selected peg
Goalγ_{708}: Subject moves green disk to peg A.	**Mental Stateσ_{710} : Orange disk is moved to peg C.**	α^{c}_{712}: Choose orange disk
Goalγ_{709}: Subject moves orange disk to peg A.	**Mental Stateσ_{711} : Purple disk is moved to peg B.**	α^{c}_{714}: Put orange disk in selected peg
Goalγ_{710}: Subject moves orange disk to peg C.	**Mental Stateσ_{712} : Red disk is moved to peg A.**	α^{c}_{724}: Choose red disk
Goalγ_{711}: Subject moves purple disk to peg B.	**Mental Stateσ_{713} : Purple disk is moved to peg C.**	α^{c}_{726}: Put red disk in selected peg
Goalγ_{712}: Subject moves red disk to peg A.		α^{c}_{741}: Choose purple disk
Goalγ_{713}: Subject moves purple disk to peg C.		α^{c}_{743}: Put purple disk in selected peg

Note. This table was adapted from: Flórez, Y. P., Jerónimo, A. J., Castillo, M. E., & Gómez, A. A. (2019, March). User-Based Cognitive Model in NGOMS-L for the Towers of Hanoi Algorithm in the Metacognitive Architecture CARINA. In *The International Conference on Advances in Emerging Trends and Technologies* (pp. 473-484). Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-32022-5_44.

Goela et al., (2001) propose algorithms which is solved with 16 steps. But in this example, the successful subject resolved the TOH in one hour with 60 steps, and the unsuccessful subject used 173 steps in two hours.

Thus, this example, is a contribution in the scientific world where the algorithm of the TOH is modeled with NGMOS-L, using cognitive model based-user through the metacognitive architecture CARINA. This has been published by Flórez et al., (2019) as a chapter book (see Appendix I). In addition, was presented in *1st International Conference on Advances in Emerging Trends and Technologies ICAETT 2019* (see Appendix J), and was presented in a national congress (see Appendix K).

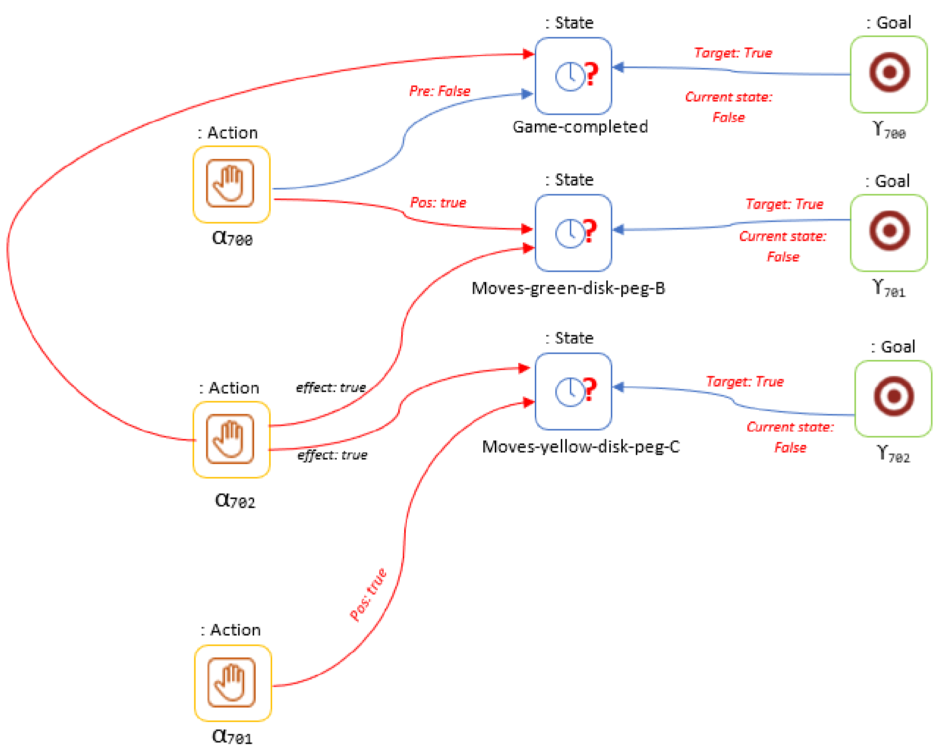
Finally, the specification in M ++ of the cognitive model in NGOMS-L to be executed in CARINA is shown in the Figure 34 (see Figure 34).

5) Cognitive Model from NGOMS-L to M++ Language

The *Goals* are achieved when the mental state it is related to this is completed. The reasoning process of CARINA's object-level searches changes in a problem from a set of initial states to a set of final states, as well.

Figure 34

Specification in M ++ of the cognitive model in NOGOMS-L of the TOH algorithm in CARINA



The Figure 35 (see Figure 35), presents all the *Goals* and *Actions* of the model expressed in M++ and the preconditions (actions to accomplish) and postcondition (actions to return when an action is accomplished).

In these terms, Figure 36, shows all the *Goals*, *Mental States* and *Actions* of the Towers of Hanoi algorithm implemented in the metacognitive architecture CARINA.

Figure 35

Pre-conditional and post-conditional actions in M++ for the TOH algorithm in CARINA

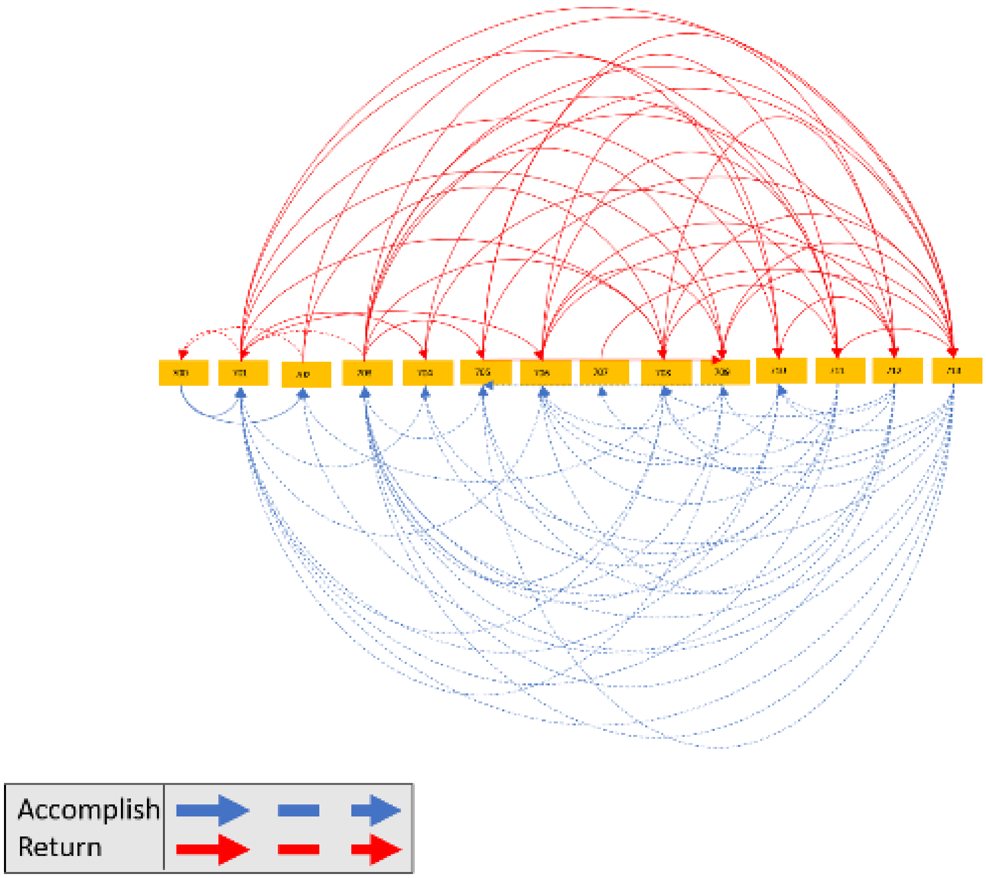
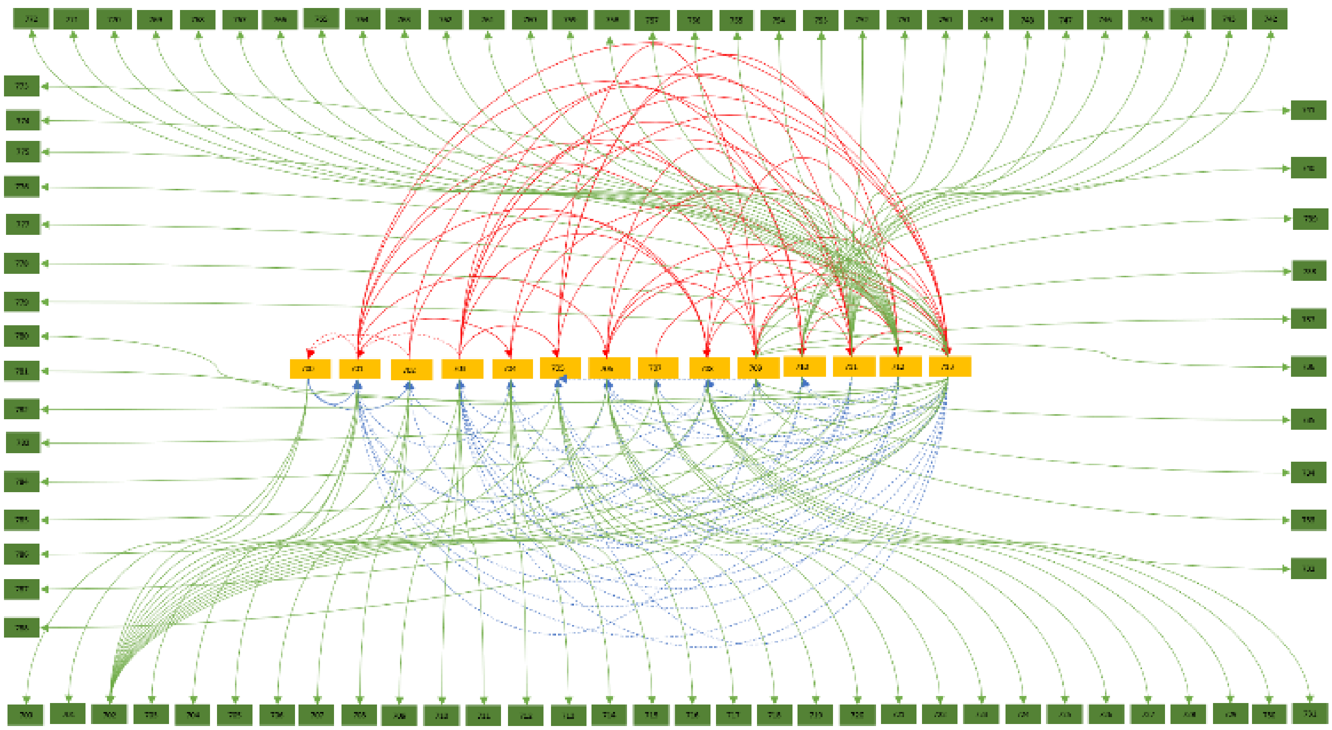


Figure 36

Representation of the NGOMS-L model in M++



Note. The blue and red lines refer to pre-conditional and post-conditional actions. The pre-conditions refer to the current state of the goal and the post-conditional actions refer to the returned state when a goal is affected by the pre-conditional action and it turns true. And the green lines refer to each action that allows to continue accomplishing all the goals of the cognitive task.

6) *Runnable cognitive model in CARINA of TOH algorithm*

The cognitive model for the excecution of a computational cognitive model for the Towers of Hanoi Algorithm in the CARINA metacognitive architecture was created through an executable code in an open standard file format, called JSON, the code fragments are described in detail (see Figure 37):

Figure 37

Runnable cognitive model in CARINA of TOH algorithm


```json
{
  "Goal": [
    "game-complete",
    "green-disk-on-peg-b",
    "yellow-disk-on-peg-c"
  ],
  "StateMental": [
      "yellow-disk-on-peg-b",
      "green-disk-on-peg-c"
  ],
  "Action": [
    "moves-green-disk-peg-b",
    "moves-yellow-disk-peg-c"
  ],
  "Object": [
    "yellow-disk",
    "green-disk",
    "peg-b",
    "peg-c"
  ],
  "_id": "5eb45c837f14160017aa09dd",
  "User": "5da6213fceac9e2488a9b285",
  "domain": "towers of hanoi",
  "createdAt": "2020-05-07T19:07:47.295Z",
  "__v": 0
}
```

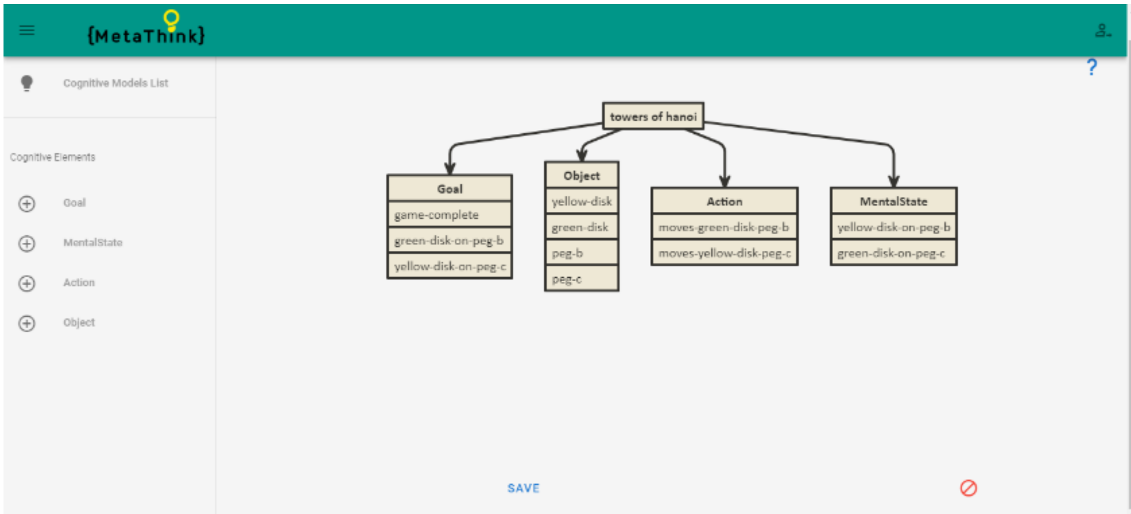

The *Mental States* indicate to CARINA how to accomplish a specific task. For this, all the *Mental States* which are part of the cognitive model contain an identifier for the system, a name, a type and an identifier of the cognitive model. Thus, the cognitive model starts with the *Mental States*, which are *Goals* to be modified from a false state to a true state. The *Mental States* then, become the tasks that the cognitive model contains to accomplish the problem.

7) Testing and Maintenance of Cognitive Model

The cognitive model developed for the Towers of Hanoi Algorithm in the CARINA metacognitive architecture was tested with a functional framework to create cognitive models in visual way, called MetaThink version 2.0 (described in chapter: 5.4 MetaThink Version 2.0). The results of the cognitive model are shown below (see Figure 38):

Figure 38

Executable cognitive model for the Towers of Hanoi Algorithm in the metacognitive architecture CARINA



This cognitive model of the Towers of Hanoi algorithm was created using MetaThink V2.0 and is structured by the main objective of the cognitive task which is: "goal:Game_complete" (in this example: towers_of_hanoi). I.e., the cognitive model has goals (three sub_steps), objects (four elements), actions (two actions, one pre-conditional and one post-conditional) and mental states (two states).

Chapter VI

Conclusions

This research showed a formal, semantic and computational representation of the necessary elements for cognitive model constructions in the metacognitive architecture CARINA, also developed a functional prototype of a software called MetaThink V2.0 that allows the visual creation of the elements to design a runnable cognitive model in CARINA.

The formal representation of a cognitive model in CARINA has the Problem, Goals, Sensor, Mental State, Procedural Knowledge and Semantic Knowledge.

The semantic representation of a cognitive model in CARINA contains the Problem which has the Mental States, Goals, sub-goals, Rules, Actions, Belief, Attention System and the Sensor that permanently monitors reasoning traces in CARINA self-model, which contains Cognitive Functions profiles that have been runnable in its *object-level*.

The computational representation contains the name, which has Problem's name, and the type, the Mental States, the Goals which contain the current Mental State, and target value, the Production Rules, which contain the rule, the condition, the attention system, the problem, the mental state, the goal, the state, the sensor, the constraint, the conclusion, and the action. Finally, the Belief, that contains typeSMU, which has, type and value.

Different illustrative examples were presented to create cognitive tasks based on the metacognitive architecture CARINA, such as the addition of two numbers, the generation of WH-question and the Towers of Hanoi Algorithm.

The methodology used in this research is the cognitive modeling methodology to create cognitive models that are executed in CARINA, performing the main representations: the formal, semantic and computational, which allows to evidence the definition of the principals elements of a cognitive model such as: the *Mental States*, *Goals*, *Actions* and *Production Rules* of a cognitive model.

In this context, cognitive models need to be stored in CARINA's Semantic Memory with the purpose to retrieve them anytime when necessary. Thus, these cognitive models will be executed after, without direct supervision of the cognitive designer. With this, the computational representation was created, using an open standard format file (JSON) based on the formal and semantic representations, containing the described elements above, and based on the computational representation, a functional prototype of a software was built, called MetaThink V2.0 for the visual creation of cognitive models.

The validation process of the functional prototype was obtained by analyzing the criteria of technical quality and usability, through six categories as: *functionality*, *reliability*, *usability*, *efficiency*, *maintainability* and *portability*. A group of eight experts evaluated the MetaThink V2.0 software. From the results, it was concluded that the prototype presented achieved *efficiency*, *functionality* and *usability*, being useful to develop cognitive models in an agile and visual way. However, the categories of *portability*, *reliability*, and *maintainability* must be improved.

6.1. Recommendations

For further research, cognitive models should be based on planning. Additionally, CARINA's current system of knowledge acquisition, i.e. beliefs, must be transformed (because are currently very limited). Additionally, CARINA's current system of knowledge acquisition, i.e. beliefs, must be transformed (because currently they are very limited). For Beliefs, to be based on planning, they should not only have declarative and semantic knowledge, but also procedural and episodic knowledge. Therefore, cognitive models should also take into account:

CARINA's reasoning cycle, which is based on cognitive functions. For this, a formal and computational representation of cognitive functions should be made (for this, an article was submitted, in *Cognitive Systems Research* journal). In this context, cognitive functions (reasoning cycle), memory and knowledge acquisition should be joined in CARINA. In these terms, for CARINA's *object -level* to work, it needs to join the cycle of reasoning (based on cognitive functions). Additionally, the representation of knowledge (based on belief), CARINA requires a knowledge acquisition mechanism based on cognitive models. In order to achieve this objective, a research project is currently being proposed by the Universidad de Córdoba - Colombia (see Appendix M).

Chapter VII

References

Aamodt, A. (1995). Knowledge acquisition and learning by experience--the role of case-specific knowledge. *Machine Learning and Knowledge Acquisition*, 197–245.

Anderson, J. R. (1996). ACT: A simple theory of complex cognition. *American Psychologist*, *51*(4), 355.

Apt, K. R., Blair, H. A., & Walker, A. (1988). Towards a theory of declarative knowledge. In *Foundations of deductive databases and logic programming* (pp. 89–148). Elsevier.

Arevalillo-Herráez, M., Arnau, D., & Marco-Giménez, L. (2013). Domain-specific knowledge representation and inference engine for an intelligent tutoring system. *Knowledge-Based Systems*, *49*, 97–105.

Atkinson, R. C., & Shiffrin, R. M. (1968). *Human memory: A proposed system and its control processes*.

Avery, E., Kelley, T. D., & Davani, D. (2006). Using cognitive architectures to improve robot control: Integrating production systems, semantic networks, and sub-symbolic processing. *15th Annual Conference on Behavioral Representation in Modeling and Simulation (BRIMS)*.

Barchini, G. E., Sosa, M., & Herrera, S. (2004). La informática como disciplina científica. Ensayo de mapeo disciplinar. *Revista de Informática Educativa y Medios Audiovisuales*, 1(2), 1-11.

Bechtel, W. (2012). *Mental mechanisms: Philosophical perspectives on cognitive neuroscience*. Psychology Press.

Boley, H., Paschke, A., & Shafiq, O. (2010). RuleML 1.0: the overarching specification of web rules. *International Workshop on Rules and Rule Markup Languages for the Semantic Web*, 162–178.

Borst, J. P., & Anderson, J. R. (2015). Using the ACT-R Cognitive Architecture in combination with fMRI data. In *An introduction to model-based cognitive neuroscience* (pp. 339–352). Springer.

Caro, Manuel F., Josvula, D. P., Gomez, A. A., & Kennedy, C. M. (2018). Introduction to the CARINA Metacognitive Architecture. *Proceedings of 2018 IEEE 17th International Conference on Cognitive Informatics and Cognitive Computing, ICCI*CC 2018*, 530–540. https://doi.org/10.1109/ICCI-CC.2018.8482051

Caro, Manuel F, & Jiménez, J. A. (2014). MOF-based metamodel for pedagogical strategy modeling in Intelligent Tutoring Systems. *2014 9th Computing Colombian Conference (9CCC)*, 1–6.

Caro, Manuel 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 Architectures*, *9*, 82–104.

Caro, Manuel F, Josyula, D. P., Jiménez, J. A., Kennedy, C. M., & Cox, M. T. (2015). A domain-specific visual language for modeling metacognition in intelligent systems. *Biologically Inspired Cognitive Architectures*, *13*, 75–90.

Caro, Manuel Fernando, Josyula, D. P., Madera, D. P., Kennedy, C. M., & Gómez, A. A. (2019). The CARINA Metacognitive Architecture. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, *13*(4), 71–90.

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

Cox, M. T., Oates, T., & Perlis, D. (2011). Toward an Integrated Metacognitive Architecture. *AAAI Fall Symposium: Advances in Cognitive Systems*.

Flórez, Y. P., Jerónimo, A. J., Castillo, M. E., & Gómez, A. A. (2019). User-Based Cognitive Model in NGOMS-L for the Towers of Hanoi Algorithm in the Metacognitive Architecture CARINA. *The International Conference on Advances in Emerging Trends and Technologies*, 473–484.

Ford, K. M., & Bradshaw, J. M. (1993). Introduction: Knowledge acquisition as modeling. *International Journal of Intelligent Systems*, *8*(1), 1–7.

Georgeff, M. (1988). A theory of action for multiagent planning. In *Readings in Distributed Artificial Intelligence* (pp. 205–209). Elsevier.

Gerasimou, S., Vogel, T., & Diaconescu, A. (2019). Software Engineering for Intelligent and Autonomous Systems: Report from the GI Dagstuhl Seminar 18343. *ArXiv Preprint ArXiv:1904.01518*.

Ghasemzadeh, M. (2010). *Constructing Semantic Knowledge Model based on Children Dictionary*.

Glass, J. M. (2008). Fibromyalgia and cognition. *The Journal of Clinical Psychiatry*, *69*, 20–24.

Gobet, F., Lane, P. C. R., Croker, S., Cheng, P. C. H., Jones, G., Oliver, I., & Pine, J. M. (2001). Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, *5*(6), 236–243.

Goela, V., Pullara, D., & Grafman, J. (2001). A computational model of frontal lobe dysfunction: Working memory and the Tower of Hanoi task. *Cognitive Science*, *25*(2), 287–313.

Gonzalez, C., & Gonzalez, C. (2002). *The Role of Cognitive Modeling in Enhancing Dynamic Decisions*. Retrieved April 2, 2018, from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.712.5953

Hatzilygeroudis, I., & Prentzas, J. (2006). Knowledge representation in intelligent educational systems. In *Web-based intelligent e-learning systems: Technologies and applications* (pp. 175–192). IGI Global.

Hatzilygeroudis, I., & Prentzas, J. (2004). Knowledge representation requirements for intelligent tutoring systems. *International Conference on Intelligent Tutoring Systems*, 87–97.

Isern, D., Gómez-Alonso, C., & Moreno, A. (2008). Methodological development of a multi-agent system in the healthcare domain. *Communications of SIWN*, *3*, 65–68.

Jensen, R., de Moraes Lopes, M. H. B., Silveira, P. S. P., Ortega, N. R. S., & others. (2012). The development and evaluation of software to verify diagnostic accuracy. *Revista Da Escola de Enfermagem Da USP*.

Jerónimo, A. J., Acosta, K., Caro, M. F., & Rodríguez, R. (2017). Protocolos verbales para el análisis del uso de estrategias metacognitivas en la elaboración de algoritmos por estudiantes de grado sexto de la institución educativa Mercedes Ábrego. In *V congreso Internacional y XIII encuentro nacional de educación en tecnología e informática*.

Jerónimo, A. J., Caro, M. F., & Gómez, A. A. (2018). Formal Specification of cognitive models in CARINA. *2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, 614–619.

John, B. E., & Kieras, D. E. (1996). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *3*(4), 320–351.

Johns, B. T., Taler, V., Pisoni, D. B., Farlow, M. R., Hake, A. M., Kareken, D. A., Unverzagt, F. W., & Jones, M. N. (2018). Cognitive modeling as an interface between brain and behavior: Measuring the semantic decline in mild cognitive impairment. *Canadian Journal of*

Experimental Psychology/Revue Canadienne de Psychologie Expérimentale, *72*(2), 117.

Kieras, D. E. (1999). A guide to GOMS model usability evaluation using GOMSL and GLEAN3. *University of Michigan*, *313*.

Kopp, S., & Bergmann, K. (2017). Using cognitive models to understand multimodal processes: The case for speech and gesture production. *The Handbook of Multimodal-Multisensor Interfaces*, 239–276.

Kotseruba, I., Gonzalez, O. J. A., & Tsotsos, J. K. (2016). A review of 40 years of cognitive architecture research: focus on perception, attention, learning and applications. *CoRR Abs/1610.08602*.

Kronenfeld, D. B. (1978). Scripts, plans, goals, and understanding: an inquiry into human knowledge structures By Roger C. Schank and Robert P. Abelson. *Language*, *54*(3), 779.

Kwasnik, B. H. (1999). *The role of classification in knowledge representation and discovery*.

Lebiere, C. (1999). The dynamics of cognition: An ACT-R model of cognitive arithmetic. *Kognitionswissenschaft*, *8*(1), 5–19.

Levesque, H. J. (1986). Knowledge Representation and Reasoning. *Annual Review of Computer Science*, *1*(1), 255–287. https://doi.org/10.1146/annurev.cs.01.060186.001351

Lieder, F., & Griffiths, T. L. (2019). Resource-rational analysis: understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*, 1–85.

Lieto, A., Bhatt, M., Oltramari, A., & Vernon, D. (2018, May 1). The role of cognitive architectures in general artificial intelligence. *Cognitive Systems Research*, *48*, 1–3. https://doi.org/10.1016/j.cogsys.2017.08.003

López, A. L. E., Calao, Y. M. V., Salgado, A. A. G., & Piñeres, M. F. C. (2018). Validación de un modelo cognitivo basado en M++ para la generación de preguntas Factoid-Wh. *Teknos Revista Científica*, *18*(2), 11–20.

Madera-Doval, D. P., Caro-Piñeres, M. F., Gómez-Salgado, A. A., Cardozo-Soto, A. M., & Jiménez-Builes, J. A. (2018). Design of metacognitive expectations of cognitive functions through ontological representations. *Dyna*, *85*(206), 194–201.

Matta, N., Corby, O., & Prasad, B. (1998). A generic library of knowledge components to manage conflicts in CE tasks. *Concurrent Engineering*, *6*(4), 274–287.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Minsky, M. (1974). *A Framework for Representing Knowledge*.

Muller, T. J., & Heuvelink, A. (2008). *Implementing a Cognitive Model in Soar and ACT - R : A Comparison*. Retrieved April 2, 2018, from https://www.researchgate.net/profile/Annerieke_Heuvelink/publication/228415515_Implementing_a_Cognitive_Model_in_Soar_and_ACT-R_A_Comparison/links/0fcfd50885793794e5000000.pdf

Murray, D. (1985). *Review of Elements of episodic memory*.

Newell, A., & Simon, H. A. (2007). Computer science as empirical inquiry: Symbols and search. In *ACM Turing award lectures* (p. 1975).

Nwana, H. S., & Ndumu, D. T. (1999). Perspective on software agents research. *Knowledge Engineering Review*, *14*(2), 125–142. https://doi.org/10.1017/S0269888999142012

O'Leary, D. E. (1998). Using AI in knowledge management: Knowledge bases and ontologies. *IEEE Intelligent Systems and Their Applications*, *13*(3), 34–39.

Olier, A. J., Gómez, A. A., & Caro, M. F. (2018). Cognitive Modeling Process in Metacognitive

Architecture CARINA. *2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, 579–585.

Paisner, M., Cox, M., Maynord, M., & Perlis, D. (2014). Goal-driven autonomy for cognitive systems. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *36*(36).

Peters, S., & Shrobe, H. E. (2003). Using semantic networks for knowledge representation in an intelligent environment. *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003.(PerCom 2003).*, 323–329.

Pew, R. W., & Mavor, A. S. (1998). *Modeling Human and Organizational Behavior-Application to Military Simulations*.

Pezzulo, G., & Calvi, G. (2004). A Pandemonium Can Have Goals. *ICCM*, 237–242.

Polk, T. A., & Seifert, C. M. (2002). *Cognitive modeling.* MIT Press.

Prezenski, S., Brechmann, A., Wolff, S., & Russwinkel, N. (2017). A cognitive modeling approach to strategy formation in dynamic decision making. *Frontiers in Psychology*, *8*(AUG). https://doi.org/10.3389/fpsyg.2017.01335

Proctor, R. W., & Capaldi, E. J. (2012). *Psychology of science: Implicit and explicit processes*. Oxford University Press.

Rao, A. S., & Georgeff, M. P. (1991). Modeling rational agents within a BDI-architecture. *KR*, *91*, 473–484.

Ritter, F. E., Tehranchi, F., & Oury, J. D. (2019). ACT-R: A cognitive architecture for modeling cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, *10*(3), e1488.

Rus, V., Wyse, B., Piwek, P., Lintean, M., Stoyanchev, S., & Moldovan, C. (2012). A detailed account of the first question generation shared task evaluation challenge. *Dialogue & Discourse*, *3*(2), 177–204.

Russell, S., & Norvig, P. (2002). *Artificial intelligence: a modern approach*.

Scheutz, M. (2001). Computational versus causal complexity. *Minds and Machines*, *11*(4), 543–566.

Schmid, U., Ragni, M., Gonzalez, C., & Funke, J. (2011). *The challenge of complexity for cognitive systems*. Elsevier.

Shi, Z., Zhou, H., & Wang, J. (1997). Applying case-based reasoning to engine oil design. *Artificial Intelligence in Engineering*, *11*(2), 167–172. https://doi.org/10.1016/S0954-1810(96)00029-5

Strube, G. (2000). Generative theories in cognitive psychology. *Theory & Psychology*, *10*(1), 117–125.

Sun, R. (2007). The motivational and metacognitive control in CLARION. *Modeling Integrated Cognitive Systems*, 63–75.

Sun, R. (2008a). Introduction to computational cognitive modeling. *Cambridge Handbook of Computational Psychology*, 3–19.

Sun, R. (2008b). *The Cambridge handbook of computational psychology*. Cambridge University Press.

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

Sun, R., & Helie, S. (2015). Accounting for creativity within a psychologically realistic cognitive architecture. In *Computational Creativity Research: Towards Creative Machines* (pp. 151–165). Springer.

Sun, R., & Naveh, I. (2004). Simulating organizational decision-making using a cognitively realistic agent model. *Journal of Artificial Societies and Social Simulation*, *7*(3).

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

Tanaka, M., Aoyama, N., Sugiura, A., & Koseki, Y. (1995). Integration of multiple knowledge representation for classification problems. *Artificial Intelligence in Engineering*, *9*(4), 243–251. https://doi.org/10.1016/0954-1810(95)00006-5

Tulving, E., & others. (1972). Episodic and semantic memory. *Organization of Memory*, *1*, 381–403.

TURING, I. B. Y. A. M. (1950). Computing machinery and intelligence-AM Turing. *Mind*, *59*(236), 433.

Unsworth, N. (2010). On the division of working memory and long-term memory and their relation to intelligence: A latent variable approach. *Acta Psychologica*, *134*(1), 16–28.

Van Lamsweerde, A. (n.d.). *Goal-Oriented Requirements Engineering: A Guided Tour*. Retrieved April 2, 2018, from http://www.panda.sys.t.u-tokyo.ac.jp/kushiro/ReferencePaper/Requirements engineering/00948567.pdf

VanPatten, B., & Williams, J. (2014). Introduction: The nature of theories. In *Theories in second language acquisition* (pp. 13–28). Routledge.

Wang, Y. (2008a). On concept algebra: A denotational mathematical structure for knowledge and software modeling. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, *2*(2), 1–19.

Wang, Y. (2008b). On contemporary denotational mathematics for computational intelligence. In *Transactions on computational science II* (pp. 6–29). Springer.

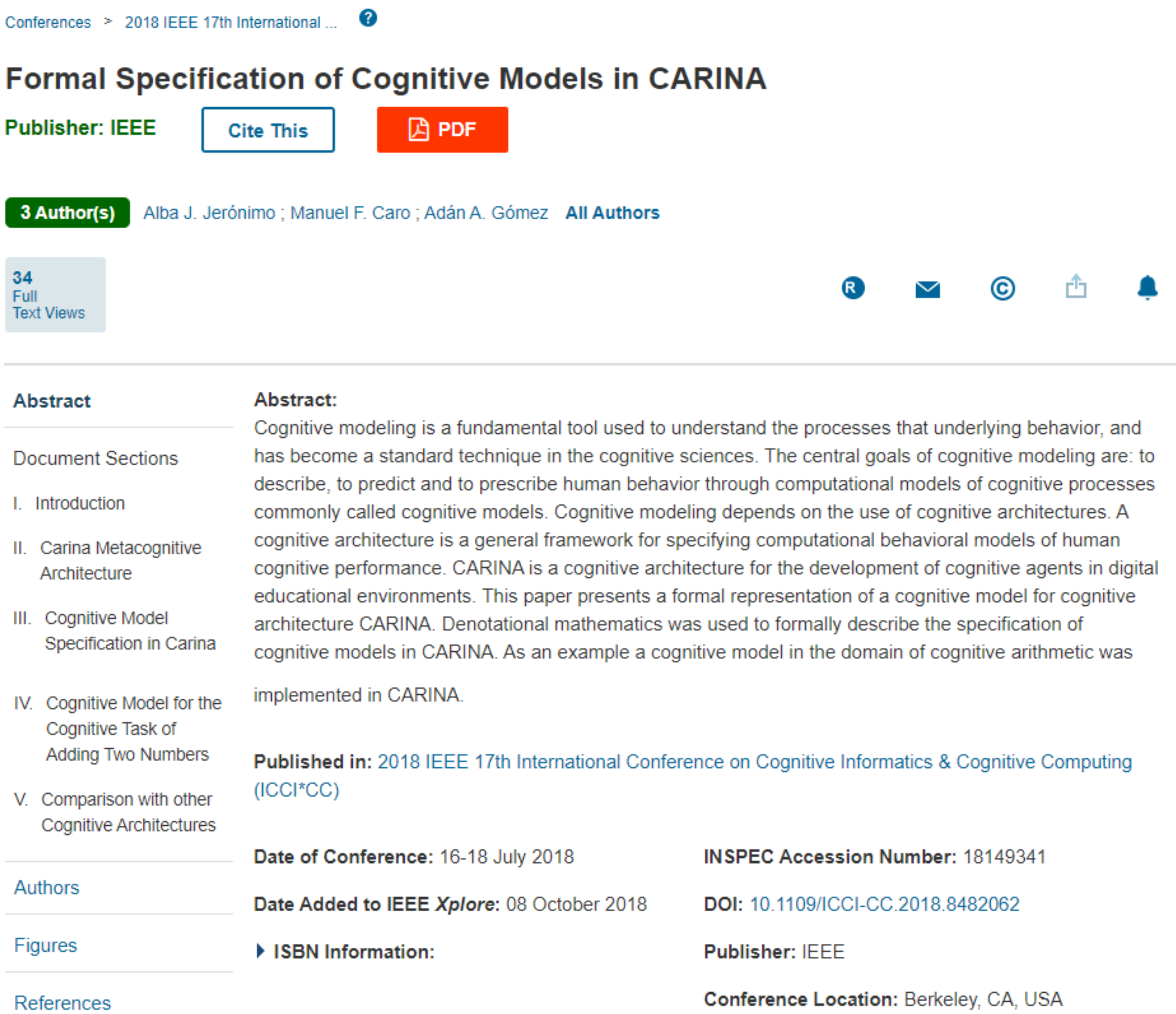
Wong, J. H., Kirschenbaum, S. S., & Peters, S. (2010). *Developing a cognitive model of expert performance for ship navigation maneuvers in an intelligent tutoring system*.

Zhou, G., Zhou, Y., He, T., & Wu, W. (2016). Learning semantic representation with neural networks for community question answering retrieval. *Knowledge-Based Systems*, *93*, 75–83.

Appendixes

Appendix A

Jerónimo, A. J., Caro, M. F., & Gómez, A. A. (2018). Formal Specification of Cognitive Models in CARINA. In 2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI CC) (pp. 614-619). IEEE.*



Conferences > 2018 IEEE 17th International ...

Formal Specification of Cognitive Models in CARINA

Publisher: IEEE   Cite This   PDF

3 Author(s) Alba J. Jerónimo ; Manuel F. Caro ; Adán A. Gómez All Authors

Abstract

Document Sections

I. Introduction

II. Carina Metacognitive Architecture

III. Cognitive Model Specification in Carina

IV. Cognitive Model for the Cognitive Task of Adding Two Numbers

V. Comparison with other Cognitive Architectures

Authors

Figures

References

Abstract:
Cognitive modeling is a fundamental tool used to understand the processes that underlying behavior, and has become a standard technique in the cognitive sciences. The central goals of cognitive modeling are: to describe, to predict and to prescribe human behavior through computational models of cognitive processes commonly called cognitive models. Cognitive modeling depends on the use of cognitive architectures. A cognitive architecture is a general framework for specifying computational behavioral models of human cognitive performance. CARINA is a cognitive architecture for the development of cognitive agents in digital educational environments. This paper presents a formal representation of a cognitive model for cognitive architecture CARINA. Denotational mathematics was used to formally describe the specification of cognitive models in CARINA. As an example a cognitive model in the domain of cognitive arithmetic was implemented in CARINA.

Published in: 2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)

Date of Conference: 16-18 July 2018

Date Added to IEEE *Xplore*: 08 October 2018

ISBN Information:

INSPEC Accession Number: 18149341

DOI: 10.1109/ICCI-CC.2018.8482062

Publisher: IEEE

Conference Location: Berkeley, CA, USA

Appendix B

17th IEEE International Conference on Cognitive Informatics & Cognitive Computing in Berkeley, California. Conference paper certificate.

ICCI*CC'18 IEEE COMPUTER SOCIETY IEEE Computational Intelligence Society

17th IEEE International Conference on Cognitive Informatics & Cognitive Computing
http://www.ucalgary.ca/icci_cc2018/
July. 16-18, 2018, University of California Berkeley, CA, USA

Certificate

July 19, 2018

This document certifies that ***Alba Judith Jeronimo Montiel*** has presented the paper **61 -*"Formal Specification of cognitive models in CARINA"*** at the 17th IEEE International Conference on Cognitive Informatics and Cognitive Computing (IEEE ICCI*CC'18) held at University of California Berkeley, California, USA during July 16-18, 2018.

Registration is handled by the University of Calgary, Canada on behalf of IEEE.

The Organization Committee
17th IEEE International Conference on Cognitive Informatics and Cognitive Computing
(IEEE ICCI*CC'18).

Registration Chair:

Omar Zataram
Dept. of Electrical and Computer Engineering
Univ. of Calgary
2500 Univ. Dr. NW, Calgary, AB T2N 1N4, Canada
Telephone: (403) 220-7596
Fax: (403) 282-6855
Email: omar.zatarainduran@ucalgary.ca

Appendix C

II Workshop 2018: Metacognition Seminar in Monteria – Colombia. Certificate of participation

Appendix D

Chapter proposal approved

eEditorial Discovery
Supporting Your Editorial Experience

Dear Alba Jerónimo,

Thank you for submitting your article proposal, "Formal and Semantic Representation of a cognitive Model in the Metacognitive Architecture CARINA," for the upcoming book, "Encyclopedia of Organizational Knowledge, Administration, and Technologies." After reviewing your recent proposal, I believe the topic of your proposed article is consistent with the theme of this book and could make a significant contribution to this project. I therefore encourage you to begin preparing your full article and submit a copy before Friday, November 30, 2018 at the following URL:

https://www.igi-global.com/submission/submit-chapter/?projectid=6a13edab-268c-4180-b12d-b3657d73d24c

Please note that you will be asked to create an account prior to uploading your article to the system. This is to ensure the security of your work and to assist you in organizing your materials for submission, receiving and providing peer reviews, and making any necessary revisions to your article. For information on creating and accessing your Web account, please see our tutorial at www.igi-global.com/publish/contributor-resources/book-submission-system/video-guide/#creating-a-user-account.

The following documents are provided by the publisher, IGI Global (www.igi-global.com), to aid in the writing of your article:

Mandatory guidelines for your full article submission:
https://www.igi-global.com/publish/resources/encyc-organization-and-formatting.docx

Ensure your article has been professionally copy edited

It is crucial that professional copy editing is conducted prior to submission to ensure proper use of the English language, proper grammatical structure, and correct spelling and punctuation. Submitted articles are considered to be in their final form and ready for publication as is. We recommend eContent Pro for copy editing (**currently offering 10% discount**): https://www.econtentpro.com/copyediting.

Details to keep in mind for your full article submission:
https://www.igi-global.com/publish/contributor-resources/before-you-write/

Image guide:
https://www.igi-global.com/publish/resources/image-guide.pdf

Your adherence to the guidelines provided in these documents is very important. Should you have any questions regarding your proposed article, please do not hesitate to contact me, Jan Travers, at jtravers@igi-global.com.

I look forward to receiving your article!

Sincerely,
Dr. Mehdi Khosrow-Pour, editor
encyclopedia@igi-global.com

Appendix E

Update on Encyclopedia of Information Science and Technology, 5th ed. (New tentative release date)

Jan Travers dom., 12 abr. 11:29
para mí

inglés > español Traducir mensaje Desactivar para: inglés ×

Dear Prof. Jerónimo

Greetings. I hope this e-mail finds you safe and in good health. As you were previously notified, your chapter that successfully underwent a double-blind peer review process, was formally accepted for publication in the Encyclopedia of Information Science and Technology, 5th edition. We are still intending to proceed to publication with the encyclopedia, however, I wanted to note that the original anticipated release date of Ju 2020 has been shifted slightly. Unfortunately, due to the global pandemic and with many organizations (libraries, booksellers, research centers, indices, printers etc.) around the world curtailing activities until the recovery occurs we are notifying you that the new tentative release date is now set for **September 1, 2020.**

This new tentative release date will allow IGI Global to better promote the publication (with your research) once libraries worldwide reopen and return to their mission of providing the most up-to-date reference sources for their library users.

Please note that if there were any items that the development staff had previously reached out to you on requiring further attention/revision, please continue forward with sending those materials over as it is very important that we ensure that the book is able to be released without delay when the time comes.

Should you have any questions, please do not hesitate contacting me.

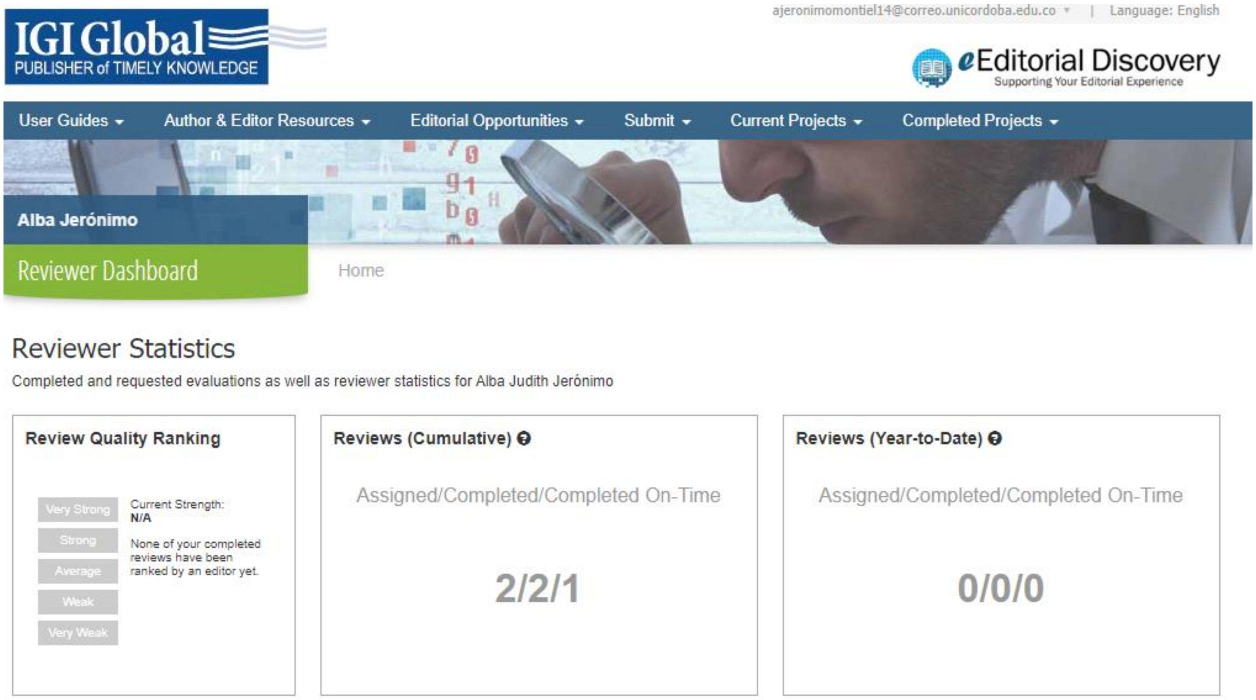
Sincerely,
Jan Travers
Director of Intellectual Property and Contracts

Sent on behalf of Dr. Mehdi Khosrow-Pour, DBA
Editor-in-Chief of the Encyclopedia of Information Science and Technology, 5th edition

(Ms) Jan Travers
Director of Intellectual Property & Contracts
IGI Global
IGI Global DISSEMINATOR OF KNOWLEDGE
PUBLISHER of TIMELY KNOWLEDGE
701 East Chocolate Avenue
Hershey, Pennsylvania 17033-1240, USA
Tel: 717-533-8845 (ext. 112); Fax: 717-533-8661
E-Mail: jtravers@igi-global.com
www.igi-global.com

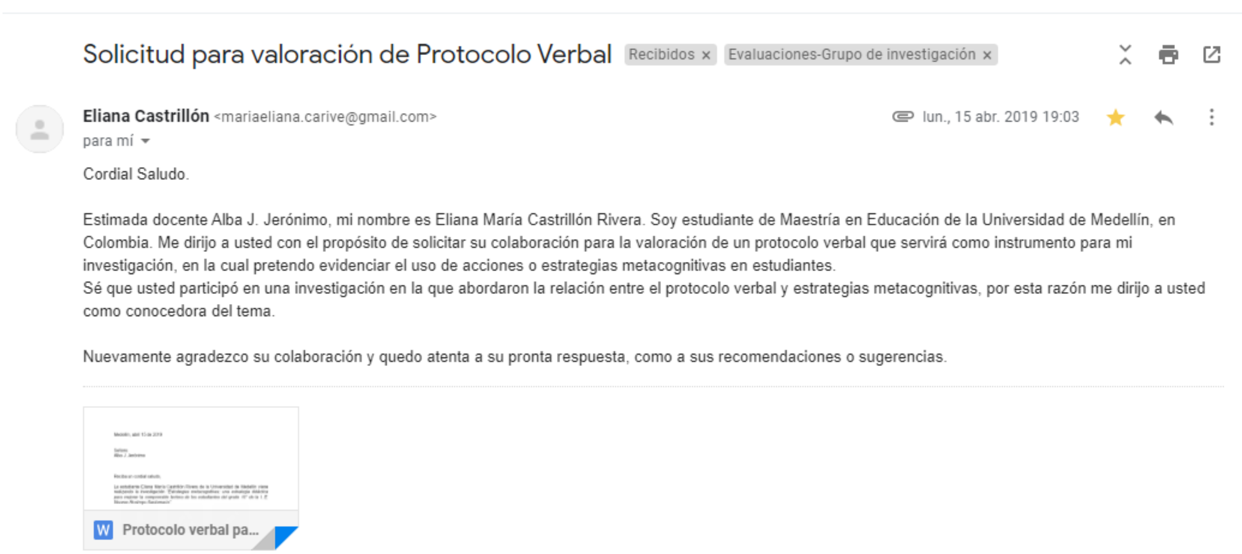
Appendix F

Completed and requested evaluations as well as reviewer statistics in IGI GLOBAL



Appendix G

Completed and requested evaluations. Verbal Protocol which was used as an instrument for research in Master's studies at the Universidad de Medellín, Colombia



Appendix H

MetaThink V2.0 Assessment Instrument. Jensen, R., de Moraes Lopes, M. H. B., Silveira, P. S. P., Ortega, N. R. S., & others. (2012). The development and evaluation of software to verify diagnostic accuracy. Revista Da Escola de Enfermagem Da USP

METATHINK ASSESSMENT INSTRUMENT

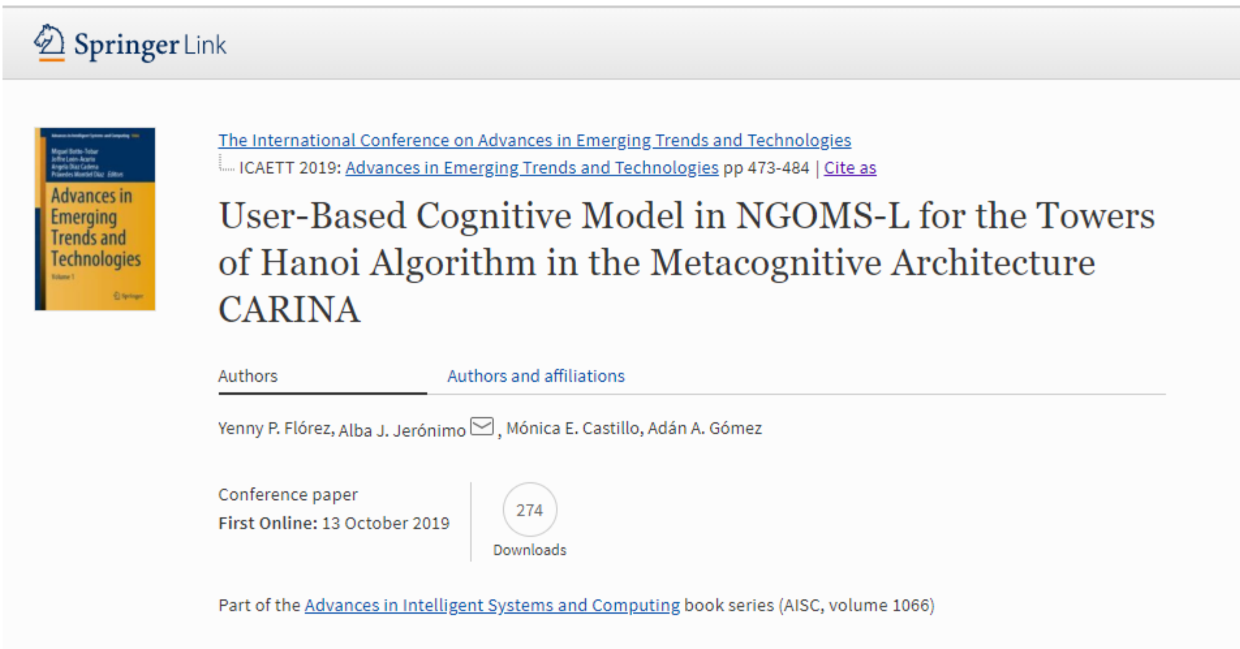
Item	Score			Recommendation
	A	D	NA	
FUNCTIONALITY				
The software does what is appropriate (adequacy)				
The software has available all the functions required for its execution (adequacy)				
The software does what was proposed correctly (accuracy)				
The software is precise in executing its functions (accuracy)				
The software is precise in its results (accuracy)				
The software interacts with the specified modules (interoperability)				
The software has capacity for multiuser processing (interoperability)				
The software has capacity to operate with networks (interoperability)				
The software complies with standards, laws, etc. (conformity)				
The software has secure access through passwords (secure access)				
RELIABILITY				
The software has frequent failures (maturity)				
The software reacts appropriately when failures occur (tolerance to failures)				
The software informs users concerning invalid data entry (tolerance to failures)				
The software is capable of recovering data in the event of failure (recoverability)				
USABILITY				
It is easy to understand the concept and application (intelligibility)				
It is easy to perform its functions (intelligibility)				
It is easy to learn how to use (learnability)				
The software facilitates the user's data entry (learnability)				
The software facilitates the user's retrieval of data (learnability)				

It is easy to operate and control (operability)				
The software provides help in a clear manner (operability)				
EFFICIENCY				
The software's response time is appropriate (time)				
The software's excecution time is appropriate (time)				
The resources used are appropriate (resources)				
MAINTAINABILITY				
It is easy to find a failure, when it occurs (analyzability)				
It is easy to modify and adapt (modifiability)				
There is a great risk when changes are made (stability)				
Changes are easy to test (testability)				
PORTABILITY				
It is easy to adapt to other environments (adaptability)				
It is easy to install in other environments (capacity to be installed)				
It is in agreement with portability standards (conformity)				
It is easy to use to replace another program (capacity to replace)				
Legend: Agreement (A), Disagreement (D) and Does not apply (NA)				

General recommendations

Appendix I

Flórez, Y. P., Jerónimo, A. J., Castillo, M. E., & Gómez, A. A. (2019, March). User-Based Cognitive Model in NGOMS-L for the Towers of Hanoi Algorithm in the Metacognitive Architecture CARINA. In The ICAETT. DOI: https://doi.org/10.1007/978-3-030-32022-5_44


SpringerLink

The International Conference on Advances in Emerging Trends and Technologies
ICAETT 2019: Advances in Emerging Trends and Technologies pp 473-484 | Cite as

User-Based Cognitive Model in NGOMS-L for the Towers of Hanoi Algorithm in the Metacognitive Architecture CARINA

Authors | Authors and affiliations

Yenny P. Flórez, Alba J. Jerónimo, Mónica E. Castillo, Adán A. Gómez

Conference paper
First Online: 13 October 2019

274 Downloads

Part of the Advances in Intelligent Systems and Computing book series (AISC, volume 1066)

Appendix J .

Certificate of participation in: 1st International Conference on Advances in Emerging Trends and Technologies ICAETT 2019

I CONGRESO INTERNACIONAL SOBRE AVANCES EN NUEVAS TENDENCIAS Y TECNOLOGÍAS

CERTIFICATE OF PARTICIPATION

This is to certify that

Alba Jerónimo

Has participated in the **1st International Conference on Advances in Emerging Trends and Technologies ICAETT 2019**, held on May 29 - 31, 2019 at Universidad Tecnológica Israel, Quito - Ecuador, and orally presented the paper entitled: **User-Based cognitive model in NGOMS-L for the Towers of Hanoi algorithm in the Metacognitive Architecture CARINA**

Miguel Botto-Tobar
Organizing Committee
SNOTRA S.A.

Joffre León-Acurio
Organizing Committee
SNOTRA S.A.

Marcelo Zambrano Vizuete
Organizing Committee
Universidad Técnica del Norte

Wladimir Paredes
Organizing Committee
Instituto Superior Tecnológico Rumiñahui

Paúl Baldeón Egas
Organizing Committee
Universidad Tecnológica Israel

Appendix K

Certificate of participation in: "Primer congreso Nacional de Semilleros de Investigación y Emprendimiento"


UNIVERSIDAD DE CÓRDOBA

VICERRECTORÍA DE INVESTIGACIÓN Y EXTENSIÓN

COMITE CENTRAL DE SEMILLEROS DE INVESTIGACIÓN

Primer Congreso Nacional de Semilleros de Investigación y Emprendimiento

Certifica que:

ALBA JUDITH JERONIMO MONTIEL

Identificado con:

Codigo de verificacion: 49b96061bdc16a9d1214659d817a5454

Participó en calidad de Ponente en el marco del:

PRIMER CONGRESO NACIONAL DE SEMILLEROS DE INVESTIGACION Y EMPRENDIMIENTO

Realizado en la Universidad de Córdoba, los dias 17 y 18 de octubre de 2019

Montería, Córdoba.

GILMAR SANTAFÉ PATIÑO

Vicerrector de Investigación y Extensión

ELIECER CARABALLO H.

Coordinador General

Acreditada INSTITUCIONALMENTE

Para verificar la autenticidad de este certificado ingrese el código de verificación en http://campusvirtual.unicordoba.edu.co/inscripciones/certificados

Appendix L

V Encuentro Interinstitucional de Semilleros de Investigación 2020. Mayo 12 y 13

Lineamientos para presentación de ponencias

Recibidos × Research Group × Semillero Intelligent Revolution ×

Cuarto encuentro interinstitucional de semilleros 28 abr. 2020 13:46

para

Buenas tardes estimado investigador,

Te agradecemos por participar en el 5 encuentro de Semilleros de Investigación de la Fundación Universitaria Horizonte, cómo se menciono anteriormente el evento se realizará de manera virtual, próximamente te enviaremos la programación con las instrucciones de acceso a las salas.

A continuación te enviamos los parametros para la presentación de tu ponencia:

GENERAL

En principio, tu ponencia puede contener la cantidad de diapositivas que desees, siempre que se respete el tiempo de la presentación que será de máximo 20 minutos.

ESPECIFICACIONES TÉCNICAS:

3:30-4:00	
	"Desarrollo de modelos cognitivos para la arquitectura metacognitiva carin"
4:00-4:10	*Universidad de Córdoba*

Appendix M

Research project sponsored by the Universidad de Córdoba

	UNIVERSIDAD DE CÓRDOBA	**CÓDIGO:** FINV-011 **VERSIÓN:** 02 **EMISIÓN:** 16/02/2012 **PÁGINA** 1 de 21
	PRESENTACIÓN DE PROYECTOS DE INVESTIGACIÓN	

INFORMACIÓN DE LA CONVOCATORIA	
NOMBRE DE LA CONVOCATORIA	**AÑO**
Convocatoria interna para los semilleros de investigación y emprendimiento	**2019**

INFORMACIÓN GENERAL DEL PROYECTO	
TÍTULO DEL PROYECTO	Modelo de representación semántica de elementos cognitivos para la arquitectura metacognitiva CARINA
DURACIÓN (MESES)	10 meses
LÍNEA DE INVESTIGACIÓN INSTITUCIONAL	Educación Cultura y Calidad de vida
GRUPO DE INVESTIGACIÓN Y CÓDIGO	EDUTLAN (Education, Technology and Language) Código Colciencias COL0065564

INFORMACIÓN DE INVESTIGADORES				
	NOMBRE	**CÉDULA**	**TELÉFONO**	**E-MAIL**
INVESTIGADOR PRINCIPAL	Adán Alberto Gómez Salgado			aagomez@correo.unicordoba.edu.co
COINVESTIGADOR	María Paula Barrera Agámez			mariapaulabarreraagamez@gmail.com
COINVESTIGADOR	Alba Judith Jerónimo Montiel			ajeronimomontiel14@correo.unicordoba.edu.co
COINVESTIGADOR	Laura Andrea Márquez García			lauramarquezg@correo.unicordoba.edu.co
COINVESTIGADOR	Heider José Zapa Reyes			hzapareyes04@correo.unicordoba.edu.co