**RESEARCH ARTICLE**

# A domain-specific visual language for modeling metacognition in intelligent systems

Manuel F. Caro [a,*], Darsana P. Josyula [b], Jovani A. Jiménez [c], Catriona M. Kennedy [d], Michael T. Cox [e]

[a] *Grupo de Investigación en Informática y Computación Cognitiva, Departamento de Informática Educativa, Universidad de Córdoba, Montería, COR, Colombia*
[b] *Department of Computer Science, Bowie State University, Bowie, MD, USA*
[c] *Departamento de Ciencias de la Computación y la Decisión, Universidad Nacional de Colombia, Medellín, ANT, Colombia*
[d] *School of Computer Science, University of Manchester, Manchester, United Kingdom*
[e] *Wright State Research Institute, Beavercreek, OH 45431, USA*

**Abstract**
Metacognition has been used in artificial intelligence to increase the level of autonomy of intelligent systems. However the design of systems with metacognitive capabilities is a difficult task due to the number and complexity of processes involved. This paper presents a domain-specific visual language specifically developed for modeling metacognition in intelligent systems called M++. In M++ the specifications of the cognitive level (object-level) and metacognitive level (meta-level) are supported in a metamodel configured according to the standard Meta-Object Facility (MOF) of Model-Driven Architecture (MDA) methodology. M++ allows the generation of metacognitive diagrams in a visual editor named *MetaThink*. A validation process was conducted to ensure the reliability of M++ in terms of quality of the notation and consistency of generated models. The validation was performed using two techniques: (i) empirical study and (ii) model tracing. The results given in the experimental study demonstrate that M++ is a useful notation for the process of modeling metacognitive components in intelligent systems. Metacognitive models generated from the validation process using the Tracing technique were consistent with the MOF-based metamodel. M++ contribute to cognitive architecture

* Corresponding author.
  *E-mail addresses:* mfcarop@unal.edu.co (M.F. Caro), darsana@cs.umd.edu (D.P. Josyula), jajimen1@unal.edu.co (J.A. Jiménez), catriona.kennedy@manchester.ac.uk (C.M. Kennedy), michael.cox@wright.edu (M.T. Cox).

research adding precision to metacognitive concepts and enabling cognitive architecture researchers to do fast and exploratory prototyping of metacognitive systems using *MetaThink* tool.

## Introduction

Metacognition from cognitive science is defined as mental awareness and regulation of one's thinking (Cox, 2005; Koedinger, Aleven, Roll, & Baker, 2009). Metacognition has two executive processes that the subject performs over its cognitive processes; these processes are the monitoring and control (Anderson et al., 2008). The term metacognition in Artificial Intelligence (AI) refers to the ability of intelligent systems to monitor and to control their own learning and reasoning processes (Anderson, Oates, Chong, & Perlis, 2006; Cox & Raja, 2012; Singh, 2005); therefore, metacognition in AI is often referred by some authors as meta-reasoning (Anderson et al., 2008; Cox, 2005).

Metacognition has been widely used in AI for designing robust systems capable of both detecting anomalies in their own reasoning and to generate strategies to solve them using two metacognitive mechanisms: *introspective monitoring* and *meta-level control*. Introspective monitoring includes mechanisms for detecting reasoning failures and Meta-level control includes rules for the recommendation of the best strategy available to correct the reasoning failures. This work focuses on the modeling the metacognitive mechanisms used to monitor and control the reasoning processes in an intelligent system. The design of an intelligent system with metacognitive support is a difficult task due to the number and complexity of processes involved (Caro, Josyula, & Jimenez, 2013; Cox, 2005).

The modeling process is even more complicated because there is no standard model for the computational design of metacognitive processes. The design of systems based on available metacognitive models is difficult, since some models are theoretical (Cox, Oates, & Perlis, 2011; Nelson & Narens, 1990) and the computational models (Alonso, Arnold, & Havasi, 2010; Kennedy, 2010; Schmill et al., 2011) do not provide formalisms of software engineering methodologies that allow systematic development. The variety of existing approaches to computational design of metacognitive processes delays the selection process of the design model for any application. These problems confirm the lack of an appropriate notation for designing and modeling intelligent systems with metacognitive support.

To address the issues pointed out, we defined a notation included in a Domain-Specific Visual Language (DSVL) for modeling metacognitive processes. A Domain-Specific Language (DSL) has notations and constructs focused on a particular application domain (e.g. Metacognitive Systems). DSVLs have a declared syntax composed of an abstract syntax and a concrete syntax. The abstract syntax describes the concepts and their relations to other concepts without any consideration of meaning. Concrete syntax is the set of notations that facilitates the presentation and construction of the language; graphical syntax offers benefits, like easier design, when modeling within certain domains.

The main objective of this paper is to introduce a novel DSVL for modeling metacognition in an intelligent system. The DSVL is called *M++* and has a core based on a metacognitive metamodel named *MISM* (Caro, Josyula, Cox, & Jiménez, 2014). *MISM* is configured according to the standard Meta-Object Facility (MOF) (OMG., 2013) of Model-Driven Architecture (MDA) methodology. MDA is an approach from the Object Management Group's (OMG) (OMG, 2013) for the development of model-driven software. In MDA, the development of a system is viewed as a sequence of transformations and model refinement (Chitforoush, Yazdandoost, & Ramsin, 2007). MOF standard provides a sequence of transformations and refinement of models. The transformations allow designers to have general, easily adaptable schemes to integrate metacognitive components in the system design process. MDA architecture is based on meta-modeling of four layers: (i) meta—Meta-modeling layer, which corresponds to MOF and defines an abstract language for metamodel specification; (ii) metamodel layer, which consists in metamodels that are defined in the standard MOF; (iii) layer model, which includes real-world models; (iv) the layer of ''real world'' which includes real-world things.

For the validation process, two metacognitive systems present in the literature were modeled using *M++*. All concepts in the original systems could be represented. Artifacts from the models generated with *M++* were consistent with the concepts of *MISM*.

This paper is structured as follows. Section 'M++: A visual language for modeling metacognition' describes the architectural components of *M++*. Section 'Validation' presents the results of validation process. Finally, the paper concludes with a brief discussion.

## M++: A visual language for modeling metacognition

M++ is a DSVL for modeling metacognition in intelligent systems and incorporates two meta-reasoning mechanisms, i.e., introspective monitoring and meta-level control. In M++, the abstract syntax is specified with MOF-based metamodels and the concrete syntax is expressed by some mapping of the abstract syntax elements to visual constructs. The main artifacts of M++ are models specified in a visual manner. M++ is available on the website: https://sites.google.com/site/rgcicc/home.