

DOI: 10.20098/j.cnki.1001-5132.2025.0105

An update on the MECA cognitive architecture

BORELLI José Renato^{1,2}, GUDWIN Ricardo Ribeiro^{1*}

(1.University of Campinas, Campinas 13083-970, Brazil;

2.Federal Institute of São Paulo, Hortolândia 13183-000, Brazil)

Abstract: The Multipurpose Enhanced Cognitive Architecture (MECA) is a cognitive framework designed to model complex, human-like processes across multiple domains. Originally focusing on implementing a Dual Process Theory approach and integrating a machine consciousness mechanism based on Global Workspace Theory, MECA has been updated to integrate a dual-layer subsumption mechanism, enabling both reactive and deliberative behaviors, dynamic goal setting and a visual-spatial memory subsystem, enhancing MECA's capacity for real-world interaction and adaptive behavior. Also, with the introduction of the new computational ideas' knowledge representation scheme, MECA proposes to organize knowledge dynamically to handle context-sensitive reasoning and flexible categorization. MECA's implementation relies on the Cognitive Systems Toolkit (CST), facilitating its integration with cutting-edge technologies. MECA and CST are being continuously developed and updated, aligned, and open to incorporate the latest AI artifacts and methodologies. This approach ensures the delivery of organized, monitorable, auditable, and controllable AI solutions, significantly reducing reliance on "black box" cognitive processes while enhancing transparency and accountability in AI-driven systems. These updates reinforce MECA's potential as a robust architecture for developing autonomous, adaptable, and context-aware AI systems capable of real-world interaction and adaptive learning.

Key words: cognitive architecture; dual-process theory; dynamic subsumption; cognitive systems toolkit

CLC number: TP18

Document code: A

Article ID: 1001-5132 (2025) 02-0029-20

1 Introduction

Cognitive architectures are general-purpose control systems inspired by theories explaining cognition in humans and animals^[1]. The purpose of such architectures is to simulate cognitive capabilities - such as perception, attention, memory, reasoning, and behavior generation - by decomposing these processes and facilitating their

integration within computational models^[2]. As both a theoretical model and a computational framework, cognitive architectures support diverse applications, including robotics, games, human performance modeling, human-machine interaction, natural language processing, and virtual agents^[3].

Over the past few decades, numerous cognitive architectures have emerged, leading to several

Received date: 2025-01-05 JOURNAL OF NINGBO UNIVERSITY (NSEE): <http://journallg.nbu.edu.cn/>

Foundation item: Supported by the São Paulo Research Foundation (FAPESP), CPE SMARTNESS (2021/00199-8) and CEPID/BRAINN (2013/07559-3) .

Biography: BORELLI José Renato, PhD student/professor, specializing in: cognitive architectures, cognitive systems, intelligent systems, artificial minds, robotics, etc. Email: jrborelli@ifsp.edu.br

***Corresponding author:** GUDWIN Ricardo Ribeiro, associate professor, specializing in: cognitive architectures, cognitive systems, computational semiotics, intelligent systems and artificial minds. Email: gudwin@unicamp.br

comparative reviews and benchmarks^[3-5]. Our early work focused on examining popular architectures^[6] and developing the Cognitive Systems Toolkit (CST), an open-source software toolkit to support the modular cognitive model construction^[2]. This toolkit enabled the creation of MECA, the Multipurpose Enhanced Cognitive Architecture^[7], a general-purpose cognitive framework influenced by established models like SOAR^[8], Clarion^[9] and LIDA^[10], using our CST^[2] as a core.

In designing MECA, we incorporated valuable insights from these architectures, such as the use of codelets (as in LIDA) and elements inspired by Global Workspace Theory^[11] to enable machine consciousness. We included dual knowledge representations, explicit and implicit, adopting rule-based processing (as in SOAR) and a subsumption-like structure^[12-15], which supports implicit processes with the option of integrating neural networks such as HTM^[16]. This approach reflects the influence of dual-process theories and hybrid models, such as SAL (Synthesis of ACT-R and Leabra)^[17], which combines rule-based and neural network architectures to enhance cognitive capabilities.

Since its original presentation, in 2017, MECA has gained new structures and received changes in its architectural conception, which are reported now in this work^[7]. Recent advancements in MECA integrate a dual-layer subsumption mechanism, enabling both reactive and deliberative behaviors, dynamic goal setting and a visual-spatial memory subsystem. These enhancements allow MECA to respond more adaptively in real-world applications, especially in mobile and humanoid robotics, where despite the flexibility of subsumption mechanisms, some sort of planning is essential. Additionally, MECA introduced a knowledge representation scheme based on computational ideas^[18], enabling dynamic knowledge organization and flexible categorization to manage context-sensitive reasoning. By grounding knowledge representation in frameworks like grounded cognition^[19] and conceptual spaces^[20], MECA enhances its capability to represent, prioritize, and adapt knowledge beyond traditional symbolic

processing and neural network models.

The subsequent sections detail the updated design and capabilities of MECA, reviewing the theoretical background in section 2, detailing on the main architecture specification in section 3, elaborating on what is missing and the next steps for MECA in section 4 and concluding in section 5.

2 Foundational background

2.1 Dual Process Theory

Dual Process Theory^[21-28] is a prominent framework in cognitive psychology, explaining higher cognition through two distinct systems: System 1 and System 2^[21]. These systems differ in speed, consciousness, and functionality, offering complementary perspectives on how humans process information and make decisions.

System 1 is fast, automatic, and largely unconscious^[22], encompassing both instinctive behaviors and learned responses. It operates in parallel and reacts swiftly to environmental stimuli, providing immediate, default responses aligned with broad goals. Shared across humans and animals, System 1 represents a collection of autonomous subsystems rather than a single unified system.

System 2, in contrast, is slow, deliberative and conscious. It enables abstract thinking, problem-solving, and future planning, often leveraging past experiences to construct mental models. Unique to humans and evolutionarily recent, it relies heavily on central working memory and facilitates decision-making in complex or exceptional scenarios. System 2 also plays a regulatory role, inhibiting System 1's default responses when necessary to address specific, time-sensitive goals^[22].

The interaction between these systems defines overall behavior, with System 1 providing rapid reactions and System 2 modulating them for refined, context-sensitive outcomes. This dynamic integration addresses potential conflicts, balancing efficiency with

adaptability.

While current models broadly describe these systems, challenges remain in developing precise computational architectures to simulate their interaction and conflict resolution. Future research aims to deepen understanding of how System 2 overrides System 1 to counteract cognitive biases, offering insights into both psychology and engineering applications for autonomous systems.

2.2 Subsumption architecture

The subsumption architecture, introduced by Rodney Brooks in the 1990s, revolutionized intelligent control systems, particularly in robotics with Behavior-based Robotics^[29]. It emerged as an alternative to the traditional AI paradigm, which relied on a sequential pipeline for perception, modeling, planning, and execution. While effective for small-scale problems, such systems struggled with scalability as the number of behaviors increased, resulting in excessive complexity and inefficiency. It provides, though, the following key features:

Modularity and parallelism: The architecture emphasizes independently developed behaviors to be integrated into a parallel system where behaviors compete for control of actuators. This design ensures robustness and adaptability in dynamic environments.

Reactive control: Instead of building complex internal models, the architecture focuses on immediate and responsive interactions with the environment.

Limitations and dynamic subsumption: The classical Subsumption Architecture employs fixed priorities among behaviors, which can lead to inefficiencies in situations requiring context-specific adjustments. To overcome this, dynamic subsumption was introduced^[13-15]. This enhancement allows the dominance of behaviors to change dynamically based on evaluation tags that assess their relevance in real-time. By dynamically assigning control to behaviors with the highest evaluation tag, the architecture becomes more adaptable to varying conditions. There are different implementations of this dynamical subsumption idea,

but a very simple comprehension of the idea can be taken from Fig. 1. In the left side of this figure, we can see a standard Subsumption Architecture, with its fixed priority among behaviors. On the right side, we can see how a dynamical subsumption can be implemented. Together with the standard control message x_i , there comes together an evaluation tag e_i which is dynamically generated by the own behavior. So, instead of choosing the output value x using a fixed set of priorities, the dynamical mechanism simply chooses the x_i that has the greatest e_i value. This e_i value can be generated dynamically according to the systems' requirements.

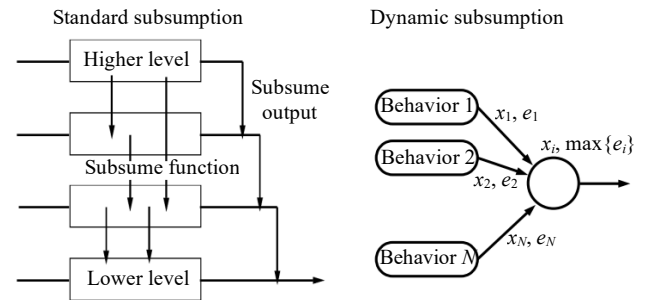


Fig. 1 The dynamic subsumption scheme

2.3 Grounded cognition

Grounded cognition, as theorized by Barsalou^[19, 30-31], challenges traditional symbolic AI approaches that use abstract, amodal symbols defined externally. Instead, it posits that knowledge is rooted in perceptual experiences and actions.

Perceptual symbols: according to Barsalou^[19, 30-31], representations are directly tied to sensory and motor states, stored and reactivated as simulations. There is somewhat of a consensus on the application of perceptual symbol systems theory as an alternative to deal with the symbol grounding problem in computer simulations. This can be viewed by the increasing number of works using this approach to support their model's elaboration (e.g. Narayanan^[32], Bergen et al^[33], Roy^[34], Cangelosi et al^[35], Dominey et al^[36], Madden et al^[37], Lalleo et al^[38], Frank et al^[39] and Pezzulo et al^[40]).

Simulators: dynamic structures that integrate multi-modal sensory information, forming the basis for

understanding and cognition. This theory addresses the symbol grounding problem by connecting abstract symbols to real-world perceptual experiences. Computational approaches like neural networks and fuzzy systems facilitate such grounding by linking directly to sensory data.

2.4 Conceptual spaces

Proposed by Gärdenfors^[20,41], conceptual spaces provide a geometric framework for organizing and understanding knowledge. They describe information, perception, attention, categorization and memory, using dimensions and domains to represent qualities and categories.

Quality dimensions: fundamental metrics like temperature or color, used to define properties within domains.

Objects and categories: objects are points in conceptual spaces, and their categories are collections of regions in these spaces. Unlike symbolic AI, conceptual spaces inherently embed semantics, offering a robust solution to the symbol grounding problem. By bridging perception and cognition, this approach complements theories like Barsalou's perceptual symbol systems while expanding into abstract and lexical semantics.

These theories and frameworks (subsumption architecture, grounded cognition and conceptual spaces) address critical challenges in AI, particularly scalability, adaptability and the symbol grounding problem. Together, they provide a foundation for creating systems that are both computationally efficient and grounded in real world interactions.

2.5 The Cognitive Systems Toolkit

The Cognitive Systems Toolkit, developed by Paraense et al^[2], provides a foundational framework for constructing cognitive architectures like MECA. Its design revolves around two primary components: codelets and memory objects.

Codelets are lightweight, non-blocking pieces of code that execute simple tasks continuously in a cyclic, parallel manner. Inspired by Hofstadter et al^[42]. Copycat model and further developed in Franklin's LIDA

architecture^[43], they align with Baars' Global Workspace Theory^[44]. Codelet-based systems are inherently parallel, with each codelet functioning as an independent agent within the architecture.

Memories act as flexible information holders for storing data and representations essential for cognitive processing. There are two different kinds of memories in CST: memory objects and memory containers. Memory objects are simple information holders. Its main structure comprises: Information (I): can range from simple numbers to complex structures like graphs or networks. Meta-information: Timestamp (T): tracks the object's last update. Evaluation (E): a numeric value for tasks like emotional appraisal or object differentiation. Memory containers are basically an array of memory objects, such that information coming from different codelets do not mix inside the memory. They work as a placeholder for multiple pieces of information of the same kind, that for some reason must be stored together. To restore information stored in a memory container, a policy is used to select the inner memory object within the container and recover its information and meta-information. Memory containers can be used for different purposes. Memories are closely coupled with codelets, storing data they process and any new data they generate.

The memory container is responsible for implementing an important element in the dynamic subsumption mechanism used in MECA, evaluating competing activations, ensuring optimal behavior, see Fig. 2.

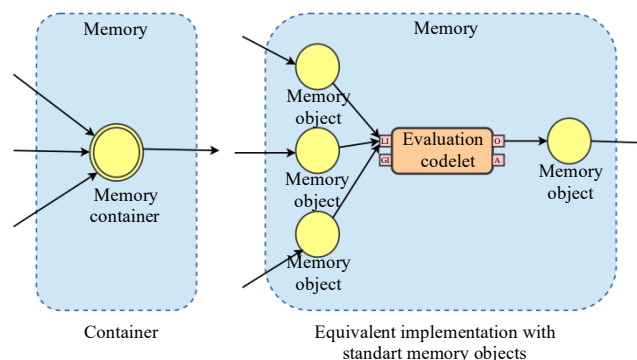


Fig. 2 A memory container and its implementation

One of the advancements in the new MECA is due to a change in CST, which expanded the number of policies used to select content from within a container. In the old implementation, there was only one policy (the max policy), which used the maximum value of E of an internal memory object to select its content. The new implementation included many other policies, including the min, the random flat, the random proportional and the iterate policies.

CST reference cognitive architecture offers a template for organizing codelets and memory objects into functional groups. These groups model cognitive functions such as sensing, perception, attention, emotions and planning.

The framework supports extensibility - architects can easily create new types of codelet and integrate them, and scalability - the architecture can evolve with additional functionalities and modules over time.

CST is an open-source project. The homepage of the CST Project can be found at <http://cst.fee.unicamp.br>, and its source code is available as open source in GitHub at <https://github.com/CST-Group/cst>.

3 The MECA cognitive architecture

The original description of MECA architecture can be found on Gudwin et al^[7]. Since its first presentation, it has been tested in a traffic control application^[45] and a transportation robotics scenario^[46]. In this work, we will focus on the updates and new mechanisms introduced since so far, highlighting them in the overall description.

The Multipurpose Enhanced Cognitive Architecture combines insights from Dual Process Theory, cognitive psychology and advanced computational models. It is divided into two main subsystems: System 1, responsible for automatic unconscious processes, and System 2, which handles deliberate conscious reasoning. This architecture builds on the CST reference architecture^[2] and integrates modular components such as codelets, memories, memory groups and codelet groups.

MECA is organized into a network of interconnected components that perform parallel and sequential operations. Five key elements define its structure, illustrated by Fig. 3.

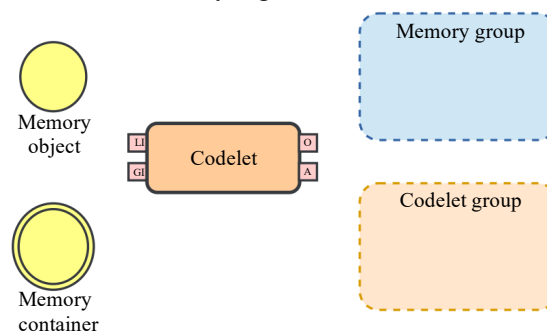


Fig. 3 The different elements in the architecture

Codelets: small, modular processes handling specific tasks in parallel.

Memory objects: simple units for information storage.

Memory containers: specialized memories equivalent to multiple memory objects sharing the same location for exchange of information. Different codelets writing in a memory container will result in their information being stored in different locations. Getting information from a memory container requires a policy for selecting among these multiple inner memory objects.

Memory or codelet groups: used to group memories or codelets of the same type, in order to organize the overall architecture.

An overview of the MECA cognitive architecture can be seen in Fig. 4. All inputs of MECA are performed via System 1, by the sensory codelets, which collects sensory data and processes it through pathways leading to motor responses, motivational activations, or conscious deliberation. Outputs are generated by motor codelets that gather information from motor memory to actuate system behaviors.

System 1 is responsible for performing a more reactive kind of behavior, or simple motivated behavior triggered by drives. Depending on the situation, it might trigger System 2 intervention, by the attention codelets and goal generation codelets.

System 2 handles deliberate reasoning, integrating

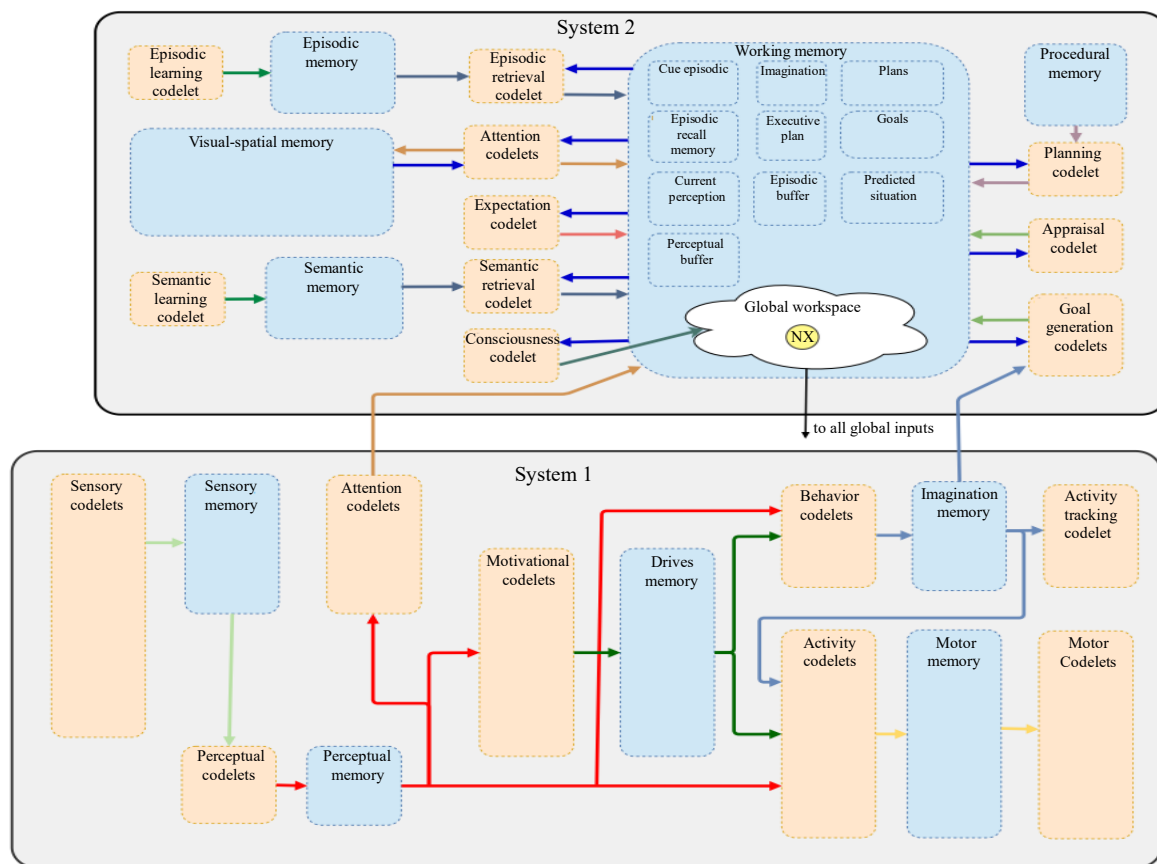


Fig. 4 An overview of the MECA cognitive architecture

inputs from System 1 into conscious, goal-oriented processes.

The integration of Systems 1 and 2 ensures MECA's ability to process sensory inputs, generates motivated behaviors and supports advanced cognitive functions such as episodic memory, learning, spatial localization, planning and problem-solving.

3.1 System 1

System 1 is responsible for fast, parallel, and largely unconscious processing. Its dynamic subsumption architecture prioritizes reactive behavior while allowing for motivational-driven actions. It is comprised by five subsystems: sensory subsystem, perception subsystem, motivational subsystem, attentional subsystem and behavioral subsystem.

Sensory subsystem: includes the sensory codelets and the sensory memory: processes raw sensory input from environment.

Perceptual subsystem: includes the perceptual

codelets and perceptual memory: abstracts sensory data into high-level percepts.

Motivational subsystem: includes the motivational codelets and drives memories: based on the long-term needs, it calculates drives measuring the level of unsatisfaction of each internal need.

Attention subsystem: includes the attention codelets from System 1 and the imagination memory: provides the triggering of System 2, while relevant information is detected, or there is a lack of valid plans to satisfy drives.

Behavioral subsystem: comprises the behavioral, activity, activity tracking and motor codelets, together with the imagination and motor memories: combines activities and plans using a double layer subsumption mechanism, providing action to the environment by the motor codelets.

3.1.1 Sensory subsystem

The sensory subsystem processes raw sensory

input (Fig. 5), collecting information from environment and storing it in the sensory memory. From a cognitive modeling perspective, sensory memory is the raw storage of information coming from visual, auditory, olfactory, tactile and other sensory modalities in a time window which generally spans over something as 50–500 ms^[47]. This memory usually comprises uninterpreted data which is used in the first steps of perception. In human beings, at least two kinds of sensory memory were identified, the iconic memory, storing visual pattern stimulus and the echoic memory, storing auditory stimulus, but there might also be other memories for other senses as well, which were not so widely investigated.

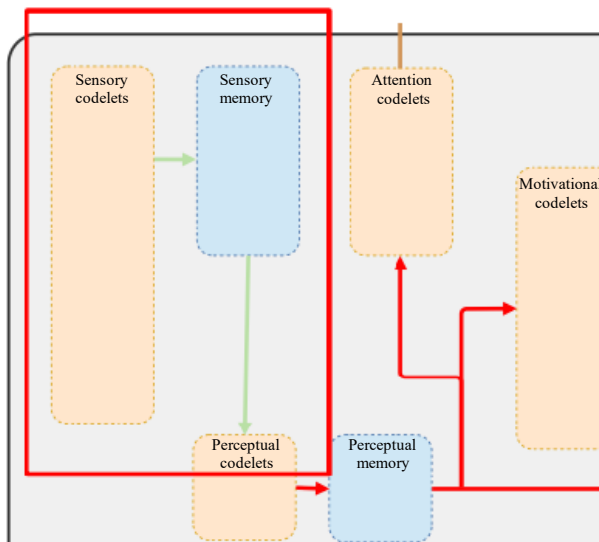


Fig. 5 A zoom on the sensory subsystem

In MECA, this memory stores memory objects carrying direct representations of system sensors. These memory objects can be simple numbers or very complex data structures, representing both scalar sensors or n -dimensional images, according to the information provided by the sensor. It might also store temporal sequences of sensor data, which can be used by perceptual codelets to create more elaborate percepts. More elaborated or derived representations from direct sensory capture are not stored here, but at the perceptual memory. The memory objects stored in the sensory memory are updated by sensory codelets.

Sensory codelets are codelets which are

responsible for grabbing information from sensors at the environment, and feeding the corresponding memory objects which might hold the sensors values. Depending on the applications (e.g. robotic applications), sensory codelets will be really reading the sensor values and creating a corresponding representation. In other applications (as e.g. in a videogame, an internet agent or a virtual world), sensory codelets will open sockets to other computer applications and will simulate the acquisition of data from the environment.

Usually, sensory codelets are application specifics, and the MECA software implementation just provides basic template classes to be reused while building an application using MECA. The sensory subsystem is basically the same as in the original MECA implementation.

3.1.2 Perceptual subsystem

The perceptual subsystem abstracts sensory inputs into higher-level concepts and percepts, capable of being system “understandable” and processable (Fig. 6).

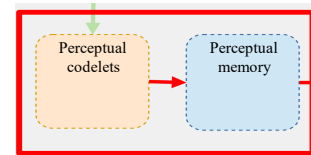


Fig. 6 A zoom on the perceptual subsystem

The role of perceptual subsystem is to generate more elaborate percepts, abstractions of sensory data, which are then used as input, to motivational codelets.

The percepts can also be tracked by attention codelets to detect special situations and send information upstream to System 2. These attention codelets are responsible for generating the current perception at the working memory, where a selected subset of the perception memory is made available for System 2 subsystems in a representation suitable to be processed within System 2. Still toward the motor subsystem on System 1, a faster and more reactive activation is performed from perception directly to the behavior/activity subsystem layer (see the red bus in Fig. 4). Just like in the case of sensory system, the perceptual subsystem is basically the same as in the older MECA

architecture.

3.1.3 Motivational subsystem

The motivational subsystem is responsible for grabbing the percepts from perceptual memory and calculate the drives, which will be used later as input to the behavioral subsystem. These drives, available in drive memory, are direct measures for the unsatisfaction of needs for the system. The System 1 motivational subsystem (Fig. 7), is basically responsible to calibrate and activate reactions to filtered perceptions, considering updated status, needs and goals, originating motivational behaviors^[48-51].

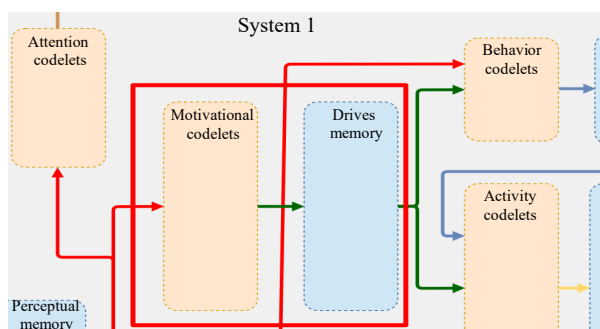


Fig.7 A zoom on the motivational subsystem

When comparing it to the old MECA architecture^[7], the current motivational subsystem was simplified to become more effective. The older MECA architecture used to include moods and emotional codelets, which were used to modulate the drives generated by the motivational codelets. After using MECA in two different study cases^[45-46], we realized that the overall motivational subsystem was too complex, and in most of the cases, moods and emotional codelets were not improving too much the architecture performance. Due to that, we decided to subtract them from the architecture. In the future, if it is necessary, they might return, depending on the new requirements that might appear in future applications of MECA.

3.1.4 Behavioral subsystem

One of the biggest changes in MECA, in the newer version reported here, was in the behavioral subsystem (Fig. 8). The new behavioral subsystem includes a dual-layer subsumption mechanism^[52] that basically provides the architecture with the ability to mix plan-

based sequences of actions to pursue goals with the ability to improvise with reactive actions, when this is necessary. The original behavioral codelets in the old MECA were split into behavior codelets (proper), and activity codelets. In this new configuration, the term behavioral codelets was reserved to the proposal of sequences of activities aiming to drive reduction, together with the help of System 2, which is responsible for the generation of plans. On the other hand, the term activity codelets is reserved for simple actions (called now activities), which work basically as steps to compose more elaborate behaviors. The whole subsystem dynamics works like the following. Each behavior codelet is responsible for a unique drive. It should propose a plan to satisfy this drive (to reduce it), which is stored internally in itself. If it does not currently have a plan, or if the current plan is not feasible anymore, based on the current percepts, the codelet drops a plan request in the imagination memory. This plan request is then directed to System 2 by the goal generation codelets. System 2 is then asked to generate a plan, which is further propagated by the consciousness mechanism, being captured by behavior codelet back, when it is ready. If the current plan is feasible, then it is also sent to the imagination memory, in a different memory container: the plan memory container. All behavior codelets will be proposing different plans to the plan memory container. This plan memory container is the input both for the activity codelets and the activity tracking codelet. The different plans proposed by each behavior codelet will assume different Eval (E) values, depending on the relevance assigned and the current percepts. Using the max policy for selecting options from the container, the best plan for the current situation is selected and returned (in the blue bus).

The full mechanism is described in more detail in [52]. The Activity Tracking Codelet is responsible for comparing the sequence of steps in the plan and deciding which part of the plan the system is currently in. Then, it signalizes, at the own plan, which is the

current plan step to perform. This is traced by the activity codelets, that decide if the plan is mentioning them, and if it is the case, decide to act, sending an output to motor memory.

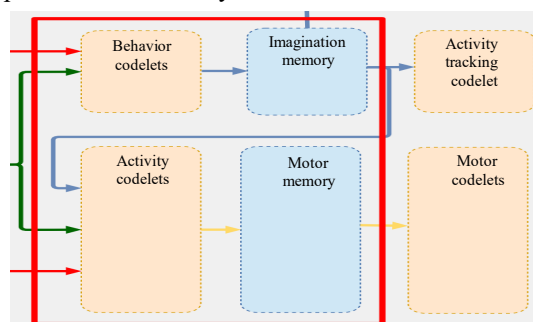


Fig. 8 A zoom on the behavioral subsystem

But the interesting thing is that the blue bus, with the data for the current selected plan, is just one of the inputs of activity codelets. Activity codelets also receive inputs from the drives memory (the green bus) and from the perceptual memory (the red bus). This is one of the most interesting features in the new MECA behavioral system. With that, the activity codelets are able to decide if they will “follow the plan” indicated by the blue bus, or will use the information given by drives and perception to improvise an opportunistic action, bypassing the plan.

3.2 System 2

Different from System 1, which is responsible for fast and automatic behaviors, according to Dual-Process theories, System 2 is responsible for slow, deliberative and conscious processing in the brain. In human beings, it is responsible for abstract thinking, problem-solving, and future planning, often leveraging past experiences to construct mental models.

In MECA, System 2 was built centered on the working memory, which is the location of short-term knowledge constructed with the purpose of storing relevant information in the long-term memories and creating plans requested by System 1.

System 2 was reorganized and updated to comprise basically five different memories. Four of them: episodic memory, visual-spatial memory, semantic memory and procedural memory are long-term

memories. Working memory is, otherwise, a short-term memory. System 2 is responsible for both storing different long-term knowledge, of different kinds, and retrieving these long-term memories for making plans, doing predictions and running the consciousness mechanism, based on Global Workspace Theory^[11]. Most of these were already present in the original MECA. The big novelty here is the inclusion of a visual-spatial memory, responsible for making sense of space and time, at the environment, and positioning objects where the system believes they are located.

The overall process of System 2 in MECA comprises a distributed system running in parallel, where an intricate set of processes are interoperating.

To provide a better understanding of the inner mechanisms in MECA System 2, we will start with a description of the working memory, which is the center for all operations in System 2.

3.2.1 Working memory

Working memory in MECA is inspired by the multi component model proposed by Baddeley^[53-54], consisting of distinct memory systems tailored for specific tasks. MECA expands this model, integrating key features for adaptive decision-making and complex cognitive tasks.

MECA working memory is a split of the following components:

Current perception memory: this is where percepts from System 1 enter System 2. It is fed by the attention codelets from System 1.

Perceptual buffer: this is a sequential buffer using the information from the current perception memory in sequential timesteps. It is used to build episodes further stored at the episodic buffer.

Episodic buffer: this is where the attention codelets build the current happening episode. It comprises an episode that just happened at the environment, and is used by the episodic learning codelet to store new episodes in the episodic memory.

Cue memory: this is where cues are presented to the episodic retrieval codelet in order to search for

episodes in the episodic memory.

Episodic recall memory: this is where episodes retrieved from the episodic memory are made available for immediate use in planning and reasoning.

Predicted situation memory: holds projections generated by the expectation subsystem.

Goals memory: this is where the goal generation codelets store goals to be used in the planning subsystem for making plans.

Plans memory: this is where the many different plans generated by the planning codelet are stored for further selection

Executive plan: this is one of the plans in the plans memory, selected for immediate execution, which is captured by the consciousness codelet and broadcasted back to System 1.

Imagination memory: hosts simulated scenarios created during planning or goal evaluation.

Global workspace: acts as a central interface where prioritized information is broadcast for system wide integration.

3.2.2 Visual-spatial memory

Visual-spatial memory is one of the novelties in this version of MECA. It is basically a list of all known objects that passed through perception in the past, with their supposed location, according to the last time they were seen. Visual-spatial memory works in conjunction with both attention codelets coming from System 1 and attention codelets of System 2. When an object is brought to working memory, attention codelets from System 2 try to detect if this is a new object, or an object that has already been seen in the past. If it recognizes the object as an already known object, its new location is updated in the visual-spatial memory.

One interesting object within the visual-spatial memory is the self object, which positions the body of an embodied system like a robot. The use of a self object is useful while trying to plan specific movements in the environment. Its location can be estimated using SLAM (Simultaneous Localization And Mapping) algorithms.

In some uses of MECA, the objects in the visual-spatial memory might also represent the region of space they occupy. Together with the self object, being used as an observer, this allows the creation of expected views from different perspectives, which can be useful for planning.

Basically, visual-spatial memory can be seen as a giant map that captures everything that is known at the environment, positioned in their expected location in space. Together with the region of space these objects occupy, it can be used to derive inferences regarding spatial relations between different objects.

3.2.3 Attention codelets and the creation of experience

One of the entries in System 2 is due to attention codelets from System 1. By this way, objects recognized by the perception subsystem are introduced in the current perception, at the working memory. After that, a sequence of steps is performed by attention codelets from System 2, which are responsible for the creation of what we call the system's experience.

First, attention codelets use the time sequence of different current perception values to create the perception buffer, which organizes these perceptions in time. This is made in parallel with the work these codelets also perform in actualizing the visual-spatial memory.

Later, this perceptual buffer is also used by attention codelets to detect episodes happening at the environment and put them on the episodic buffer.

Finally, episodes on the episodic buffer will compete for going to consciousness by the consciousness subsystem.

3.2.4 Episodic subsystem

Episodic memory is a neurocognitive mechanism for accessing time delimited contextualized information that naturally makes part of the human process of decision making, usually enhancing the chances of a successful behavior. This assertion is supported by several human psychological pieces of research, which indicate that the knowledge of his/her personal history enhances one's person ability to accomplish several cognitive capabilities in the context of sensing,

reasoning and learning^[53-58].

In MECA, the episodic subsystem (Fig. 9) interacts with the working memory to provide support for planning and decision-making. It is composed basically of the episodic learning codelet, which provides storage within the episodic memory, and the episodic retrieval codelet, which provides retrieval services. We also detected a requirement for maintenance, responsible for the mechanisms of forgetting and reorganization of the episodic memory, due to its ever-growing nature, to avoid an expansion beyond the available data storage limitation. But this maintenance feature is not yet implemented in MECA, being just a requirement for future developments. This means that the current versions of MECA might suffer from memory limitations, in a long-term run. We expect to be able to address this in the future.

To understand the storage mechanism performed by episodic learning codelet, we need to review the process by which episodes are created.

As pointed out in the last section, attention codelets are responsible for detecting episodes and dropping them in the episodic buffer. As a full episode is detected and released in the episodic buffer, this episode becomes available to the consciousness mechanism (section 3.2.7) to compete and eventually gain access to the global workspace, and then being broadcast for all the codelets in the architecture. This is how the events might enter the episodic memory. If in the broadcast of consciousness, the episodic learning codelet detects an episode being broadcast, it stores the episode in the episodic memory. This means that not all the episodes are stored in the episodic memory, but just those that

gained consciousness.

From the other side, as soon as there are episodes stored in the episodic memory, they can be retrieved using the service provided by the episodic retrieval codelet. This codelet is always monitoring whatever is stored in the cue memory, inside the working memory. These cues are used to trigger episodes related to the cue. If the episodic retrieval codelet can find episodes related to the cue, these are then copied into the episodic recall memory, also within the working memory, becoming available to whatever it is useful for.

3.2.5 The System 2 motivational subsystem

The System 2 motivational subsystem provides a second source of connection between System 1 and System 2. It is depicted in Fig. 10.

As already described in section 3.1.4, when the behavioral codelets (in System 1) don't have a valid plan to be proposed, they send a plan request to the imagination memory (the one in System 1, not the one with the same name within working memory). These plan requests are detected by the goal generation codelets, in System 2, which will start the process of requesting a plan. The first step in this endeavor is to create a goal in the goals memory, within the working memory. In principle, there might be a specific goal generation codelet for each behavior codelet in the behavioral system, as there might be a specific motivation behind each behavior (and also a drive). The motivation subsystem from System 2 will then generate multiple goals, which are sent to the goals memory, within the working memory. It is also a responsibility of each goal generation codelet, to evaluate the importance of its goal, which is tagged in the Eval (E) associated to

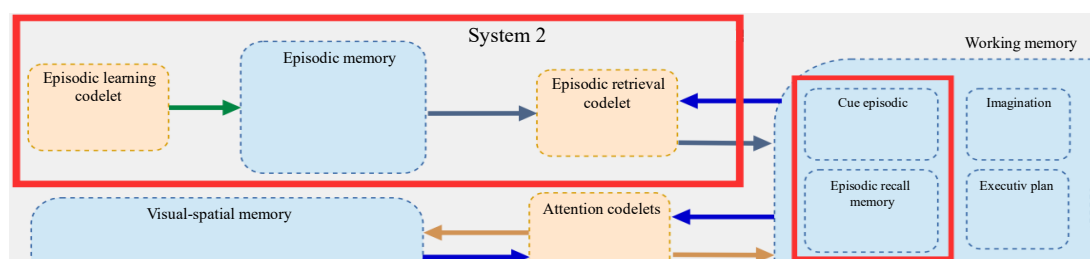


Fig. 9 A zoom on the episodic subsystem

the goal, such that the planning system is further able to prioritize the goals, while providing plans to satisfy them.

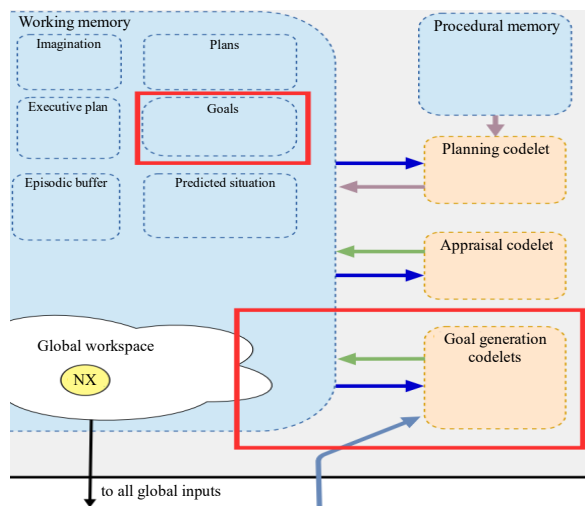


Fig. 10 A zoom on the goals generation mechanism - System 2 motivational subsystem

The appraisal codelet can also be used to appraise and change the Eval (E) value of knowledge available in the working memory.

3.2.6 Planning subsystem

Planning subsystem represents a critical component for deliberative reasoning, scenario simulation, and goal-driven behavior. It enables the architecture to move beyond immediate, reactive behaviors by forecasting potential actions and evaluating outcomes through an advanced computational imagination mechanism. This subsystem supports both short-term goals and long-term strategies by simulating alternative scenarios before committing to action. An overview of the planning subsystem can be viewed in Fig. 11.

The MECA planning subsystem is a pluggable component. This means that different planners can be used, depending on the situation. In the experiments we did with MECA, both SOAR^[8] and OR-Tools^[59] were used in different experiments, as planners.

The planning subsystem constructs and evaluates “what-if” scenarios, simulating various future outcomes using inputs from current perception, episodic recall, semantic knowledge, and working memory. Based on goals generated by the motivational subsystem, the

planning subsystem devises action plans comprising sequences of activities required to reach a specific target state. The subsystem dynamically integrates knowledge from episodic, semantic, and procedural memories, aligning immediate actions with previously learned experiences, rules, and strategies.

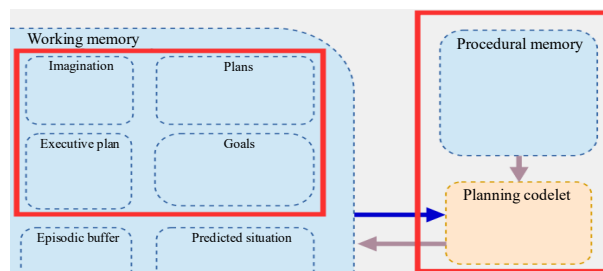


Fig. 11 A zoom on the planning subsystem

The MECA planning subsystem provides a common interface that can be used by multiple possible planners. The following slots are available:

Goals memory: the goals provided by the goal generation codelets are the specification to be used while generating a plan.

Imagination memory: this slot in working memory is a workspace where simulated future scenarios are created and stored temporarily.

Plans memory: all generated plans are stored temporarily in the plans slot of working memory until an action sequence is selected for execution. Each plan is associated with an evaluation score derived from metrics like efficiency, resource optimization, and likelihood of goal success.

Executive plan memory: after plan evaluation, one plan is selected as the “active plan” and transferred to the executive plan sub-memory for execution. The active plan details the sequence of actions, contingencies, and target outcomes.

The use of these slots is not mandatory for all planners. Depending on the planner requirements, they can be used or not. The only hard constraint is that after a plan is scheduled for execution, it should be made available at the executive plan memory slot within working memory, because this is where the MECA consciousness subsystem will look for something to

compete for consciousness and broadcasting the plan, such that the behavior codelets are able to receive it and start executing it, if this is the case.

The planning subsystem interacts extensively with the other parts of MECA for cohesive functionality:

Episodic subsystem: the planning subsystem retrieves stored episodes from episodic memory to learn from past experiences and simulate alternative futures.

Semantic subsystem: semantic knowledge provides general rules, conceptual relationships, and heuristics for action plans. For example, prior learned strategies influence optimal behaviors in dynamic environments.

Expectation subsystem: predictive models from the expectation subsystem supply immediate foresight into environmental changes, improving plan evaluations.

Global workspace: key outputs, such as goals or actionable plans, are broadcast via the global workspace to synchronize decisions across all subsystems.

3.2.7 Consciousness subsystem

The consciousness subsystem implements a variation of Global Workspace Theory, where selected information from the working memory is broadcast for global access. The consciousness subsystem is depicted in Fig. 12. Basically, the consciousness codelet observes all different knowledge available within the working memory and based on the Eval (E) value of each of them, decides what to broadcast, at each instant of time.

Typically, there is a competition between current perception, episodic recall memory and imagination memory, but if a plan is made available at the execution plan slot of working memory, it will have the top priority for gaining consciousness.

There are many subsystems that depend on the consciousness mechanism to work. Both the episodic learning codelet and the semantic learning codelet only update the episodic and semantic memories with some content coming from consciousness.

But the most important subsystem depending on the consciousness subsystem is the behavioral subsystem. This is the standard way it receives the plans requested to System 2.

3.2.8 Expectation subsystem

The expectation subsystem is illustrated in Fig. 13. Basically, it uses whatever is available at the working memory to predict the next situation, which is stored in the predicted situation slot of working memory. Based on these predictions, the overall system can evaluate if things are going as previewed, and in the case of a surprise, might trigger some specific reactions.

3.2.9 Semantic subsystem

The semantic subsystem (Fig. 14) has a mechanism that is similar to the episodic subsystem. It is split, basically, into a storage mechanism, provided by the semantic learning codelet, which receives its input from the broadcast performed by the consciousness

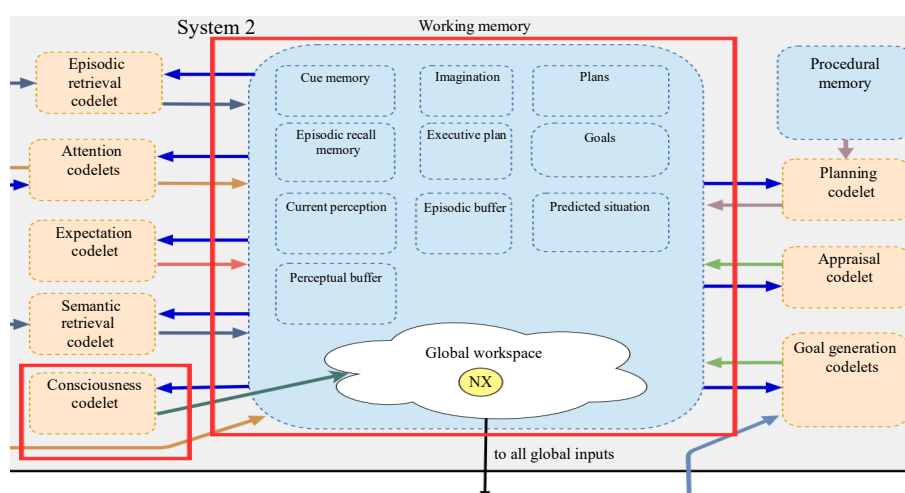


Fig. 12 A zoom on the consciousness subsystem

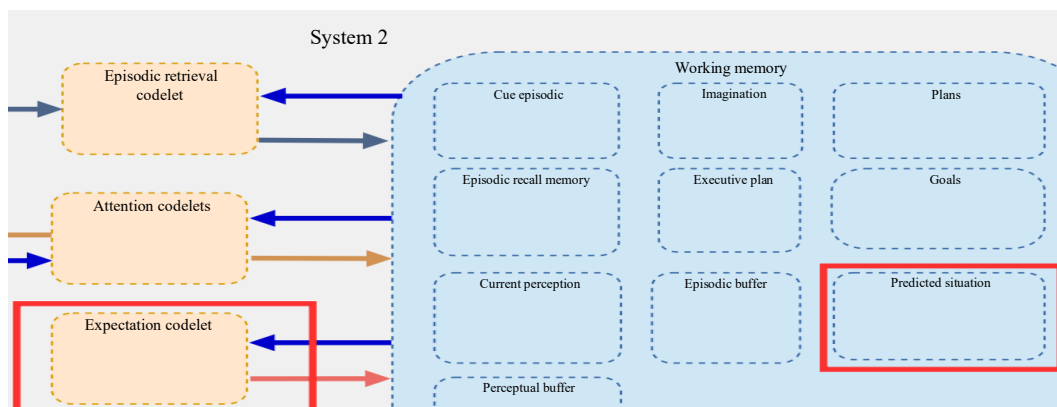


Fig. 13 A zoom on the expectation subsystem

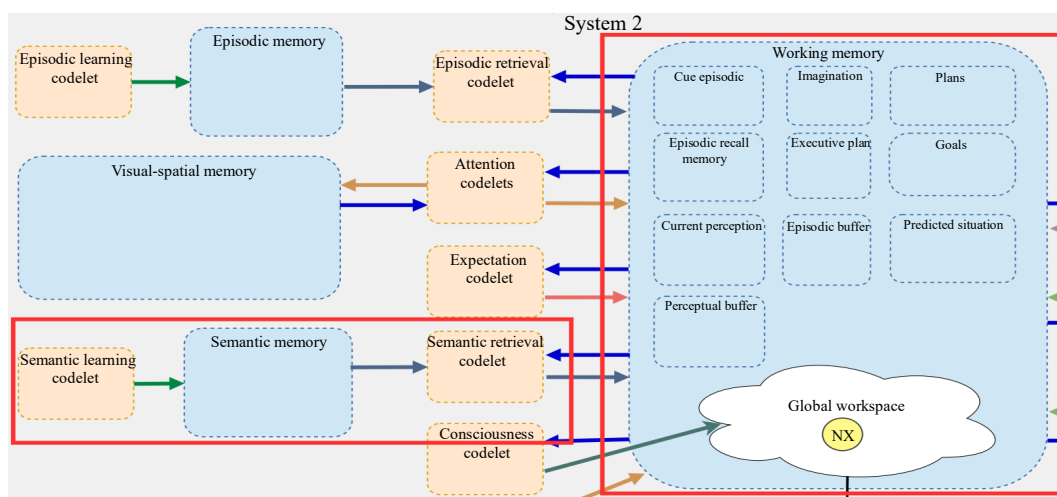


Fig. 14 A zoom on the semantic subsystem

mechanism, and then checks if there is some semantic knowledge available in it. This semantic knowledge is compared to whatever is already available in semantic memory, and if this is new knowledge it stores this new information there.

The second mechanism in this subsystem is the retrieval mechanism. The semantic retrieval codelet keeps monitoring the cue memory slot in working memory, and if there is a match between the cue and some internal semantic knowledge, this piece of knowledge is brought to working memory.

3.3 Computational ideas: the new knowledge representation scheme in MECA

In the old MECA^[7] there was no particular format for knowledge representation. MECA basically inherited the CST directive of using any Java class as a possible piece of knowledge to be stored in memories. Since then,

a new knowledge representation scheme has appeared in CST^[18], and MECA started using it. It is not mandatory, but the new directive is to strongly recommend the use of computational ideas to represent any piece of knowledge, also in MECA.

The notion of computational idea was first introduced in the article “existence, hypotheses, and categories in knowledge representation”^[18] as a general structure for representing different kinds of knowledge, and is available in the CST, used by MECA. Computational ideas have their background from philosophy of mind, particularly using the philosophy of Locke, Kant and Peirce. A computational idea is a computational version of any kind of idea that might appear in the human mind. In principle, these ideas can be categorized as from three different worlds (Fig. 15), which have some internal resemblance, but must be

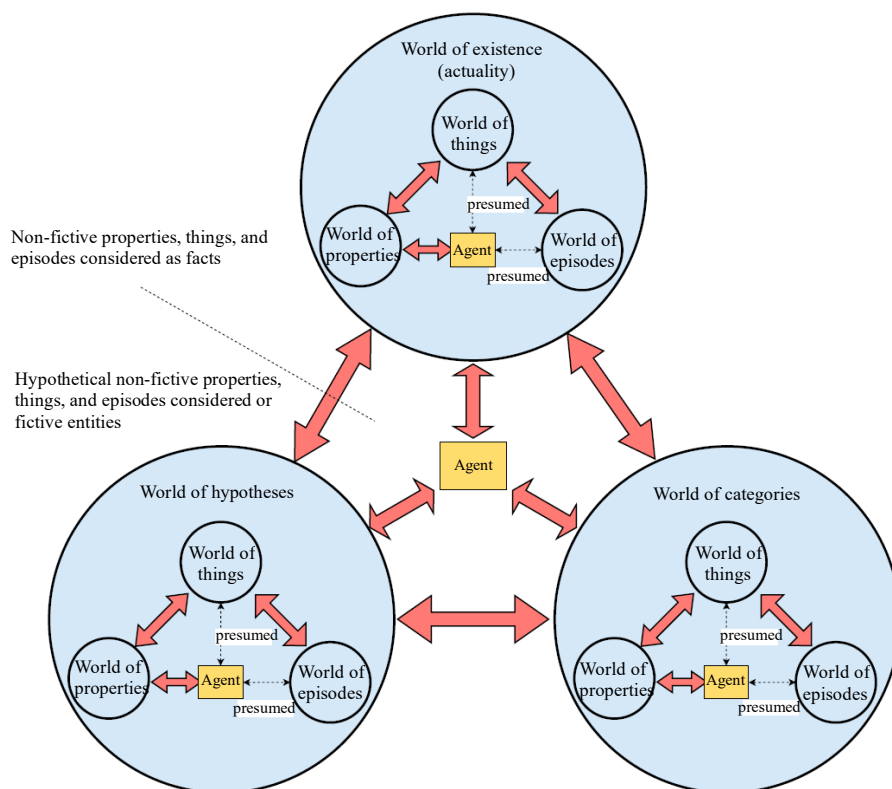


Fig. 15 The worlds of ideas and the fragments of reality

differentiated in any kind of mind (natural or artificial).

In this conceptualization, reality can be split into existence, and the laws operating on it. So, the laws are not part of existence, but a part of reality. As Peirce states: laws do not exist but are real.

The first world to be considered here is the world of existence. This comprises the properties, things and episodes that our mind understands as really happening in our environment. It includes our experience of space, time and everything our mind assumes is really happening there. Ideas from the world of existence are related to whatever our mind interprets as something really there. The things are things our mind assume as real things, their properties are those that were particularly sensed (or precisely estimated), and the episodes comprise the flow of time that already happened. It is interesting to compare the ideas from the world of existence with the ideas from the world of hypothesis. The world of hypothesis, differently from the world of existence, comprises ideas of purely hypothetical things, properties and episodes. These can

be hypothetical imaginations or plans for the future. It also includes things, properties and episodes, but now these are not assumed as being a part of existence. In some diseases of the human mind, like e.g., schizophrenia, the mind confuses elements from the world of hypothesis as being from the world of existence and assumes that imaginary things are real. Using computational ideas, which specifically differentiates its elements as being under the scope of hypothesis or existence, allows for the representation of this difference in cognitive architectures.

Finally, the world of categories comprises generic ideas that categorize properties, things and episodes, both in the world of hypothesis and in the world of existence. Categories are crucial for organizing and grouping entities, making knowledge management efficient and scalable.

A general description of a computational idea can be seen, as UML diagrams, in Fig. 16 and Fig. 17. In Fig. 16, we show the structure of a computational idea, and on Fig. 17, different kinds of ideas are presented.

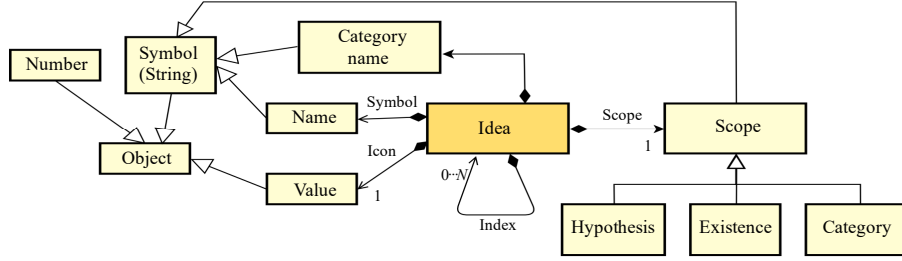


Fig. 16 Computational idea

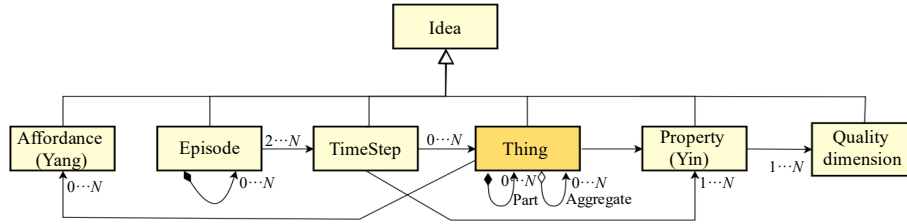


Fig. 17 Types of ideas

It is interesting to observe how an idea is represented. A computational idea has a name, a value, a category and a scope. The name is a symbolic representation for the idea and should be unique within the cognitive architecture using it. The value, in principle, should be either a number or a string, but can be, in principle, any Java object, of any class (remembering that CST implements computational ideas in Java). This allows the representation of values as arrays, tensors, graphs, trees, or any other kind of structure that is necessary. The category is also a string, and the scope can be: hypothesis, existence or category. But the biggest and most powerful capability of a computational idea is its ability to refer to other ideas as a part of its representation. So, some ideas (not everyone) might require an index to other ideas to be fully understood.

More information on computational ideas and some examples of their use can be seen in Camargo et al^[18].

3.4 Discussion

Some issues on computational ideas in MECA and CST are deeply inspired by the theory on grounded cognition^[19,30-31] and conceptual spaces^[20,41]. These provide influential theories for understanding knowledge representation in AI.

Grounded cognition emphasizes perceptual symbols tied to sensory experiences, which store multi-

modal states in simulators that re-enact these experiences to convey meaning. This perceptual grounding allows systems to represent concepts with direct sensory analogs, facilitating abstract thinking without human-imposed symbols.

Conceptual spaces theory posits that the mind organizes knowledge geometrically, where concepts (like object categories) map to specific regions within a multi-dimensional metric space, constructed from quality dimensions. These dimensions may represent various sensory or qualitative properties, and their spatial relationships define conceptual similarity. This theory aligns with grounded cognition by establishing that conceptual meaning arises from proximity and structure within a perceptually grounded space, rather than from external symbols.

Computational ideas of incremental complexity, binding properties into things, and things into episodes along time steps are the perfect playground for implementing the theory of both grounded cognition and conceptual spaces.

But an important reflection must be made. In the current literature on large language models, a similar but not so explicit model is being created with the aim of grounding symbols into something else. These are usually called embeddings, and their understanding is still full of missing points.

Neural network-based word vectors, e.g., Word2Vec^[60] and BERT^[61] are good examples of how these embeddings can be used to learn semantic relationships through contextual co-occurrences in large text corpora, enabling AI to capture analogies and semantic relationships by mapping words to positions within a continuous vector space. This setup resembles Gärdenfors' conceptual spaces, as word vectors implicitly ground meaning through spatial proximity, without explicit symbolic rules. By capturing semantic relationships as geometrical distances, these word vectors provide a scalable mechanism for linguistic grounding, like Barsalou's simulators, which represent words through associations with perceptual experiences.

The similarities between grounded cognition, conceptual spaces, and modern AI artifacts open new possibilities for integrating word vectors with multi-modal information, such as visual and sensory data. This could lead to representations that more closely align with human cognition, moving from purely statistical co-occurrence patterns to perceptually enriched conceptual spaces capable of nuanced, flexible understanding across contexts.

Currently, there is still no policy for addressing embeddings into MECA. Differently from embeddings in large language models, computational ideas are more explainable, as the multiple connections from some ideas to other ideas are explicitly encoded in its inner structure. But we feel that there should be some way of encoding computational ideas into embeddings, and decoding embeddings into computational ideas. This raises an important question we will address in the next section: what is still missing in MECA, and what are the next steps in the development of MECA?

4 Missing points and the next steps for MECA

Even though MECA provides learning for both episodic memory and semantic memory, and possibly some sort of learning mechanism for its pluggable

planning subsystem (as e.g. in SOAR), there is still no standard mechanism for learning new activities. Even though the planning subsystem might combine activities in different behaviors, the current MECA is still stuck with the activity codelets created in design time. It is true that these activity codelets might be neural networks, subject to some sort of learning, but this does not have any support from MECA itself. This happens because in its current implementation, activities are still tied to codelets, which need to be provided during design time. One possible solution would be to turn activities into computational ideas, and then create specialized codelets for learning them. Our team is already investigating this possibility, but not yet with fruitful results. So, learning new activities is one of our next steps.

A second missing point is the use of natural language. With the development of large language models, there is a growing interest in including language capabilities into MECA. As it is today, it is possible to create activity codelets that connect with a large language model and receive some feedback from it, which might allow the use of language inside MECA. But we feel that, just like other cognitive capabilities that are embedded in a specialized subsystem, language also deserves a specialized subsystem under MECA. This is also under investigation in our team.

Finally, there is this issue we pointed out in section 3.4. We would like to investigate how MECA might use embeddings (as in LLMs) and how to convert computational ideas to/from embeddings. Embeddings might be important both while considering the issue of learning, and while considering language capability, especially if using some sort of neural network for that purpose. Even though we still don't have a good solution to that, this is also under our considerations for MECA next steps.

5 Conclusion

Since its first introduction in 2017, MECA has evolved and integrated new functionalities. MECA's

dual-process approach, combining fast, reactive System 1 operations with slow, deliberative System 2 reasoning, models not only the intricacy but also the explainability of human thought. Its ability to plan systematically, integrate multimodal information via the episodic buffer, and dynamically adapt through global workspace communication provides a robust framework for creating rational, transparent, and trustworthy AI systems.

In this article, we covered the main new features introduced in MECA, together with an overall account of the many cognitive subsystems within it, providing the readers with an updated reference to its newest implementation, together with a roadmap for future developments.

Among the most important changes are the introduction of the visual-spatial memory, the complete redesign of the behavioral subsystem, using the new dual-layer subsumption mechanism, which allowed both plan-seeking behavior and opportunistic reactions, the simplification of the motivational subsystem in System 1 and the introduction of computational ideas as the main knowledge representation scheme.

The source code of MECA is open-source and can be found at: <https://github.com/CST-Group/meca>.

References:

- [1] LANGLEY P, LAIRD J E, ROGERS S. Cognitive architectures: research issues and challenges[J]. *Cognitive Systems Research*, 2009, 10(2):141-160.
- [2] PARAENSE A L O, RAIZER K, DE PAULA S M, et al. The cognitive systems toolkit and the CST reference cognitive architecture[J]. *Biologically Inspired Cognitive Architectures*, 2016, 17:32-48.
- [3] KOTSERUBA I, TSOTSOS J K. A review of 40 years of cognitive architecture research: core cognitive abilities and practical applications[EB/OL]. [2024-11-12]. <https://arxiv.org/abs/1610.08602v3>.
- [4] SAMSONOVICH A V. Toward a unified catalog of implemented cognitive architectures[C]//*Proceedings of the First Annual Meeting of the BICA Society*. Amsterdam, Netherlands: IOS Press, 2010:195-244.
- [5] GOERTZEL B, PENNACHIN C, GEISWEILLER N. Brief survey of cognitive architectures[C]//*Engineering General Intelligence, Part 1*. Paris, France: Atlantis Press, 2014:101-142.
- [6] LUCENTINI D F, GUDWIN R R. A comparison among cognitive architectures: a theoretical analysis[J]. *Procedia Computer Science*, 2015, 71:56-61.
- [7] GUDWIN R, PARAENSE A, DE PAULA S M, et al. The multipurpose enhanced cognitive architecture (MECA)[J]. *Biologically Inspired Cognitive Architectures*, 2017, 22: 20-34.
- [8] LAIRD J. The soar cognitive architecture[M]. Cambridge, USA: MIT Press, 2012.
- [9] SUN R. A tutorial on clarion 5.0[EB/OL]. [2024-11-16]. http://www.cogsci.rpi.edu/public_html/rsun/sun.tutorial.pdf.
- [10] FRANKLIN S, MADL T, D'MELLO S, et al. LIDA: a systems-level architecture for cognition, emotion, and learning[J]. *IEEE Transactions on Autonomous Mental Development*, 2013, 6(1):19-41.
- [11] BAARS B J. A cognitive theory of consciousness[M]. Cambridge, England: Cambridge University Press, 1988.
- [12] BROOKS R. A robust layered control system for a mobile robot[J]. *IEEE Journal on Robotics and Automation*, 1986, 2(1):14-23.
- [13] NAKASHIMA H, NODA I, NAKASHIMA H, et al. Dynamic subsumption architecture for programming intelligent agents[C]//*Proceedings of the 3rd International Conference on Multi Agent Systems*. Paris, France, 1998: 190-197.
- [14] HAMADI Y, JABBOUR S, SAÏS L. Learning for dynamic subsumption[C]//*2009 21st IEEE International Conference on Tools with Artificial Intelligence*. Newark, USA, 2009:328-335.
- [15] HECKEL F W P, YOUNGBLOOD G M, HECKEL F W P, et al. Multi-agent coordination using dynamic behavior-based subsumption[C]//*Proceedings of the Sixth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. California, USA, 2010:132-137.
- [16] GEORGE D, HAWKINS J. Towards a mathematical theory of cortical micro-circuits[J]. *PLoS Computational Biology*, 2009, 5(10):e1000532.
- [17] JILK D J, LEBIERE C, O'REILLY R C, et al. SAL: an explicitly pluralistic cognitive architecture[J]. *Journal of Experimental & Theoretical Artificial Intelligence*, 2008, 20(3):197-218.
- [18] CAMARGO E, SAKABE E Y, GUDWIN R. Existence, hypotheses and categories in knowledge representation[J]. *Procedia Computer Science*, 2022, 213:496-503.

- [19] BARSALOU L W. Grounded cognition: past, present, and future[J]. *Topics in Cognitive Science*, 2010, 2(4): 716-724.
- [20] GÄRDENFORS P. *Geometry of meaning: semantics based on conceptual spaces*[M]. Cambridge, USA: MIT Press, 2014.
- [21] OSMAN M. An evaluation of dual-process theories of reasoning[J]. *Psychonomic Bulletin & Review*, 2004, 11(6):988-1010.
- [22] ST B T EVANS J. In two minds: dual-process accounts of reasoning[J]. *Trends in Cognitive Sciences*, 2003, 7(10): 454-459.
- [23] ST B T EVANS J. Dual-processing accounts of reasoning, judgment, and social cognition[J]. *Annual Review of Psychology*, 2008, 59:255-278.
- [24] ST B T EVANS J, STANOVICH K E. Dual-process theories of higher cognition: advancing the debate[J]. *Perspectives on Psychological Science*, 2013, 8(3):223-241.
- [25] FRANKISH K. Dual-process and dual-system theories of reasoning[J]. *Philosophy Compass*, 2010, 5(10):914-926.
- [26] KAHNEMAN D. *Thinking, fast and slow*[M]. New York, USA: Macmillan, 2011.
- [27] SLOMAN S A. The empirical case for two systems of reasoning[J]. *Psychological Bulletin*, 1996, 119(1):3-22.
- [28] STANOVICH K E, WEST R F. Individual differences in reasoning: implications for the rationality debate?[J]. *The Behavioral and Brain Sciences*, 2000, 23(5):645-726.
- [29] ARKIN R C. *Behavior-based robotics*[M]. Cambridge, USA: MIT Press, 1998.
- [30] BARSALOU L W. Perceptual symbol systems[J]. *Behavioral and Brain Sciences*, 1999, 22(4):577-660.
- [31] BARSALOU L W. Grounded cognition[J]. *Annual Review of Psychology*, 2008, 59:617-645.
- [32] NARAYANAN S. Moving right along: a computational model of metaphoric reasoning about events[EB/OL]. [2024-11-29]. <https://cdn.aaai.org/AAAI/1999/AAAI99-018.Pdf>.
- [33] BERGEN B, CHANG N, NARAYAN S. Simulated action in an embodied construction grammar[C]// *Proceedings of the Twenty Sixth Annual Conference of the Cognitive Science Society*, 2004:108-113.
- [34] ROY D. Semiotic schemas: a framework for grounding language in action and perception[J]. *Artificial Intelligence*, 2005, 167(1/2):170-205.
- [35] CANGELOSI A, RIGA T. An embodied model for sensorimotor grounding and grounding transfer: experiments with epigenetic robots[J]. *Cognitive Science*, 2006, 30(4): 673-689.
- [36] DOMINEY P F, HOEN M, INUI T. A neurolinguistic model of grammatical construction processing[J]. *Journal of Cognitive Neuroscience*, 2006, 18(12):2088-2107.
- [37] MADDEN C, HOEN M, DOMINEY P F. A cognitive neuroscience perspective on embodied language for human-robot cooperation[J]. *Brain and Language*, 2010, 112(3):180-188.
- [38] LALLEE S, MADDEN C, HOEN M, et al. Linking language with embodied and teleological representations of action for humanoid cognition[J]. *Frontiers in Neurorobotics*, 2010, 4:8.
- [39] FRANK S L, VIGLIOCCO G. Sentence comprehension as mental simulation: an information-theoretic perspective [J]. *Information*, 2011, 2(4):672-696.
- [40] PEZZULO G, CALVI G. Computational explorations of perceptual symbol systems theory[J]. *New Ideas in Psychology*, 2011, 29(3):275-297.
- [41] GÄRDENFORS P. *Conceptual spaces: the geometry of thought*[M]. Cambridge, USA: MIT Press, 2004.
- [42] HOFSTADTER D, MITCHELL M. *The copycat project: a model of mental fluidity and analogy-making*[M]. New York, USA: BasicBooks, 1994:205-268.
- [43] FRANKLIN S, KELEMEN A, MCCAULEY L. IDA: a cognitive agent architecture[C]// *SMC'98 Conference Proceedings 1998 IEEE International Conference on Systems, Man, and Cybernetics*. San Diego, USA, 2002: 2646-2651.
- [44] BAARS B J, FRANKLIN S. An architectural model of conscious and unconscious brain functions: global workspace theory and IDA[J]. *Neural Networks*, 2007, 20(9):955-961.
- [45] GUDWIN R, PARAENSE A, DE PAULA S M, et al. An urban traffic controller using the MECA cognitive architecture[J]. *Biologically Inspired Cognitive Architectures*, 2018, 26:41-54.
- [46] GUDWIN R, ROHMER E, PARAENSE A, et al. The TROCA Project: an autonomous transportation robot controlled by a cognitive architecture[J]. *Cognitive Systems Research*, 2020, 59:179-197.
- [47] BADDELEY A. *Human memory: theory and practice*[M]. Hove, USA: Psychology Press, 1997.
- [48] MASLOW A H. A theory of human motivation[J]. *Psychological Review*, 1943, 50(4):370-396.
- [49] SUN R. Motivational representations within a computational cognitive architecture[J]. *Cognitive Computation*,

- 2009, 1(1):91-103.
- [50] SUN R, WILSON N. Motivational processes within the perception: action cycle[M]. New York: Springer, 2010: 449-472.
- [51] MCCALL R. Fundamental motivation and perception for a systems-level cognitive architecture[EB/OL]. [2024-12-01]. <https://digitalcommons.memphis.edu/cgi/viewcontent.cgi?article=2098&context=etd>.
- [52] GUDWIN R, ROHMER E, PARAENSE A, et al. A double-layer subsumption mechanism for enforcing sequential behaviors in a cognitive architecture[C]//2021 IEEE Symposium Series on Computational Intelligence. Orlando, USA, 2021:1-7.
- [53] BADDELEY A. The episodic buffer: a new component of working memory?[J]. Trends in Cognitive Sciences, 2000, 4(11):417-423.
- [54] BADDELEY A D. Is working memory still working?[J]. The American Psychologist, 2001, 56(11):851-864.
- [55] TULVING E. Concepts of human memory[EB/OL]. [2024-12-06]. <https://alicekim.ca/concepts91.pdf>.
- [56] TULVING E. Episodic memory: from mind to brain[J]. Annual Review of Psychology, 2002, 53:1-25.
- [57] HOWARD M W, FOTEDAR M S, DATEY A V, et al. The temporal context model in spatial navigation and relational learning: toward a common explanation of medial temporal lobe function across domains[J]. Psychological Review, 2005, 112(1):75-116.
- [58] CABEZA R, CIARAMELLI E, OLSON I R, et al. The parietal cortex and episodic memory: an attentional account[J]. Nature Reviews Neuroscience, 2008, 9(8): 613-625.
- [59] CUVELIER T, DIDIER F, FURNON V, et al. OR-tools' vehicle routing solver: a generic constraint-programming solver with heuristic search for routing problems[EB/OL]. [2024-11-26]. <https://hal.science/hal-04015496v1/document>.
- [60] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space [EB/OL]. [2024-11-26]. <https://arxiv.org/pdf/1301.3781>.
- [61] GABSI I, KAMMOUN H, MTAR R, et al. Word2Vec-GloVe-BERT embeddings for query expansion[C]// Intelligent Systems Design and Applications. Switzerland: Springer Nature, 2024:167-177.

MECA 认知架构的更新

BORELLI José Renato^{1,2}, GUDWIN Ricardo Ribeiro¹

(1.坎皮纳斯大学, 巴西圣保罗 坎皮纳斯 13083-970; 2.圣保罗联邦研究所, 巴西圣保罗 奥托兰迪亚 13183-000)

摘要: 多用途增强认知架构(MECA)是一个旨在对“类人过程”这一跨领域复杂过程进行建模的认知框架。MECA 最初专注于实施双过程理论方法, 以及集成基于全局工作空间理论的机器意识机制, 现已更新为集成双层包容机制, 支持反应和审议行为、动态目标设定和视觉空间记忆子系统, 增强了其对现实世界的交互性和自适应行为能力。此外, 随着对计算思想的知识表达新方案的引入, MECA 提出对知识进行动态组织, 以处理情景敏感推理和灵活分类。MECA 的实施依赖于认知系统工具包(CST), 该工具包使其可与尖端技术实现集成。MECA 和 CST 正在不断被开发和更新、协调和开放, 以纳入最新的 AI 器件和方法。这一举措确保了有组织、可监视、可审计和可控制的人工智能解决方案的产生, 大大减少了对“黑匣子”认知过程的依赖, 同时提高了受人工智能驱动的系统透明度和可解释性。这些更新增强了 MECA 作为一种稳固架构的潜力, 使其可用于开发能够进行现实世界交互和自适应学习, 具有自主性、可适应性、情境感知的人工智能系统。

关键词: 认知架构; 双过程理论; 动态包容; 认知系统工具包

(责任编辑 史小丽)