The Agent Working Cycle in CATALINA*

Massimo Cossentino[1], Guido Averna[1], Giovanni Pilato[1], Myrto Mylopoulos[2] and John Mylopoulos[3]

[1]National Research Council of Italy (CNR), Via U. La Malfa, 153, Palermo, 90146, Italy
[2]Carleton University, 1125 Colonel By Drive, Ottawa, Canada
[3]University of Toronto, Toronto, Canada

Abstract
CATALINA is an agent architecture conceived to extend a Practical Reasoner architecture inspired by Bratman's BDI paradigm with capacities that are central to agentive guidance: executive function, attention modulation, and the global availability of desire-relevant information [1, 2]. This paper discusses the CATALINA agent working cycle that exploits the desired architectural features and is an extension of the classical MAPE (Monitor-Analyse-Plan-Execute) loop. We also propose an implementation of our architecture and an experimental setup developed to test CATALINA's agent features in two different scenarios.

Keywords
Global Workspace Theory, Executive Functions, BDI Agent, Practical Reasoning, Agent Metamodel, Goal-Oriented Reasoning, Agent Working Cycle

1. Introduction

Bratman's Belief-Desire-Intention (BDI) paradigm [3, 4] proposes a cognitive architecture for practical reasoning and constitutes one of the foundational references in Artificial Intelligence. The literature reflects many attempts to extend this cognitive architecture and apply it to specific domains. Several implementation frameworks incorporate this paradigm and implement it in various ways.

In the last few years, many authors have addressed the question of how an agent guides their goal-directed behaviour by exercising cognitive capacities associated with executive function, attention, and the global availability of information often linked to conscious awareness. This project explores extending the classical BDI paradigm with such capacities.

More specifically, we propose to incorporate within a classical BDI architecture, guidance capacities inspired by two theoretical accounts: *(i)* Baars' Global Workspace Theory [1], and (ii) Buehler's account of the Executive System [5, 6]. The result is the Cognitive AgenT prActicaL reasonINg Architecture (CATALINA) that implements a practical reasoner exhibiting features of Executive Function, attention modulation, and global availability of information inspired by Baars' theory of consciousness. The architecture is based on the central role of Baars' Global Workspace that acts as a working memory storing the conscious information shared with the suite of Executive Functions, inspired by Buehler's account of the Executive System. The focus of the current release of this work-in-progress project is on the GWT-related global availability and the attention modulation mechanism. The practical reasoner is, for now, rather simple, but further enhancements are planned, including the capacity to perform trade-off reasoning and accepting partial goal satisfaction as a strategy for solving problems for which a fully satisfying solution cannot be found.

In the following sections, we describe the CATALINA architecture and the specific agent working cycle we conceived to allow a fluid interaction among the Executive Functions (including practical reasoner functions, among them) and the Global Workspace.

26^{th} *Workshop "From Objects to Agents"*
massimo.cossentino@icar.cnr.it (M. Cossentino); guido.averna@icar.cnr.it (G. Averna); giovanni.pilato@icar.cnr.it (G. Pilato); myrto.mylopoulos@gmail.com (M. Mylopoulos); jm@cs.toronto.edu (J. Mylopoulos)
0000-0003-1258-9767 (M. Cossentino); 0000-0002-0935-5866 (G. Averna); 0000-0002-6254-2249 (G. Pilato); 0000-0003-2260-7068 (M. Mylopoulos); 0000-0002-8698-3292 (J. Mylopoulos)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

CEUR-WS.org/Vol-4028/paper11.pdf

The rest of the paper is organised as follows: Section 2 describes the three main theories we referred to in this work: Bratman's BDI architecture, Baars's Global Workspace, and Buehler's account of the Executive System and its functions. Section 3 reports the current status of the CATALINA architecture; this supports an agent working cycle that is described in Section 4. The experimental setup we developed for testing the desired features (mainly attention and consciousness in terms of global information availability) is described in Section 6, while Section 7 draws some interim conclusions and proposes some future work.

2. Research baseline

In this section, we briefly summarise the theoretical models that inspired the CATALINA architecture, which is based on three key elements: Bratman's Belief-Desire-Intention (BDI) model, Baars' Global Workspace Theory, and Buehler's account of the Executive System and its functions. Furthermore, we briefly recap the attention modulation mechanism, which plays a key role in CATALINA.

2.1. The Belief-Desire-Intention Architecture

The BDI model developed by Bratman [7] is an architecture for practical reasoning. An agent is resource-bounded since it usually has limited computational power and time constraints. The BDI model provides an abstract framework for plan-based, means-end reasoning, enabling the agent to formulate intentions and execute them effectively. In the Bratman model, three elements constitute the core ontology of practical reasoning: *beliefs*, *desires*, and *intentions*.

Beliefs represent the agent's knowledge of the state of the world and play a key role in guiding the formation of intentions and the selection of actions to achieve specific goals. They are typically expressed as logical predicates representing what the agent considers true regarding itself, other agents, and the environment. Beliefs are dynamic: they evolve over time in response to changes in the world. Moreover, they are updated based on the agent's perceptions, including information received from other agents, as well as through a priori reasoning.

Desires are mental states directed toward outcomes the agent considers to be worthwhile or valuable. They represent changes in the world that the agent views as good. Moreover, they serve as motivation for deciding how to act and thereby forming intentions. Since not all desires can be pursued simultaneously, the agent must be capable of deciding which desires to promote to intentions.

Intentions reflect an agent's commitment to pursue particular desires to achieve a desired change in the world. They are realised through plans (options in Bratman's terms), i.e. structured sequences of actions that serve as means towards the agent's intended desires. These plans may include sub-intentions and are designed to fulfil specific desires. Executing these plans enables the agent to advance towards its objectives. Of course, to be effective, intentions must remain coherent with the agent's beliefs and aligned with its desires.

Bratman's BDI model includes several psychological components that perform different operations that contribute to the processing of beliefs, desires, intentions, and the final execution of the selected intentions. Among these components, we mention here the *Means-End Reasoner*, the *Opportunity Analyser*, the *Filtering Process*, and the *Deliberation Process*. In the following, we briefly overview each component and its role for clarity and ease of understanding.

- The **Means-End Reasoner** is a component responsible for identifying plans or sub-plans that the agent can execute to achieve its intentions based on its beliefs. It uses the agent's current beliefs and desires to retrieve suitable plans from its repository or to generate new ones when necessary. These plans represent available options for the agent to fulfil its intentions.
- The **Opportunity Analyser** is a component that identifies and proposes alternative plans aligned with the agent's desires, distinct from those generated by the Means-End Reasoner, by recognising different opportunities present in the environment. It continuously evaluates the agent's desires

and monitors changes in the world to enhance existing intentions or propose new ones. The Opportunity Analyser operates in parallel with the Means-End Reasoner.

- The **Filtering Process** evaluates all incoming options from the Means-End Reasoner and the Opportunity Analyser. It rejects options that conflict with currently held intentions and their associated options, while also allowing for the revision of existing decisions in response to environmental changes. This process consists of two sub-components: the Compatibility Filter and the Filter Override Mechanism. The Compatibility Filter eliminates options incompatible with the agent's current intentions. In contrast, the Filter Override Mechanism reviews all options, including those discarded by the Compatibility Filter, and may reinstate useful ones back into the deliberation process. Both sub-components operate in parallel.
- The **Deliberation Process** evaluates all possible incoming options from the Filtering Process. When several options are considered, they are examined to see if they are suitable for execution. The aim of this component is to determine which options are worth pursuing: this allows the promotion of options to intentions. The selected intentions are executed by carrying out the actions they specify, allowing the agent to fulfill its desires.

2.2. Baars' Global Workspace Theory

Baars' Global Workspace Theory (GWT) of consciousness [8, 1] plays a key role in CATALINA [9], as it delineates the conscious processes we aim to endow our agents with. GWT proposes that an agent becomes conscious with respect to some information when this is globally broadcast, i.e., made accessible to different psychological modules involved in functions such as action planning and verbal report. In contrast, unconscious information remains confined to isolated modules. Baars illustrates this using a stage metaphor [8]: consciousness is akin to a spotlight illuminating a specific area of the stage (representing immediate memory), directed by selective attention. Only what falls under this spotlight is consciously experienced; the rest of the stage remains in darkness, representing unconscious processes.

From a cognitive architecture perspective, the Global Workspace Theory (GWT) describes a shared memory system that facilitates both information storage and communication across different functional modules [8] [10].

Central to this architecture is the Global Workspace (GW), typically aligned with working memory and a set of specialised architectural components responsible for different tasks, often operating in coordination, such as sensory processing, environmental evaluation, motor control, and language. GWT is built on three key principles [10]: 1) Each component must be specialised in its function; 2) Components compete for access to the GW 3) Once accessed, information in the GW is broadcast globally to all components. We will later report that in CATALINA, components are the Executive Functions and these are further decomposed into modules responsible for specific tasks (like deliberation, stimulus inhibition, and so on).

The GW acts like a network hub, efficiently routing information, while individual components act as connected devices. Components interact with the GW to retrieve or store information, which makes the GW fundamental for filtering, processing, and facilitating access to long-term memory. Information in the GW is temporary and subject to decay if not reinforced or replaced. Frequently used information can be secured in long-term memory through reinforcement. According to [10], each element in working memory has an associated strength value, which determines its persistence. This value decays more rapidly than memory traces in episodic memory, a component of long-term storage. When architectural components respond to a broadcast signal from the GW, they reinforce the strength of that memory, making it more likely to persist. This selective retention is crucial, as the GW has limited capacity and must discard irrelevant or unused information to remain effective.

2.3. Buehler's Executive System

Another fundamental contribution to agentive guidance is the capability of attention orientation. For this property, our work is inspired by Buehler's account of attentional modulation and the Executive System [11]. As he highlights, attention can be directed in two primary ways. When guided intentionally by the agent's goals, it is considered endogenous (aka top-down) attention. This form of control reflects a deliberate, voluntary focus.

In contrast, attention can also shift automatically in response to sudden or prominent stimuli, such as a loud noise or sharp pain, driven by exogenous (aka bottom-up) sources.

A central concept in understanding attentional control is bias: the mechanism by which certain environmental information is prioritised and influences behaviour. Bias can arise from either bottom-up or top-down processes. Salient stimuli trigger bottom-up bias, which is often modelled using a saliency map that highlights what stands out in the sensory field. Top-down bias, on the other hand, stems from the agent's intentions or planned actions. Biases can also be shaped by prior experience or learned associations. In our architecture, we refer to Endogenous Attention Modulation as the mechanism behind top-down attention, while Exogenous Attention Modulation governs bottom-up attention. Buehler [11],[2] proposes that an agent's psychological architecture includes an Executive System responsible for coordinating several specialised subsystems (architectural components labelled Executive Functions), each handling a distinct functionality. These four functions are:

- **Executive Switching Function**: This function activates relevant representations and capacities for performing some goal-related task.
- **Executive Inhibition Function**: This function suppresses distractions or actions that might interfere with the agent's goals, working closely with the switching system to regulate attention.
- **Executive Resource Allocation Function**: This function allocates the available resources to allow achievement of the agent's goals.
- **Working Memory Maintenance Function**: This function controls the information flow between Long-Term Memory and the Global Workspace; furthermore, it ensures that relevant data is available for conscious processing.

2.3.1. The Attention Modulation Mechanism

The Executive Inhibition Function is responsible for two core operations: regulating attention and defining inhibition regions, which are areas of the environment outside the agent's current focus that are temporarily excluded from the current processing. Attention modulation is central to Global Workspace (GW) theory, as only information that receives focused attention can enter the GW and become globally available. In our architecture, this modulation is managed by the Agent_Focus module that is part of the Inhibition Function; this module exploits saliency levels and attention thresholds to determine what information is relevant.

The inhibition regions are designed to limit the agent's perceptual focus to a particular area of the environment to improve processing efficiency. Furthermore, to avoid cognitive overload, this function also suppresses the internal states that interfere with the agent's current goal, e.g., distracting thoughts or emotions. In our architecture, two types of attention modulation are distinguished, each one with a unique mechanism:

- **Endogenous Attention Modulation** (*top-down*): Directed by the agent's desires, this process enables focused attention aligned with intentional objectives. It also contributes to generating new desires, though not all desires can be pursued due to practical limitations. Only a selection of them are promoted to intentions. These functionalities in CATALINA are obtained by the contributions of the Switching Function Desire_Promotion module, together with the Desire_Deletion and Intention_Deletion modules of the Inhibition Function.
- **Exogenous Attention Modulation** (*bottom-up*): This kind of modulation is triggered by unexpected or salient sensory inputs, such as sudden brightness or movement. These inputs can

override existing attention filters and lead to belief updates, often without conscious intent. When a new stimulus is significant enough, the agent may form an epistemic desire, i.e., a motivation to investigate the source of that perception, which may evolve into a new desire [12]. Exogenous attention modulation in CATALINA is implemented in the Switching Function by the Switching_To_Stimulus module.

When the Executive Switching Function receives a belief update from the Global Workspace, it compares the saliency of the new information to the current attention threshold.
The agent may modify its set of desires by adding new, more pertinent ones or removing those that are now considered less significant if the new information is thought to be more salient. In this manner, bottom-up attention can prioritise particular goals (desires) and promote adaptive behaviour by reshaping the agent's motivational structure in response to changing environmental conditions.

3. The CATALINA Architecture

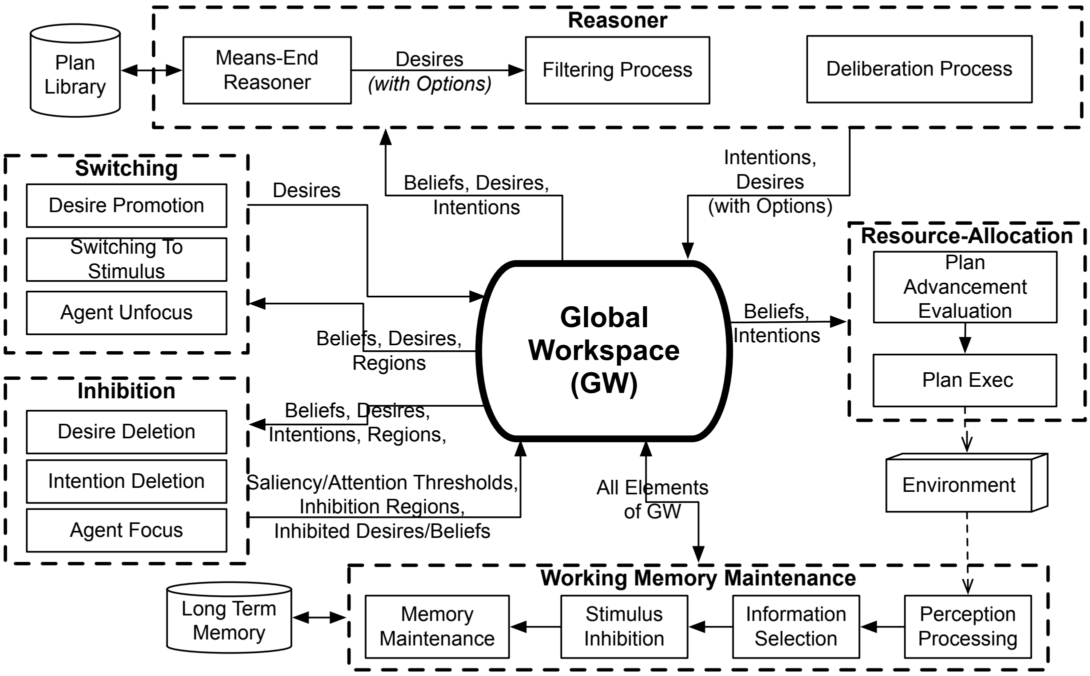

Figure 1: The Cognitive AgenT prActicaL reasonINg Architecture (CATALINA), v. 0.2

This section describes the CATALINA architecture that is composed of five components (four Executive Functions and the Global Workspace) plus one Long-Term Memory (the Plan Library, although depicted separately in Fig. 1, is part of the Long-Term Memory).

The Executive Functions are further decomposed into modules (a kind of sub-function), each one responsible for some specific task (like means-end reasoning, perception processing, and so on).

It is worth noting that this version of the architecture (v.0.2) is an improvement of the previously presented v.0.1 [9], where a few modules of the Inhibition and Switching Functions have been split into smaller components and redistributed to optimise the working cycle we will propose in Sect. 4.

The following subsections will detail a few fundamental concepts, the CATALINA's structure, scope and behaviour of each of the main modules.

3.1. Fundamental Concepts

We defined a few concepts as the basis of our proposed architecture. First of all, we distinguish two types of desires: *Practical Desires* are mental states directed toward a state of the world that the agent considers to be good or valuable and is motivated to bring about; *Epistemic Desires* are mental states directed toward acquiring some piece of knowledge (a belief) the agent considers valuable. Both Practical and Epistemic Desires also have a saliency representing the relevance or urgency to achieve them.

Practical Desires are defined using a linear temporal logic, and often they are constrained by *quality desires* and *green desires*. Quality desires regard non-functional aspects and, because of their qualitative nature, their satisfaction criterion is often hard to fix. For this reason, we adopt an operationalisation of such desires as described by [13]. Green desires represent constraints coming from some environmental respect policy, often they descend from local laws or rules, and the agent cannot violate them.

Desires are normally injected into the agent by the designer so the agent 'finds' them already defined at the beginning of its life. Joining this specification phase with the classical Bratman's desire concept (where desires are inborn in the agent) passes through the saliency/attention thresholds: we assume the agent receives several *Standing Desires* by the designer, they are 'promoted' to *Active Desires* when their saliency is greater than the saliency threshold (the filtering threshold of attention when the agent is not focused) or the attention threshold (the filtering threshold when the agent is already focused on pursuing some intention). It is worth noting that the agent's Means-End Reasoner soon starts looking for options that can satisfy any Active Desire once that is promoted.

Finally, Active Desires are promoted to *Intentions* (and pursued by the agent) if the reasoner finds a suitable option (that respects temporal, quality and green constraints) and the desire's saliency is higher than the current saliency/attention thresholds.

A detailed description of the agent metamodel and more specifically of its goal model is available in [14]. In the following, we will describe the main components of the architecture.

3.2. The Global Workspace

This module implements the Global Workspace working memory conceived by Baars' theory [1], where, according to GWT, consciousness information is localised. The GW is a shared memory for all the executive functions. In CATALINA, all executive functions and, consequently, their modules, can access the GW at the same time and be notified of changes in its content; for this reason, we implemented it as a publish-subscribe dashboard, where functions may subscribe to a specific piece of information and be notified of any change in that. Other architectural solutions (tuples, shared memory,...) could be considered for implementing the GW.
We preferred the publish-subscribe pattern because it allows a good decoupling of the components, good scalability, good modularity, and it is naturally event-driven (as it is useful for a working memory that is solicited by incoming stimuli). The common defect of requiring a middleware for its implementation comes with the solution of implementing the GW functioning mechanism, and its inevitable bottleneck is the natural limits of the working memory, which is exactly what Baar's theory postulates.

3.3. The Executive Switching Function

A desirable feature for an intelligent agent is the ability to change its mind and adopt new plans in reaction to opportunities or obstacles in the environment. This is one of the purposes of the Executive Switching Function (sometimes also called cognitive flexibility, mental flexibility, or mental set shifting and closely linked to creativity [6]). In the proposed architecture, this function is responsible for triggering the consideration of new standing desires (by promoting them to active desires) even if the agent is already committed to pursuing some other intention. To dramatically simplify the role of this function, we can say it is devoted to activating what is relevant for the agent at a specific time. In CATALINA, the Switching Function comprises three modules:

- The Desire_Promotion module is responsible for the promotion of standing desires to active desires. This happens for desires whose precondition is verified and whose saliency is greater than the current attention threshold.
- The Switching_to_Stimulus module realises what we can identify as the typical reaction agent behaviour. When a salient stimulus is received by the Working Memory Maintenance Function

and posted to the GW, this module is responsible for processing it and promoting to active desire any standing desire that reacts to that.
- The Agent_Unfocus module is invoked when the agent is not devoted to any task; in this case, the saliency and attention thresholds are set at their default values, and the inhibition lists of beliefs, desires and perception regions are all reset. In this way, the agent is clear of any mental constraint for accepting new tasks, i.e., consider the activation of some standing desire, calculate available options and adopt the most promising one in the deliberated new intention.

3.4. The Executive Inhibition Function

The Executive Inhibition Function mainly deals with attention modulation. According to Buehler's theory, this function is responsible for removing from the agent's working memory what is not relevant to the tasks (and intentions) at hand. This is very relevant for a good rational agent architecture; it enables the agent to focus on what is relevant and reduces the computational cost of continuously processing every single perception (better stimulus), even if irrelevant.

The Inhibition Function is composed of three main modules encompassing relevant algorithms for the agent's behaviour:

- The Desire_Deletion module is responsible for the deletion from the GW of all the active desires that have a low saliency value; in this way, the agent, at each time, considers only desires that are at least as relevant as the currently selected intentions. Of course, deletion propagates to the corresponding intention, so if an agent switches to a more salient intention, the less relevant ones are stopped and will be pursued later on.
- The Intention_Deletion module removes the intentions linked to every desire deleted by the previous module.
- The Agent_Focus module is responsible for the attention modulation mechanism. When the agent deliberates to pursue an intention, this module computes the new saliency/attention thresholds, it deletes from the GW all beliefs that are not relevant, defines the perception inhibition regions and inhibits all desires that could clash with the new intention.

At this stage of the development of our proposed architecture, we considered the saliency of desires to be static. This allowed us to focus on the framework's correct functioning and create case studies to verify the agent's correct behaviour. In the future, we plan to make the saliency of desires adaptive and dynamic, in accordance with neuroscience theories. In the current implementation, the saliency of each stimulus is defined in the code as it happens for desires. We acknowledge that as a simplification since the same stimulus may be more or less relevant in different agents' conditions. Correctly defining such saliency is a complex task with implications involving cognitive studies, and moreover, it depends on the application domain. In our architecture, the agent's saliency threshold and attention threshold fall within the range [0,1), and the latter is always greater than or equal to the former. As regards the relationship between the selected intentions (each one having the same saliency of the related desire) and the saliency/attention thresholds, they are defined by a simple algorithm:

- **Saliency Threshold**: if the agent is focused, the saliency threshold takes the value of the most salient intention; otherwise, if the agent is not focused, it is set at a default low value.
- **Attention Threshold**: if the agent is focused, it is set at a value that is more than the Saliency Threshold. In formula: AttentionThreshold = SaliencyThreshold + (1-SaliencyThreshold)/2. If the agent is not focused, the Attention Threshold is equal to the Saliency Threshold.

3.5. The Executive Reasoner Function

Our Reasoner Function is inspired by Bratman's reasoner [7] and inherits from that its general structure. It computes options (plans) that could be used to achieve active desires, it filters options that do not satisfy quality constraints, the deliberation process considers the set of available active desires with options and commits the agent to pursue some of them in form of intentions. This function is composed of the following modules:

- The Means-End_Reasoner is a planning module that generates options for achieving the state of the world addressed by the active desires. In CATALINA, desires may include temporal constraints specified by first-order temporal logic operators; therefore, this module also takes care that delivered options respect such constraints.
- The Filtering_Process module performs a quality filter on the options delivered by the previous module. More specifically, the Filtering Process ensures that options will satisfy quality and green desires that constrain the desire.
- The Deliberation_Process considers the set of current intentions, the active desires with available options and decides about the opportunity to select some of these desires for promotion to intentions. The decision process considers saliency as a prioritising factor, but also considers the available resources and the agent's state.

3.6. The Executive Working Memory Maintenance Function

In the current implementation of the architecture, this executive function plays three roles:

- The classical functionalities of the Memory Maintenance function that are devoted to maintaining the knowledge in the short-term (GW).
- The maintenance of long-term memory (saving information to long-term memory, like the successful adoption of some options for achieving a specified desire and the corresponding benchmark, if available).
- The processing of perceptions that includes extracting a semantic representation from the raw data and filtering significant information before posting it to the GW. This means that stimuli (beliefs with a saliency value) are extracted from the processed perception data. The Stimulus Inhibition module filters such stimuli and ensures that only relevant stimuli are posted to the GW. In this way, we avoid overloading the GW with irrelevant stimuli.

Because of the significant features it is responsible for, this function is composed of several modules:

- The Perception Processing module controls perceptors at the hardware level and forwards raw that to the next module.
- The Information Selection module processes perception data and extracts semantic information from that. For instance, it defines the truth value of some beliefs from the analysis of the perceived values of temperature. Such beliefs are complemented with a saliency value according to their significance, thus becoming a stimulus.
- The Stimulus Inhibition module filters all stimuli that fall inside one of the inhibition regions or whose saliency is less than the attention threshold. This ensures that irrelevant stimuli are not posted to the GW.
- The Memory Maintenance module posts to the GW all stimuli that passed the inhibition filtering, also it ensures the saving of relevant information to a permanent database. For instance, CATALINA saves the successful outcome of intention pursuit with a specific option so that it can be reused.

3.7. The Executive Resource Allocation Function

The Executive Resource Allocation (RA) Function is responsible for executing the agent's intention (more specifically, the option selected for that intention). In its current implementation, the RA Function performs a straightforward sequential execution of the option's actions by invoking the corresponding agent behaviours. Future developments aim to integrate a workflow engine that could support more complex plans, including those with parallel and concurrent action flows.

The next section will report how the modules discussed in this Section interact to perform the expected agent behaviour. This happens according to a specific working cycle that extends a classical approach.

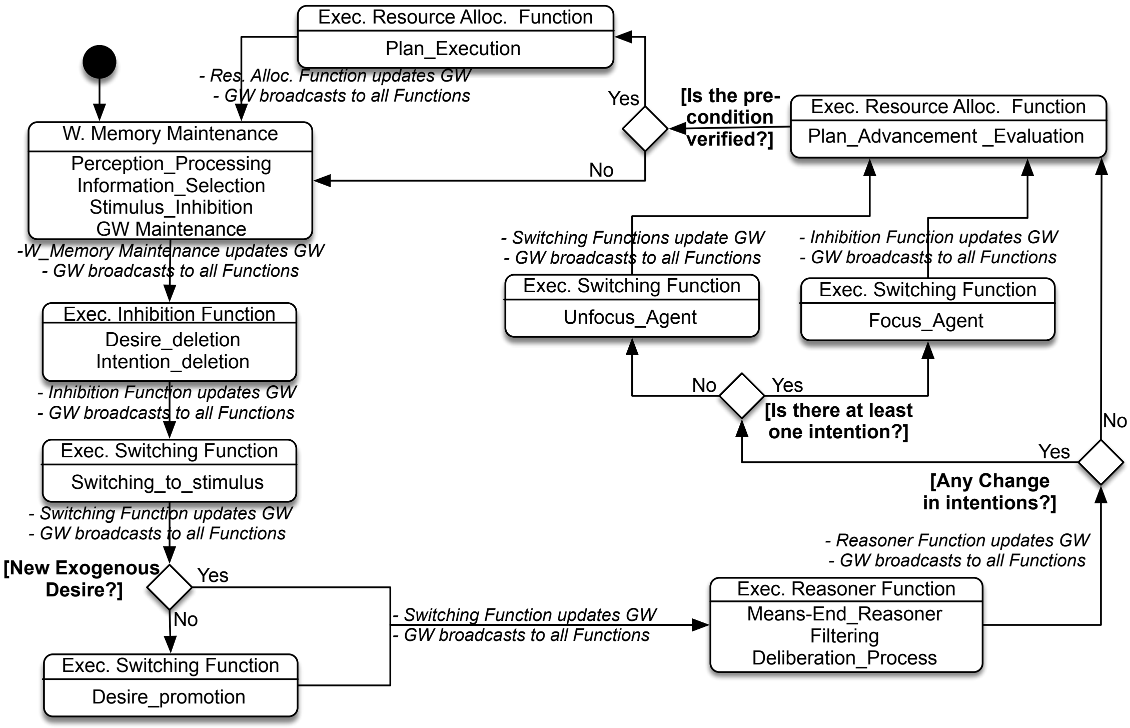

Figure 2: The Agent Working Cycle in the CATALINA Architecture

4. The Agent Working Cycle

This section describes the agent working cycle in the CATALINA architecture. Defining that has been a significant challenge since Bueheler's theory of executive functions supposes that the functions operate concurrently. We made the strategic choice to postpone the implementation of such concurrency because it would make it more difficult to test and disambiguate the specific behaviours we were aiming to define. We are already developing the next release of CATALINA that will fully support the parallel execution of the different functions and of the Global Workspace. The challenge in defining this working cycle consists in ensuring the correct functionality of the different functions, their interactions with the GW and at the same time avoiding the overloading of sensitive components like the GW, which is at the centre of all the interactions.

The working cycle we tuned after several tests is represented in Fig. 2 and is a specialisation of the classical MAPE (Monitor-Analyse-Plan-Execute) agent cycle.

The cycle starts with a monitor phase performed by the Working Memory Maintenance Function. We have assembled the perception capabilities of the agent in this function. Indeed, in the implemented scenarios that simulate the travel of an autonomous vehicle (see Section 6), this function is able to:

- Perceive the status of the route ahead, the autonomous vehicle (perception consists of asking the user so that the user can alter the simulation and test different scenarios). Possible answers by the user:
  - The road ahead is clear and safe.
  - The road is closed, and there is some danger on the road.
- Contact the Traffic Information Service and ask how long the road will be closed.

The perception modules (Perception Processing and Information Selection) process the information and generate the corresponding stimulus. A stimulus is a belief (a predicate with some truth value) that also has a saliency value (according to the importance of the perception). The Stimulus Inhibition module filters the stimuli according to their relevance (i.e. saliency) and also considers if they belong to inhibited perception regions or not (as described by Buehler in [2]. Finally, the GW Maintenance module updates the values of these beliefs in the Global Workspace (if they are not inhibited).

The GW broadcasts the updated stimulus to the subscribers to this specific event (other executive functions). Indeed, the process is more complex: the GW broadcasts an event saying to each subscriber that some information of their interest has changed. Each Function asks for the information when it is ready (in this way, we do not continuously interrupt the work of the functions).

The Desire_Deletion module of the Inhibition Function deletes (from the GW) all the active desires whose saliency is lower than the saliency/attention thresholds; more specifically, the attention threshold is considered for desires related to inhibited desires, at the first run of the working cycle, no desire is inhibited; desires inhibition happens when the agent focuses on some task, in order to remove from the GW all desires that clash with the current intentions. Consequently, the Intention_Deletion module of the Inhibition Function deletes (from the GW) all intentions related to the desires removed at the previous step. The GW broadcasts the changes made by the Inhibition Function in the list of stimuli, desires and intentions.

The Switching_To_Stimulus module of the Switching Function compares the stimulus's saliency with the saliency/attention thresholds, looks for any standing desire triggered by this stimulus and promotes that to an (exogenous) active desire.

Now the working cycle has a decision node: if the Switching_To_Stimulus module has not promoted any standing desire to a new active desire, then the Desire_Promotion module of the Switching Function is invoked. It performs a process of both inhibited and non-inhibited desires (where inhibited desires are those that are not coherent with the selected intentions). Regarding inhibited desires, they are promoted to (endogenous) active desires if their precondition holds and their saliency exceeds the Attention threshold (this means they are very salient desires that may, potentially, move the agent to change its current intentions). Conversely, non-inhibited desires are promoted if their saliency exceeds the Saliency threshold. The Switching Function posts the changes to the GW, which notifies all subscribers.

If new (active) desires have been posted to the GW, the Means-End Reasoner module of the Reasoner Function starts looking for options that could satisfy them. Such options not only allow the agent to reach the desired state of the world, but also should obey the temporal constraint specified in the desire definition.

The Filtering module of the Reasoner Function performs two fundamental operations for the satisfaction of the agent's desires:

- Options that do not satisfy the green desires attached to the pursued desire are removed from the list of options suitable for that desire.
- The list of remaining options, for each desire, is ordered according to their results in terms of validation of the quality desires.

Finally, the Deliberation_process of the same Function decides whether to promote some of the active desires to an intention and selects which one will be executed soon. The Reasoner Function posts the changes (active desires with new options, intentions, and selected intentions) to the GW, which notifies all subscribers.

If the Reasoner has performed some changes in the current intentions, two alternatives may occur:

- If there are no more intentions, the Unfocus_Agent module of the Switching Function will set the Saliency and Attention Thresholds to their (idle) default value, and void the list of inhibited beliefs, regions and desires.
- If there is still at least one active intention, the Focus_Agent module of the Inhibition Function defines the new values of Saliency and Attention Thresholds, it removes from the GW all the beliefs that are not related to the selected intention, it properly defines the inhibition regions that will mask future perceptions during the achievement of the currently selected intentions and it inhibits all desires that would clash with the intended states of the world.

All changes are posted to the GW by the Switching Function; the GW notifies the other Functions.

If the Reasoner did not perform any change in the current intentions, or after the previous steps in case of changes, the execution goes to the following steps.

The Resource Allocation Function is devoted to the execution of the option actions; the Advancement_Evaluation module compares the current state of the world with:

- The desired one, and if they coincide, cancels the currently executed intention from the GW (since it has been achieved).
- The pre-condition of the next option action. If they do not meet, the last action failed, and therefore, the plan cannot continue. It is necessary to compute a recovery plan (i.e., a novel option starting from the new state of the world). Therefore, the Plan_Execution module will not be executed.

Finally, if the precondition of the next action is verified, the Plan_Execution module performs the next step in the current option. As usual, results are posted to the GW that notifies concerned Functions. The loop restarts at a new time step.

5. Comparison With Existing Agent Architectures

Positioning the contribution of CATALINA in the scientific literature may be done by looking at existing cognitive architectures as well as agent implementation frameworks. Several cognitive architectures exist in the literature, among the others ACT-R [15], SOAR [16], LIDA [17]. Likewise, CATALINA may be compared with agent development frameworks (like Jason [18]) that are not specifically focused on consciousness features but support the BDI practical reasoning. In the following, we will briefly discuss the similarities and differences among these architectures.

Act-R (Adaptive Control of Thought-Rational) aims to model human cognition as a set of modules that interact through a central production system. Modules implement a declarative memory devoted to storing facts and a procedural memory containing the rules that operate on facts by means of pattern matching. This architecture exhibits a sound grounding on psychological theories and may be used to simulate human performance in cognitive tasks. Similarities exist between Act-R and CATALINA. We may consider that CATALINA beliefs somehow stand for the declarative memory of Act-R, and the reasoning mechanism of CATALINA is procedural, explicitly goal-oriented and largely directed by the satisfaction of quality goals. No support for consciousness or attention modulation mechanism is known to exist for Act-R.

SOAR is a general-purpose framework for building intelligent agents conceived to model and replicate general human cognitive abilities such as problem-solving, learning, and decision-making. The learning capability differentiates SOAR from CATALINA, although this is part of future work for the latter. Again, no support for consciousness or attention modulation mechanism is known to exist for Act-R, and it is unlikely to be found considering the age of the architecture.

LIDA architecture shares several objectives with CATALINA; they both aim to support consciousness, attention modulation and goal-directed behaviour. CATALINA has a richer goal model that allows to complete the functional specification with quality constraints that are of paramount importance in state-of-the-art software implementations. The Metric-Interval Temporal Logic (MITL) language used to represent CATALINA's practical desires [14] is another interesting feature that allows for modelling time constraints in a formal way.

Finally, Jason [18] shares the same referring goal-directed paradigm (BDI and Bratman's practical reasoning) with CATALINA, and it remains one of the most well-known agent development frameworks in the agent community. Jason supports functional goals that are implemented as practical desires in CATALINA. Jason's epistemic goals (test goal is their label) deeply resemble CATALINA's epistemic desires. Jason does not support quality and green desires, whereas CATALINA does (more about them in [14]). Moreover, the adoption of a formal temporal logic in CATALINA is another significant difference.

6. The Experimental Setup

The challenge of theoretical development of the CATALINA architecture is accompanied by the challenge of concrete and real development of a software[1] that adopts the CATALINA model. This software is necessary both to test our proposed architecture and because it constitutes the basis for the development of our planned future version of CATALINA as a concurrent architecture. Furthermore, to verify the correct working of our theory, it was necessary to create an experimental simulation in which the agent could work. CATALINA was developed using the Java language, version 22.0.2, and the Java runtime version is 22.0.2+9. The architecture is composed of 69 files: 57 class files (more than 57 classes) and 12 enumeration files. The code lines developed are more than 13000. For the development of the architecture, we used standard Java libraries without using third-party libraries. The integrated development environment is Eclipse, version 2024-06, and its Build ID is 20240606-1231. The operating system is Windows 11.

In our simulation, an autonomous vehicle, led by an agent, moves along a path from one city to another. The agent moves on a map composed of 47 cities, connected by routes. Each route is composed of one or more segments, called steps, their number roughing representing the distance between the two cities. A path connects two or more cities via one or more routes. The path is planned by the agent for its intentions that the agent wants to pursue over time, and by the possible obstacles that it may encounter while moving. Each route has several characteristics: a maximum speed, a quality value of the panorama, a type of vehicle that can be used to cross that route, and a maximum amount of pollution allowed to the vehicle that crosses the route. In our simulation, the agent has three standing practical desires (Visit_Paris, Visit_Frankfurt and Visit_Rome). They are satisfied when the agent visits Paris, Frankfurt and Rome, respectively. Visit_Paris and Visit_Frankfurt have no preconditions, while Visit_Rome has a precondition, and it must be verified before promoting this desires to an active desire and pursuing that: the agent must have visited Paris. There are two quality desires (Panorama_is_Great and Drive_Safely) and one green desire (Limit_Pollution_to_08) that we associate with the desires. The quality desire Panorama_is_great indicates that the option that satisfies the desire should have the highest possible panorama quality value. Instead, the Driving_Safely quality desire specifies that the agent's driving should be safe, referring to an already cited approach ([13]), we operationalise that as a constraint that limits the maximum speed of the vehicle. The green desire Limit_Pollution_to_08 requires that the total path performed by the agent must not generate more than an average of 0.8 Kg of pollution in driving along each route of its path (a route connects two cities).

We suppose the agent may select different types of vehicles, an electric one (but not all routes have enough recharge stations for that), an hybrid one (polluting more than the electric one but this needs fewer recharging stations or even no ones), and a conventional gasoline engine (the vehicle do not needs recharging stations, the travel time is shorter because there is no need to stop for recharging but the vehicle produces much more pollution).

In the experimental setup, we created two scenarios. In the first scenario, we want to show the different behaviour of our architecture when the saliency values assigned to the desires vary. In the second case, we want to show that quality desires have a critical role when they are assigned to the desires, and significantly affect the agent's behaviour.

The working cycle in our architecture is sensitive to the values of saliency/attention thresholds and is very important for the agent's behaviour. In fact, according to the current values of saliency/attention thresholds, some modules in the working cycle are executed rather while others are not, and some standing desires may or may not become intentions with consequent execution of the Focus (Executive Inhibition Function class) or Unfocus (Executive Switching Function class) Agent module, or a simple execution of the intentions of the previous cycle via Plan_Advancement_Evaluation (Executive Resource-Allocation Function class). For example, if there are some standing desires that have saliency lower than the saliency/attention thresholds, these cannot be promoted to active desires, and consequently,

[1]The current version of the architecture is available for download from the CATALINA GitHub repository: https://github.com/CATALINA-Architecture/CATALINA_Model

the Reasoner does not execute any of its modules since it does not receive a signal from the Global Workspace of a desire update message. The first scenario is useful to understand this behaviour; Table 1 shows the values of saliency for each standing desire.

In the second scenario, we show that the agent's planning decision behaviour varies significantly according to the quality desires that are associated with the desires. In fact, the quality desires that are associated with the desires provide a preference in choosing which of the available options they have to use to reach the desire. The quality desires can be very different from each other. In our simulation, this translates into the ability to choose different paths according to the different quality desires that are associated with the desires. The Table 2 reports the quality desires used in this second scenario. Note that cases 1 and 2 of the two scenarios are not related, but are cases of two distinct and independent scenarios. Now we will discuss the simulation and then the results of the experimental simulation, also varying the data, as already mentioned.

6.1. Experimental Simulation

Let us consider the previously indicated desires inserted at design time in the agent. When the agent starts, it tries to analyse the latest perceptions acquired from the environment, but at this stage, no perceptions exist. Consequently, the Switching_to_stimulus module is not executed because it does not receive any stimulus update notification from the Global Workspace. The Desire_promotion module analyses the standing practical desires inserted at design time and checks whether some of them have a true precondition (if the precondition exists for that desire) and a saliency higher than the saliency/attention thresholds of the agent. In our simulation, Visit_Paris and Visit_Frankfurt are promoted to active desire, while Visit_Rome is not promoted because its precondition is not true at this stage. The Desire_promotion module posts these active desires into the Global Workspace, which sends an active desire update notification to all interested components. Consequently, the Reasoner Function executes its Means-end module, which acquires the new active desires and analyses them. For each active desire, this module computes all possible options that satisfy the active desire, and it removes all options that do not satisfy the temporal operator in the desire definition. Next, the remaining options are filtered by the Filtering Process module of the Reasoner. For each active desire, it removes all options that do not meet the Green Desire Limit_Pollution_to_08, and it sorts the remaining options based on the Quality Desire preferences. Finally, the Reasoner executes the Deliberate Process and, for each active desire, it can deliberate an intention (which has at most one option) to satisfy its related desire.

In our simulation, the active practical desire Visit_Paris has 6 surviving options. Therefore, its related intention will have one executable option. In contrast, for the active practical desire Visit_Frankfurt, there are no surviving options from the end of the Filtering Process, so the Deliberation Process creates a related intention without any options to satisfy this active desire. Finally, the Reasoner updates the deliberated intentions to the Global Workspace, which sends an intention update notification. The Focus Agent module (Executive Inhibition Function class) receives this message and updates the saliency/attention thresholds of the agent, computes inhibition regions, inhibited beliefs, and desires, and moves them into the long memory, while maintaining essential information in the Global Workspace. Finally, the Plan_Advancement_Evaluation module (Executive Resource-Allocation Function class) evaluates if the preconditions of the next planned action (at this stage, the first action) of the deliberated option for the intention with the highest saliency are correct. If they are, the Plan_Exec (Executive Resource-Allocation Function class) executes the action. So, the working cycle can continue by starting again from the Perception Processing module (Executive Working Memory Function class).
Now, the agent has no perception, so the Switching_to_stimulus module is not performed. The Desire_promotion module is also not performed because all standing practical desires have been analysed. Hence, the Reasoner is not performed because it does not acquire an active desire update message, and only the Plan_Advancement_Evaluation and Plan_Exec modules are performed, handling the next action of the current intention. Therefore, the next loop of the Working cycle restarts. This mechanism continues until either the current intention has been fully pursued and the active practical desire (Visit_Paris) has been satisfied or the agent discovers a danger that blocks the road during its travel to

Table 1
Scenario 1: Saliency values for the Visit_Rome practical desire are different in the two cases considered for scenario 1. Moreover, this is the unique desire that has a precondition.

Scenario 1					
Desire	Case 1	Case 2	Precondition	Saliency Th.	Attention Th.
Visit_Paris	saliency = 0.5	saliency = 0.5	null	0.5	0.75
Visit_Rome	saliency = 0.7	saliency = 0.8	The agent visited Paris	0.5	0.75
Visit_Frankfurt	saliency = 0.4	saliency = 0.4	null	0.5	0.75

Table 2
Scenario 2: Visit_Rome has different quality desires in cases 1 and 2 of scenario 2.

Scenario 2		
Desire	Case 1	Case 2
Visit_Paris	quality desire = Panorama_is_Great	quality desire = Panorama_is_Great
Visit_Rome	quality desire = Panorama_is_Great	quality desire = Drive_Safely
Visit_Frankfurt	quality desire = Panorama_is_Great	quality desire = Panorama_is_Great

Paris, and it arises a standing epistemic desire to understand the type of danger and how long the road will be closed.

6.2. Experimental Results

The scenarios start when the agent arrives in Paris and deliberates on moving towards Rome. Specifically, scenario 1 starts when the agent promotes the standing practical desire Visit_Rome to active practical desire and subsequently to intention, while scenario 2 occurs when the agent chooses an option (a path to take) that is strongly influenced by the quality desire that is assigned to the practical desire Visit_Rome.
Let's consider that the agent has arrived in Paris. The active practical desire Visit_Paris has been satisfied, so it is removed from the list of desires to satisfy, and the agent updates its beliefs. Specifically, the agent updates the belief "Belief_Visited_City" to true. Further, due to the removal of the active practical desire Visit_Paris and the related intention, the Global Workspace sends a broadcast message of intention change to all Executive Functions.

When the agent does this, the agent's current saliency threshold is equal to the saliency of the removed active desire, saliency threshold = 0.5, and the agent's current attention thresholds have the value 0.75. At this stage, when the agent executes the Desire_promotion module, it verifies that the precondition of the standing practical desire Visit_Rome is true. At this stage, scenario 1 occurs.
Table 1 can be considered as a snapshot of the practical desires in the two cases considered for scenario 1 at the beginning of the new working cycle. In case 1, the saliency of the standing practical desire Visit_Rome is equal to 0.7, so when the Desire_promotion module is executed, it checks the remaining standing practical desire Visit_Frankfurt. At this stage, Visit_Rome is an inhibited standing practical desire, and its saliency is lower than the agent's attention threshold. Therefore, the Desire_promotion module does not promote any standing desire to active desire. Consequently, the Reasoner is not executed because there are no active desire updates, and therefore, the Unfocus Agent module (Executive Switching Function class) is executed because there has been a change in intentions and there are no more intentions to pursue. The Unfocus Agent module lowers the saliency and attention thresholds to the default values (both 0.3) and eliminates the inhibition regions and the inhibited beliefs and inhibited standing desires because the agent is no longer focused on some intention. Therefore, the standing desires Visit_Rome and Visit_Frankfurt are both uninhibited and reside in the Global Workspace. Therefore, the Plan_Advancement_Evaluation module does not execute anything because there are no intentions to pursue, and consequently, a new cycle of the Working is executed. In this new phase, the Desire_promotion module promotes both standing desires to active desires. Specifically, Visit_Rome (being uninhibited and having its precondition true) has a saliency higher than the agent's saliency

Table 3
Scenario 2: Different quality desires influence the options to choose to pursue an intention. The selected options can produce significantly different paths.

Scenario 2		
Feature of path	Case 1: Panorama_is_great	Case 2: Driving_Safely
Cities	Paris, Marseille, Rome	Paris, Bruxelles, Frankfurt, Munchen, Venice, Rome
Id Routes	125, 120	34, 84, 104, 88, 74
Total time (hours)	7.5	10.0

threshold (0.7 > 0.3).
In case 2, when the active practical desire Visit_Paris has just been satisfied, the Desire_promotion module verifies that the inhibited standing practical desire Visit_Rome has saliency equal to 0.8, which is greater than the agent's attention threshold, and so it promotes Visit_Rome to active practical desire. Consequently, the Reasoner computes some options to satisfy this active practical desire, finds more than one, and filters them, keeping some options to satisfy the active practical desire. Then, the Deliberation Process module deliberates on an intention with one option. Consequently, the Focus Agent module modifies the saliency threshold from 0.5 to 0.8 and the attention threshold from 0.75 to 0.9, and recreates the inhibition regions and the inhibited beliefs and inhibited standing desires according to the new intention.
Scenario 1 shows the different functioning of the working cycle according to the variation of the saliency value associated with the active desires, and that saliency significantly influences the working cycle by allowing the execution of some modules, rather than others, in the working cycle. In fact, in case 1, the working cycle executes a cycle in which it deconcentrates the agent's attention by executing only some modules. Instead, in case 2, it amplifies the agent's concentration towards active desires that were inhibited by executing all the modules necessary to pursue a new intention.
In scenario 2, we show the key role and influences that different quality desires have on the Reasoner. In fact, the two cases taken as examples show a clear influence on the choice of the paths that the agent must take to achieve its active desires. In table 3, we show the cities, routes, and the total time to complete the whole path of the options chosen by the Reasoner according to the two different quality desires: Panorama_is_great and Driving_Safely.

Let's consider again the practical desire Visit_Rome immediately after the Desire_promotion module has promoted it to active practical desire. Then, after the Reasoner has calculated all the possible options and has performed the green desire filter, it examines the current quality desire and keeps all the options that satisfy the quality desire criteria, and that the Deliberation Process module will then have to consider to decide which option to adopt for the new intention. In case 1, the quality desire associated with the practical desire Visit_Rome is Panorama_is_great. All the options kept will be those in which the quality value of the Panorama throughout the path will be high.

In case 2, the quality desire associated with the practical desire Visit_Rome is Driving_Safely. All the options retained will be those in which the qualitative value of the average speed along the entire route can be considered as safe driving.

In table 3, we see that after associating the quality desire Panorama_is_great to the active practical desire, the chosen option by Reasoner is the one in which the agent chooses a path consisting of three cities, including the departure and destination ones, and to cross 2 routes, spending a travel time for the entire path of 7.5 hours. Instead, associating the quality desire Driving_Safely, the agent decides a path consisting of 6 cities, and crosses 5 routes for a total time of 10.0 hours. Thus, different quality desires can significantly influence the choice of options that the Reasoner assigns to intentions to satisfy the active desires.

7. Conclusions and Future Work

Developing cognitive agents whose capacities go beyond the mere deliberation of what desire to pursue is one of the challenges faced by many contributions in the literature. Attention and awareness/consciousness play prominent roles among the most studied features. The CATALINA architecture proposes an extension of the classical BDI paradigm with contributions coming from the well-known theories of Baars (the Global Workspace Theory) and Buehler (the Executive System). We attempted a blending of these theories, thus aiming at the conception of an agent architecture that supports practical reasoning, attention modulation and global availability of information linked to consciousness (according to Baars' view of that). Moreover, the CATALINA reasoner has some interesting and innovative features since it supports reasoning on quality desires that constrain the options generated by the means-end reasoner and reasoning on green desires that oblige the agent to respect environment-friendly rules.

We propose an experimental setup based on the simulation of an autonomous vehicle travelling across the European map and operating under a few quality and green desire constraints.

The current version of CATALINA is still a work-in-progress. We plan to extend it with several innovative features, namely the capability to reason on partial satisfaction (both for agent's practical and quality desires). We also plan to experiment with concurrent execution of the executive functions, thus better implementing Buehler's conception of the Executive System.

Acknowledgments

G. Averna, M. Cossentino, G. Pilato acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 9 - Green-aware AI, under the NRRP MUR program funded by the NextGenerationEU.

Declaration on Generative AI

During the preparation of this manuscript, the author(s) used Grammarly in order to aid with grammar and spelling. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] B. J. Baars, The global workspace theory of consciousness: Predictions and results, The Blackwell companion to consciousness (2017) 227–242.
[2] D. Buehler, Psychological Agency-Guidance of Visual Attention, Ph.D. thesis, UCLA, 2014.
[3] M. Bratman, Intention and means-end reasoning, The Philosophical Review 90 (1981) 252–265. doi:10.2307/2184441.
[4] M. Bratman, Intention, Plans, and Practical Reason, Harvard University Press, 1987.
[5] D. Buehler, The central executive system, Synthese 195 (2018) 1969–1991.
[6] A. Diamond, Executive functions, Annual review of psychology 64 (2013) 135–168.
[7] M. E. Bratman, D. J. Israel, M. E. Pollack, Plans and resource-bounded practical reasoning, Computational intelligence 4 (1988) 349–355. doi:10.1111/j.1467-8640.1988.tb00284.x.
[8] B. J. Baars, Global workspace theory of consciousness: toward a cognitive neuroscience of human experience, Progress in brain research 150 (2005) 45–53.
[9] C. Massimo, G. Pilato, A. Guido, M. Myrto, J. Mylopoulos, Practical reasoning with attention mechanisms, in: I. Press (Ed.), In proc. of the IEEE SIMPAR Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots, 2025.
[10] W. Huang, A. Chella, A. Cangelosi, A cognitive robotics implementation of global workspace theory for episodic memory interaction with consciousness, IEEE Transactions on Cognitive and Developmental Systems 16 (2023) 266–283.

[11] D. Buehler, Agential capacities: a capacity to guide, Philosophical Studies 179 (2022) 21–47.
[12] W. Wu, We know what attention is!, Trends in Cognitive Sciences 28 (2024) 304–318.
[13] F.-L. Li, J. Horkoff, J. Mylopoulos, R. S. Guizzardi, G. Guizzardi, A. Borgida, L. Liu, Non-functional requirements as qualities, with a spice of ontology, in: IEEE 22nd International Requirements Engineering Conference (RE), 2014, pp. 293–302.
[14] M. Cossentino, G. Pilato, G. Averna, M. Mylopoulos, J. Mylopoulos, The agent metamodel in catalina (cognitive agent practical reasoning architecture), in: In proc. of the 22nd European Conference on Multi-Agent Systems (EUMAS 2025), 2025.
[15] F. E. Ritter, F. Tehranchi, J. D. Oury, Act-r: A cognitive architecture for modeling cognition, Wiley Interdisciplinary Reviews: Cognitive Science 10 (2019) e1488. doi:10.1002/wcs.1488.
[16] J. E. Laird, The Soar cognitive architecture, MIT press, 2019.
[17] S. Franklin, T. Madl, S. D'Mello, J. Snaider, Lida: A systems-level architecture for cognition, emotion, and learning, IEEE Transactions on Autonomous Mental Development 6 (2013) 19–41. doi:10.1109/TAMD.2013.2277589.
[18] R. Bordini, J. Hübner, M. Wooldridge, Programming multi-agent systems in AgentSpeak using Jason, volume 8, Wiley-Interscience, 2007.