



# Methodological aspects for cognitive architectures construction: a study and proposal

Juan P. Jiménez<sup>1</sup> · Luis Martin<sup>1</sup> · Iván Axel Dounce<sup>1</sup> · Cynthia Ávila-Contreras<sup>1</sup> · Félix Ramos<sup>1</sup>

Published online: 24 September 2020  
© The Author(s) 2020

## Abstract

In the field of Artificial Intelligence (AI), efforts to achieve human-like behavior have taken very different paths through time. Cognitive Architectures (CAs) differentiate from traditional AI approaches, due to their intention to model cognitive and behavioral processes by understanding the brain's structure and their functionalities in a natural way. However, the development of distinct CAs has not been easy, mainly because there is no consensus on the theoretical basis, assumptions or even purposes for their creation nor how well they reflect human function. In consequence, there is limited information about the methodological aspects to construct this type of models. To address this issue, some initial statements are established to contextualize about the origins and directions of cognitive architectures and their development, which help to outline perspectives, approaches and objectives of this work, supported by a brief study of methodological strategies and historical aspects taken by some of the most relevant architectures to propose a methodology which covers general perspectives for the construction of CAs. This proposal is intended to be flexible, focused on use-case tasks, but also directed by theoretic paradigms or manifestos. A case study between cognitive functions is then detailed, using visual perception and working memory to exemplify the proposal's assumptions, postulates and binding tools, from their meta-architectural conceptions to validation. Finally, the discussion addresses the challenges found at this stage of development and future work directions.

**Keywords** Methodology · Bio-inspired cognitive architectures · Cognition · Cognitive sciences · Neurosciences

---

✉ Félix Ramos  
felix.ramos@cinvestav.mx

Juan P. Jiménez  
jppjimenez@gdl.cinvestav.mx

Luis Martin  
ldmartin@gdl.cinvestav.mx

Iván Axel Dounce  
iadounce@gdl.cinvestav.mx

<sup>1</sup> Nature Inspired Computing Lab, Cinvestav I.P.N. Unidad Guadalajara, Av. del Bosque 1145, Col. El bajo, 45019 Guadalajara, JAL, Mexico

## 1 Introduction

In recent years, the field of Artificial Intelligence (AI) has developed several study branches for generation, representation, analysis, and gathering of cognition and behavior of intelligent computational agents.

Although it could be supposed that artificial intelligence -as an effort to model brain function- follows a relatively homogenous abstraction path to emulate human mind, some of the study areas of AI have opted for traditional learning and modeling approaches. These are often supported by knowledge-based systems, learning classifiers, neural networks, deep learning, statistical learning and other similar learning-based methods (Arbib 2003, 2007); while some others, which use holistic *cognitive approaches*, non-computational or mixed scopes, are focused on development of the agent itself and how it transforms the data to obtain, manage and use knowledge through experience and interaction with its environment. Most approaches strive to attend to complex functional and structural criteria, with the objective of more realistic and human-like responses (Anderson 2013).

Cognitive architectures have evolved over the past 50 years to become a solid option for the representation and modeling of intelligent behaviors searching to emulate natural human behavior, which in turn allows to provide synthetic agents with multi-level reasoning capabilities. Achieving this natural behavior implies the development of human cognitive functions in the CAs such as memory, attention, planning, and decision-making, through the study, structuration, and integration of domains in the field of cognitive sciences (psychology, philosophy, biology, etc.)

It is possible to assume a distinction between the explanatory and hierarchical forms that can be used for the classification of the domains of cognitive sciences that defines limits of hierarchical levels from biophysics and neuropsychology, (observable in the case of neurosciences and unobservable in the case of cognitive sciences) that each of them makes about cognitive phenomena (Tête 1994). This makes a differentiation between cognitive architectures focused on cognitive learning that seeks general intelligence and neurally based approaches. (Laird 2008).

There have been crucial advances by various research groups in the field. Their contributions are related mainly to aspects of classification, construction and implementation of general cognitive approaches. These contributions conform a collaboration between architectural proposals with different robustness and scalability degrees, in terms of modeling and implementation between cognitive sciences and their technological possibilities on computer science (Lieto et al. 2018).

Despite such contributions, a comprehensive criterion that unifies and encompasses the plethora of research visions that the involved cognitive domains can offer, is still an unsolved issue. Here, it is discussed that cognitive architectures still have some unity problems when it comes to accurately representing the limits of their action and scope, and how they approach complex structures of knowledge and behavior exhibited by a computational agent, when compared to the knowledge and behavior that may be obtained in a natural way by a human being.

Election of a cognitive structure to model reasoning and behavior in intelligent agents can be seen as an epistemological issue among representative, declarative and formalistic capacities of the systems based on knowledge, and other cognitive structures that imply those capacities, and capacities from different ontological nature (Lieto et al. 2018).

Moreover, in the domain of biologically inspired cognitive architectures (**BICA**), the inner complexity of cerebral functions has been a critical constraint and a permanent

debate topic for developing an unified methodology, which causes that, most of the time, our vision of integrative high-level cognitive functions is heavily linked to synthetic sets of tasks to be solved from pragmatic and parsimonious abstractions (Varma 2011).

All these difficulties make an opportunity field, that lead us to study the main methodological aspects for the creation of cognitive architectures, either traditional or bioinspired. It is not intended to develop different cognitive architectures, but helping to guide developers to generate solutions to problems of different levels of application in the field of cognitive and computer science. Thus, this proposed methodological structure is supported by cognitive sciences as context and other cognitive architectonical schema -determined as unifying frameworks- for cognitive architectures construction, using all the components required for a set of cognitive processes involved in a task, according to a specific approach, in a concurrent, synchronic and multilevel manner, considering the (maximally local) work of completeness (Vernon 2019), which provides means of representation in a cognitive system.

A definition of objectives for this paper is set to distinguish borderlines between the aforementioned standpoints (neurobiological, philosophical, psychological and computational instances), which implies a structural conformation and multimodal, transdisciplinary association in order to achieve a more complex architecture:

- Characterize the elements that enable the construction of a cognitive architecture, regardless of its focus or origin.
- Distinguishing in a qualitative way, the building and operative block, which allows assuming a desirable set of functions to explain a behavior, or a particular case study.
- Design of individual models for a particular cognitive function (such as vision or perception) and their collaboration towards more complex models, following the same methodological sequence in either cases.
- Integrate the knowledge of different scientific fields, in order to conceptualize the design of a cognitive architecture.

It is important to stress some initial conflict points about modeling and engineering cognitive architectures, primarily, what is related to the main constructive and operational guidelines mentioned above. It is assumed that an approach to the unification of methodological proposals for the construction of CAs can be carried out, by resolving the conflict between the representation of schemes or patterns that comprise the system, and the course of action for the solution of the problem. Nonetheless, heterogeneity in the choice of the guiding criteria of traditional guidelines of software architecture and data treatment, may be limited by non-congruent abstraction levels, that *strongly affects the explanatory power of computational representations* (Lieto 2017). However, the approaches' heterogeneity allows to offer suitable options for designing and developing cognitive architectures.

This article exposes a summary of historical relevant aspects of bioinspired cognitive architectures. From this overview, an analysis of the compositive methodological aspects of those relevant examples is presented. Then a taxonomical classification of architectures according to methodological constraints and a diagram of our methodological proposal and comprehensive study cases for the working memory and cognitive perception functions followed by the integration of these components to show the properties of our proposal and a discussion about the current state and further development of cognitive architectures from the exposed standpoint.

## 2 Considerations on cognitive architectures development

These considerations are intended to describe how cognitive architectures have been developed, in order to explore their general methodological features, and how can they provide relevant information in this topic.

In the early years of cognitive architectures development, Cooper et al. (1996) remarked the need for methodological sophistication in the face of evidence about the complexity of human behavior. To facilitate integration, it is important to make a difference between the notion of cognitive sciences and their practical results with the lack of consensus, mainly in their theoretical assumptions, by information in the representation of knowledge, and computational models (symbolic, connectionist or emergent).

Kitamura (2001) made a problematization about the foundations of modeling cognitive functions and stressed the main differences between intelligence and behavior in a computational agent. One of the most relevant premises is about the necessary aprioristic knowledge that conventional AI is related, and a neuroscience-inspired subsumptional architecture, providing non-symbolic loosely related cognitive functions, more suitable for expecting emergent behavior.

Weitzenfeld et al. (2002) proposes three elements for the construction of neural networks that have direct applicability with the design of cognitive architectures: modularity, object-orientation and concurrency.

*Modularity* is related to requirements for structuring software systems and allows addressing the entire system's inherent complexity. It anticipates and legitimizes the fragmentation of a complex whole into smaller parts (Weitzenfeld et al. 2002; Garrido 2009). Conceptual models of cognitive functions are presented, which represent a structural description of the brain areas involved. On modularity, two advantages stand out: the facilitation of understanding and the reusability of a module in other models.

*Object-Orientation* supports modularity when functional abstraction is done, adding as an abstraction level to the object itself and avoiding undesired effects by the modification of the system on a large scale. In terms of computer science, object-orientation analysis and its related techniques, prioritize the formation of a system that can reduce conceptual complexity through modules (Miller 2008), whose behavior can be analyzed and decomposable into their individual elements. This analysis' most important aspect is that it serves for the modeling of real-world systems and entities, which are modeled as objects and categorized in classes (Garrido 2009).

*Concurrency* is related to the dynamic capacity to handle parallel and distributed processing modules. A concurrent model is suitable for managing "active" units and prioritizing distributed work. Concurrency can be considered as a systemic property by which more than one execution context can be active at the same time (Mauro 2015).

Both in Weitzenfeld's neural networks proposal as in the design of cognitive architectures, concurrency helps to describe the interaction between components according to the determination of their temporality, their hierarchy and their presence in the physical nucleus of the system.

It is important to emphasize the relevance of the applicability of biological complex systems in the cognitive architectures context. In computer science, a first convention is following the *divide and conquer* paradigm as design requirement; the second, refers to the transition of levels of detail through informational exchanges between different sub-systems, sometimes designing the function isolated; in others, considering the interaction dynamics between them.

This complex functionality is assumed possible in a setting of collaborative multiagents. Maes (1991) proposed a model for an Agent Network Architecture, decentralizing cognitive functions by distinguishing between *Behavior-based IA* schema and *Knowledge-based IA*. Some proposals of cognitive architectures have emerged in the field of multiagents to try to solve conflictive negotiations between agents (Maes 1991).

There exist agent-based cognitive architectures whose components are designed by theoretical foundations of cognitive sciences (Torres 2013). Multi-Agent Systems (MAS) can benefit from CAs, because of their *ad-hoc* structure to execute multilevel, concurrent and context-sensitive processes. Such agents can also work in collaboration (or conflict) with other purely cognitive ones or strictly knowledge related ones.

Sun (2004) determines the initial assumptions that mediate in the construction of cognitive architectures, defining them as a set of structures and processes (which should be constituted simultaneously), oriented to the solution of multilevel and multi-hierarchy cognitive problems, and highlights the great difference between the assumptions sets that each research group should use to construct a CA according to their particular design intentions.

In Duch et al. (2008), proposes that elementary objectives and theoretical principles, which determine and methodologically guide cognitive architectures (phylogenetic properties), are universal only at the beginning; and through development time and refinement by practice (ontogenetic properties) certain paths are improved according to use, specific context and the problem they intend to solve.

Krichmar (2012) applied, in the field of cognitive robotics design, the three creational (constitutive) principles for cognitive agents stated by Pfeifer and Bongard (2006). Krichmar suggests that these design principles are suitable for the development of Bioinspired Cognitive Architectures. Those principles are: definition of an ecological niche, a defined behavior or task, and an agent design. Krichmar poses what can be seen as an *economy statement* for agents design abstraction, and links behavioral components to the reasoning process in the bottom-up construction of a cognitive agent.

The differences between the aforementioned approaches and principles imply searching necessary and sufficient conditions to legitimize a homogeneous form of construction. Hence, there is an essential paradox between the monitoring of expectations to achieve a general artificial intelligence through a cognitive architecture, and the representation of instruments, both conscious as non-conscious, on design of models of representation and use of knowledge.

Vernon et al. (2016) follows the line of Sun (2004). They both frame a limited number of necessary and sufficient conditions, in order to consolidate a cognitive cycle similar to that of the human being. In addition, there must be a set of theoretical elements to allow outlining the so-called “architecture template”. From this template’s existence, it is possible to establish the specific creation approach according to the research’s purpose and the problem to face, the cognitive function’s operational level, and the degree of interaction with other functions.

We argue that the “architecture template” should be expanded and explained as a methodological condition for the creation of cognitive architectures, regardless of the context and domain in which they are located. This topic will be briefly addressed further below.

Vernon (2019) establishes, from a pragmatic perspective of computational science, a criterion for designing cognitive architectures, which makes a mutual exclusion of two investigative approaches or agendas. Those agendas, that arise around the construction of CAs, refer to design based on *desiderata* and design based on *use cases* (Vernon et al. 2016; Vernon 2017). In the first one, desiderata is based on cognitivist approach as a structured set of essential functional requirements. It is a list of desirable

aspects to be found in a CA and obeys an integrated architectonic schema. This method is useful when the research problem is clearly identified. The second one (use cases), fulfills practical and specific user requirements and can be seen as a downgrade from a cognitive architecture to achieve an architectural system.

Lieto et al. (2018), before Vernon, remarked the importance of the generality of the structure and definition of the cognitive architectures design perspectives, according to the nature of their origins, application approach, and final purposes. The definition of their exhibited behavior's limits, allows to establish a basic characterization to model the human cognition's invariant mechanisms, which intend to guide the human mind's study. On the other hand, biological processes have a central epistemological role to characterize the nature of intelligent behavior.

### 3 Methodological levels of cognitive architectures

Before presenting a brief review of the most relevant cognitive architectures' methodological aspects, it is important to clarify the relation *Cognitive Architecture–Methodology*. Maes (1991) postulates that a cognitive architecture *is a methodology*; this is a valid statement because the architecture itself provides systematic, functional and structural guidelines on how to perform cognitive processes.

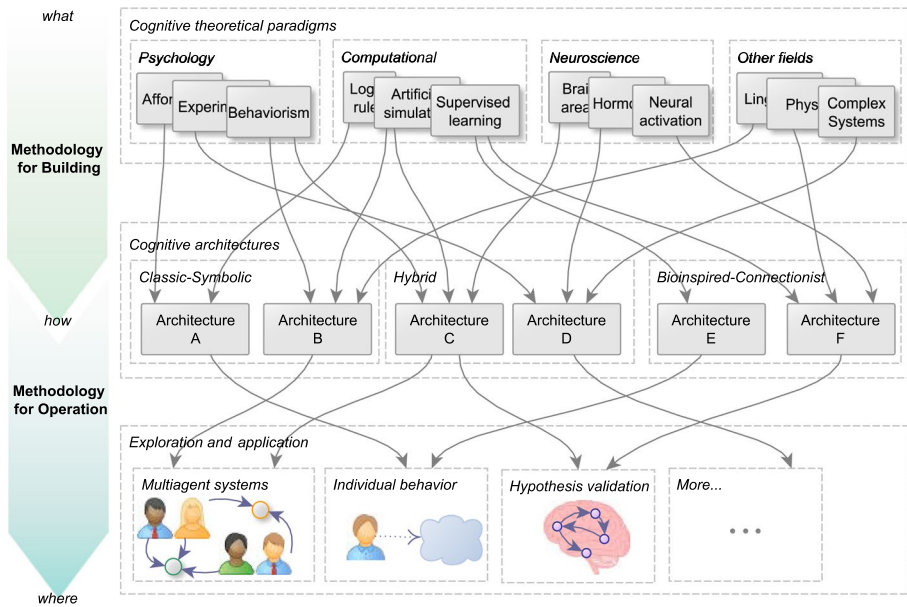
However, the idea of *architecture-as-methodology* does not define how that architecture is set in the first place. Consequently, is required a conceptual distinction between two taxonomic levels for the same term: *Methodology for building* cognitive architectures, and *Methodology for operation* of cognitive architectures as is shown in Fig. 1.

*Methodology for building* comprehends a set of guidelines and recommendations that help to gather and filter the pertinent theoretical or computational approaches to fulfill the requirements and intentions, which form the structure of a cognitive architecture. It proposes basic processes for objective-oriented decisions, establishment and monitoring of essential action policies, as well as choosing the most suitable general knowledge domains for the architecture.

*Methodology for operation* encompasses the descriptive aspects of how cognitive processes are performed, mainly in a synthetic entity or a computational intelligent agent. In this way, the architecture is seen as a methodology when it is applied to any domain, because it provides a framework for making assumptions, following conditions and executing functions about cognitive processes.

Methodology of building is carried out at the beginning and it is during this process that cognitive paradigms that aid to form the architecture are chosen. The reasoning is that this initial stage helps to go from *what* is desired to obtain, to *how* to achieve it. In this way the outputs, both at building and operational level, can be considered as architectures in themselves, where both follow the same proposed design principles.

This methodological process's output is a cognitive architecture which, according to its theoretical foundation, can be described as classic, biologically inspired, hybrid, or some other. From that point, one can go further and enter the methodology for operation level, by taking the architecture's structural and functional descriptions and putting them in diverse applications and domains for exploration purposes. This is going from *how* the architectures works, to *where* it is required to be deployed.



**Fig. 1** Two taxonomic levels are distinguished: *Building level* for every theoretical and intentional aspect to achieve meta-architectural foundations, and *Operational level* for those aspects developed to solve specific tasks

## 4 Methodological aspects

In this section, the most relevant aspects to propose methodological criteria to design a cognitive architecture are determined, according to the set of considerations presented in the previous sections. The purpose of an architecture is to adequately determine the knowledge and behavioral inputs and outputs, in agreement with the expectations of the designer (architectonic principles); and the criteria for using the design itself, according to the *basic principles of operation built into a cognitive system* (engineering principles) (Brooks 1962; Anderson 2013).

In methodological terms, cognitive architectures experience communicational and representational issues, making harder to successfully explain the integration between cognition, anatomy and behavior (Pachalska et al. 2012). Moreover, through several directives from decision making, perceptual and emotional instances, and consequently, building requirement categories, solutions are offered, in one hand, concerning the closest correlates to the ineffability of the mind and, on the other hand, fulfilling the need for immediacy in the expected results.

In the context of bioinspired cognitive architectures, it is important to discuss about the level of biological contribution to the development of processes and computational agents, which essentially constitute an architecture (Holland et al. 2013).

An important assumption for the objectives of this proposal is that the defining agendas for construction profile of cognitive architectures do not necessarily have to be mutually exclusive, as stated by Vernon (2017), as long as the level of modeling for intelligence to be addressed is clear. This is because subfunctionalities require validation at micro level,

supported by particular case studies; nevertheless, they need, as a whole, to be linked to a set of functionalities which together build the aforementioned desiderata.

For a general revision of the most relevant aspects for the construction of cognitive architectures, we consider five main categories, which comprehend crucial points about the theoretical delimitation, a specific decompositional scope, architectonical structure, representational language and the general validation criteria of the exemplified cases.

## 4.1 Generalities

General bases for cognitive architectures set up theoretical and practical instances according to the defined research goal.

We search for a defined research scope in reviewed working agendas, according to affordability in the realization of specific-to-general tasks; general *guidelines* offered by developers in order to get a comprehensive epistemological frame about the architecture and how it is created; and collaborating *theoretical foundations* on how all the aforementioned aspects were abstracted and synthesized in a consistent model.

Scope This parameter strives to align the orientation and the boundaries that an architecture can have at the time of its conceptualization, and what is the type of tasks that can be achieved with the chosen configuration.

Some BICAs propose to cover complex reasoning or behavior structures, whose cognitive components interact with each other at several levels, in order to set and follow a main knowledge domain (methodology for building). On the other hand, some others look to accomplish specific groups of tasks or abilities. Such tasks are subordinated to the domain, compatible with a proper set of features (methodology for operation).

The construction approach also considers the way in which architecture leads the agent to reason or behave, according to the level of inspiration and independence in the performance of the tasks, that the function should satisfactorily carry out and the theoretical approaches they are based on (Sun 2009).

Statement of guidelines and level The ontological level of methodology can be for building or for operation, as shown in Fig. 1. Design criteria are needed to direct how the procedures that build up the architecture should be carried out as a *whole*: references as a *manifesto*, a general manual (some of the reviewed architectures in this work only have technical operation manuals), or else, a trackable set of elements, that allows making inferences about consistency, unity, motivation and intentionality. It is advisable to specify their position regarding the levels mentioned above.

Definition of theoretical paradigm foundation This parameter refers to the description of the elements that support the architecture's specific design. This design delimits the CA's field of action by what its meta-architecture determines.

The concept of *paradigm* can be used to model cognitive functions as templates or building blocks for the interaction models to produce cognitive rules (Taatgen et al. 2006). Nonetheless, our meaning is closer to *normal science* definitions (Kuhn 2012), where a *paradigm* is the most pertinent action policy that is executed as a norm, until it is replaced by a better one in its referential frame.

Here we note that the set of paradigms commonly associated with the development of CAs, includes information of the guidelines to adopt given the architecture's formative structure, representational language, cognitive cycles, and rigor with which the interaction between its integrating elements is constituted. Some important paradigms used in computer science are: cognitivist, emergent and hybrid (Vernon et al. 2007; Kingdom 2008).



Therefore, the observance of the theoretical assumptions chosen at the architectural level is important, because it channels fundamental aspects of what is expected according to the disciplinary body chosen to build the architecture, the way of integration and representation of the models, its validation mechanisms, and the tools available for the tests.

**Modeling basis** This parameter begins by ordering the theoretical principles to consolidate a scheme that allows solving a given problem within the framework of the architecture. This parameter entails understanding of the studied phenomena, cognitive implications, and formal statements. An accurate modeling base is the first transition channel between the assumptions and the chosen set of functionalities, supposed to be following the architecture's initial approach.

This criterion also sets the knowledge representation levels in a case study to solve a specific task, desiderata or set of interactions between cognitive functions for a desirable behavior modeling, or any similar suitable schema.

## 4.2 Division into functions and integration mechanisms

Despite the ever-growing amount of studies on biological phenomena, there is not enough evidence in the literature to clearly define the initial processes from which a formal systematization emerges. It is possible to think, as a preliminary solution, of an algorithmic structure close to linearity; however, it is quite important to explain their inner constitution and how it is formally related with other subsystems in the same, lower and upper levels. Formal models of cognitive functions create a first work set, where the interpretation of biological phenomena is rigorously done in a first abstraction layer.

**Statement of desired abilities** In this parameter, the set of skills that form an integrated system of reasoning or behavior is determined. For this set, we consider the competency areas (Kotseruba et al. 2016), mainly focused on psychological experiments, human performance modeling (HPM), categorization and clustering, computer vision and human-computer interaction (HCI). To maintain modeling validity and the congruence of results, pragmatic approaches can be used (Varma 2011), (Adams et al. 2012).

The architecture should be extensible, in other words, capable of consistently integrating new functionalities, even in previously defined skill sets. Nonetheless, a successful integration of cognitive functions considering a general intelligence scheme has had little research (Langley et al. 2009).

**Components or modules** A modular configuration allows to determine a cognitive process's degree of centralization and its subfunctionalities' inter-operability, related to the brain as an organized system in a complex way (Fodor 1983). Modularity or community structure is a crucial element to design biological networks. In biological systems, such models arise in several processes and may enhance their adaptability and robustness to perturbations (Zeng and Ma 2013).

Frequently, a modular approach is coupled with decompositional analysis, which considers the analysis of the system in terms of its components (Anderson et al. 2004a). Nonetheless, we consider that modularity is a setup that helps to understand abstractions in a multimodal-stimuli setting (Lefort et al. 2011) which represent a locally operational model, aimed at global behavior.

### 4.3 Architectonic design

In the conformation of the architecture's components, differences between the more consolidated proposals in the state of the art were found. The stronger, more robust architectures that were revised, focused on the abstract representation of memory and learning. Variations in their interaction and hierarchy levels are also evident, like the preference for low granularity (numerous functions per module) subsystems over high granularity (single function per module) ones.

Cognitive architectures play a double role, one in the representation of cognitive phenomena, and one in the modeling of an informational system (Varma 2011). Regarding this second role, similar to the architecture of buildings, the definition of styles provides guidelines and restrictions for the structure's design (Malan and Bredemeyer 2005). Additionally, as proposed by Varma (2011), we consider parsimony and pragmatism to be design requirements for CAs; but we also highlight the possible (yet expected) contradiction between them, caused by the demand of functional emergence.

*Structural pattern* Structural representation involves knowing a cognitive function's constitutive composition, its formal modeling in a complex context, and how it relates to other functions at different levels, hierarchies and times (how it is composed and where it is directed to).

In the case of biologically inspired cognitive architectures, the level of interaction and linkage of functions can be determined at their lower levels; for instance, structural patterns could define the assumptions about network connections at neuronal activation level, which take into account phylogenetic, ontogenetic and morphogenetic properties (Bressler and Tognoli 2006) for specific problem resolution. The brain functions' structural pattern suggested by neurobiology is complex, which may be unsuitable to approach with the currently available computational methods and tools, hence we exhort other researchers to re-evaluate the technical possibilities of architectures to achieve human-like behavior.

*Functional approach* An architecture's functional approach determines the hierarchical levels, abstractions and simplifications of the cognitive models arranged as modules, in order to offer more comprehensive solutions.

### 4.4 Representation

An architecture implies an organized communicative expression or language (in broader terms than the eminently computational one) in which cognitive functions, and the data they use, are represented. This study also contemplates the conceptual or logical graphic schemes and the semantics that bind them.

*Graphic scheme* It refers to the graphic representation that describes an architecture's systematization, hierarchy and interaction between components. Through schemes, graphs or different conceptual approaches, it is possible to determine the proposal's general functioning, the adoption of cognitive cycles, or the importance of cognitive functions whose hierarchy is imposed to others.

In the case of neuroscience based CAs, a cortical parcellation is required to determine the concurrence and parallelism of the processes carried out by multiple cognitive functions, but that are located in the same cortical areas and that define complex high-level functions. Parcellation for such cases may be through comprehensive atlases such as those

proposed by Brodmann (2007), or with derivative systems such as *Talairach* (Lancaster et al. 2000) or *Desikan* (Desikan et al. 2006).

*Knowledge Language* This parameter refers to the form of communication and expression to be used by the architecture. Each expression might imply a different structure, which is why the choice of a proper knowledge language is crucial to bridge those dissimilarities. It is possible to establish languages based on first-order logics, ontologies or formal generative grammars, to name a few. The precision and clarity in the language used to understand and model cognitive phenomena are fundamental to avoid ambiguities.

## 4.5 Evaluation

In software engineering, the processes of *verification* and *validation* (commonly known as V&V) are used to evaluate the operation and determine the quality of the developed software. Verification consists on checking that the operations executed and their data (usually at the implementation level) are consistent with previously described requirements and specifications. On the other hand, validation goes beyond the execution and ensures that the developed system corresponds to the ideal or desired behavior.

*Validation process* Bringing V&V into the methodology of development of cognitive architectures/models, we can say that verification is primarily related to the functional design and implementation stages, while validation is especially important to determine if the overall design meets the main objective of achieving human-like behavior.

Most methodologies do not express how (or even if) V&V should be carried out as part of an evaluation process. However, some of them commonly perform validation through simulation of case studies to check if the model behaves as expected.

*Tools* They represent the software platforms where the actual implementations of a cognitive architecture are running. These implementations can be built ad hoc or be designed as a scenario for multiple related tests. Tools are useful for verification and validation processes. Some examples of them are simulation interfaces, frameworks and interpreters.

## 5 Review of methodologies

In this section, we briefly review some of the representative cognitive architectures. Then, a comparative schema is presented based on the previously described methodological aspects found in these works.

### 5.1 ACT-R

Adaptive control of thought-rational (ACT-R) (Anderson 2013; Anderson and Bellezza 1993; Taatgen et al. 2006), with nearly 40 years of development, is perhaps the most known cognitive architecture and has been broadly applied to perform different tasks. Despite its large list of publications and derived works, its methodological guidelines and recommendations are not concentrated in a unified proposal: while methodology of construction needs to be inferred through studying the evolution of the architecture itself; methodology of operation is exhaustively described in tutorial-like manners. Because of that, its scope goes from particular models within ACT-R, to more general recommendations on cognitive systems development.

Initially it heavily relied on using symbolic AI, but later it went through more cognitive paradigms while mapping its modules to both behavioral and neural evidence. The architecture covers various cognitive functions (memory, attention, visual processing, problem solving and declarative knowledge) and encourages the development of simple and complex tasks models (like language processing or spatial reasoning), so cognitive capabilities of the system could be increased (Juvina et al. 2018).

ACT-R has several predefined modules: visual module, visual buffer, manual module, manual buffer, execution, selection, matching, procedural module, goal buffer, intentional module, retrieval buffer and declarative module. Its structural pattern shows a central place for procedural module (which includes selection and matching), with bidirectional connections to input and output modules and their respective buffers. Functionally, it works in a cyclic manner, in which information held in the buffers is processed, then a single production rule fires, and finally the buffers are updated. Normally, an ACT-R model describes many of these processing cognitive cycles (Bothell 2017).

The evaluation of ACT-R models consists on running agent-based simulations in its software interpreter (the original uses Lisp language, but there are other implementations in Python), and observing if the model behaves as psychological data suggests. Nonetheless, in most cases, it is evaluated solely with respect to how well it fits the case study, what psychological theories it covers, and its own predictions (Ritter et al. 2017, 2019)

## 5.2 Soar

Soar (Laird et al. 1987; Langley et al. 2009; Laird 2012a) is other well-known symbolic cognitive architecture, focused on general intelligence. It has some common principles with ACT-R, even in the cognitive cycle aspect, but also has major theoretical differences, especially in control of conflict resolution (Johnson 1997; DiFilippo and Jouaneh 2018). Nonetheless, we focus in the methodological aspects it describes, rather than the architecture itself.

Soar architecture is based in the Problem-Space Computational Model (**PSCM**), a theory of decision making and problem-solving proposed by Newell (Laird et al. 1986, 2017b). Laird (2012a) describes an example of methodology to construct cognitive architectures, that finally derives in Soar. These guidelines are well explained and useful to construct any other cognitive system (Laird et al. 1986). The *research strategy* followed in Soar involves a firm selection of its theoretical approach, which is mostly psychological, compatible with evidences in neurosciences, and AI for development and performance. The main idea is to capture a subset of cognitive functions which is likely to be compatible with broader tasks, to achieve general intelligence.

Models are built in an agent-based way to understand their behavior in complex environments, and so they can span wide ranges of tasks, mainly related to action-selection. A practical tip to build a model is by asking ‘*what will Soar do with a task?*’ to complete it (Ritter and Young 1994; Ritter et al. 2019).

Thus, Soar is structured by multiple modules that interact with a central one: working memory. Soar’s functional approach is a processing cycle based on PSCM. Each cycle consists of four phases: input, operator selection, operator application, and output (Laird et al. 2017b).

The architecture has some basic modules, like declarative memory (episodic and semantic), working memory, decision procedures, production memory or chunking, learning and perception; and extension modules such as visuospatial buffer or motor module. The

memory modules are responsible for representing the knowledge through symbolic structures and production rules. Moreover, their developers consider additional modules like attention, sense of time, sense of self, language, emotions and motivation to be integrated into Soar.

The evaluation of Soar models consists on running the agents simulations in its current platform (Soar 9.6.0) and observe if the system is able to achieve the desired goal (Laird et al. 2017a).

### 5.3 LIDA

The Learning Intelligent Distribution Agent (**LIDA**) is a cognitive architecture developed by Stan Franklin and colleagues (Cognitive Computing Research Group) at the University of Memphis. The LIDA model is an extension of the IDA (Intelligent Distribution Agent) model (Franklin and F.G. Patterson 2006; Franklin et al. 2013, 2016).

The LIDA model is a conceptual and computational model attempting to cover a large portion of human cognition. It is based mainly on Baars's Global Workspace Theory (**GWT**), the most widely accepted psychological and neurobiological theory of the role of consciousness in cognition (Baars and Franklin 2009; Franklin and F.G. Patterson 2006; Snider et al. 2011), and a number of other psychological and neuropsychological theories. The LIDA model and its architecture are grounded in the LIDA cognitive cycle, which is produced by the collective, coordinated actions of its subsystems (sensation, perception, working memory, episodic memory, consciousness, learning and action selection). This cycle consists of three phases where the agent must sense the environment and select an appropriate action (perceive-interpret-act) (Franklin and F.G. Patterson 2006; Franklin et al. 2016).

In LIDA, knowledge is represented by different memory modules and learning mechanisms. Perceptual knowledge uses ontologies and is organized as a slipnet, a semantic network with passing activation, while episodic knowledge is based on content-addressable memories, which are related to Kanerva's sparse distributed memory (**SDM**) (Franklin and F.G. Patterson 2006; Snider et al. 2011).

The development of software agents and robots for specific problem domains is carried out through the LIDA framework, a software implementation of the LIDA model in Java language. The implementations' evaluation is based on the development of cognitive agents that can replicate the results obtained from experiments on human subjects (Snider et al. 2012; Franklin and the Cognitive Computing Research Group 2018).

### 5.4 SiMA

In the context of the SiMA project (Simulation of Mental Apparatus and Applications), Schaaf (2016a) proposes a case-driven methodology to develop mental architectures, composed by five main stages: analysis, specification, functional modeling, implementation and evaluation. This is the only methodology (in this review) that is described independently of the architecture in which it was conceived, so we consider it more general than the others.

The first stage: *analysis*, states the importance of setting guidelines and selecting theoretic paradigms concerning the research question (as required by the cognitive process to model), while the researcher or developer is responsible of such decisions. It encourages the paradigm selection considering the specific goals of the research program. Meanwhile,

the second stage: *specification*, sets the modeling basis by proposing a composition of consistent processes' descriptions into exemplary cases (use-cases).

Next, the third stage: *functional modeling* integrates previous assumptions and considerations about the cognitive process into a unified function. In this stage, the methodology only suggests making decisions about the model's level of detail, however, it does not provide information on how to functionally form the model, and does not even discuss its structural aspect.

The fourth stage: *implementation*, the inputs, outputs, knowledge representations and algorithms to be used by the function are defined. A software is generated, considering the functional model obtained from the previous stage (Schaat 2016a). In the last stage: *evaluation*, the resulting model is compared in virtual simulations against the model specifications and the previously specified simulation case scenarios, used as templates.

SiMA relies on a holistic functional model of the human mind and is based on two foundations from psychoanalysis. The first one is resolving conflicts between motivations; the second one, is the separation of functionalities that process conscious and unconscious data (Wendt et al. 2015; Schaat 2016b). Multiple subsystems like embodiment, drive track, perception track, selection of actions, Super-Ego rules, defense mechanisms and selection of needs, compose the SiMA architecture (Fittner and Brandstatter 2018).

This cognitive architecture's approach is needs-driven, the embodiment subsystem generates desires that are used to create goals. To achieve these goals, SiMa uses a multi-cycle approach, which means it can take several cycles before evoking an external action (Wendt et al. 2015). In SiMA, knowledge is represented as an ontology, stored in memory and helps in the decision-making process. However, it cannot store experiences, whereby a learning mechanism is required (Fittner and Brandstatter 2018).

The implementations and evaluations are carried out in the MASON framework, a multi-agent simulation library written in Java (Luke et al. 2004; Fittner and Brandstatter 2018), to prove the correctness of the functional model and its underlying concepts.

## 5.5 NEF (SPAUN)

The Neural Engineering Framework (NEF), proposed by Eliasmith and Anderson (2004), introduces a mathematical theory for the construction of bioinspired models, based on neural behavior for a wide variety of dynamic functions. It is strongly based on computational neuroscience and neurobiology (instead of classic approaches which opt for psychology), and it uses neural-populations (not single neurons) or brain areas as modeling unit.

From *How to build a brain* (Eliasmith 2013), it is possible to highlight the major aspects of their methodological approach to construct models. We consider its scope as general-to-particular, since despite the methodological aspects can be applied to almost any bioinspired neural model, it is mainly intended to build models within NEF. The most complete and sophisticated functional brain model constructed and described in NEF is Spaun (Semantic Pointer Architecture Unified Network), which is directed by eight specific desired tasks that cover diverse kinds of challenges for bioinspired cognitive systems (copy drawing, recognition, reinforcement learning, serial working memory, counting, question answering, rapid variable creation and fluid reasoning) (Eliasmith et al. 2012). However, it does not claim that a general cognition will be achieved once the tasks are successfully completed.

Overall, NEF's functional approach is stated by a set of principles for constructing the neural models, which involves representation, transformation and dynamics (Sharma et al.

2016; Voelker et al. 2017). Following these principles, the components of a model are the brain-area-inspired functional elements, and its structural pattern is formed by their connections (information flow) as well as the organization of components into hierarchies, subsystems or central mechanisms.

Evaluation is carried out through execution of simulation tests using Nengo (Bekolay et al. 2014; Sharma et al. 2016), an implementation tool derived from NEF, which allows the simulation of large-scale models. Moreover, Eliasmith (2013) gives some *core cognitive criteria* to assess “how good” a cognitive architecture is, which include representational structure, performance concerns and even scientific merit. Even though the criteria list is not complete, it stands out among other methodologies that do not offer this type of practical-to-philosophical discussion.

## 5.6 iCub

iCub is both an open systems platform and a child-like humanoid robot testbed for research of human cognition and artificial intelligence (Metta et al. 2019). It incorporates two levels of biological emulation: the constitutive properties of living entities *phylogenesis and ontogenesis*. These are considered among the essential requirements modeled through the agent’s experience with the environment, which prioritize sensory modalities for the consolidation of what the authors call *developmental cognitive systems* (Vernon et al. 2011). These systems interact with other influences to build a set of *roadmaps* that define the architecture’s orientation. iCub develops an enactive and embodiment approach, whereby a cognitive system develops its own understanding of the world around it through its interactions with the environment (Metta et al. 2010, 2019).

This cognitive architecture’s design is founded on humans’ developmental psychology and neurophysiology. It mainly considers neuroscientific knowledge about action, perception and cognition. It consists of multiple components that operate concurrently, thus, it is not governed by a state machine as many CAs are. iCub has thirteen components: procedural memory, episodic memory, attention selection, egosphere, exogenous salience, endogenous salience, gaze control, vergence, affective state, action selection, locomotion, reach and grasp, and the iCub interface (Vernon et al. 2011). Despite most of these components have been implemented, there are pending modules that require further development (Metta et al. 2010, 2019).

As mentioned in Metta et al. (2010), it uses the phylogeny and ontogeny to define the innate skills with which the iCub must be equipped, so that it is capable of ontogenetic development, of defining the ontogenetic process itself, and of showing exactly how the iCub should be trained or to what environments should it be exposed in order to accomplish this ontogenetic development.

The iCub architecture is built on top of YARP, a middleware that helps in abstracting algorithmic modularity and hardware interfacing. It does not define validation criteria; however, they plan to make testing in the same manner as a developmental psychologist would test an infant in a laboratory experiment (Metta et al. 2010; Cangelosi and Schlesinger 2018).

## 5.7 SEMLINCS

SEMLINCS (SEMantic, SEnsory-Motor, SElf-Motivated, Learning, INtelligent Cognitive System) is a cognitive architecture developed with the main goal of investigating how a

control structure can learn production rule-like conceptual structures from sensorimotor experiences, and how these structures are able to generate human-like behavior and cognition. It is built mainly based on Butz's subsymbolic theory of cognition (Butz 2016). However, it also uses ideas from AI, machine learning, psychology, biology, linguistics, computational neuroscience and cognitive modeling (Schrodt et al. 2017a, b).

In order to create an autonomous and self-motivated behavior system, SEMLINCS uses a methodology based on the principle of *autopoiesis* and intrinsically motivated systems. It incorporates modules for a motivational system, schematic knowledge, schematic planning, sensorimotor planning and speech comprehension. However, these modules require improvements in the current implementation, as well as the addition of new components, like episodic memory, which will allow the agent to move back in the environment. It also requires abilities for cooperation between agents to achieve a goal (Schrodt et al. 2017b).

Its cognitive cycle works through self-motivated continuous learning of conceptual structures from sensorimotor experiences. The learning mechanisms are based on formalizations of free energy-based inference and part of this knowledge is represented using production rule-like structures (Schrodt et al. 2017b).

SEMLINCS was developed in a Super Mario-based virtual environment, however, it can be applied to other virtual environments. The validation depends on if the agent is able to successfully accomplish a level, and its criteria are practical (Schrodt et al. 2017a, b; Shapshak 2018).

## 5.8 Summary

After the general revision of the representative cognitive architectures, in Table 1, we detail and classify a set of elements used as a guide to obtaining the respective cognitive architecture. We use these elements to propose a conciliatory method. A key problem with much of the literature is the missing of methodological documents associated with CAs, in consequence, a comparison of general methodologies on a larger scale becomes unattainable.

## 6 A methodological proposal to develop cognitive architectures

Considering the review of methodological aspects present in some relevant CAs, we propose a methodology to construct cognitive architectures. This proposal is suitable for cases where multiple functionalities or abilities (such as perception, decision-making, attention, planning, etc.) interact with each other.

This approach is useful because it allows splitting up a broad cognitive process into more tractable sub-systems, and this division can be re-applied iteratively, resembling how most neuroscientific-based approaches work (Cooper et al. 1996).

Figure 2 shows a diagram with all the components, flow, and stages of our proposed methodology. Below, we present a detailed explanation for each stage and its associated processes.

### 6.1 Meta-architecture

The meta-architecture stage comprises various steps at the beginning of the development. We propose two decision levels in this stage: a general one, intended for the complete

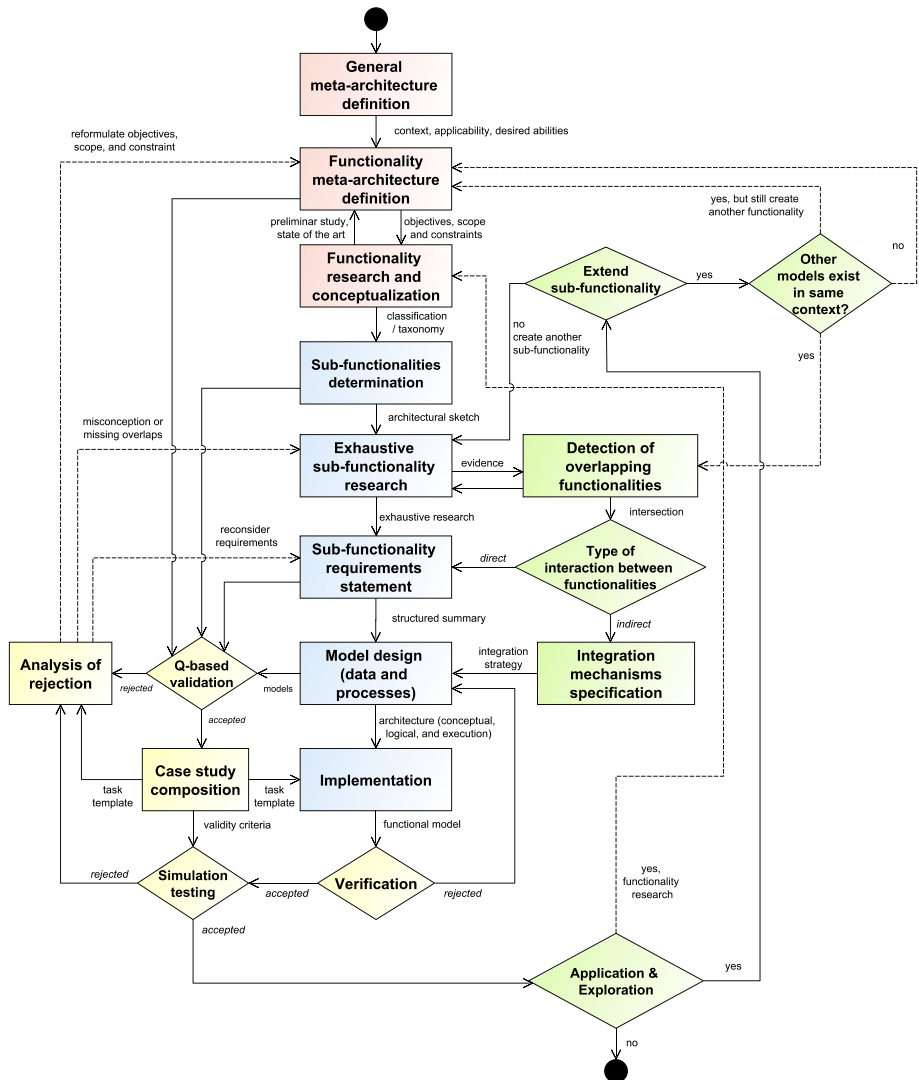


**Table 1** Comparative table of methodologies to construct cognitive architectures

Methodology	SiMA Schaat et al. (2014)	NEF-SPAUN Elia-smith (2007, 2013)	SOAR Laird (2012)	ACT-R Anderson et. al (2004)	iCUB Vernon et al. (2007)	LIDA Franklin et al. (2013)	SEMLINCS Schrodt et. al. (2017a, b)
<i>Generalities</i>							
Scope	General	General and par-ticular	General and par-ticular	General and par-ticular	Particular	General and par-ticular	General and Par-ticular
Guidelines and level	Yes, building.	Yes, building and operation	Yes, building and operation.	Yes, building and operation.	Yes, building.	Yes, building and operation.	Yes, building and operation.
Theoretic paradigm	Psychoanalysis	Neuroscience, Biology and A.I.	Symbolic A.I, Cognitive sciences	Symbolic AI, Cognitive sciences.	Developmental psychology, Neurophysiology, and Enaction	Neural. Cognitive sciences.	Subsymbolic cognition/
Modeling basis	Use-case driven	Brain-area based (Neuralpopulation)	Use-case, action-selection tasks	Use-case, action-selection tasks	Predefined innate map Developmental mental functionalities	Use-case-Per-ceiveinterpret-act	Autopoietic sustainable system
<i>Divisions</i>							
Desired abilities	Stated, but flexible	Stated initial tasks	Stated, but flexible	Stated, but flexible	Stated, but flexible	Stated, but flexible	Stated, but flexible.
Components	Predefined	As required (brain-area based)	Predefined	Predefined	Predefined	Predefined	Predefined
<i>Architectural</i>							
Structural pattern	Layers and Subsystems	Flexible -Hierarchies, systems and subsystems	Components interacting through a central working memory	Specific -Central, peripheral and buffer modules	Metastructural/ sensorial instances for modeling adaptive behavior	Coordinated sub-systems	Flexible
Functional approach	Multi-cycle	Predefined, following Principles of NEF	Specific, processing cognitive cycle	Specific, processing cognitive cycle	Enactive developmental cognition	Specific, processing cognitive cycle	Self-motivated learning of conceptual structures from sensorimotor experiences.

**Table 1** (continued)

Methodology	SiMA Schaaf et. al. (2014)	NEF-SPAUN Elia-smith (2007, 2013)	SOAR Laird (2012)	ACT-R Anderson et. al (2004)	iCUB Vernon et al. (2007)	LIDA Franklin et al. (2013)	SEMLINCS Schrodt et. al. (2017a, b)
<i>Representation</i>							
Graphic scheme	Undefined	Yes, for brain areas	Yes,	Yes, module and brain-area	Yes.	Yes, module.	Yes, module.
Knowledge language	Ontology	Semantic pointer	Productionrule (chunks)	Productionrule	YARP framework	Ontology	Formalizations of free energybased experience
<i>Evaluation</i>							
Validation Criteria	Stated, practical	Stated, philosophical and practical	Non stated, but practical	Stated, practical	Practical	Replication of experiments	Practical
Tools	MASON framework work	Virtual simulation. Nengo neural platform	SOAR 9.6.0	Virtual Agentbased simulation. ACT-R interpreter	YARP	LIDA Framework (Java)	Virtual Agentbased simulation.
				Declarative Chunks		Sparse distributed memory	Production rule-like



**Fig. 2** A general sequence diagram of the proposed methodology for the development of cognitive architectures. Stages of the meta-architecture (red shade), general architecture and Specific architecture (blue shade), evaluation and validation (yellow shade), intersection and integration (green shade). (Color figure online)

cognitive system and where it is possible to define information cycles or flows; and a specific one, to study a modular cognitive function individually.

*General meta-architecture definition* An architecture’s theoretical basis and structural patterns, as defined in Sect. 4, cannot be easily changed at posterior development stages. Because of that, it is important to determine the aspects that will define and direct the expected results.

A meta-architecture is the set of high-level decisions that are related with the system’s structure and integrity, although it is not the structure itself (Malan and Bredemeyer 2005).

This specification defines architectural patterns as well as theoretical and philosophical principles that guide the architecture's structure.

General meta-architecture definition is a process directly related to intentions, hypothesis and experience of a research group. It sets up the context, applicability, desired abilities, architectural patterns and styles for the whole cognitive architecture.

In the case of bioinspired CAs, neuroscience is one of the central sources of evidence, adopting brain area-based studies rather than neuronal-based ones. For our proposed methodology, we additionally consider psychological studies about expected behaviors, mainly to support establishing design assumptions, to create graphical and logical schematizations and to do validations. Additionally, our architecture is composed by several cognitive functions working on different hierarchical levels, like sensory-motor system, attention, perception, motivation, emotion, memory, planning and decision-making. In this context, we aim to model core brain areas and their proposed operations, adopting computational approaches to exteriorize them. All these definitions are part of our general meta-architecture.

*Functionality meta-architecture definition* This scheme is guided—but not strictly restricted—by the general meta-architecture. Its purpose is to define specific objectives, scopes and limitations for each cognitive function. For this reason, a revision of the state of the art is required. The outcomes of this process can be refined according to the evidence found during further steps.

*Functionality research and conceptualization* The first research and conceptualization iteration is carried out to gather information about the current understanding of a specific functionality in humans or non-human primates. It is crucial to study the perspectives and theories present in the research fields established during the meta-architectural stages, and their evolution over the years, to explain behaviors associated with the function.

Nonetheless, as the development progresses and new knowledge is considered, it is possible to define more suitable guidelines and general categories according to the research objectives.

## 6.2 General and specific architecture

This stage leads to the generation of computational models which can be integrated in the general architecture. It comprises three main activities highly related to software engineering: analysis of requirements, model design and implementation.

*Sub-functionalities determination* An undetermined number of iterations between the *functionality meta-architecture definition* and the *research and conceptualization* stages may be required to successfully execute this step and, even then, it can be adjusted as the development progresses.

This step consists in the realization of the architectural sketch, in which the main processes or *sub-functionalities* are identified; supported by the information about the functionality gathered in the meta-architectonic stage. The number of sub-functions is bounded by the previous methodological stages' results.

It requires the creation or adoption of a taxonomy for the functionality among other concurrent functionalities, along with its internal partitions to build a sub-functionality-based general structure for the architecture.

*Exhaustive sub-functionality research* Once the sub-functionalities are determined, one of them must be selected to be subject of an exhaustive analysis, which involves collaboration and evaluation with a multidisciplinary team of experts in neurosciences, psychology,

cognitive sciences and other experts of areas involved in the design to obtain a broader overview of the chosen functionality. Since each sub-functionality may imply different processes, the following steps should be applied to each of them.

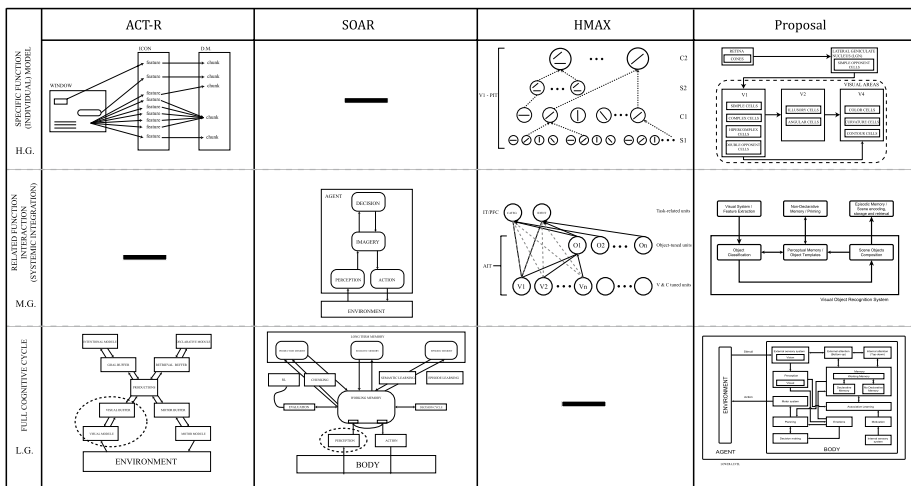
In our case, the neuroscientific study serves to detect specific components (brain areas) involved in a sub-function, and, along with psychology, to define their possible processes and dynamics. As can be seen in Fig. 3, it is necessary to recognize the path that the function will take.

*Sub-functionality requirements statement* This step is executed based on the information gathered during the *exhaustive research*. In this stage, the developers detect and describe each of the functional and non-functional requirements for each component of the system, which must be validated by experts. This is further discussed in Sect. 6.4.

In our proposal, we suggest a direct mapping between a singular component to brain structures or nuclei, in order to prove that requirements are precise enough to associate theoretical or biological evidence to a component. Then, we use the set of functional requirements to make a *model design*. In the case we are developing the integration of two different sub-functionalities, the defined requirements of each one must be joined in such a way that both are compatible.

*Model design (data and processes)* In this step, the proposed methodology takes in consideration three types of design: *conceptual design*, *logical design* and *execution design*. The first, builds up system's components using connection diagrams, supporting those connections with evidence from the fields selected during the meta-architectural stage.

Logical design describes the type of information that is transmitted over these connections and the proposed operations of each component, given a specific processing flow, using collaboration and activity diagrams. Execution design portrays how the components will be implemented as a software system using process and nucleus diagrams.



**Fig. 3** Specificity of the visual system model in terms of low (LG), medium (MG) and high (HG) granularity: a) As an individual function appearing as a component of a cognitive cycle (ACT-R) [LG]; b) Implicitly considered in the perceptual module (Soar) [MG]; c) As highly detailed models from computer vision, that seek to solve specific problems (HMax) [HG]; d) As an inherent part of a general cognitive complex flow, at the three proposed levels. Dashed circles highlight where the visual function is located at

As described above, the interaction and hierarchical levels of the studied cognitive functions are considered. In BICAs, the anatomical connections between brain areas can be used to guide the conceptual design. For example: Human visual system contains processes and input data types that support representations with varying degrees of granularity, involved in low-level visual processing and selective attention.

Models from neurosciences and bioinspired computer vision agree that the initial activations of the visual pathways include transformations from photoreceptors in the retina, which pass through thalamus, and then to the occipital cortex; this information can help direct the logical design instance. These areas are responsible, to different extents, of extracting features, taking in account the stimuli's physical properties (bottom-up way), and enhancing them according to task-relevance, given their features or localization (top-down way); this can guide the execution design step.

As shown in Fig. 3, a vision module could be seen either as an individual high granularity (**HG**) (or high level of detail) instance, like in computational vision systems similar to *HMax*; or as a component of a more complex sensory system, which describes relationships with proximal functions at the same level, in a medium granularity (**MG**) range, conforming a subsystem; or as an integral part of a general scheme that represents a cognitive cycle or a more comprehensive system (low granularity) (**LG**).

*Implementation* This is the last process of the *general and specific architecture* stage. Due to our methodology's approach to handle cognitive processes by their parts, the resultant architecture is a set of structured components, with their own operations and information transformations, and that are able to interact among them at any time. For that reason, we propose it should be implemented as a distributed system.

Furthermore, we use a custom middleware to develop distributed cognitive architectures (Jaime et al. 2015), which has two types of template nodes: Big nodes (**BN**), that represent the components and carry out communication processes between nodes; and small nodes (**SM**), which carry out specific processes in the execution design.

### 6.3 Intersection and integration

As mentioned before, this approach considers that the architecture is composed by various individual, but highly interconnected, cognitive functions. So, the integration's main objective is a progressive and consistent aggregation of functionalities or sub-functionalities.

In order to develop and mature the architecture, the first step is **detecting overlapping functionalities**, this implies analyzing which other functions (or sub-functionalities) are related to the current model.

This means the functionalities could be developed as separated systems, but an **integration mechanisms specification** must define strategies to connect them. We argue these strategies should answer four main questions:

1. *What can we currently do to add the other functionality?* If it has not been developed yet, a black-box integration can be considered, or enough resources can be assigned to start this new independent function's complete development. In case that the functionality (or part of it) already exists, the model and the corresponding research documentation should be gathered to analyze the remaining questions.
2. *Where are the connection points among the functionalities?* This is to identify the components in one function which send or receive information from another function's components, or to find the ones that obtain information from both.

3. *What kind of information is generated, stored and shared by the functionalities?* It is important to identify the format, types and data structures of such information, to conform function connection interfaces for current and future modules.
4. *How do functional and structural based interactions make possible emergent behavior at several cognitive levels?* A modularity principle alone is insufficient to make an effective integration of sub-functionalities and to offer a satisfactory explanation of complex cerebral function; therefore, it is important to precise where and how the corresponding links are established.

*Extend sub-functionality* With the exploration of the proposed model; we must be able to specify possible improvements or extensions to it. It can extend the model by creating another sub-functionality from the proposed taxonomy and start from the research basis stated in Exhaustive sub-functionality research. Nevertheless, if we want to extend an overlapped cognitive function to our model that is not part of the proposed taxonomy, it must be decided the way to proceed with the development.

*Other models exist in same context?* To extend the model with a function that is not part of the taxonomy, we have to make a research of existing models in the state of the art that fulfill the context specified in the general meta-architecture definition is required. If there is not a model that covers the context, the development of a new function is required. Nevertheless, if we found some models that cover our requirements, we must choose the most appropriate one and proceed to the detection of overlapping functionalities between it and our model. However, even if we found these models, we can still choose to start the creation of another functionality instead of using an existing one.

*Detection of overlapping functionalities* The search for overlaps in modeled cognitive functions, caused by the heterogeneity between the research areas and theories involved in cognitive sciences, is carried out both in the processes analyzed and in the descriptions of cerebral cortex (if applicable). The functions that appear concurrently, but which may present intersection points, must be more strictly monitored, especially to prevent different interpretations from the same field of study. Structurally, it means keeping registry of the participating cortex, responsible for multiple cognitive processes.

In the case of the integration of two different sub-functionalities (different taxonomies). It must be specified the common overlaps both on theoretical and cerebral cortex levels. In this way, we can start to consider the integration of both functionalities as a single but complex sub-functionality.

*Type of interaction between functionalities* In all the aforementioned scenarios, it is necessary to analyze the existent type of interaction between functionalities. We consider *direct* and *indirect* interactions.

Direct interaction points out that functionalities are highly dependent one to each other, regularly sharing information. In such conditions, the development team should consider both functionalities' requirements, to make sure that their bases (biological evidence) and constraints are satisfied. In the case of the intersections detected during the fusion of two different sub-functionalities, all the common overlaps are considered as direct.

On the other hand, an indirect interaction assumes that functionalities are not strongly related in most of their components, even though they share information at one point or another.

*Integration mechanisms specification* If the type of function interaction is direct, proceed to establish the sub-functionality requirements statement, as shown in Fig. 2.

If it is determined that there is an indirect interaction between the modeled cognitive functions, an intermediate model is proposed which considers the intersection areas, where only the function of interest's functionalities need to be properly modeled, while the others can be replaced by black-box abstractions or similar simplifications. It is crucial to attend *What kind of information is generated, stored and shared by the functionalities?* at this point, establishing consistent and suitable data types and structures for such abstractions.

## 6.4 Evaluation (verification and validation)

According to the summary of software architecture analysis (Dobrica and Niemela 2002), there are two classes of techniques to evaluate an architecture: the first is *questioning*, which extracts qualitative properties; and the second is through *measurement*, which obtains results for specific qualities. Our proposal is a combination of both at different development stages of cognitive architectures and models, as follows:

Since the earliest development stages (from meta-architecture definition to determination of sub-functionalities, and then statement of requirements), a recurrent and general-to-specific **question-based (Q-based) validation** will help to ensure the understanding of the functionality and its intended model. The idea is to determine if the context, objectives, classification, etc. appear to be theoretically accurate, and under which circumstances that is true.

The Q-based validation is proposed by the development team and addressed mainly to an interdisciplinary team of experts in cognitive sciences, although it could be performed with other sources of knowledge. This will help to refine meta-architectural information through various iterations, even before the *model design* step begins.

*Verification* In this stage the model must be applicable (validation) and correct (verification). This assumes the precision of the transformation of the model to a computer implementation (Balci 1994). In the literature, there are methodologies for validation, verification and testing for cognitive functions models.

*Q-based validation* It refers to the legitimacy made by the body of expert advisors to the conceptual, logical and execution designs presented, considering their consistency, as well as their individual and collective integrity.

*Case study composition* It consists of the experiment's proposal that will be used to validate the model's implementation. When building the experiment, it must specify the procedure, the required inputs, the expected results and the variables of interest to be evaluated. However, in some cases, and depending on the function, the experiment must be taken directly from the most common ones performed in human beings.

*Simulation testing* Once the cognitive model and its implementation have passed the verification processes, proceed to execute the computational simulation following the procedure described in the proposed case study. Finally, the variables of interest taken from the results obtained from the execution are evaluated against the expected results defined in the experiment, which represents the validation process.

*Analysis of rejection* If a fault or inconsistency is detected in any of the previous steps, an analysis of rejection is performed, which requires a complete revision of the evaluated models, to determine what originated it. If there is sufficient evidence about the inconsistency, the evaluated models are sent back to the analysis phase. Otherwise, we can proceed to the next stage.

*Application and Exploration* If the results obtained from the simulation testing are positive, we can state that the obtained system is ready and can be deployed in other



experiments related to the developed cognitive function. Also, We could decide to mark as finished the current development of the function, or we can continue with the exploration of new applications, possible improvements to the model or changes in its conceptualization.

## 7 Case study

The proposed methodology has been used to develop models for a specific bioinspired cognitive architecture, such as *perception* (González-Casillas et al. 2018), *attention* (Avila-Contreras et al. 2014; Martin et al. 2016), *recognition* (Jaime et al. 2014) and *visual system* (Torres et al. 2011). In order to show the methodology's ability to handle a modular approach for cognitive architectures development, next, we apply the methodology for creating models of working memory and visual perception cognitive functions, then is illustrated how our proposal is useful for linking both cognitive functions in order to fulfill more complex behaviors.

### 7.1 Case study 1 Working memory

Next, we describe the development of a cognitive model of working memory using the proposed methodology. We explain the underlying design decisions taken at each stage of our proposal.

One of our main goals is that the model must be able to perform one of the basic working memory tasks: a Sternberg task (Sternberg 1966), and its results can be compared against human results in the same task. However, a more complex case may require extensive research, therefore it is beyond this work's scope.

*General meta-architecture definition* First, it is necessary to establish the general meta-architectural aspects in which the model is immerse. In our case, it is part of a broader (but not complete) cognitive architecture, whose main purpose is to provide human-like behaviors for computer systems, like virtual agents. The architecture is composed by several cognitive functions working on different hierarchical levels, like sensory-motor system, attention, perception, motivation, emotion, memory, planning and decision-making.

As we determined, neuroscience is the central source of evidence we consider, preferably brain-area based studies instead of neural-based ones. In addition, we support our work with psychologic studies about expected behaviors, mainly for validation purposes. In this context, we pursue modeling core brain areas and their operations, and we adopt computational approaches to model those operations.

*Functionality meta-architecture definition* In this case, we focus on the **working memory** function. It refers to the capacity to maintain temporarily a limited amount of information in mind, which can be used to support various abilities, including learning, reasoning, and preparation for action (Baddeley and Hitch 1974; Squire 2009). It is considered to hold only the most recently activated, or conscious, portion of long-term memory, and it moves these activated elements into and out of brief, temporary memory storage (Sternberg 2011).

Initially, the model's scope was to identify the basic working memory sub-functions and its interaction mechanisms with other cognitive functions. Then, to describe and make a prototype addressing its basic aspects. We built it as a system with both, short-term storage and information manipulation capabilities. However, due to this function's broad range of interactions, this work was limited to model the ones with the declarative memory, non-declarative memory, planning, decision-making and perception systems.

*Functionality research and conceptualization* In this stage, we present a preliminary study to define, according to the research fields and theories we adopted, what working memory is, how it behaves and what are the processes it performs.

As we stated before, working memory manipulates and temporarily maintains a limited amount of information in mind, which can then be used to support various abilities, including learning and reasoning. Unlike short-term memory, working memory is not exclusively a storage site, but also a framework of interacting processes that involves the temporary storage and manipulation of information in the service of performing complex cognitive activities (Baddeley et al. 2011).

Due to interest in explaining working memory's processes, several models have been proposed in the literature. Some of these models are purely psychological; others try to establish the neural basis of working memory. In order to develop our model's proposal, we took into account works from Baddeley (Baddeley and Hitch 1974; Baddeley 2000), Oberauer (Cowan 1999; Oberauer 2002, 2007), Goldman-Rakic (Wilson et al. 1993; Levy and Goldman-Rakic 2000; Goldman-Rakic 2011) and Mark D'Esposito (D'Esposito 2007; D'Esposito and Postle 2015). Thus, the models considered were:

- Multi-component model, by Baddeley.
- Active memory, by Cowan and Oberauer.
- Domain-based segregation of Prefrontal Cortex.
- Neurophysiologic model, by Mark D'Esposito.

Working memory is a specific type of memory, therefore it requires the basic functions (encoding, storage, retrieval and forget) described in Schacter and Wagner (2013) and Jaime (2016). However, several studies, for instance (Bledowski et al. 2010), (Baddeley 2000) and (Oberauer 2002), have found that there are additional operations exclusive to working memory:

- Update: Working memory's content is constantly being updated with new information.
- Filtering/Inhibition: Prevents the storage of irrelevant information.
- Goal update: It stores and updates the current goal's representation.
- Decay: Stored information loses relevance through time, until it is removed from memory.
- Maintenance/Refreshing: It increases the relevance of memory traces, to prevent decay.
- Manipulation: It allows making changes to the stored information, like sorting by priorities.

Besides these functions, working memory has some characteristics that distinguish it from long-term memory: *limited capacity*, *distributed nature* and *abstraction levels*.

In the field of cognitive architectures, there are several research groups working on their own models, which were designed based on different approaches, from biologically inspired models to purely computational. Table 2 summarizes the main features of some related CAs.

*Sub-functionalities determination* Despite the plethora of information about the tasks performed by working memory, there isn't a defined classification or taxonomy of it. However, considering the models and characteristics described in the *general research and conceptualization*, we proposed one. This classification is geared towards the sensory modalities and the complexity of the information handled by the different brain structures involved.

Our model consists of several buffers used to handle different information types. Each buffer represents a working memory sub-functionality, see Fig. 5. The proposed sub-functions are:

- **Declarative working memory:** It is responsible for keeping online and performing the proposed functions over the task-relevant information coming from semantic and episodic memory. The information represented by this sub-function has the highest abstraction level.
- **Perceptual working memory:** It holds relevant objects' features from different sensory modalities such as visual, auditory and tactile. It sends top-down signals in order to maintain or reactivate information in the areas that originally processed it. Also, this function may be used to provide additional information to declarative working memory. The information represented by this sub-function has the lowest level and is sensory modality dependent.
- **Non-declarative working memory:** It holds information about the most relevant responses associated with a perceived stimulus. By storing the Stimulus-Response (**S-R**) mapping, it allows faster responses in high-level cognitive functions like planning and decision-making.
- **Blackboard:** Stores references to current and previously presented stimuli and actions. It also stores task rules, generated plans, desired goals and any associations between stimuli created by other functions. The blackboard may be considered as a superset of the task-set or the task-set's source of information.
- **Emotional working memory:** There is additional information coming from other systems which are not part of the previous sub-functions. Thus, we propose an additional sub-function to maintain stimuli coming from systems like emotional and motivational.

#### *Exhaustive sub-functionality research and detection of overlapping sub-functionalities*

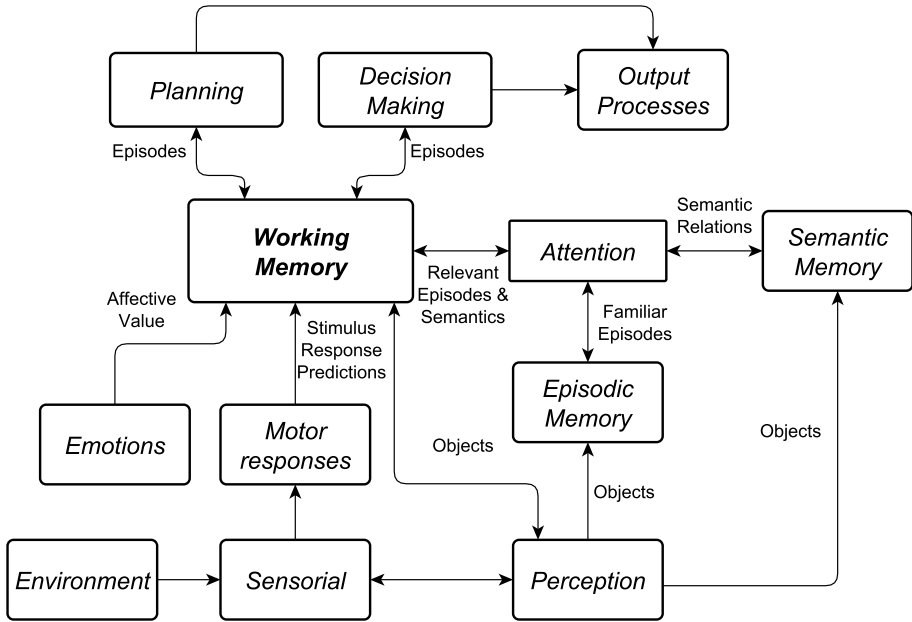
Once we defined the classification of sub-functionalities, we chose to develop the **declarative working memory (DWM)** sub-function, which we break down and explain in detail below.

We use the term declarative working memory to denominate the part of working memory that receives and holds information directly from both types of declarative memory: semantic and episodic (high-level information). It must be able to provide and integrate information of semantic relations, scenes and episodes. It means that DWM has the most abstract information.

DWM's design starts with an analysis of related cognitive functions that could impose development limitations. Then, we present the alternatives to solve such limitations.

As mentioned before, working memory is required to perform numerous goal-oriented tasks. This means that it is linked directly to those functions that, like it, help achieve goal-oriented behavior. In order to attain a goal, functions like **planning** and **decision-making** require the use of information held in working memory, but which was created by other functions across multiple brain areas. Therefore, working memory is also linked to those sources of information.

Hence, functions like declarative memory, non-declarative memory, perception, emotions and motivation represent sources of information that reach working memory, as shown in Fig. 4. Of all these mentioned intersections, only *planning*, *decision-making* and *declarative memory* have **overlapping processes** with DWM. Due that long-term declarative memory represents DWM's main information input, and that planning and



**Fig. 4** Functions that are related to working memory

**Table 2** This table shows the main features of the related cognitive architectures: ACT-R (Anderson et al. 2004b), Soar (Laird 2012b), LIDA (Franklin and F.G. Patterson 2006), CHREST (Gobet and Lane 2010/06), CLARION (Sun 2006), and EPIC (Kieras and Meyer 1997)

Cognitive Architectures						
Function	ACT-R	Soar	LIDA	CHREST	CLARION	EPIC
Explicit working memory	x	Yes	x	Yes	x	Yes
Buffers by modality	x	Perceptual General	x	Auditory Visual General	Sensory Goals	Perceptual General
Limited capacity	x	x	x	Yes	Yes	Partial
Approach	Biology	x	Psychology	x	Psychology	x
Can work on isolation	x	x	x	x	x	x
Distributed	x	x	x	x	x	x
Interaction with other functions	x	Yes	Yes	x	x	x
Different abstraction levels	x	x	Yes	x	Yes	Yes

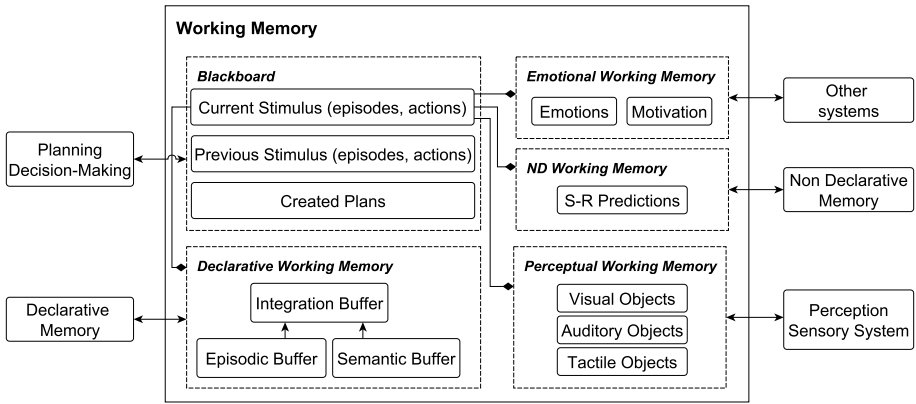


Fig. 5 Working memory sub-functionalities and their interactions with other systems

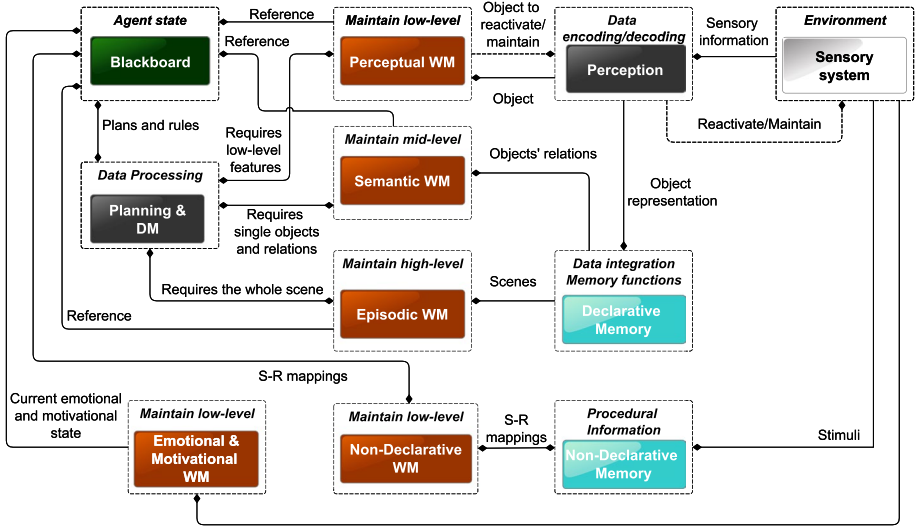


Fig. 6 Conceptual diagram of declarative memory and its relation to non-declarative memory, the sensory system in different modalities and the decision-making system

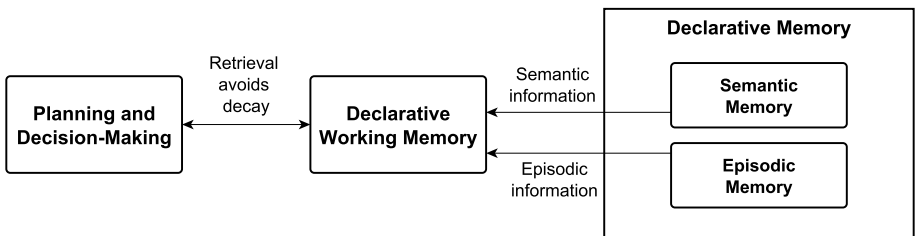


Fig. 7 Direct and indirect intersections

decision-making are the main consumers of such information held in DWM, they are required components to consider during our model's construction. See Figs. 6 and 7.

Type of interaction between functionalities and integration mechanisms specification The two possible types of interactions are direct and indirect. An interaction is **direct** when a function is considered necessary, by the research group, to continue the development of another function; it is **indirect** when a function is linked to other function, but it is not considered necessary for its development or for a given task. Next, we present the type of interaction for each function linked to working memory:

- **Planning and Decision-Making (PDM):** They have an indirect relation. They are the main receivers of working memory's output. These modules only retrieve and update information from working memory. However, information retrieval from a DWM buffer prevents its decay.
- **Declarative Memory (DEM):** It has a direct relation. It is working memory's main source of information and defines the data types to be used.

A cognitive function is considered related to another one through shared requirements. The **functional requirements (FR)** shared by each intersected function and DWM are:

- **Planning and Decision-Making:**
  - **FR-PDM-1:** It retrieves information from the different DWM buffers.
  - **FR-PDM-2:** It updates information from the different DWM buffers.
  - **FR-PDM-3:** It encodes information from the different DWM buffers.
- **Declarative Memory:**
  - **FR-DEM-1:** It encodes, retrieves, stores and forgets semantic information.
  - **FR-DEM-2:** It encodes, retrieves, stores and forgets episodic information.
  - **FR-DEM-3:** It encodes, retrieves, stores and forgets object information.
  - **FR-DEM-4:** All the storage-related brain structures require the mid-term level of memory.
  - **FR-DEM-5:** It defines the data types for object, semantic and episodic information.

Considering the intersection type and the shared requirements, we proceed to decide each function's integration mechanisms.

- **Planning and Decision-Making:** Due that they have an indirect relation with the main function, a black-box modeling strategy was chosen. Outputs to this system will be developed, as well as the retrieval, encoding and update functions. The complex processes this function performs are beyond this work's scope.
- **Declarative Memory:** We determined it can't be considered as a black box, since we don't know the data type handled by this memory and the working memory model requires this function. Thus, we proceed to the development of the declarative memory module.

Once we defined the integration mechanisms, we continued to develop the required sub-functions. Details about the *declarative memory development* can be found in Jaime (2016).

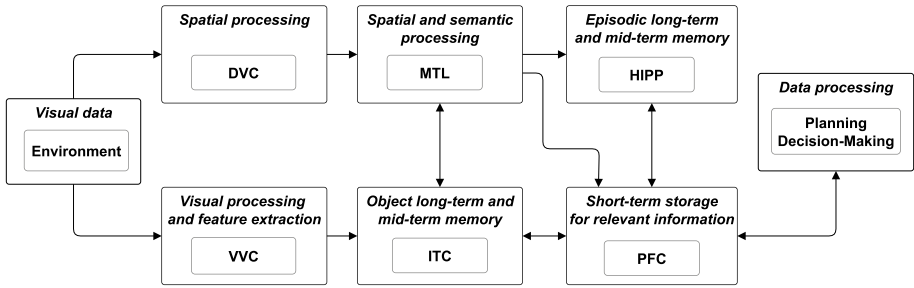
*Sub-functionality requirements statement* Through an extensive study of the neuroscientific evidence, we found the brain structures and their processes involved in the declarative

working memory function. Then, we defined the following functional requirements (FR) for each cortical area.

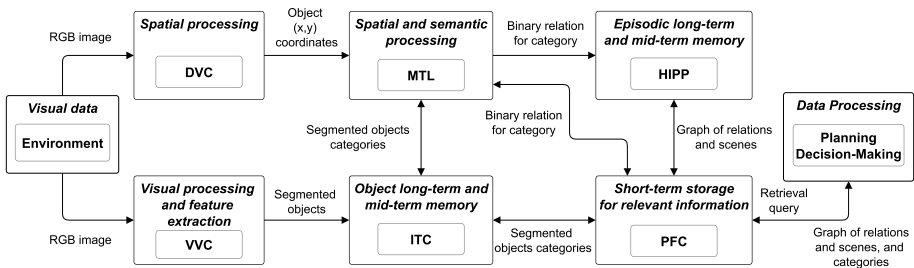
- Prefrontal Cortex (PFC).
  - **FR-PFC-1:** It encodes, stores, retrieves and forgets object, semantic and episodic information.
  - **FR-PFC-2:** It selects task-relevant information from mid-term and long-term memory.
  - **FR-PFC-3:** It manipulates stored items by sorting them based on contextual priorities.
  - **FR-PFC-4:** It sends top-down signals to maintain objects in memory.
  - **FR-PFC-5:** Stored information decays through time until it is deleted.
  - **FR-PFC-6:** Its storage is limited to 4 items.
  - **FR-PFC-7:** It updates object, episodic and semantic information.
- Inferior Temporal Cortex (ITC).
  - **FR-ITC-1:** It encodes, stores, retrieves and forgets object information.
  - **FR-ITC-2:** It retrieves a subset of long-term information stored in ITC and makes it faster to retrieve.
  - **FR-ITC-3:** Its available information subset decays through time until it is deleted.
  - **FR-ITC-4:** Its storage is limited to 8 items.
- Medial Temporal Lobe (MTL).
  - **FR-MTL-1:** It encodes, stores, retrieves and forgets semantic information.
  - **FR-MTL-2:** It retrieves a subset of long-term information stored in MTL and makes it faster to retrieve.
  - **FR-MTL-3:** Its available information subset decays through time until it is deleted.
  - **FR-MTL-4:** Its storage is limited to 8 items.
  - **FR-MTL-5:** It transforms spatial information.
- Hippocampus (HIPP).
  - **FR-HIPP-1:** It encodes, stores, retrieves and forgets episodic information.
  - **FR-HIPP-2:** It retrieves a subset of long-term information stored in HIPP and makes it faster to retrieve.
  - **FR-HIPP-3:** Its available information subset decays through time until it is deleted.
  - **FR-HIPP-4:** Its storage is limited to 8 items.
- Dorsal Visual Cortex (DVC).
  - **FR-DVC-1:** It extracts spatial information.
- Ventral Visual Cortex (VVC).
  - **FR-VVC-1:** It extracts visual features.

**Model design** In this stage, we present different architectonic views (conceptual, logical and execution) required to help in the proposal's validation and verification processes.

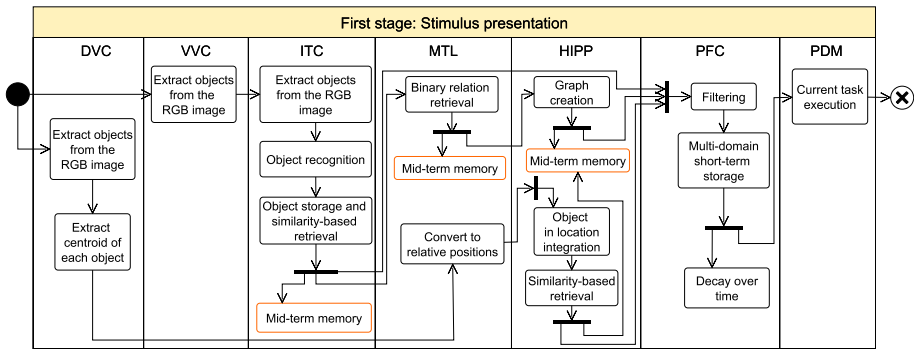
The **conceptual view** identifies the system's high-level components and their connections. Then, a diagram is constructed considering the neuroscientific evidence and the determined requirements. The brain structures, related functions and black boxes represent the components of the diagram (see Fig. 8).



**Fig. 8** The *connection diagram* is built based on the specified requirements. It shows the identified brain structures' inputs and outputs



**Fig. 9** The *collaboration diagram* adds the type of information transferred between areas

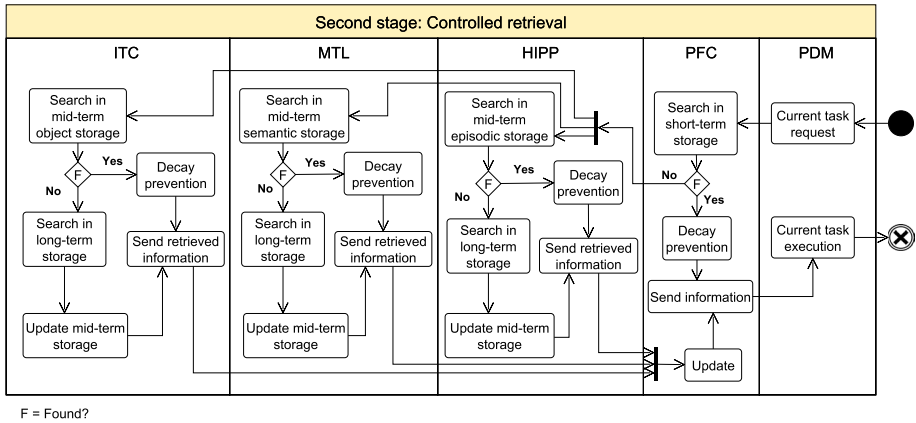


**Fig. 10** Before a stimulus reaches short-term storage on declarative working memory, it automatically triggers several processes through the system

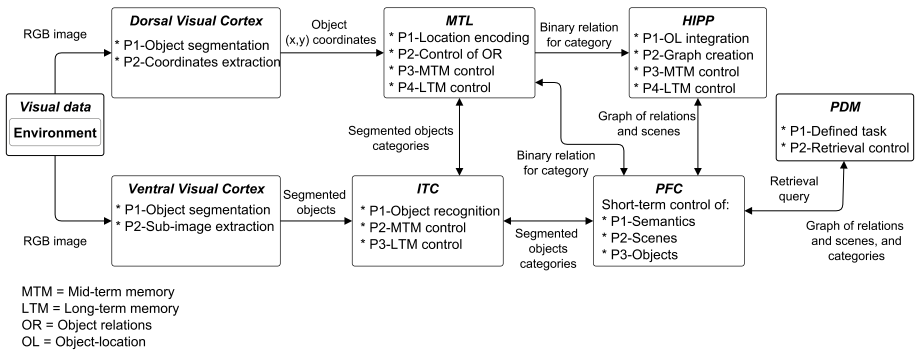
The **logical view** adds more detail to the description of interactions between the system's components. It sets the information flow, the processes and the type of information transferred among areas (see Fig. 9). In this case, we identified two main flows: when a stimulus is present in working memory (PFC) and when it must be searched in memory (see Figs. 10 and 11). The data types used are the same for both cases.

The **execution view** provides an approach to the system's physical structure. It maps components and processes to the system nodes [abstractions of the brain structures (Jaime et al. 2015)]. The *process diagram* extends the collaboration diagram by adding the





**Fig. 11** If certain information is not available in short-term storage, working memory requires to update its content with data from mid-term or long-term storage



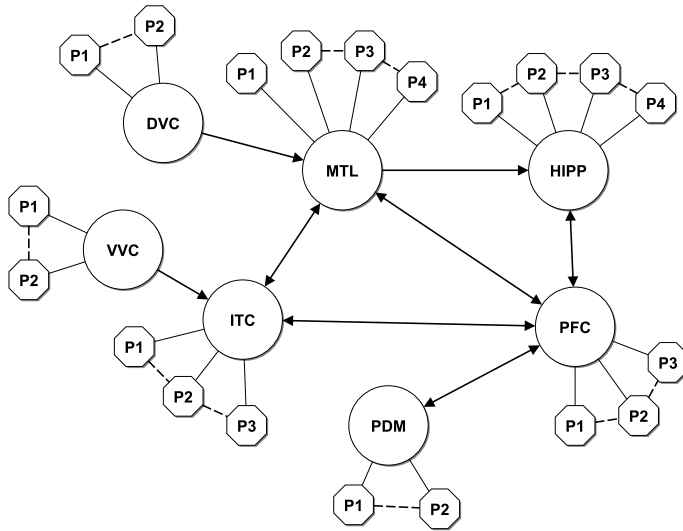
**Fig. 12** Processes carried out by each component of the model

processes each system node performs (Fig. 12). The *nucleus diagram* is useful to make an implementation in the middleware proposed by Jaime et al. (2015), and shows the Big Nodes, Small Nodes and their interactions (Fig. 13).

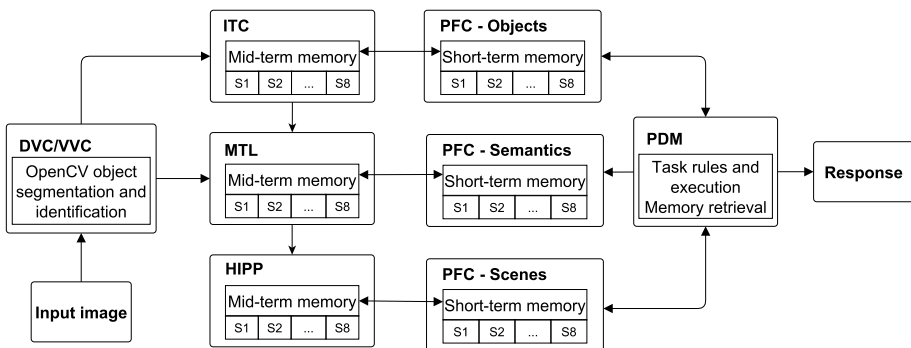
**Implementation** To validate the proposed model, a software implementation was developed based on the execution view diagrams, using the middleware mentioned earlier. In this case, each brain structure such as ITC, HIPPP and PFC, is represented as a Big Node and is responsible of communicating with other Big Nodes; while Small Nodes represent the processes carried out by each area and they allow distributed behavior, as proposed in (Jaime et al. 2015).

Due that the middleware was built on Java, all the processes were developed using this language. Also, some additional libraries like OpenCV were required to implement the model’s visual processing stages. Figures 15 and 16 show screenshots of the software.

Figure 14 presents a simplified view of the implementation flow. The core structure is a priority queue in both short and mid-term memory. The short-term one can store a maximum of 4 items, while what we propose as mid-term memory is used to store those elements that couldn’t be stored in short-term memory. The implementation flow for the presentation stage is:



**Fig. 13** Nucleus diagram used for the current implementation. Circles represent Big Nodes, while octagons are Small Nodes



**Fig. 14** Implementation's general flow

- An RGB image is presented to the system and it is sent to DVC and VVC.
- DVC receives the image and OpenCV performs an object segmentation to extract the centroids. It then sends the extracted objects to MTL.
- VVC receives the image and OpenCV performs an object segmentation. Then, it identifies each extracted object's class, which is finally sent to PFC and MTL.
- MTL encodes the locations and sends them to HIPP. It also relays the objects received from ITC to HIPP.
- HIPP integrates the locations from MTL with the identified objects from ITC. It then sends the integrated scene to PFC.
- PFC stores the received information in different buffers. However, it can only store 4 items per modality. If the buffer is full, it removes the oldest item and sends it to mid-term memory of the received area.

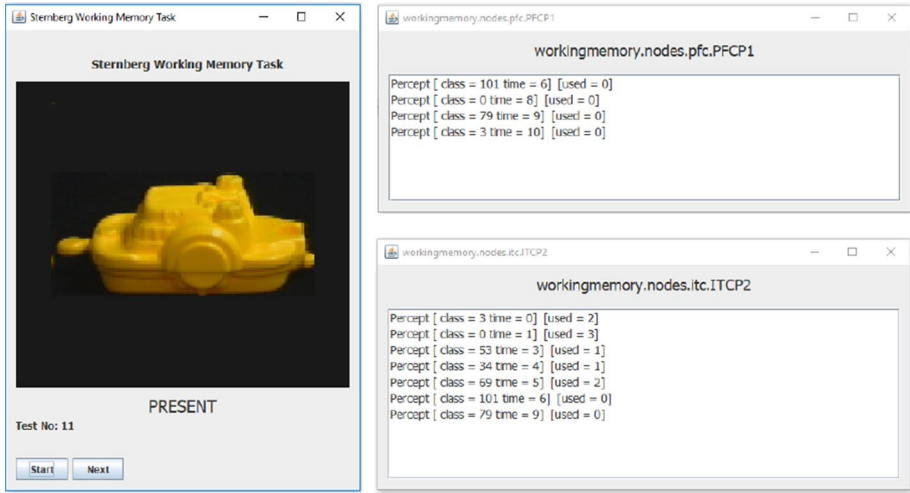
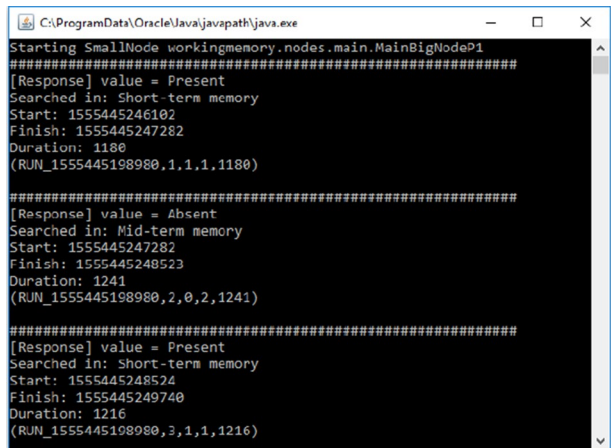


Fig. 15 Software implementation using the middleware for cognitive architectures

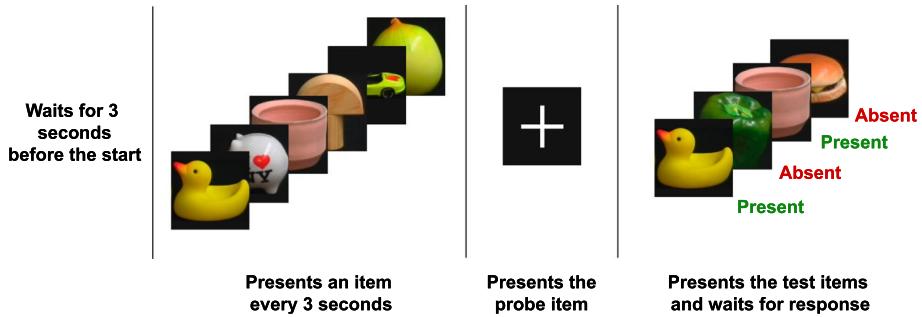
Fig. 16 Example of system's output after the presentation of the test items



- PFC sends each received item to Planning and Decision-Making.

The controlled retrieval flow proposed is:

- PDM defines the task rules such as the searched item or object class, the probe item class or the current execution mode.
- PDM receives an item. If the item is the probe item class, then, it starts search mode.
- PFC receives an item. If PDM is in search mode and the item class is not a probe item, then, it searches the class in short-term memory. If the class exists, it responds *Present*. However, if the class does not exist, it searches in mid-term memory. If the class exists, it responds *Present*; otherwise, *Absent*.



**Fig. 17** Sternberg working memory task used in this case study

**Table 3** Response time (RT) obtained for every memory access condition after 100 executions

Response	Storage buffer	Avg. RT (ms)	Min RT	Max RT
Present	Short-term memory	539	482	592
Present	Mid-term memory	588	525	622
Absent	Both memories	540	505	619

*Case study composition* The main goal of this methodology's stage is illustrating how to compose an appropriate case study. For simplicity, we chose a basic working memory task. However, it is worth noting that the proposed model is not limited to this task, it can be used to perform other experiments but that is beyond this article's scope.

The case study used to validate the model is based on a *Sternberg working memory task* (Sternberg 1966). This task consists on presenting to the system a list of items to memorize, followed by a maintenance period, during which such list must be retained in memory. Then, this maintenance period terminates by the onset of a probe item, which indicates that the subject must be ready for the test cues. When a test item is presented, the subject must respond whether the item was in the previous list or not. Figure 17 shows the experiment description.

In this experiment, each item is a 128\*128px RGB image and our model's implementation represents a person. The trial starts with the presentation of each one of the list's 6 elements every 3 seconds. Then, a special item (cross-shaped image) is presented and the system waits for 4 seconds. Finally, 4 test items are presented to the system and it responds if the item is *present* or *absent* in the list.

The variable of interest is the speed with which the system decides whether the item is a member of the set of items held in memory, by responding as quickly as possible.

*Simulation testing and analysis of rejection* Our results are consistent with the evidence found in human experiments, which states that the response time increases linearly with the size of the memory set (Sternberg 1966). In this case, we obtained a shorter response time when the item is stored in the short-term storage buffer. However, when the item was present in the mid-term memory storage buffer or it was absent, it took more time to evoke an answer. Table 3 shows the obtained response times for every condition.

Although our results differ to some extent with those in human subjects, it can be argued that this difference in response time is mainly due to the additional processing performed by the computer vision algorithms and network communication.

We believe that our model can be improved by considering additional variables like interaction with other systems such as attention, emotions and motivation, and by increasing the visual processing algorithms' performance.

We found no testing problems during development, however, in the case of unexpected results, an analysis of rejection must be performed to identify what originated them. Depending on the causes, it could imply stepping back to add or remove a requirement (sub-functionality requirements statement), to include evidence of other systems or cortical areas (exhaustive research of sub-functionality), or even to redefine the whole proposal (meta-architecture of functionality definition), as shown in Fig. 2.

*Application* The applications of this system are limited due to the fact it represents just a small part of working memory. However, it can be used to reproduce other tasks or experiments associated with this specific sub-function.

*Exploration* We can stop the development of the cognitive architecture, or we can extend it by considering the next scenarios taken from the methodology. We can proceed with the development of a new working memory sub-function from the proposed taxonomy. Or extending the declarative working memory sub-function by developing a related sub-function (see case study 2). Or, finally, extending it with an existing model (see case study 3).

## 7.2 Case study 2: Visual object recognition

In this study case, it is presented a synthesized development of a cognitive architecture for a visual object recognition system. The main goal of this model is to create a visual representation of objects present in an environment. The resulting model uses the replication of a neuroscientific experiment as a validation test. More specifically, the experiment is that of feature reduction method (Kobatake and Tanaka 1994), which shows how object representations in primates use critical features. Research on this matter and other object recognition functionalities may require more exploration out of this scope.

*General meta-architecture definition* The meta-architecture designed here has the same definition as the one described in Sect. 7.1.

*Functionality meta-architecture definition* In this architecture we focus on **perception**. That is the conscious awareness of our environment or our bodies that arise from activity in sensory pathways (Mason 2012). In this case, the perception will use **visual modality**. Since this functionality has broader sub-functions to work with, in this scope, we limit the model to those involved in visual object recognition.

The objective in this architecture is to model the processes that integrate the main visual ventral stream of the brain (Kravitz et al. 2013), in particular those of high-level processing, and the interactions with other cognitive functions. As said before, we limit this model to high-level visual object recognition, functionalities that interact with these processes are modeled as black boxes.

*Functionality research and conceptualization* To design the architecture of visual object recognition, it is essential to define the perception on which it will be based. Therefore, we analyzed the central visual processing of the brain from neuroscientific evidence.

Before visual processing takes place, visual sensing takes raw data from the environment, then projects this data to the thalamus, and finally, it takes this preprocessed information into visual cortices in the neocortex (Zeki 1978), where perception takes place (Mason 2012). From this point, the brain has two main visual streams: the dorsal "Where/How" and ventral "What" pathways, which process visual space and guided action and recognizes an object from the scene, respectively (Goodale and Milner 1992; Tanaka et al.

1991). The ventral stream takes information in every visual cortex progressively, being selective from essential to complex features, then this information is projected to the inferior temporal cortex (ITC), which reacts to even more complex visual features. Neurons in ITC have unique properties; some are:

- Neuronal activation clusters represent related groups of object complex features (Tsunoda et al. 2001; Lehky and Tanaka 2016; Rolls et al. 2003), meaning objects has some form of feature representation;
- Neural activity is modulated by experience and training (Kravitz et al. 2013; Gilbert and Li 2013; Baker et al. 2002), meaning it has a learning function;
- Some neurons are selective to transformations such as retinotopic position of object parts (Kravitz et al. 2010; Yamane et al. 2006; DiCarlo 2003), meaning that, although object recognition is position invariant, there is some form of retinotopic representation; and
- It receives top-down projections for filtering and priming from expectancy (Gilbert and Li 2013; de Lange et al. 2018; Oliva and Torralba 2007).

We found a few models for object recognition that take some of these properties into account, and are represented in Table 4. In Table 5 are also shown related cognitive architectures.

*Sub-functionalities determination* With the information that is described above in *functionality research and conceptualization*, we determined a taxonomy of visual perception. Next, it is listed the sub-functionalities from this cognitive functionality.

- **Object recognition** This concerns processes that contribute to single object categorization, identification and features representation in memory, composing the visual ventral stream in neuroscience. This is divide in low and **high-level object recognition**, and it may be category oriented, like body parts, tools, faces, etc.
- **Scene recognition** This sub-functionality has processes like *object recognition*, but it focuses in scene features (those which surround the single object recognized).
- **Spatial recognition** It follows the visual dorsal stream, which processes spatial egocentric input from agent visual perspective. These generate maps of the spatial environment and, then, translate them to allocentric maps that serve to other processes, such as navigation and episodic memory.

*Exhaustive sub-functionality research and detection of overlapping sub-functionalities* From the sub-functionalities defined, object recognition was chosen, in particular **high-level general object recognition (GOR)**.

This sub-functionality generates high-level of visual object features based on those extracted from earlier low-level processing, these are mainly basic feature extraction, and color and texture segmentation, then the segmented proto-object is classified. Classification takes place in a hierarchical manner, processing general features first and local features later. When it classifies a familiar input, it activates features related to the input, at contrary, when a novel input is classified, it generates and stores this new features. When classification succeeds, the final identified object class is given to declarative memory, including semantic and episodic, which creates various data associations types.

In the whole recognition process, feedback and feedforward flow of information takes place in the same areas from attentional function, giving various filters to all processes.

**Table 4** In this table we list related models for visual object recognition and their main features: VisNet (Rolls 2012), Fei-Fei & Perona extended model (Fei-fei and Perona 2005), Ullman model (Ullman 2007), HMAX (Serre et al. 2007), Edelman model (Edelman and Intrator 2000), Olshausen, Anderson & Van Essen model (Olshausen et al. 1995), CNNs (Güçlü and van Gerven 2015), AHRM (Petrov et al. 2005), RBC (Biederman 1987)

Visual object recognition models					
Model	Approach	Representation form	Retinotopy representation	Top-down influences	Learning capabilities
VisNet	Hebbian learning	Artificial neurons	Yes, external	Yes	Yes
Fei-Fei & Perona	Codebook & codewords	Bag of codewords	No	No	Yes
Ullman	Fragment-based hierarchy	Image fragments	No	No	Yes, through training stage
HMAX	MAX operator	Artificial neurons	No	No	Yes, through training stage
Edelman	Coarse coding of shape fragments	Image fragments	Yes	No	Yes, through training stage
Olshausen, Anderson & Van Essen	Dynamic Routing Circuit	Artificial Neurons	Yes	Yes	No
CNN	Deep Neural Network	Artificial Neurons	No	No	Yes, through training stage
AHRM	Reweighting Hebbian Learning	Artificial Neurons	No	Yes	Yes
RBC	Recognition by Components	Geones	No	No	No

**Table 5** In this table are shown related cognitive architectures and their features for visual object recognition: SAL (Lebiere et al. 2008), LIDA (Faghihi and Franklin 2012), and DUAL (Kokinov 1994)

Cognitive Architectures						
Architecture	Approach	Representation Form	Retinotopy Representation	Top-Down Influences	Learning Capabilities	
SAL	ACT-R reasoning with Leabra learning	Artificial neurons and symbolic rules	No	Yes	Yes, through training stage	
LIDA	Distributed structures	Nodes	Yes	Yes	Yes	
DUAL	Emergence from agent coalitions	Agents	No	Yes	Yes	

For ACT-R and Leabra see (Ritter et al. 2019) and (O'Reilly et al. 2012), respectively



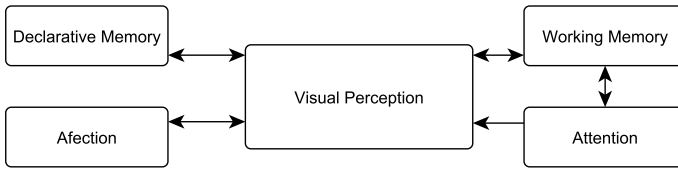


Fig. 18 Cognitive functionalities that interact with perception

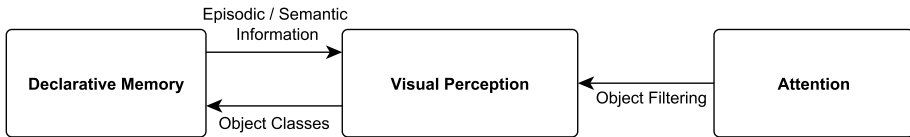


Fig. 19 Diagram representing overlapping functionalities and their interaction

From the cognitive functionalities represented in Fig. 18, we determined *attention* and *declarative memory* to be **overlapping processes** (see Fig. 19).

Type of interaction between functionalities and integration mechanisms specification Concerning the sub-functionality of **GOR**, there some other sub-functionalities that has **direct** interaction:

- **V1/V2 low-level features extraction (FE)** This sub-function has processes the main low-level data from image input of the agent. This information is the basic features set that **GOR** will work with.
- **V4 color/texture segmentation (SEG)** After main basic low-level features are extracted, segmentation takes place. At one of these segments, **GOR** will take place.
- **Attention (ATT)** This functionality is required after the segmentation process, and it will filter segments that are near to the fovea.

There are also few **indirect** interactions:

- **Declarative Memory (DEM)** This functionality receives the **GOR** output and stores it.
- **Working Memory (WM)** It also takes **GOR** output, stores it in short-term memory and manipulates this for other functionalities of the agent.
- **Affection (AFF)** Takes **GOR** data and creates emotional associations.

*Sub-functionality requirements statement* We next list the functional requirements (**FR**) of the brain structures for high-level visual object recognition based on neuroscientific evidence.

- Primary/Secondary Visual Cortex (V1/V2).
  - **FR-V1/V2-1:** It receives an image as input.
  - **FR-V1/V2-2:** Extracts basic features from input image and creates feature maps.
  - **FR-V1/V2-3:** Feature maps are sent to V4.
- Extrastriate Visual Area 4 (V4).
  - **FR-V4-1:** It receives feature maps as input.

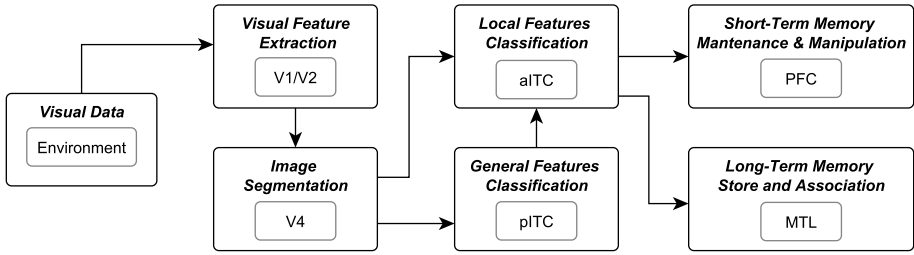


Fig. 20 Connection diagram of the conceptual model

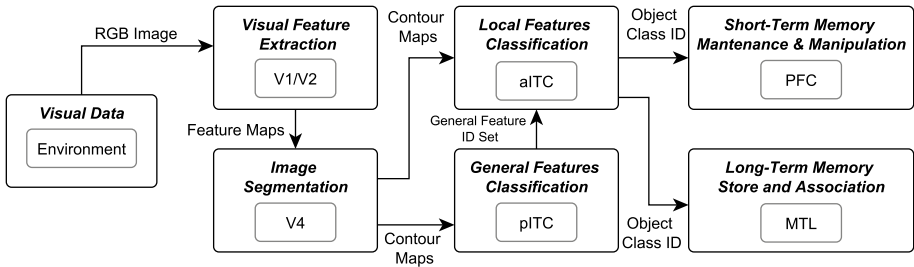


Fig. 21 Information flow for the logical model

- **FR-V4-2:** Creates contour maps from feature maps.
- **FR-V4-3:** Contour maps are available for pITC.
- Posterior Inferior Temporal Cortex (pITC).
  - **FR-pITC-1:** It receives a subset of contour map, these compose a proto-object.
  - **FR-pITC-2:** It compares the general features of the contour with those in visual perceptual memory and, then, takes feedback.
  - **FR-pITC-3:** It assigns an ID to general features based on comparison results.
  - **FR-pITC-4:** General feature IDs are available to aITC.
- Anterior Inferior Temporal Cortex (aITC).
  - **FR-aITC-1:** It receives a subset of contour map, these compose a proto-object. It also receives general feature IDs from pITC.
  - **FR-aITC-2:** It compares the local features of the contour with those in visual perceptual memory and, then, takes feedback.
  - **FR-aITC-3:** It assigns an ID to local features based on comparison results.
  - **FR-aITC-4:** Creates a class ID from general and local feature IDs that is available to MTL and PFC.
- Medial Temporal Lobe (MTL).
  - **FR-MTL-1:** It receives a class ID from aITC.
- Prefrontal Cortex (PFC)
  - **FR-PFC-1:** It receives a class ID from aITC.

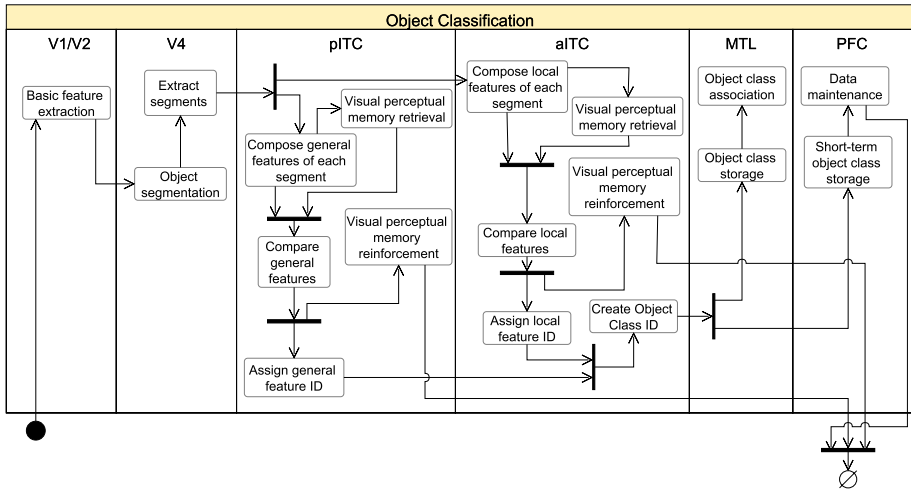


Fig. 22 Activity diagram for the logical model

Model design: For the **conceptual view** is a connection diagram presented in Fig. 20. For the **logical view** we have in Fig. 21 a representation of the information flow, and in Fig. 22 is an activity diagram representing a general object recognition task. Finally, the **execution view** is presented with a process diagram in Fig. 23 and with the nucleus diagram in Fig. 24.

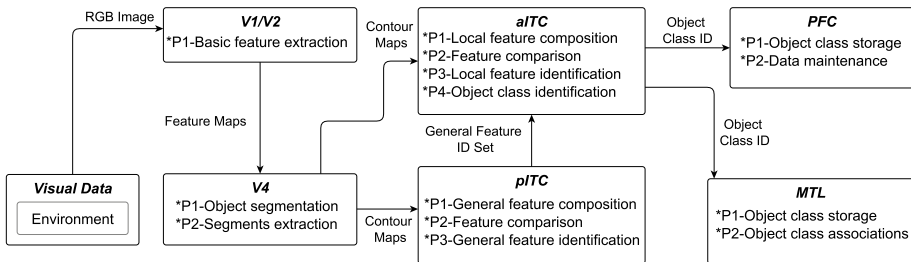


Fig. 23 Diagram representing processes for the execution model

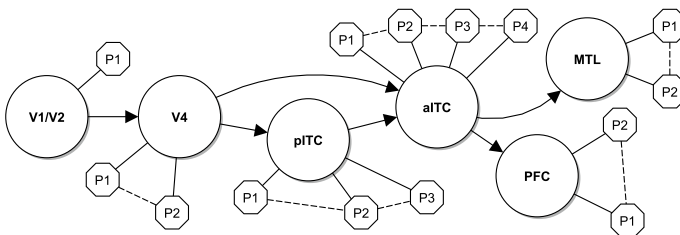


Fig. 24 Nucleus diagram for the execution model

*Implementation* Based on the nucleus diagram (Fig. 24), we implemented the model using a middleware just like the previous study case. This was accomplished using Java programming language and OpenCV libraries for visual processing. Following, it is listed the main elements of the processes carried on the system.

- Initially, an RGB image of is presented as input of V1/V2.
- V1/V2 processes the RGB image converting it to a gray scale feature map.
- V4 takes the feature map and detects contours. This contours map are sent to pITC and aITC.
- pITC takes contours and compose complex general features using a blur over the contours map. Then it compares with complex general features in visual perceptual memory. Finally, it gives an ID for the features and send this IDs to aITC.
- aITC takes contours and compose complex local features using invariant points over the contours map. Then it compares with complex local features in visual perceptual memory. Later, it gives an ID for the features and, combining with general features IDs, creates a class ID and sends it to MTL and PFC.
- MTL and PFC are programmed as black box that store object class IDs.

*Case study composition* The case study used as validation is based on the object reduction method (Kobatake and Tanaka 1994). This experiment consists in to see how a neuron responds to a complex visual object. Then, we simplify it by eliminating step by step a part of the features present in the object, creating simplified samples, while observing the same neuron initially activated. The sample that activates this neuron the most is then passed to another iteration of simplification. At last, the simplest feature (most simplified object) that maximally activates the neuron is considered a critical feature that describes that neuron. In this case, the neurons are represented as complex feature IDs, and, when comparing with features in memory, the best match is a neuron maximally activated. In Fig. 25 is a representation of this case study.

*Simulation testing and analysis of rejection* In our results, we observed that the reduction method could be applied in the proposed software, giving evidence that the system behavior is similar to those in primates and even in humans. An input image was processed in the system, and the output features were then analyzed. The next iteration output was compared with the last iteration and so on.

Although the simulation testing passed, this is still a reduced approach and it need for some other extensions like affection and attention to be a more realistic and complete behavior.

*Application* Taking in consideration that this is a reduced implementation of a perceptual system that simulates primate behavior, its application will be limited. Despite this, the model can be used as part of experimental architectures that require real-time perceptual learning functions.

*Exploration* Further this point of the development and research in this cognitive functionality, we explore the options in which the project may take course. Some are adding specific learning sub-functionalities, modeling other functionality interaction like attention or affection, research about other complex behavior that take part on visual perceptual processes, or integrating with other developed architecture like declarative working memory (see subsect. 7.3).

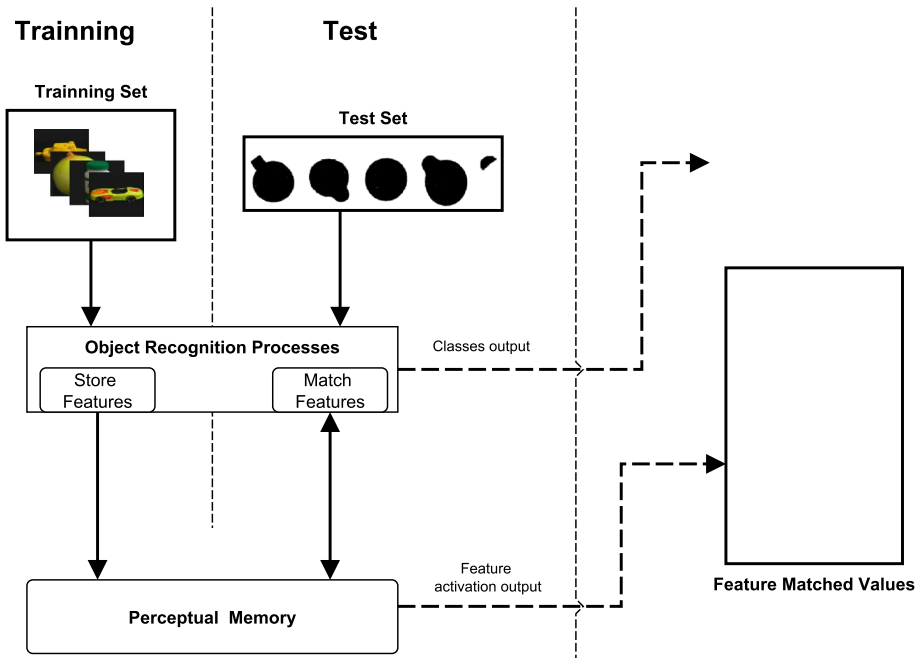


Fig. 25 A representation of the current case study

### 7.3 Case study 3: Integration of cognitive functions

As we stated in step 6.3, a developed cognitive architecture can be extended by determining which other functions are related to the current model. In this case study, we proceed to extend the cognitive architecture of declarative working memory developed in case study 1. Therefore, some of the first methodological steps might remain the same as for the previous cases.

Specifically, the steps *General meta-architecture*, *Functionality meta-architecture definition*, *Functionality research and conceptualization*, and *Sub-functionalities determination* will remain the same as for case study 1. This is due to that we are going to extend a model from an already researched sub-function of working memory, and we are not going to start a new cognitive function. Therefore, our methodological development starts from the end of the case study 1.

*Application and Exploration* Through an exploration of the developed sub-function, we determined that we can extend the model by continuing the bio-inspired development of certain overlapped functions which were modeled but implemented in a non-bio inspired manner.

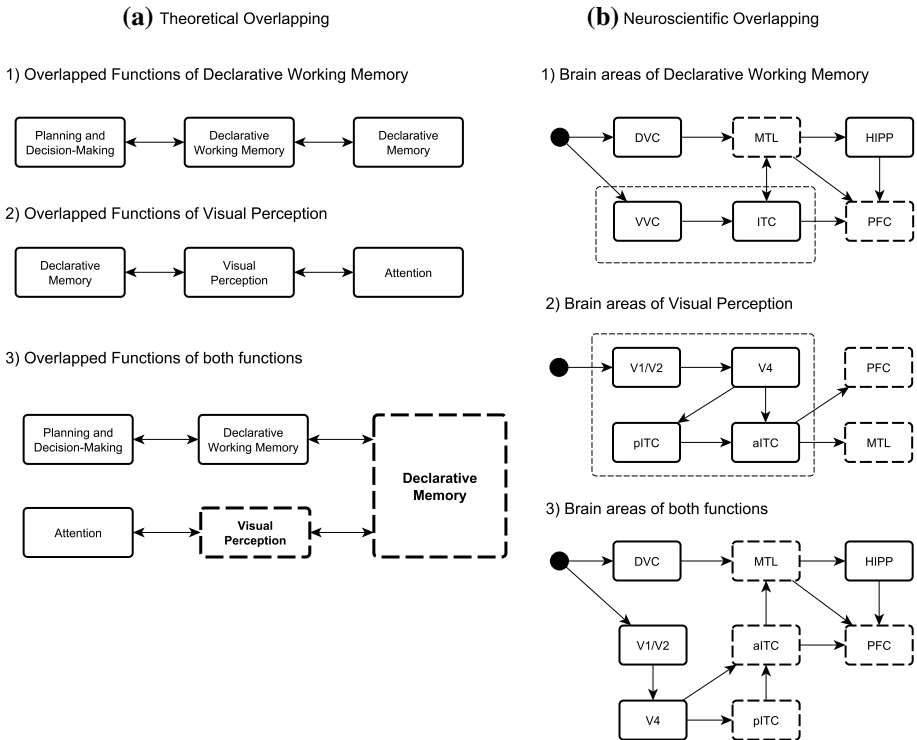
*Extend sub-functionality* In this case, we decided to extend the DWM system by adding an appropriate object recognition system, not just a template matching that works as a black-box. Thus, the addition of a Perceptual function is required. As this function is not part of the taxonomy defined in case study 1 we must specify the next steps to proceed.

*Other models exist in the same context?* To avoid full function development from scratch we review if there exist other models that fulfill the criteria set in the

meta-architecture. At this point, we can see the case study from two perspectives. The first one implies that either we want to or we have to make the Perceptual function from zero; thus, we can see the case study 2 as the development of a non-existing function before we continue with the extension of the memory model. The second perspective implies that a 3rd party developed the Perceptual function of case study 2 and we can take its model as a basis. Either case, we proceed to detect the overlapping functionalities between the DWM model and the Perceptual one.

*Exhaustive sub-functionality research and detection of overlapping functionalities* Comprehensive research for both perception and DWM has been covered in its respective case study. In this step, we focus on presenting, which are the common processes and common brain structures obtained from both pieces of research.

Figure 26 shows a join of the overlapping functions taken from both functions. In this stage, we first determine in a theoretical the overlaps between both functions model. Then, once we have found the theoretical connection points, we use them to see the intersections in a brain structure level. From both models, declarative memory is the main input



**Fig. 26** In part (A), (1) and (2) show the related functions in a theoretical level of each function taken from their study case. Diagram (3) joins both diagrams, and the dashed rounded rectangles represent the critical points of connection. Part (B) presents the same connection points but a cerebral cortex level, (1) and (2) show the connection diagram of each function. In (1), the dashed rounded rectangle around the areas represents the areas covered by the visual perception function. In contrast, the dashed rounded rectangles represent the brain areas that could be affected by a replacement. The dashed rounded rectangle around the areas in (2), represent the set of regions that can replace those in (1). Finally, (3) shows the integration of both models, and the dashed rounded rectangles are the areas that require to be analyzed for compatibility

to declarative working memory, and it is the main output from visual perception. Therefore, we conclude that the critical points of connection are declarative memory and visual perception.

After exhaustive research of the brain structures associated with the intersection of both connection points, we obtain the first sketch of the connections diagram, and we specify which are the common areas between both models that require some adaptation in the requirements.

*Type of interaction between functionalities* Because it is an integration of two functions, once the connection points are identified, they must be treated as direct interaction type and the requirements must be adjusted to be compatible.

*Sub-functionality requirements statement and integration mechanisms specification* Taking into account the established requirements for both the perception and memory function, additional requirements were established. Specifically, we define the requirements for the critical areas obtained from the *detection of overlapping functionalities*: PFC, MTL, aITC, and pITC. The rest of the requirements of the areas remains the same.

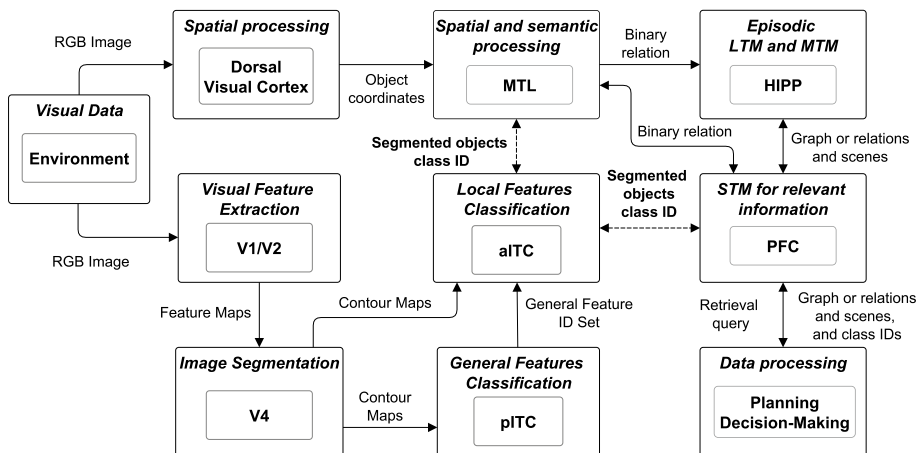
- Prefrontal Cortex (PFC).
  - **FR-PFC-1:** It encodes, stores, retrieves and forgets object, semantic and episodic information.
  - **FR-PFC-2:** It selects task-relevant information from mid-term and long-term memory.
  - **FR-PFC-3:** It manipulates stored items by sorting them based on contextual priorities.
  - **FR-PFC-4:** It sends top-down signals to maintain objects in memory.
  - **FR-PFC-5:** Stored information decays through time until it is deleted.
  - **FR-PFC-6:** Its storage is limited to 4 items.
  - **FR-PFC-7:** It updates object, episodic and semantic information.
  - **FR-PFC-8:** It receives a class ID from aITC.
- Medial Temporal Lobe (MTL).
  - **FR-MTL-1:** It encodes, stores, retrieves and forgets semantic information.
  - **FR-MTL-2:** It retrieves a subset of long-term information stored in MTL and makes it faster to retrieve.
  - **FR-MTL-3:** Its available information subset decays through time until it is deleted.
  - **FR-MTL-4:** Its storage is limited to a defined amount of items.
  - **FR-MTL-5:** It transforms spatial information.
  - **FR-MTL-6:** It receives a class ID from aITC.
- Posterior Inferior Temporal Cortex (pITC).
  - **FR-pITC-1:** It receives a subset of contour map, these compose a proto-object.
  - **FR-pITC-2:** It compares the general features of the contour with those in visual perceptual memory and, then, takes feedback.
  - **FR-pITC-3:** It assigns an ID to general features based on comparison results.
  - **FR-pITC-4:** General feature IDs are available to aITC.
  - **FR-pITC-5:** It encodes, stores, retrieves and forgets object information.
  - **FR-pITC-6:** It retrieves a subset of long-term information stored in ITC and makes it faster to retrieve.
  - **FR-pITC-7:** Its available information subset decays through time until it is deleted.

- **FR-pITC-8:** Its storage is limited to a defined amount of items.
- Anterior Inferior Temporal Cortex (aITC).
  - **FR-aITC-1:** It receives a subset of contour map, these compose a proto-object. It also receives general feature IDs from pITC.
  - **FR-aITC-2:** It compares the local features of the contour with those in visual perceptual memory and, then, takes feedback.
  - **FR-aITC-3:** It assigns an ID to local features based on comparison results.
  - **FR-aITC-4:** Creates a class ID from general and local feature IDs that is available to MTL and PFC.
  - **FR-aITC-5:** It encodes, stores, retrieves and forgets object information.
  - **FR-aITC-6:** It retrieves a subset of long-term information stored in ITC and makes it faster to retrieve.
  - **FR-aITC-7:** Its available information subset decays through time until it is deleted.
  - **FR-aITC-8:** Its storage is limited to a defined amount of items.

*Model design* In this stage, we present different architectonic views (conceptual, logical and execution) required to help in the proposal’s validation and verification processes.

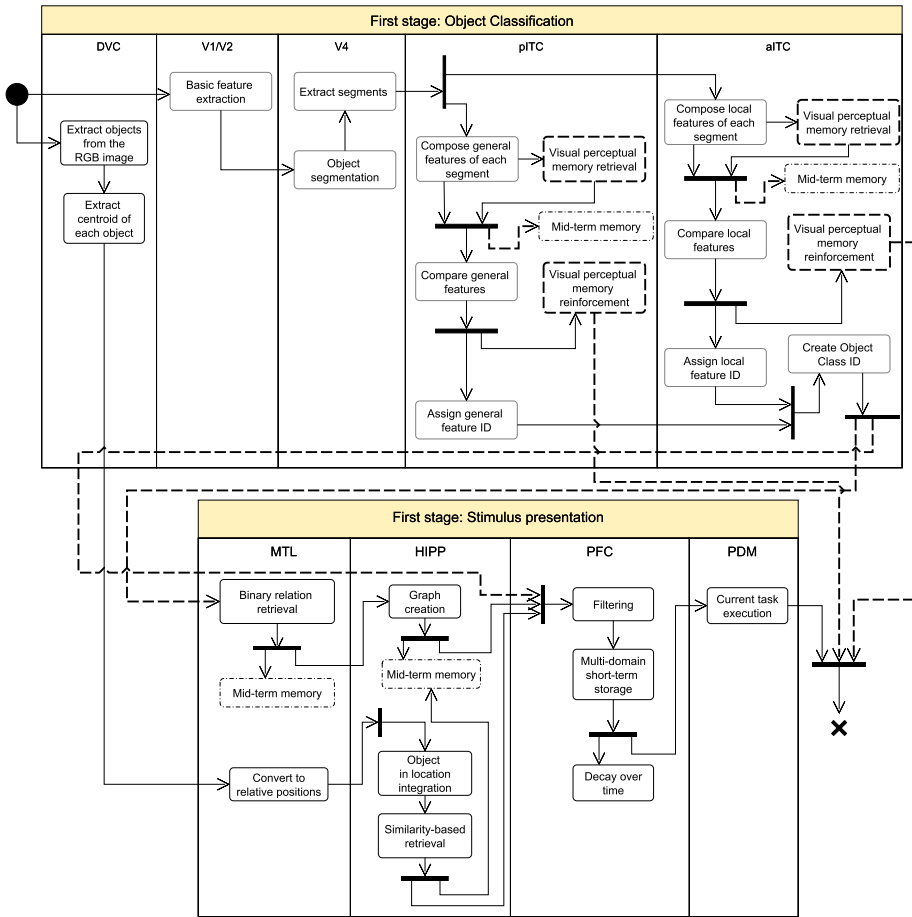
The **conceptual view** obtained from the integration of the requirements and neuro-scientific evidence taken from both functions is shown in Fig. 27.

The **logical view** covers the details about the data transferred between the brain areas, also shown in Fig. 27. The dotted lines with bold labels represent the data types that were adapted to be compatibles between the two functions. Figure 28 and 29 present the flow of the stages obtained from the integration of the functions. In this case, the main stages correspond to those defined by DWM but with visual perception processes embedded. Like the previous diagram, dotted lines, and dotted rounded rectangles represent intersection points between the functions that were modified to be compatibles.



**Fig. 27** This diagram sums both the connection and collaboration diagram. It shows the brain areas obtained from both functions, their connections, and the data type transferred. The type **Segmented objects class ID** send by aITC provides more details about the objects, thus, MTL and PFC were modified to be able to process this type of data





**Fig. 28** The object classification stage of visual perception connects directly to the stimulus presentation of declarative working memory. The dotted lines show the point of connection between the output of processes taken from one model to another. The dotted rounded rectangles correspond to processes that were initially proposed in the DWM model but were added to the visual perception model for compatibility

The diagrams of the **execution view** required for the deployment are the *process diagram* (Fig. 30) and the *nucleus diagram* (Fig. 31).

*Implementation* Most of the implementation of both systems remained the same. Although the data structures used in both systems were compatible, the software framework wasn't. Therefore, two bridge nodes were added in the DWM system as a connection point to the perceptual systems (Fig. 32). These nodes open a socket where the data coming from perception are transformed into a representation that the DWM can process.

*Case study composition* To validate the integration of both functions, we choose to use a hybrid case study taken from the proposed for each function. Hence, we split the case into two stages. The first stage will be focused on visual perception, and its primary goal is that the system learns the images that will represent the items in the next stage. Once the system has learned the items, the next stage is to validate the declarative working memory by using the same *Sternberg working memory task*. The first stage is the case study defined to

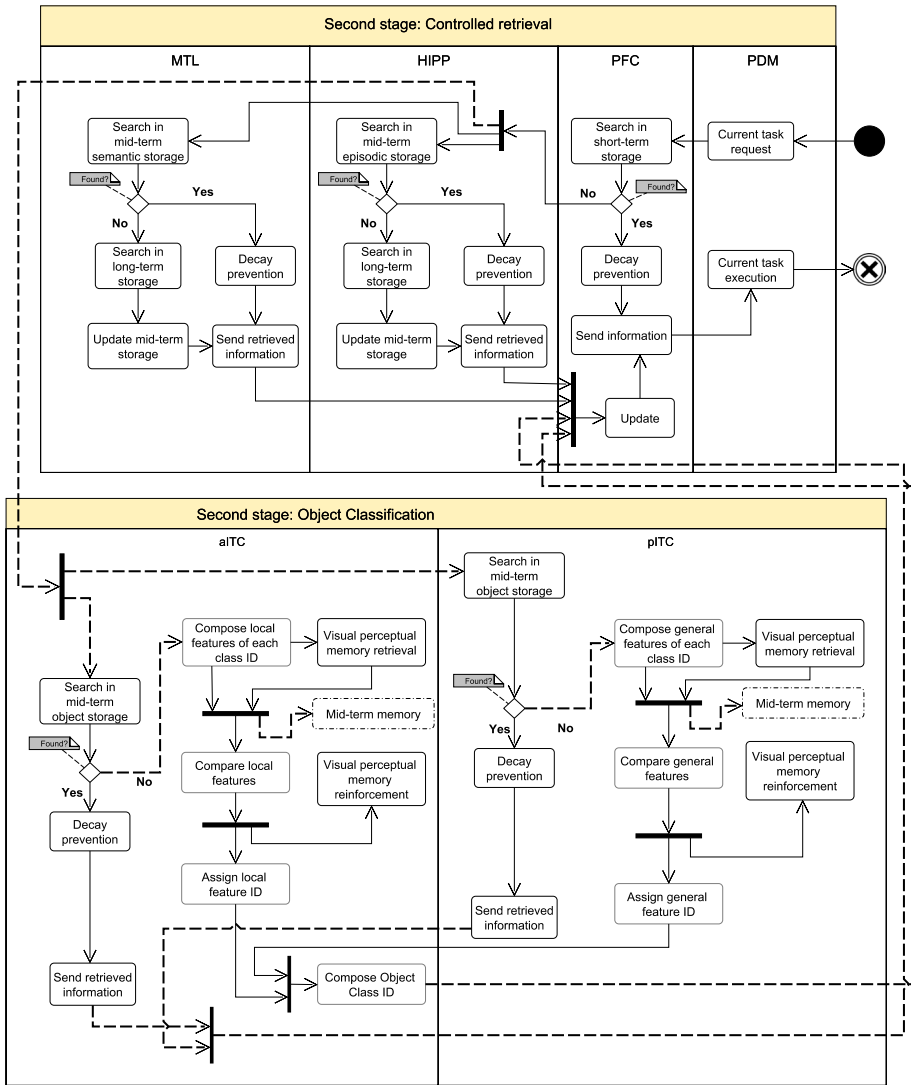


Fig. 29 Direct and indirect intersections

evaluate the visual perception and the second stage, the case study used to validate DWM but with both systems integrated.

*Simulation testing and analysis of rejection* The results obtained for the experiment were divided by stage.

*Stage 1* Firstly, like in case study 2, the perceptual system went into a training sub-stage, this gave us a total of 553 visual features from which object classes were built. With this training result, we were able to begin the first stage of the experiment. As we were passing images to the perceptual system, we observed in its output, a set of classes that were based on those visual features from the trained perceptual memory,

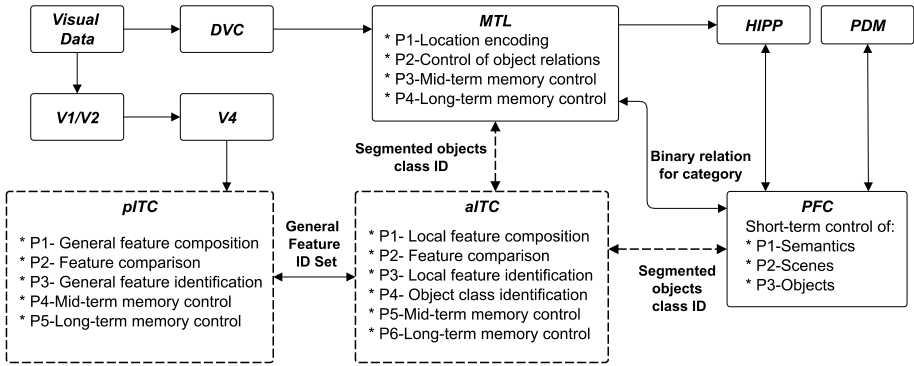


Fig. 30 Processes of the integrated models

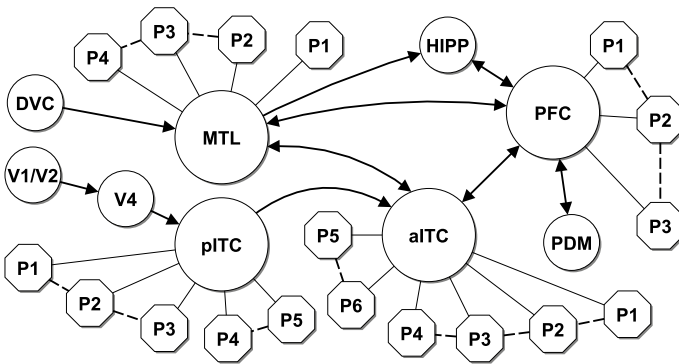


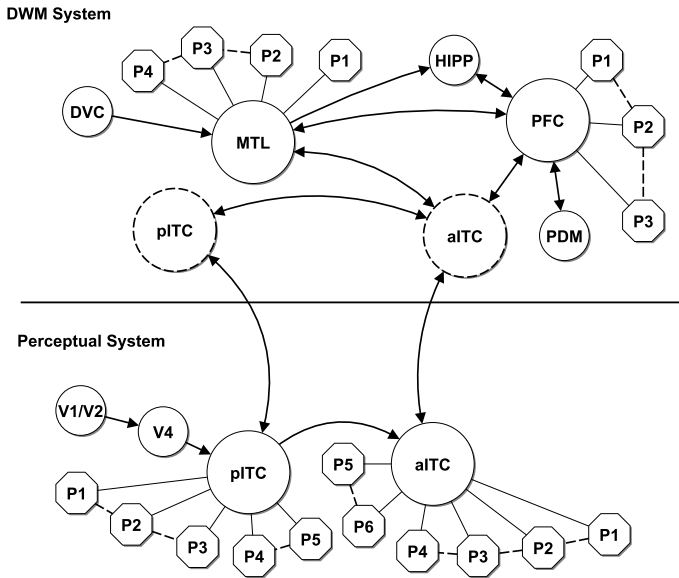
Fig. 31 Processes of the integrated models in a nucleus diagram

and that these classes were consistent with respect to repeatability (i.e. if an image was given for second occasion during this stage). Therefore, with this correct behavior of the perceptual system, we determined to continue with the second stage of this case study.

*Stage 2* As in case study 1, the results for the second stage remained consistent with the evidence found in human experiments. We also obtained a shorter response time when the item is stored in the short-term storage buffer, and when the item was present in the mid-term memory storage buffer or it was absent, it took more time to evoke an answer. Table 6 shows the obtained response times for every condition. As we can see, due to the extension of processes taken from both functions and the addition of bridge nodes, the response time took about twice the time compared with the case study 1. However, the response patterns remained the same for each type of memory.

*Application* Considering the limitations of both systems, the applications of this integration are limited but can be used to reproduce other kinds of experiments associated with both functions.

*Exploration* Despite we can go further with the modeling and improvement of our architecture. We decide to stop the development at this point.



**Fig. 32** Extended nucleus diagram: shows the nodes that act as bridges between the systems. The dashed circles pITC and aITC, open a socket to receive data from one system, transform the data, and transfer it to the other system

**Table 6** Response time (RT) obtained for every memory access condition after 100 executions

Response	Storage buffer	Avg. RT (ms)	Min RT (ms)	Max RT (ms)
Present	Short-term memory	1128	561	1434
Present	Mid-term memory	1146	672	1433
Absent	Both memories	1187	1077	1451

## 8 Discussion

The interactions between anatomic, cognitive and behavioral knowledge domains for mind modeling, imply approaching it with non-conventional representation forms to make more accurate approximations to artificial comprehensive human behavior.

These concurrent multi-level approaches overcome all possible solutions projected over a single domain in cognitive sciences. Examples are consciousness as epiphenomena of biological activity (Weber and Weekes 2009) or the emergence of behaviors caused by the interaction of cognitive functions.

Consequently, we argue that the study of the interactions, in multiple levels, between modular cognitive functions, as well as their associated coordination and observation systems that support the analysis of complex phenomena, is essential for consistently developing cognitive architectures. Conciliating knowledge from several research fields is not a trivial endeavor; it is still difficult to build human-like artificial agents, without considering the high complexity present at different disciplinary fields (computer

science, neurosciences, psychology, philosophy, social sciences). For this reason, this non-trivial endeavor is considered in our proposal.

Another relevant discussion topic is human-like behavior. Computational proposals in the field must address a principle of adaptability, which permits animal species to survive in an environment. This discourse allows us to point out a problematic binomial *body/mind*, where “the adaptive relation with the environment is replaced with a computational treatment of the information through algorithms” (Tête 1994).

Furthermore, this notion of adaptability directly affects the design of cognitive architectures, since it implies limitations when modeling computational agents with human-like behavior. It is also necessary to consider the biological and anatomical biases, which might contribute to the general modeling of an agent’s behavior, to make it more “similar” to that of the human being.

Moreover, it is possible to determine the expected forms of intelligence and behavior according to the general paradigms of cognition, and to propose the methodological foundations that, independently of the research approaches used and the specific orientation of each architecture, can be considered as determinant elements to the domain in which they develop or to the problem they solve. On the other hand, simplification for operational purposes of the complexity of the cognitive functions that the architectures have chosen to develop, and consequently sacrifice precision in the capture of the global phenomenon by the fulfillment of a specific cognitive task.

We presented a methodology to construct cognitive architectures. Our proposal can be used to develop both complete architectures and individual cognitive functions. We proposed a distinction between the construction and the operational levels of cognitive architectures, by an explicit division during methodological meta-design steps. This allows us to distinguish specific work scenarios for each of them, according to the *context, applicability, desired abilities, objectives, scope and constraints* considered.

The case study selected, as a proof of concept for our proposed methodology, focused on developing two single high granularity cognitive functions and its interactions and interdependencies, also the linking of these two functions in the context of obtaining a broader cognitive architecture. The modularity managed by our proposed methodology leads us to a design position, where we observe two integration levels: the emerging cognition, which arises by interactions between the participating subsystems, and individual functional behavior for task solving.

The previous case study followed the proposed methodology in order to achieve a cognitive architecture, which is highly tied to psychological and neuroscientific theories of working memory. This result was in part due that each methodological step demands an ever-increasing study of the function. Therefore, once we have a better understanding, we will be able to create a model and define the suitable tasks to evaluate every sub-functionality of the architecture.

Taken together, *meta-architecture of functionalities, general research and conceptualization, sub-functionalities classification and exhaustive research of sub-functionality*, constitute a fundamental part of the methodology. They all contribute to define the problem’s scope and the theories, both psychological and neuroscientific, required to understand the function, which are useful guides to construct a model and to determine if the function could or could not be divided into sub-functions.

In case study 1, for instance, after the proper research of the function, a Sternberg working memory task (Sternberg 1966) was used to validate the developed model. The results obtained from the implementation were matched against data from experiments

in human subjects, showing promising results. These results support our claim that by following the methodology, we can achieve, to some extent, human-like behavior.

However, it is worth noting that the validation task may change for different processes executed by the architecture. In our view, these results constitute an excellent initial step toward a robust methodology. Nonetheless, further developments are required to identify possible limitations, which will help to refine this proposal.

In our proposal, we have intentionally attenuated the role that learning and memory units have taken traditionally in the context of cognitive architectures, in order to clarify the progressive biological and behavioral interrelations between cognitive functions for the systematization of integrity assumptions, and highlight their importance.

Thus, a hybrid standpoint is required to embrace our proposed methodology, which is intended to manage progressive integration of a whole conceptual mapping of brain structures, to possess a comprehensive cognitive cycle, and to exhibit behavior as a result of the modules' interaction. This is evidenced in case study 3, and it contrasts with the *bottom-up* research stance usually associated with bioinspired cognitive architectures.

Our position described in this paper slightly differs from the considerations presented in Sect. 2, which is carried out in a very similar way and exposes concepts akin to the ones discussed here. Nevertheless, we consider two differentiating key-points in our proposal:

1. *The systematization of the assumptions of integrity:* Our proposal requires information exchanges from low-level to higher-level cognitive function models; the former's subsequent behavior is influenced by feedback received from the latter, and vice versa. Its concurrent nature requires a dynamic that transcends the unidirectional levels of scientific research. Each reported cognitive function –whatever its level– has a model with functionalities and sub-functions, as seen in the presented case study.
2. *The observation point for the architecture's construction:* This systematization is hybrid, because it uses not only reports of cortical mapping at brain level, but also the description of what has been done in the field and how it is documented both biologically and behaviorally; despite the heterogeneity of the multiple knowledge fields referred.

During this research's development, the construction of abstraction levels have been oriented to the phenomenological possibilities of cognitive functions, considering the limitations of what abstractions at computational level can achieve.

In Sect. 5, we presented a table that describes some key features of the most relevant cognitive architectures. Overall, it shows that most of these CAs are for general use and strive to develop an execution framework, based on certain building guidelines taken from cognitive sciences; however, during our review, we seldom found explicit descriptions about the origin of such guidelines (from the chosen theoretic paradigms), and how following them influences development until the corresponding architecture's implementation and validation. Thus, this lack of well-defined specific guidelines led us to propose our methodology for the construction of cognitive architectures. It is worthwhile noting that due to this same absence of explicitness and the fact that every CA uses its own and specific design directives, we do not consider it appropriate to compare a whole methodological process against other non-structured design processes or a list of guidelines.

What is expected from this proposal is, in the short-term, to cover the essential models that are part of a minimum cognitive cluster. This kind of clusters is crucial to verify

the effectiveness of the interaction between functions, and to integrate further modules into increasingly complex cognitive models. This also implies verifying the integrity and consistency of functionalities and sub-functions.

In the medium-term, we anticipate this will highlight the importance of adopting multi-disciplinary approaches, specially the relationship between the bioinspired cognitive architectures' structural nature, determined by the brain's anatomy, and their functional aspects, given by the interpretation of biologic phenomena to computational implementations.

In the long-term, we hope this contributes to conform an integral approach for studying the great complexity of the human mind. The conceptual framework and structural mapping of cerebral functions, considering brain cartography, which are emphasized throughout our proposed methodology, seek to position the science of the mind as an integral set of behavioral, biological and cognitive knowledge. This set has an important role in the relationship between the meta-architectonic constructions that define how a cognitive compound works, and the interaction of its lower level sub-functionalities, which progressively contribute to the formation of cognitive clusters in higher-level areas.

**Acknowledgements** This work is supported in part by Consejo Nacional de Ciencia y Tecnología (CONACYT) via the Post-Doctoral and Doctoral scholarship program (CVU-490378/SNI-76678) (CVU-585726), and SEP-CINVESTAV FIDSC2018/93.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Adams S, Arel I, Bach J, Coop R, Furlan R, Goertzel B, Hall JS, Samsonovich A, Scheutz M, Schlesinger M et al (2012) Mapping the landscape of human-level artificial general intelligence. *AI Magazine* 33(1):25–42
- Anderson J (2013) *The architecture of cognition*. Taylor & Francis. <https://books.google.com.mx/books?id=zL0eAgAAQBAJ>
- Anderson J, Bellezza F (1993) *Rules of the Mind*. L. Erlbaum Associates. <https://books.google.com.mx/books?id=xYvLraxYTZoC>
- Anderson JR, Bothell D, Byrne MD, Douglass S, Lebiere C, Qin Y (2004a) An integrated theory of the mind. *Psychol Rev* 111(4):1036
- Anderson JR, Bothell D, Byrne MD, Douglass S, Lebiere C, Qin Y (2004b) An integrated theory of the mind. *Psychol Rev* 111(4):1036–1060. <https://doi.org/10.1037/0033-295X.111.4.1036>
- Arbib MA (2003) Towards a neurally-inspired computer architecture. *Nat Comput* 2(1):1–46
- Arbib MA (2007) How neural computing can still be unconventional after all these years. In: *Proceedings of the 6th international conference on Unconventional Computation*. Springer, pp 1–1
- Avila-Contreras C, Medina O, Jaime K, Ramos F (2014) An agent cognitive model for visual attention and response to novelty. In: *Agent and multi-agent systems: Technologies and applications*, Springer, pp 27–36
- Baars BJ, Franklin S (2009) Consciousness is computational: the lida model of global workspace theory. *Int J Mach Conscious* 1(1):23–32
- Baddeley A (2000) The episodic buffer: a new component of working memory? *Trends Cognit Sci* 4(11):417–423. [https://doi.org/10.1016/S1364-6613\(00\)01538-2](https://doi.org/10.1016/S1364-6613(00)01538-2)
- Baddeley AD, Hitch G (1974) Working memory, psychology of learning and motivation, vol 8. Academic Press, Cambridge, pp 47–89. [https://doi.org/10.1016/S0079-7421\(08\)60452-1](https://doi.org/10.1016/S0079-7421(08)60452-1)

- Baddeley AD, Allen RJ, Hitch GJ (2011) Binding in visual working memory: the role of the episodic buffer. *Neuropsychologia* 49(6):1393–1400
- Baker CI, Behrmann M, Olson CR (2002) Impact of learning on representation of parts and wholes in monkey inferotemporal cortex. *Nat Neurosci* 5(11):1210–1216. <https://doi.org/10.1038/nn960>
- Balci O (1994) Validation, verification, and testing techniques throughout the life cycle of a simulation study. *Ann Oper Res* 53(1):121–173
- Bekolay T, Bergstra J, Hunsberger E, DeWolf T, Stewart TC, Rasmussen D, Choo X, Voelker A, Eliasmith C (2014) Nengo: a python tool for building large-scale functional brain models. *Front Neuroinform* 7:48
- Biederman I (1987) Recognition-by-components: a theory of human image understanding. *Psychol Rev* 94(2):115–147. <https://doi.org/10.1037/0033-295X.96.1.2>
- Bledowski C, Kaiser J, Rahm B (2010) Basic operations in working memory: contributions from functional imaging studies. *Behav Brain Res* 214(2):172–179
- Bothell D (2017) Act-r 7 reference manual. Available at [act-r psy cmu edu/wordpress/wpcontent/themes/ACT-R/actr7/reference-manual.pdf](http://act-r.psy.cmu.edu/wordpress/wpcontent/themes/ACT-R/actr7/reference-manual.pdf), Accessed February
- Bressler SL, Tognoli E (2006) Operational principles of neurocognitive networks. *Int J Psychophysiol* 60(2):139–148
- Brodmann K (2007) Brodmann's: localisation in the cerebral cortex. Springer, Berlin
- Brooks FP (1962) Architectural philosophy. Planning a computer system, pp 5–16
- Butz MV (2016) Toward a unified sub-symbolic computational theory of cognition. *Front Psychol* 7:925–925. <https://doi.org/10.3389/fpsyg.2016.00925>
- Cangelosi A, Schlesinger M (2018) From babies to robots: the contribution of developmental robotics to developmental psychology. *Child Develop Perspect* 12(3):183–188
- Cooper R, Fox J, Farrington J, Shallice T (1996) A systematic methodology for cognitive modelling. *Artif Intell* 85(1–2):3–44
- Cowan N (1999) An embedded-processes model of working memory. *Models Working Memory Mech Active Maintenance Executive Control* 20:506
- Desikan RS, Ségonne F, Fischl B, Quinn BT, Dickerson BC, Blacker D, Buckner RL, Dale AM, Maguire RP, Hyman BT et al (2006) An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage* 31(3):968–980
- D'Esposito M (2007) From cognitive to neural models of working memory. *Philos Trans R Soc Lond B Biol Sci* 362(1481):761–772
- D'Esposito M, Postle BR (2015) The cognitive neuroscience of working memory. *Annual Rev Psychol* 66:115–142
- de Lange FP, Heilbron M, Kok P (2018) How do expectations shape perception? *Trends Cognit Sci* 22(9):764–779. <https://doi.org/10.1016/j.tics.2018.06.002>
- DiCarlo JJ (2003) Anterior inferotemporal neurons of monkeys engaged in object recognition can be highly sensitive to object retinal position. *J Neurophysiol* 89(6):3264–3278. <https://doi.org/10.1152/jn.00358.2002>
- DiFilippo NM, Jouaneh MK (2018) Using the soar cognitive architecture to remove screws from different laptop models. *IEEE Trans Autom Sci Eng* 16(2):767–780
- Dobrica L, Niemela E (2002) A survey on software architecture analysis methods. *IEEE Trans Softw Eng* 28(7):638–653
- Duch W, Oentaryo RJ, Pasquier M (2008) Cognitive architectures: Where do we go from here? *AGI* 171:122–136
- Edelman S, Intrator N (2000) (Coarse coding of shape fragments) plus (retinotopy) approximate to representation of structure. *Spatial Vision* 13(2–3):255–264. <https://doi.org/10.1163/156856800741072>
- Eliasmith C (2007) How to build a brain: From function to implementation. *Synthese* 159(3):373–388
- Eliasmith C (2013) How to build a brain: a neural architecture for biological cognition. Oxford University Press, Oxford
- Eliasmith C, Anderson CH (2004) Neural engineering: computation, representation, and dynamics in neurobiological systems. MIT press, Cambridge
- Eliasmith C, Stewart TC, Choo X, Bekolay T, DeWolf T, Tang Y, Rasmussen D (2012) A large-scale model of the functioning brain. *Science* 338(6111):1202–1205
- Faghghi U, Franklin S (2012) The LIDA model as a foundational architecture for AGI. In: Wang P, Goertzel B (eds) Theoretical foundations of artificial general intelligence. Atlantis Press, Paris, pp 103–121. [https://doi.org/10.2991/978-94-91216-62-6\\_7](https://doi.org/10.2991/978-94-91216-62-6_7)
- Fei-fei L, Perona P (2005) A Bayesian hierarchical model for learning natural scene categories. *Computer vision and pattern recognition. IEEE, San Diego*, pp 524–531



- Fittner M, Brandstatter C (2018) How human inspired learning enhances the behavior of autonomous agents. *JCP* 13(2):154–160
- Fodor J (1983) *Modularity of mind*. MIT Press, Cambridge
- Franklin S (2018) the Cognitive Computing Research Group The mind according to lida—brief account. <http://ccrg.cs.memphis.edu/tutorial/mind-according-to-lida.html>
- Franklin S, FG Patterson J (2006) The lida architecture: Adding new modes of learning to an intelligent, autonomous, software agent. *Software Agent IDPT-2006 Proceedings (Integrated Design and Process Technology)*: Society for Design and Process Science (2006)
- Franklin S, Madl T, D’mello S, Snaider J (2013) Lida: a systems-level architecture for cognition, emotion, and learning. *IEEE Trans Automon Mental Develop* 6(1):19–41
- Franklin S, Madl T, Strain S, Faghihi U, Dong D, Kugele S, Snaider J, Agrawal P, Chen S (2016) A lida cognitive model tutorial. *Biol Inspired Cognit Architect* 16:105–130
- Garrido JM (2009) *Object oriented simulation*. Springer, Berlin
- Gilbert CD, Li W (2013) Top-down influences on visual processing. *Nat Rev Neurosci* 14:350–363. <https://doi.org/10.1038/nrn3476>
- Gobet F, Lane P (2010/06) The chrest architecture of cognition: The role of perception in general intelligence. In: 3d Conference on Artificial General Intelligence (AGI-2010), Atlantis Press. <https://doi.org/10.2991/agi.2010.20>
- Goldman-Rakic PS (2011) Circuitry of primate prefrontal cortex and regulation of behavior by representational memory. *Comprehens Physiol*, pp 373–417
- González-Casillas A, Parra L, Martín L, Avila-Contreras C, Ramirez-Pedraza R, Vargas N, del Valle-Padilla JL, Ramos F (2018) Towards a model of visual recognition based on neurosciences. *Proc Comput Sci* 145:214–231
- Goodale MA, Milner AD (1992) Separate visual pathways for perception and action. *Trends Neurosci* 15(1):20–25. [https://doi.org/10.1016/0166-2236\(92\)90344-8](https://doi.org/10.1016/0166-2236(92)90344-8)
- Güçlü U, van Gerven MAJ (2015) Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *J Neurosci Official J Soc Neurosci* 35(27):10005–14. <https://doi.org/10.1523/JNEUROSCI.5023-14.2015>
- Holland O, Diamond A, Marques HG, Mitra B, Devereux D (2013) Real and apparent biological inspiration in cognitive architectures. *Biol Inspired Cognit Archit* 3:105–116
- Jaime K (2016) *Declarative memory model based on neuroscience for virtual creatures*. PhD thesis, CINVESTAV del IPN Unidad Guadalajara, Zapopan, Jalisco
- Jaime K, Torres G, Ramos F, Garcia-Aguilar G (2014) A cognitive architecture for visual memory identification. *Int J Softw Sci Comput Intell* 6(2):63–77
- Jaime K, Cervantes A, Medina O, Ramos F, Rosales JH, Torres G, Ramos M (2015) A middleware for integrating cognitive architectures. In: *International conference on brain informatics and health*, Springer, pp 435–444
- Johnson TR (1997) Control in act-r and soar. In: *Proceedings of the nineteenth annual conference of the cognitive science society*, pp 343–348
- Juvina I, Larue O, Hough A (2018) Modeling valuation and core affect in a cognitive architecture: the impact of valence and arousal on memory and decision-making. *Cognit Syst Res* 48:4–24
- Kieras DE, Meyer DE (1997) An overview of the epic architecture for cognition and performance with application to human-computer interaction. *Hum-Comput Interact* 12(4):391–438
- Kingdom R (2008) *A review of cognitive architectures*. Tech rep, ISO Project Report
- Kitamura T (2001) *What should be computed to understand and model brain function?: from robotics, soft computing, biology and neuroscience to cognitive philosophy*, vol 3. World Scientific, Singapore
- Kobatake E, Tanaka K (1994) Neuronal selectivities to complex object features in the ventral visual pathway of the macaque cerebral cortex. *J Neurophysiol* 71(3):856–67. <https://doi.org/10.1152/jn.1994.71.3.856>
- Kokinov B (1994) The DUAL cognitive architecture: a hybrid multi-agent approach. In: *Proceedings of the 11th European conference on artificial intelligence*
- Kotseruba I, Gonzalez OJA, Tsotsos JK (2016) A review of 40 years of cognitive architecture research: focus on perception, attention, learning and applications. *CoRR abs/161008602*
- Kravitz DJ, Kriegeskorte N, Baker CI (2010) High-level visual object representations are constrained by position. *Cerebral Cortex* 20(12):2916–2925. <https://doi.org/10.1093/cercor/bhq042>
- Kravitz DJ, Saleem KS, Baker CI, Ungerleider LG, Mishkin M (2013) The ventral visual pathway: an expanded neural framework for the processing of object quality. *Trends Cognit Sci* 17(1):26–49. <https://doi.org/10.1016/j.tics.2012.10.011>

- Krichmar JL (2012) Design principles for biologically inspired cognitive robotics. *Biol Inspired Cognit Architect* 1:73–81
- Kuhn TS (2012) *The structure of scientific revolutions*. University of Chicago press, Chicago
- Laird JE (2008) Extending the soar cognitive architecture. *Front Artif Intell Appl* 171:224
- Laird JE (2012a) *The soar cognitive architecture*. The MIT Press, Cambridge
- Laird JE (2012b) *The soar cognitive architecture*. The MIT Press, Cambridge
- Laird JE, Rosenbloom PS, Newell A (1986) Chunking in soar: the anatomy of a general learning mechanism. *Mach Learn* 1(1):11–46
- Laird JE, Newell A, Rosenbloom PS (1987) Soar: an architecture for general intelligence. *Artif Intell* 33(1):1–64
- Laird JE, Congdon CB, Assanie M, Derbinsky N, Xu J (2017a) *The soar user's manual version 9.6. 0*
- Laird JE, Lebiere C, Rosenbloom PS (2017b) A standard model of the mind: toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. *Ai Magazine* 38(4):13–26
- Lancaster JL, Woldorff MG, Parsons LM, Liotti M, Freitas CS, Rainey L, Kochunov PV, Nickerson D, Mikiten SA, Fox PT (2000) Automated talairach atlas labels for functional brain mapping. *Human Brain Map* 10(3):120–131
- Langley P, Laird JE, Rogers S (2009) Cognitive architectures: research issues and challenges. *Cognit Syst Res* 10(2):141–160
- Lebiere C, O'Reilly R, Jilk DJ, Taatgen Na, Anderson JR, O'Reilly R (2008) The SAL integrated cognitive architecture. *Biologically inspired cognitive architectures: papers from the AAAI 2008 fall symposium* (2014):98–104. <http://www.aaai.org/Papers/Symposia/Fall/2008/FS-08-04/FS08-04-027.pdf>
- Lefort M, Boniface Y, Girau B (2011) Coupling bcm and neural fields for the emergence of self-organization consensus. In: *From brains to systems*, Springer, pp 41–56
- Lehky SR, Tanaka K (2016) Neural representation for object recognition in inferotemporal cortex. *Current Opin Neurobiol* 37:23–35. <https://doi.org/10.1016/j.conb.2015.12.001>
- Levy R, Goldman-Rakic PS (2000) Segregation of working memory functions within the dorsolateral prefrontal cortex. In: *Executive control and the frontal lobe: Current issues*, Springer, pp 23–32
- Lieto A (2017) Representational limits in cognitive architectures. *Cognitive Robot Architectures*, pp 16
- Lieto A, Bhatt M, Oltramari A, Vernon D (2018) The role of cognitive architectures in general artificial intelligence. *Cognit Syst Res* 48:1–3. <https://doi.org/10.1016/j.cogsys.2017.08.003>
- Luke S, Cioffi-Revilla C, Panait L, Sullivan K (2004) Mason: A new multi-agent simulation toolkit. In: *Proceedings of the 2004 swarmfest workshop*, Michigan, USA, pp 316–327
- Maes P (1991) The agent network architecture (ana). *Acm Sigart Bull* 2(4):115–120
- Malan R, Bredemeyer D (2005) *The visual architecting process*. Architecture Resources for Enterprise Advantage, Bredemeyer Consulting
- Martin L, Torres G, L'opez-Tolsa G (2016) Bio-inspired cognitive model for sustained attention. In: *2016 IEEE 15th international conference on cognitive informatics and cognitive computing (ICCI\* CC)*, IEEE, pp 551–557
- Mason P (2012) *Medical neurobiology*, 2nd edn. Oxford University Press, Oxford
- Mauro J (2015) Constraints meet concurrency. *Constraints* 20(4):488–489. <https://doi.org/10.1007/s10601-015-9218-6>
- Metta G, Natale L, Nori F, Sandini G, Vernon D, Fadiga L, Von Hofsten C, Rosander K, Lopes M, Santos-Victor J et al (2010) The icub humanoid robot: an open-systems platform for research in cognitive development. *Neural Netw* 23(8–9):1125–1134
- Metta G, Sandini G, Vernon D, Natale L, Nori F (2019) icub: the open humanoid robot designed for learning and developing complex cognitive tasks. In: *IEEEERSJ International Conference on Intelligent Robots and Systems*, Citeseer
- Miller R (2008) *C# for Artists: The art, philosophy, and science of object-oriented programming*. Pulp Free Press
- Oberauer K (2002) Access to information in working memory: exploring the focus of attention. *J Exp Psychol Learn Memory Cognit* 28(3):411
- Oberauer K (2007) Activation, binding, and selective access. In: Osaka N, Logie RH, D'Esposito M (eds) *The cognitive neuroscience of working memory*, 1st edn. Oxford University Press, Oxford, pp 1310–1370
- Oliva A, Torralba A (2007) The role of context in object recognition. *Trends Cognit Sci* 11(12):520–527. <https://doi.org/10.1016/j.tics.2007.09.009>
- Olshausen BA, Anderson CH, Van Essen DC (1995) A multiscale dynamic routing circuit for forming size- and position-invariant object representations. *J Comput Neurosci* 2(1):45–62. <https://doi.org/10.1007/BF00962707>

- O'Reilly R, Hazy T, Herd S (2012) The leabra cognitive architecture: how to play 20 principles with nature and win! The Oxford Handbook of Cognitive Science pp 1–31, <http://psych.colorado.edu/~oreilly/papers/oreillyhazyherdip.pdf>
- Pachalska M, MacQueen BD, Brown JW (2012) Microgenetic theory: brain and mind in time. In: Encyclopedia of the history of psychological theories, Springer, pp 675–708
- Petrov AA, Doshier BA, Lu ZL (2005) The dynamics of perceptual learning: an incremental reweighting model. *Psychol Rev* 112(4):715–743. <https://doi.org/10.1037/0033-295X.112.4.715>
- Pfeifer R, Bongard J (2006) How the body shapes the way we think: a new view of intelligence. MIT press, Cambridge
- Ritter F, Young R (1994) Practical introduction to the soar cognitive architecture: tutorial report. *AISB Quart* 88:62
- Ritter FE, Kim JW, Oury JD (2017) Act-r: Frequently asked questions list. Electronically. <http://acs.ist.psu.edu/projects/act-r-faq/act-r-faq.html>
- Ritter FE, Tehrani F, Oury JD (2019) ACT-R: a cognitive architecture for modeling cognition. *Wiley Interdiscip Rev Cognit Sci* 10(3):1. <https://doi.org/10.1002/wcs.1488>
- Rolls ET (2012) Invariant visual object and face recognition: neural and computational bases, and a model VisNet. *Front Comput Neurosci* 6:70. <https://doi.org/10.3389/fncom.2012.00035>
- Rolls ET, Aggelopoulos NC, Fashan Z (2003) The receptive fields of inferior temporal cortex neurons in natural scenes. *J Neurosci* 23(1):339–348
- Schaat S (2016a) A case-driven methodology for the interdisciplinary development and examination of mental architectures. *Proc Comput Sci* 88:429–437
- Schaat S (2016b) Sima-c: a foundational mental architecture. *Proc Comput Sci* 88:371–378
- Schaat S, Wendt A, Jakubec M, Gelbard F, Herret L, Dietrich D (2014) ARS: an AGI agent architecture. In: International conference on artificial general intelligence. Springer, pp 155–164
- Schacter DL, Wagner AD (2013) Learning and memory. In: Kandel ER, Schwartz JH, Jessell TM, Siegelbaum SA, Hudspeth AJ (eds) Principles of neural science, 5th edn. Mc Graw Hill, New York, pp 1441–1460
- Schrodt F, Kneissler J, Ehrenfeld S, Butz MV (2017a) Mario becomes cognitive. *Topics Cognit Sci* 9(2):343–373
- Schrodt F, Röhm Y, Butz MV (2017b) An event-schematic, cooperative, cognitive architecture plays super mario. *Cognit Robot Architect* 10:10–15
- Serre T, Wolf L, Bileschi S, Riesenhuber M, Poggio T (2007) Robust object recognition with cortex-like mechanisms. *IEEE Trans Pattern Anal Mach Intell* 29(3):411–426. <https://doi.org/10.12816/0047609>
- Shapshak P (2018) Artificial intelligence and brain. *Bioinformation* 14(1):38
- Sharma S, Aubin S, Eliasmith C (2016) Large-scale cognitive model design using the nengo neural simulator. *Biol Inspired Cognit Architect* 17:86–100
- Snaider J, McCall R, Franklin S (2011) The lida framework as a general tool for agi. In: International conference on artificial general intelligence, Springer, pp 133–142
- Snaider J, McCall R, Strain S, Franklin S (2012) The lida tutorial
- Squire LR (2009) Memory and brain systems: 1969–2009. *J Neurosci* 29(41):12711–12716. <https://doi.org/10.1523/JNEUROSCI.3575-09.2009>
- Sternberg RJ (2011) Cognitive psychology. Cengage Learning, Boston
- Sternberg S (1966) High-speed scanning in human memory. *Science* 153(3736):652–654. <https://doi.org/10.1126/science.153.3736.652>
- Sun R (2004) Desiderata for cognitive architectures. *Philos Psychol* 17(3):341–373
- Sun R (2006) The clarion cognitive architecture: Extending cognitive modeling to social simulation. Cognition and multi-agent interaction. Cambridge University Press, New York, pp 79–99
- Sun R (2009) Motivational representations within a computational cognitive architecture. *Cognit Comput* 1(1):91–103
- Taatgen NA, Lebiere C, Anderson JR (2006) Modeling paradigms in act-r. From cognitive modeling to social simulation. *Cognit Multi-Agent Interact*, pp 29–52
- Tanaka K, Saito H, Fukada Y, Moriya M (1991) Coding visual images of objects in the inferotemporal cortex of the macaque monkey. *J Neurophysiol* 66(1):170–189. <https://doi.org/10.1152/jn.1991.66.1.170>
- Tête A (1994) Entre neurosciences & psychologie cognitive: une frontière en question. *Revue de Synthèse* 115(3–4):485–502
- Torres G, Jaime K, Ramos F, Garcia G (2011) Brain architecture for visual object identification. In: IEEE 10th international conference on cognitive informatics and cognitive computing (ICCI-CC'11), IEEE, pp 33–40

- Torres GA (2013) Metodología para el desarrollo de arquitecturas cognitivas basada en neurociencias. PhD thesis, Cinvestav unidad Guadalajara
- Tsunoda K, Yamane Y, Nishizaki M, Tanifuji M (2001) Complex objects are represented in macaque inferotemporal cortex by the combination of feature columns. *Nature Publ Group* 4(8):832–8. <https://doi.org/10.1038/90547>
- Ullman S (2007) Object recognition and segmentation by a fragment-based hierarchy. *Trends Cognit Sci* 11(2):58–64. <https://doi.org/10.1016/j.tics.2006.11.009>
- Varma S (2011) Criteria for the design and evaluation of cognitive architectures. *Cognit Sci* 35(7):1329–1351
- Vernon D (2017) Two ways (not) to design a cognitive architecture. *Cognit Robot Architect*, vol 42
- Vernon D (2019) The architect's dilemmas. In: *Cognitive architectures*, Springer, pp 59–70
- Vernon D, Metta G, Sandini G (2007) A survey of artificial cognitive systems: implications for the autonomous development of mental capabilities in computational agents. *IEEE Trans Evolut Comput* 11(2):151–180
- Vernon D, Von Hofsten C, Fadiga L (2011) A roadmap for cognitive development in humanoid robots, vol 11. Springer, Berlin
- Vernon D, von Hofsten C, Fadiga L (2016) Desiderata for developmental cognitive architectures. *Biol Inspired Cognit Architect* 18:116–127
- Voelker AR, Benjamin BV, Stewart TC, Boahen K, Eliasmith C (2017) Extending the neural engineering framework for nonideal silicon synapses. In: 2017 IEEE international symposium on circuits and systems (ISCAS), IEEE, pp 1–4
- Weber M, Weekes A (2009) Process thought as a heuristic for investigating consciousness. In: *Process approaches to consciousness in psychology, neuroscience, and philosophy of mind*, pp 37–56
- Weitzenfeld A, Arbib M, Alexander A (2002) *The neural simulation language*. MIT Press, Cambridge
- Wendt A, Gelbard F, Fittner M, Schaat S, Jakubec M, Brandstätter C, Kollmann S (2015) Decision-making in the cognitive architecture sima. In: 2015 Conference on Technologies and Applications of Artificial Intelligence (TAAD), IEEE, pp 330–335
- Wilson FA, Scaldidhe S, Goldman-Rakic PS (1993) Dissociation of object and spatial processing domains in primate prefrontal cortex. *Science* 260(5116):1955–1958
- Yamane Y, Tsunoda K, Matsumoto M, Phillips AN, Tanifuji M (2006) Representation of the spatial relationship among object parts by neurons in macaque inferotemporal cortex. *J Neurophysiol* 96(6):3147–3156. <https://doi.org/10.1152/jn.01224.2005>
- Zeki SM (1978) Functional specialisation in the visual cortex of the rhesus monkey. *Nature* 274(August):423–428
- Zeng AP, Ma HW (2013) *Modularity*. Springer, New York, pp 1441–1442

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.