Learning in LIDA

Article *in* Cognitive Systems Research · November 2020

DOI: 10.1016/j.cogsys.2020.11.001

2 authors:

Sean Kugele
University of Memphis

Stan Franklin
University of Memphis

Learning In LIDA

Sean Kugele[1] and Stan Franklin
Department of Computer Science and Institute for Intelligent Systems,
University of Memphis, Memphis, TN 38152 USA
skugele@memphis.edu, franklin.stan@gmail.com

Abstract

LIDA is a systems-level, biologically-inspired cognitive architecture. More than a decade of research on LIDA has seen much conceptual work on its learning mechanisms, and resulted in a set of conceptual commitments that constrain those mechanisms; perhaps the most essential of these constraints is the Conscious Learning Hypothesis from Global Workspace Theory, which asserts that all significant learning requires consciousness. Despite these successes, many conceptual challenges remain, and bridging the divide between LIDA's conceptual model and its implementations has been challenging.

The contributions of this paper are threefold: We present a detailed survey of learning in LIDA, during which we clarify, elaborate on, and synthesize together ideas from numerous papers, using updated terminology that reflects the continuing evolution of LIDA. We explore foundational issues in learning, such as, "What must be innate or built-in?" versus "What can be learned?", the nature of LIDA's representations, and the relationship between the LIDA conceptual model and its computational realizations. Finally, we provide a roadmap for future work. We believe that this paper will direct and catalyze our research endeavors, and provide a thorough introduction to the conceptual foundations of LIDA's learning mechanisms that will be useful to anyone that would like a deeper understanding of LIDA or for those that plan to implement LIDA-based agents.

Keywords
Human-like learning; Cognitive architecture; Cognitive model; LIDA model; Bio-inspired computing; Artificial consciousness

[1] Corresponding author. FedEx Institute of Technology rm 410, 365 Innovation Dr., Memphis, Tn, 38152, USA. skugele@memphis.edu

1 Introduction

The Learning Intelligent Decision[2] Agent (LIDA) is a systems-level, biologically-inspired cognitive architecture for creating and modeling minds (see Section 2.1). More than a decade of research on LIDA has resulted in the creation of a wide range of learning mechanisms including attentional, declarative, motivational, procedural, perceptual, spatial, and sensorimotor. These learning mechanisms can differ considerably in their processes and implementations; however, all of them must conform to a set of core principles that we refer to as LIDA's *conceptual commitments* (Franklin et al., 2013).

LIDA differentiates itself from many other biologically inspired cognitive architectures in several ways: First, LIDA implements, and fleshes out, many important aspects of Global Workspace Theory (GWT) (Baars, 1988), a cognitive theory of consciousness. GWT is also the source of one of LIDA's most essential conceptual commitments: The Conscious Learning Hypothesis, which states that no significant learning occurs in the absence of consciousness (see Section 3.1). While the scientific study of consciousness has become more acceptable in recent years, research in machine consciousness and the attempted construction of conscious artifacts (Franklin, 2003) has been largely neglected. Second, LIDA emphasizes the integral role of feelings in supporting many cognitive processes. Contrary to the view of feelings as maladaptive responses to an environment, we adopt the position that feelings are necessary for rapid situational appraisal, efficient action selection, directed attention, and the modulation of learning. The role of feelings as modulators of learning has become particularly important to our research program, which we have elevated to one of our conceptual commitments (see Section 3.2). Finally, LIDA embraces a multitude of distinct long-term memory modules rather than viewing long-term memory as a monolithic and uniform storage media. This multiplicity of memory modules reflects a previously undocumented conceptual commitment in LIDA that different types of knowledge structures (e.g., perceptual, autobiographical, spatial, procedural, etc.) require different representations and are supported by different processes (including learning mechanisms); therefore, they should be modeled as belonging to distinct long-term memory modules. This approach to long-term memory lends a great deal of flexibility to the LIDA architecture at the expense of increased complexity.

Despite our great progress and the maturity of the LIDA conceptual model, LIDA's learning mechanisms remain underspecified and largely unimplemented. A primary goal of this paper is to survey this frontier of learning in LIDA, and to identify directions and opportunities for future exploration. During this journey, we will convey the idiosyncrasies of each learning mechanism, and elaborate on the LIDA conceptual commitments relevant to learning. We will also explore foundational issues that influence learning such as "What must be innate or built-in?" versus "What can be learned?", the nature of LIDA's representations (i.e., data structures), and the relationship between the LIDA conceptual model and its computational realizations (i.e.,

[2] Until recently, LIDA stood for Learning Intelligent Distribution Agent due to its historical origins in the Intelligent Distribution Agent (IDA) project. IDA was a software agent that emulated the role of a Navy detailer, assigning new billets to naval personnel at their end of their tours; this assignment process is called *distribution*.

implementations). Our hope is that this paper will direct and catalyze our research endeavors, and introduce the conceptual foundations of LIDA's learning mechanisms to anyone seeking a deeper understanding of learning in LIDA or that plans to implement LIDA-based agents.

The remainder of this paper is structured as follows: Section 2 provides background on LIDA including our definitions of minds and autonomous agents, and the LIDA conceptual model. Section 3 examines each of LIDA's conceptual commitments that are relevant to learning. Section 4 explores general issues relevant to learning in LIDA. Section 5 examines each of LIDA's learning mechanisms in detail. Section 6 concludes the paper with a retrospective and roadmap for future work.

2 Background

LIDA is both a conceptual and computational model of minds. LIDA draws inspiration from many theoretical and experimental results in psychology and neuroscience about brains; however, LIDA does not model brains, which are but one possible incarnation of minds. In this section, we will provide a brief description of LIDA's conceptual model and related concepts, which will be necessary to understand the remainder of the paper. (Section 4.3 discusses the relationship between LIDA's conceptual and computational models.)

For additional information on LIDA and its components see Franklin et al. (2016).

2.1 Minds and Autonomous Agents

LIDA is a cognitive architecture supporting the creation and modeling of *minds*. In this context, we define minds as control structures (Newell, 1973) for *autonomous agents*, where

> *an* ***autonomous agent*** *is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda, and so as to effect what it senses in the future.* (Franklin & Graesser, 1997)

There are many examples of autonomous agents including human and most non-human animals (i.e., biological systems) and some software agents. A fundamental function of a mind is to answer the continuingly recurring question, "What do I do next?"

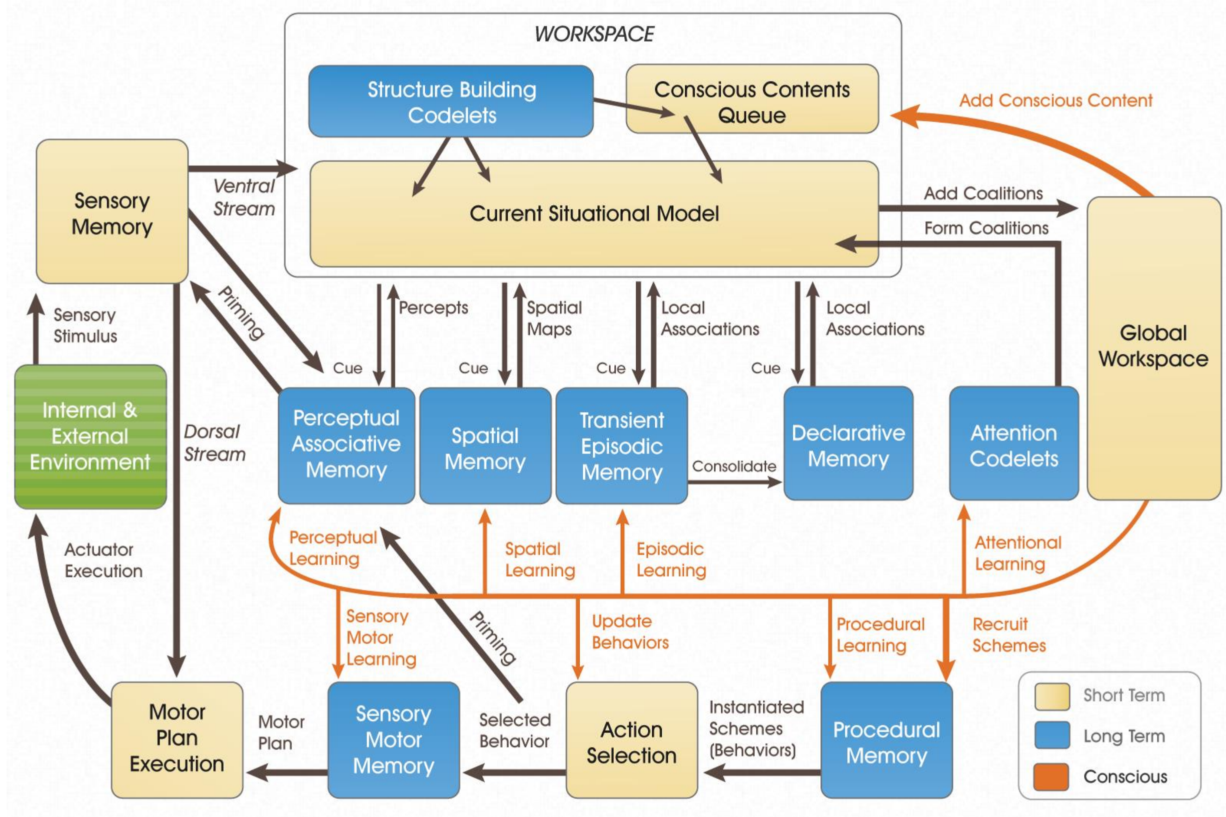

Figure 1 – Diagram of the LIDA conceptual model. The conscious broadcast initiates learning in all long-term memory modules. Figure reproduced from Franklin et al. (2016).

2.2 The LIDA Conceptual Model

LIDA agents are autonomous agents (see Section 2.1) that implement the LIDA conceptual model, which is the subject of this section. The LIDA conceptual model views cognition as emerging from a continual series of potentially overlapping *cognitive cycles* (see Section 2.2.1). Each cognitive cycle relies on a set of underlying functional components, which includes short-term memory (STM) modules (see Section 2.2.2), long-term memory (LTM) modules (see Section 2.2.3), and special-purpose processors called *codelets* (see Section 2.2.4). The *conscious broadcast* is the basis for functional consciousness in LIDA, and it is a necessary and sufficient condition for all learning in LIDA (see Section 3.1). The conscious broadcast is generated by one of LIDA's STM modules called the Global Workspace (see Section 2.2.2), and is received by all LIDA modules.

Figure 1 summarizes LIDA's modules and their interactions. Note that not every LIDA module (or learning mechanism) must be implemented in a LIDA agent for it to be considered a LIDA agent. Depending on the nature of the agent one is trying to model (e.g., insect vs. rodent vs. human cognition), or the research question one is trying to explore (e.g., replicating hippocampal lesion studies), it may be beneficial to focus on a subset of LIDA modules.

2.2.1 The LIDA Cognitive Cycle

Autonomous agents must continually sense, perceive, and act on their environments. This is conceptualized in LIDA as occurring by a series of *cognitive cycles*, each of which is divided into three phases: perception and understanding, attention, and action and learning (see Figure 2). Complex cognitive processes such as planning, deliberation, and problem solving may require multiple cognitive cycles.

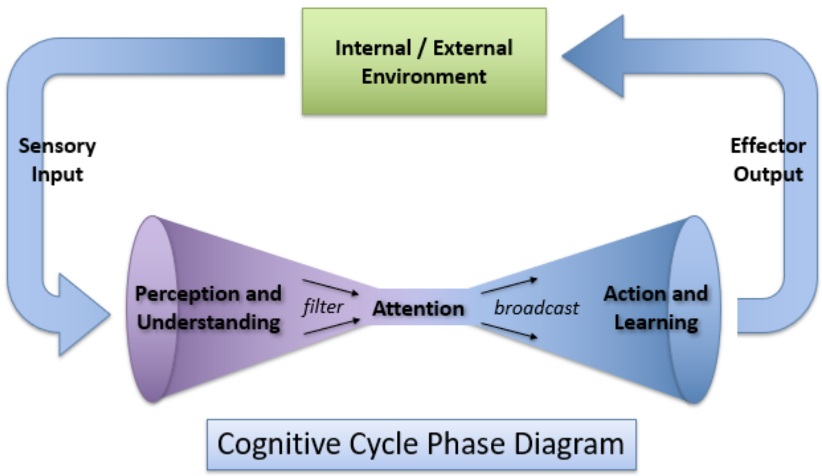

Figure 2 - The Cognitive Cycle Phase Diagram. LIDA views cognition as emerging from a continual series of overlapping cognitive cycles, each of which is divided into three phases: perception and understanding, attention, and action and learning.

2.2.2 Short-Term Memory (STM) Modules

Sensory Memory

Sensory Memory encodes incoming sensory stimuli from the environment into rapidly decaying, modality-specific representations. This sensory content is typically an attenuated version of the original sensory stimuli due to selective feature extraction by *feature detectors*.

Workspace (Preconscious)

The Workspace (preconscious) contains two submodules – the Current Situational Model (CSM) and the Conscious Contents Queue (CCQ) – that support situational understanding and preconscious thought (see Figures 1 and 6).

The CSM is a short-term, high-capacity memory module that integrates sensory content from Sensory Memory with memories from LTM and “preconscious” representations created by structure building codelets (see Section 2.2.4). The cognitive content in the CSM can also *cue* many of LIDA’s LTM modules, allowing the retrieval and integration of additional long-term memories into the CSM.

The CCQ is a short-term, low-capacity memory module that contains the contents of a limited number of recent conscious broadcasts. Each new conscious broadcast is added to the “front” of the CCQ, and previously added content gradually shifts towards the “back” of the CCQ. Despite its name, the CCQ is not a typical queue since structure building codelets have direct access to any position in the queue. (Queues typically do not support random access to their elements.) The CCQ primarily supports the recognition of time-dependent relationships (e.g., causality) and the perception of time (Snaider et al., 2012).

Global Workspace

The Global Workspace supports the filtering and broadcast of the most salient content in the CSM by means of a winner-take-all competition among *coalitions*. Coalitions contain content advocated for by one or more attention codelets (see Section 2.2.4), and represent a temporary alliance between attention codelets for the purpose of jointly endorsing content. Coalitions have a single parameter, activation, that is a function of many factors (see Franklin et al. (2016) for details) including the content’s activation and the attention codelets’ base-level activation. The coalition with the highest activation wins the competition, and that coalition’s content is broadcast to all LIDA modules. We refer to this event as the global (or conscious) broadcast.

Action Selection

Action Selection selects a behavior from a set of candidate behaviors (i.e., instantiated schemes (see Procedural Memory in Section 2.2.3)), and sends it to several modules including the Sensory Motor System (see Figure 5) for execution. Some of the factors that may influence a candidate behavior’s likelihood of selection include:

(1) The relevance of the behavior to the current situation; i.e., its *current activation*.
(2) The reliability of the behavior in increasing the likelihood of its result; i.e., its *base-level activation*.
(3) The desirability of the behavior’s result; i.e., its *total incentive salience*.
(4) The preconditions that must be satisfied before the behavior’s action is executable.
(5) The consequences of the result should the behavior’s action be executed.

LIDA supports four modes of Action Selection: volitional (i.e., deliberative), consciously-mediated, automatized, and via alarms.

Motor Plan Execution

Motor Plan Execution supervises the execution of a motor plan, which results in the selection of a series of motor commands to be executed by a LIDA agent’s actuators. Motor plan execution occurs outside of a LIDA agent’s conscious awareness, though the agent may become aware of its low-level actions indirectly by monitoring incoming sensory content.

The design of the Motor Plan Execution module was heavily influenced by the Subsumption Architecture (Brooks, 1986). Following the instantiation of a motor plan from a motor plan template (MPT) in Sensory Motor Memory (see Section 2.2.3), Motor Plan Execution operates in a purely reactive fashion (similar to the Subsumption Architecture) using a process referred to as *online control* (see Figure 3). During online control, one or more motor commands from the executing motor plan are “triggered” based on incoming sensory content from Sensory Memory over the dorsal stream (see Figure 1). The motor plan’s *choice function* chooses one of these motor commands to execute and sends the *command value* of the selected motor command to the designated actuator.

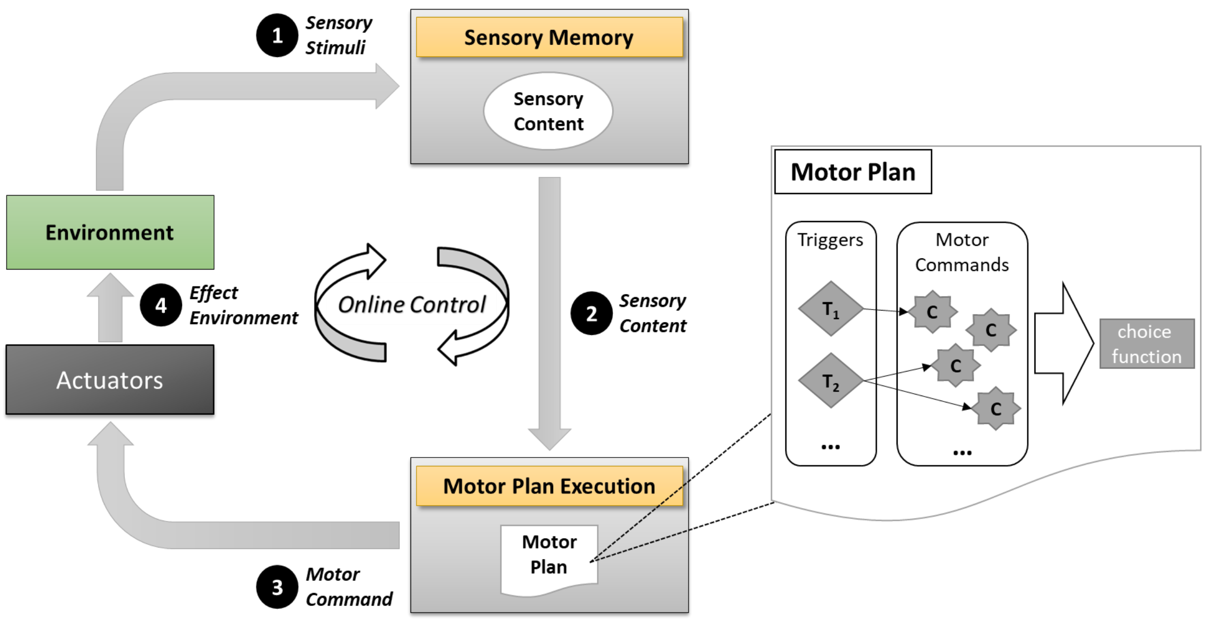

Figure 3 - Motor Plan Execution’s Online Control process. Sensory stimuli from the environment is used by Online Control to “trigger” and “choose” appropriate motor commands and update motor command values that are sent to the agent’s actuators for execution.

2.2.3 Long-Term Memory (LTM) Modules

LIDA’s long-term memory modules support a diverse set of representations (i.e., data structures) and processes. Many LTM modules can be *cued* by content in the CSM, allowing the formation of “preconscious” representations: memories and percepts of which an agent can become consciously aware. Other LTM modules, such as Procedural Memory and Sensory Motor Memory, are not cueable, so an agent can never become consciously aware of those representations. (See Franklin & Baars (2010) for additional information regarding varieties of unconscious processes.)

Perceptual Associative Memory (PAM)

Perceptual Associative Memory contains representations, referred to conceptually as *nodes* and *links*[3], that support LIDA’s ability to quickly (i.e., non-deliberatively) recognize objects, events, entities, concepts, etc., and relationships between them (for example, “X *is above* Y” or “X *is larger than* Y”). All PAM nodes have a *base-level activation* that quantifies the relative, empirical frequency of their corresponding objects, events, entities, concepts, etc., occurring in the agent’s environment. PAM nodes can be activated (i.e., have their current activations increased)

---

[3] Links are typically *directed* in LIDA, indicating an asymmetric relationship between a *source* node and a *sink* node. Activation propagates across links from source to sink.

either by feature detectors or by the propagation of activation from other “linked” nodes. PAM nodes that have a total activation (i.e., current + base-level activation) above an activation threshold will be integrated into the CSM as a *percept*.

Declarative Memory

Declarative Memory contains autobiographical and semantic memories. Autobiographical memories are referred to as *episodes*. Semantic memories, which correspond to facts and rules, are a form of degenerate episode in which the autobiographical contexts (e.g., place, time, feelings) have been interfered with, as well as decayed, leaving only detached factual content.

Transient Episodic Memory (TEM)

Transient Episodic Memory contains recent episodes for a short duration of time (typically on the order of hours to a day or so). Episodes are added to TEM from the conscious broadcast. Episodes that have not decayed away will be transferred to Declarative Memory during offline consolidation.

TEM serves two purposes: It supports the modeling of a Declarative Memory consolidation process (Section 3.4), and it reduces interference between recent and more remote (i.e., consolidated) episodes that are highly similar.

Spatial Memory

Spatial Memory contains *cognitive maps*, which are allocentric, topographic representations depicting the location of objects within an agent’s environment. They rely on a special class of PAM nodes called *place nodes*[4], which represent specific locations in the environment. Cognitive maps associate (i.e., “link”) place nodes with object nodes to encode the relative locations of those objects within the environment (see Figure 4). A cognitive map can encode an agent’s own location by associating a “self” PAM node with a place node. Cognitive maps are typically hierarchically organized: a single geographic region may be represented by many cognitive maps (concurrently) with different scales and resolutions. Cognitive maps support agent localization and route planning, among other things.

Attention Codelet Memory (Attentional Memory)

Attention Codelet Memory supports the attention codelet lifecycle: creation, parameter maintenance, and termination.

Structure Building Codelet Memory

Structure Building Codelet Memory supports the structure building codelet lifecycle: creation, parameter maintenance, and termination.

---

[4] Place nodes are inspired from “hippocampal place cells” that occur in animal brains, which are believed to encode an animal’s belief about its current spatial location.

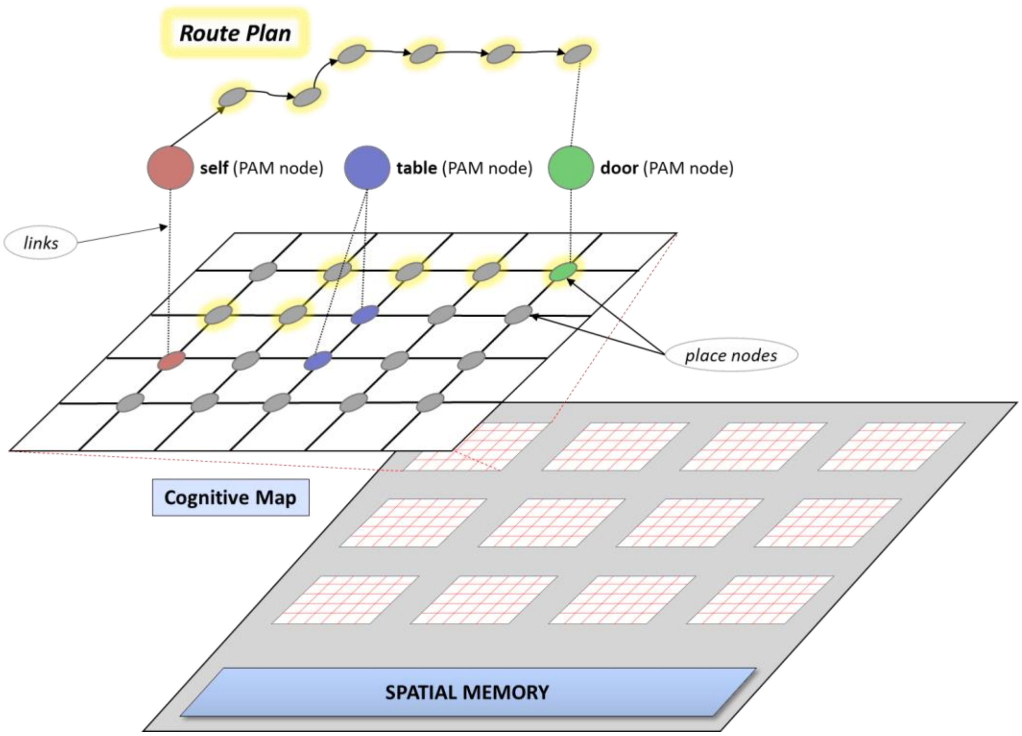

Figure 4 - A conceptual depiction of Spatial Memory, including an expanded view of a cognitive map. Place nodes are PAM nodes that represent locations in an agent's environment. PAM object nodes (in this case, table and door) are associated with ("linked to") place nodes to represent their location within a cognitive map. A special PAM "self" node can be associated with a place node to indicate an agent's location. A "route plan" is depicted showing how cognitive maps support route determination via a series of place nodes. Figure reproduced (with modification) from Madl et al. (2018).

Procedural Memory

Procedural Memory contains representations called *schemes*[5], which encode correlations between contexts (i.e., states), actions, and their results. A *base-level activation* is associated with every scheme that quantifies its *reliability*; that is, how likely its action is to obtain its result when taken from its context.

When Procedural Memory receives a conscious broadcast, it activates schemes based on their *relevance* to the contents of that broadcast. Procedural Memory determines a scheme's relevance by comparing the context and result of each scheme to the received conscious content, and updates the scheme's *current activation* accordingly. The greater the similarity between a scheme's context and result and the contents of the conscious broadcast the greater the increase in its current activation. Schemes with current activations above a configurable threshold will be instantiated as *candidate behaviors*.

Instantiation can be viewed as the process of creating a specific instance (behavior) from an abstract template (scheme). During instantiation, unbound variables in a scheme's context,

---

[5] Inspired by Drescher (1991). Drescher referred to these tripartite data structures as *schemas*.

action[6], and result are specified in accordance with the contents of the current conscious broadcast. If a scheme does not specify a context (which will generally be the case for an agent's initial "action only" schemes), then Procedural Memory will supply a new context to that candidate behavior based on the contents of the conscious broadcast that initiated the behavior's instantiation.

Sensory Motor Memory (SMM)

Sensory Motor Memory contains *motor plan templates* (MPTs) that specify abstract motor plans. SMM selects a single MPT, based on the *selected behavior* from Action Selection, and instantiates the selected MPT into a concrete *motor plan*. Once instantiated, the motor plan is sent to Motor Plan Execution. The instantiation of MPTs into motor plans is analogous to the instantiation of schemes into candidate behaviors (see Procedural Memory) though the data structures are very different. SMM and Motor Plan Execution have closely aligned concerns; therefore, we frequently refer to them collectively as the Sensory Motor System (SMS) (see Figure 5).

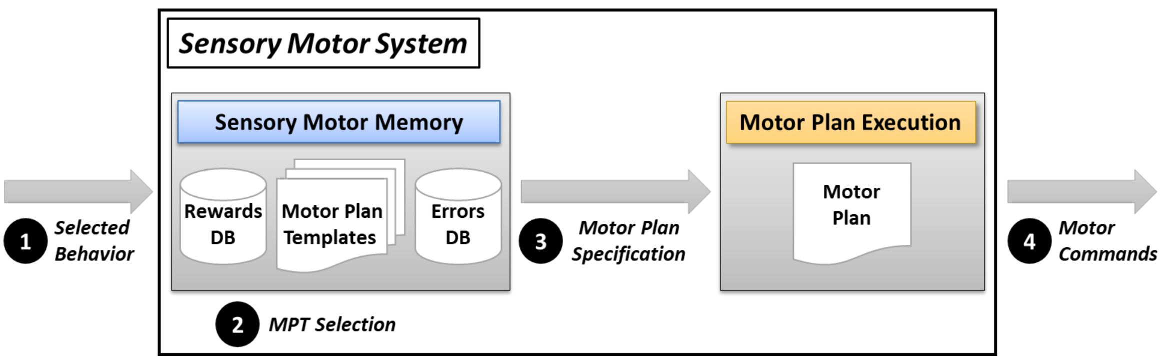

Figure 5 - The Sensory Motor System, illustrating the Motor Plan Template (MPT) Selection and Motor Plan Specification processes. The Sensory Motor System receives a selected behavior from Action Selection and outputs a series of motor commands that are directly executable by an agent's actuators.

2.2.4 Codelets

Codelets are special purpose processors that perform various (internal) operations based on the presence of content of interest in the Workspace (preconscious). There are two types: attention codelets and structure building codelets. Each type of codelet belongs to a supporting LTM module that manages their lifecycles and implements their selectionist and instructionalist learning mechanisms (see Figure 6).

Attention Codelets

Attention codelets monitor the CSM looking for salient content of interest (based on their individual concerns). If content is found that matches an attention codelet's concerns, it sends this selected content to a coalition forming process. The coalition forming process creates a new

---

[6] Actions may have unbound variables that qualify and modulate how an action is executed; for example, "walk rapidly" and "walk slowly" could be implemented in a single scheme with an action of the form "walk <how fast>", where "<how fast>" modulates the walking speed.

coalition that includes this content, and sends it to the Global Workspace to compete for the conscious broadcast.

Attention codelets vary in the kinds of content they consider salient. *Specific attention codelets* advocate for a narrow range of content; for example, specific types of objects or events. By contrast, the *default attention codelet* advocates for a wide range of content: its selection criteria are based solely on the content's total activation and total incentive salience. *Expectation codelets* are attention codelets created in response to selected behaviors (see Action Selection in Section 2.2.2) that advocate for content in the CSM corresponding to the expected results (or non-results) of that selected behavior. For example, if an agent is executing a selected behavior for "flipping a light switch" that includes an expected result of "light on" then an expectation codelet may temporarily bias the agent's attention towards content in the CSM representing the state of the light bulb (e.g., "light on" or "light off").

Structure Building Codelets (SBCs)

Structure Building Codelets monitor the CSM and CCQ looking for content of interest (based on their individual concerns). If content is found that matches a structure building codelet's concerns, it creates and/or modifies representations in the CSM.

Structure building codelets support an agent's ability to interpret its current situation by generating or manipulating "preconscious" content: for example, by creating relational/associative links (e.g., categorical or causality relationships) or entirely new representational structures, such as mental simulations (i.e., preconscious mental images).

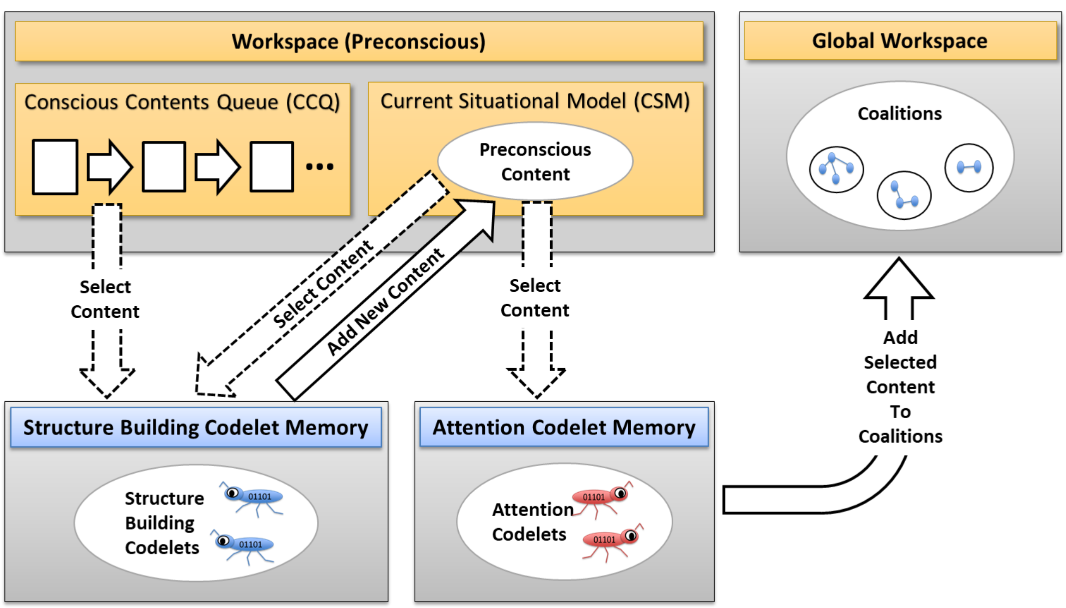

Figure 6 - Codelets and their relationship to the (preconscious) Workspace and Global Workspace. Structure building codelets select content from the CCQ and CSM, and add new content to the CSM. Attention codelets select content from the CSM and add their selected content to coalitions in the Global Workspace (via a coalition forming process).

3 Learning - Conceptual Commitments

Franklin et al. (2013) proposed a set of *conceptual commitments* that made explicit LIDA's commitment to various theoretical principles. These commitments constrain all of LIDA's conceptual processes and their implementations. In the remainder of this section, we will examine the conceptual commitments that are pertinent to learning.

3.1 Conscious Learning Hypothesis

LIDA adheres to the Conscious Learning Hypothesis of GWT, which states that all *significant* learning requires consciousness. LIDA and GWT further contend that learning occurs as a direct result of the conscious broadcast; therefore, consciousness is both a *necessary* and *sufficient* condition for significant learning. This causal relationship between conscious broadcasts and learning does not preclude the possibility of a substantial delay between the onset of a conscious broadcast and subsequent learning based on that conscious content; for example, as would occur during offline consolidation in support of declarative learning. It's also important to clarify that the module-specific learning mechanisms are themselves unconscious: an agent is never aware of the internal processes responsible for learning, only the resulting side-effects (e.g., access to new memories and more reliable motor skills).

3.2 Feelings as Motivators and Modulators of Learning

Feelings can influence a LIDA agent's cognitive processes in at least three ways. They can increase the overall salience of preconscious cognitive content, making it more likely to come to consciousness. They can bias an agent's action selection towards behaviors that lead to desirable outcomes, and they can modulate learning and the retention of long-term memories.

LIDA implements feelings via PAM nodes with *affective valence*, which we call *feeling nodes*. Affective valence quantifies an agent's immediate hedonic response to events[7] (e.g., eating ice cream), indicating liking (positive affective valence) or disliking (negative affective valence). For example, a hunger feeling node would have negative affective valence, whereas a satiety feeling node would have positive affective valence. The magnitude of the affective valence is conveyed by the *total activation* of a feeling node, and the direction (like or dislike) is conveyed by its valence sign (either $+$ or $-$). Feeling nodes, like all PAM nodes, can be activated either by feature detectors in Sensory Memory (e.g., the presence of sugars in foods) or by *cueing* from the CSM (e.g., recognizing that you have a straight flush in poker).

We distinguish two types of feeling nodes: *drive feeling nodes* and *interpretative feeling nodes*. Drive feeling nodes reflect an agent's essential bodily needs (e.g., food, water, oxygen) and are typically activated by homeostatic receptors (specialized internal sensors that capture aspects

---

[7] In this paper, we will typically refer to *events* as the "targets" of LIDA's motivational constructs; for example, the things that are liked or wanted. While we contend that events are the most common representational type associated with an agent's motivations, we don't exclude the possibility that objects, concepts, etc. could have associated feelings, and attractive or repulsive force.

of an agent's bodily state). Hunger and satiety are examples of drive feelings. *Interpretative feeling nodes*, by contrast, provide affective appraisals of environmental stimuli that can serve as heuristics for operating in an environment. The positive and negative feelings typically associated with eating sweet and bitter foods are examples of interpretative feelings. Interpretative feelings may indirectly support an agent's drives; for example, liking to eat sweet foods may bias the agent towards eating ripe fruit, resulting in increased satiety and decreased hunger. Interpretative feelings are only adaptive, however, if the interpretations they provide remain beneficial within the agent's environment. If the environment changes, those feelings could become maladaptive; for example, if sweetness were frequently correlated with poisonous substances or unhealthy foods. Drive feelings, on the other hand, will remain adaptive in any external environment so long as the agent's bodily needs remain constant.

Kringelbach and Berridge (2009) argued that liking and disliking should be distinguished from wanting and dreading, and that they are implemented in brains using distinct neural pathways – the former indicates an immediate hedonic response to an event while the latter is an attractive or repulsive force associated with an event. Wanting and dreading are quantified in LIDA by an event's *incentive salience*. Incentive salience is further divided into a *base-level incentive salience* and *current incentive salience*. *Base-level incentive salience* is a context-invariant attraction or repulsion to an event, which is learned from repeated exposure to that event. *Current incentive salience* is a context-sensitive attraction or repulsion associated with an event that is modulated by the agent's current situation.

Events are complex structures that are central to LIDA's implementation of feelings and motivations. Each event contains nodes for the actor(s), action(s), object(s), etc. that comprise that event. Bi-directional "activation" links associate each node with its parent *event node*. These links allow top-down and bottom-up flow of activation, and represent the *thematic roles* (Fillmore, 1968) of each of the nodes within the event's node structure. Figure 7 illustrates event node structures and LIDA's motivational constructs in detail.

In general, learning (of all varieties) is facilitated by additional incentive salience (or activation); however, this only holds up to a point. Too much arousal may impair both learning and cognitive performance. This notion is captured in the Yerkes-Dodson law (Yerkes & Dodson, 1908), which postulates an "inverted U" relationship between arousal and learning.

3.3 Profligate Learning, Comprehensive Decay, and Forgetting

All LIDA long-term memory modules support learning, typically with every cognitive cycle. This eager and opportunistic learning strategy would rapidly result in a proliferation of representations if it were not counter-balanced by an equally relentless and comprehensive decay strategy: all parameters decay, and irrelevant and ineffectual representations are pruned (i.e., forgotten). We refer to this Darwinian-style of "generate and test" learning as *profligate learning* since most representations are created and then discarded with seemingly reckless abandon. Decay and forgetting processes are required for all LIDA modules, and they are a necessary companion to LIDA's learning processes.

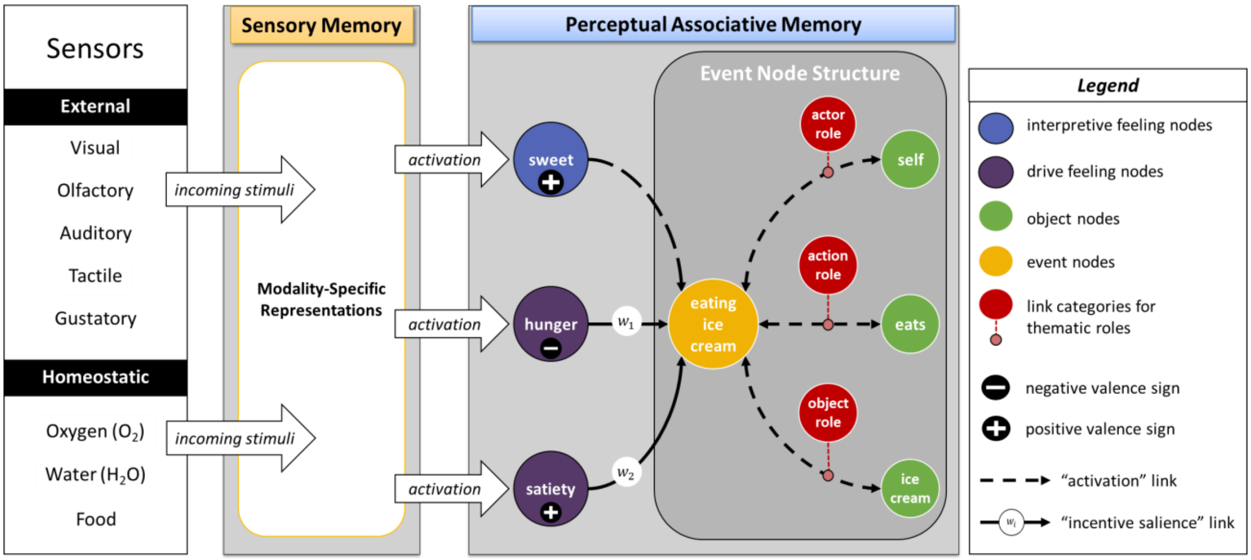

Figure 7 – Illustration of LIDA's motivational constructs. Feeling nodes are PAM nodes with either positive or negative affective valence (e.g., sweetness, hunger, and satiety). Feeling nodes can be activated by incoming sensory content (for example, from external sensors or homeostatic sensors) or cued from content in the CSM (not depicted). A feeling node's activation combines with the node's positive or negative valence sign to give the magnitude and "direction" (i.e., like or dislike) of a stimulus response. Current activation is passed from feeling nodes to event nodes over "activation links". Current incentive salience is passed from drive nodes to event nodes over "incentive salience" links. Unlike "activation" links, "incentive salience" links have associated weights that modulate the degree and direction of the increase in current incentive salience. Event node structures contain a set of associated nodes that are connected to the event node by links representing "thematic roles", such as the "actor" and "action" characterizing an event. In this example, the LIDA agent is the actor in the "eating ice cream" event; therefore, the "self" node has been associated with the event using a link indicating the actor thematic role.

3.4 Transient Episodic Memory and Consolidation

The long-term retention of episodic and semantic memories requires consolidation. This assertion, while not universally accepted, is well supported by a substantial body of evidence (Born & Wagner, 2006; McGaugh, 2000; Nadel, Hupbach, Gomez, & Newman-Smith, 2012; Stickgold & Walker, 2005).

LIDA's transient episodic memory module supports the temporary retention of episodic memories between offline consolidation events. Consolidation is hypothesized to occur during sleep in humans; therefore, we typical consider TEM retention periods in the range of a few hours to a day. Ramamurthy and Franklin (2011) proposed ideas on how a consolidation process could be implemented using Sparse Distributed Memory; however, the specifics of the consolidation mechanism, including the triggers that might initiate consolidation in a software agent, are still an open area of research.

4 General Concerns for Learning

4.1 A Tabula Rasa?

Nature versus Nurture has been a topic of perennial debate among philosophers and psychologists for centuries. While we don't have any intention of resolving that debate here, a paper on learning would be remiss without addressing “What *must* be innate or built-in?” and “What *can* be learned?” by a LIDA agent.

All LIDA agents *must* be endowed with the following:

(1) Intrinsic Motivations

Evolution provides biological agents with a set of intrinsic motivations that tend to increase their probability of survival and reproduction. Similarly, all LIDA agents begin life with built-in feelings including homeostatic drives (i.e., drive feeling nodes) and interpretive affective responses (i.e., interpretative feeling nodes) that can serve as heuristics for operating in an environment (see Section 3.2). All additional motivations, such as wants and dreads, are ultimately derived from this built-in set of intrinsic motivations (see Section 5.3).

(2) Primitive schemes and motor plans

All LIDA agents require innate or built-in procedural knowledge that minimally includes a set of primitive (action only) schemes. These initial, primitive schemes, and their corresponding motor plans, support basic action selection and execution (e.g., “move eyes left” or “reach right arm forward”), and form the basis for an agent’s early environmental exploration.

(3) Default attentional processes

All LIDA agents require a minimal, built-in attentional mechanism that can discriminate and filter incoming sensory stimuli, and subserve the process of bringing cognitive content to consciousness. This could be implemented, for example, using a single default attention codelet and/or a set of specific concern attention codelets that respond to salient stimuli from the agent’s sensors (e.g., bright lights, loud sounds, etc.).

(4) Primitive feature detectors

There is a growing body of experimental evidence that humans (Turati et al., 2006; Simion et al., 2008) and other animals (Wood, 2013) are born with a set of innate primitive feature detectors that gradually become more sophisticated and attuned to the physical world through experience. In LIDA, these primitive feature detectors correspond to nodes in PAM that are activated directly from modality-specific representations in Sensory Memory and the Current Situational Model (as opposed to receiving activation from other PAM nodes). An example of a primitive feature detector would be a node that is receptive to (i.e., activated by) colors in the red spectrum of light.

It is technically possible for an autonomous agent to learn all of its feature detectors directly from experience, for example by applying a convolutional neural network

(Krizhevsky et al., 2012); therefore, we do not rule out the possibility that a LIDA agent could also learn all of its primitive feature detectors directly from online, conscious experience. That being said, current machine learning approaches typically require a massive amount of data (i.e., experiences) to learn useful features, so we suspect that many designers of LIDA agents will choose to supply their agents with a set of built-in primitive feature detectors since a fully automated approach may be infeasible in practice.

(5) Learning mechanisms

All LIDA agents require a set of innate or built-in, module-specific learning mechanisms. The parameters of these learning mechanisms, such as learning rates, may change over time, but the basic processes must be provided by a designer, evolutionary mechanism, etc.

Given this modest endowment of initial competencies, a LIDA agent can learn essentially everything else. This includes (but is not limited to) the recognition of new objects, events, and concepts, new autobiographical memories, new facts, new schemes and motor plans, new allocentric and egocentric cognitive maps, new attentional and creative processes, and new affective associations. Whether a LIDA agent *should* be designed to learn these things is another matter: learning necessitates exploration, which involves costs and risks – great or small. The nature of an agent's environment (dynamic versus static; hostile versus hospitable) and the feasibility of engineering built-in knowledge and competencies will need to be considered on an agent by agent basis.

4.2 The Nature of LIDA's Representations

LIDA is committed to the principles of embodied (situated) cognition, which we interpret as a structural coupling between autonomous agents and their environments (Franklin et al., 2013). We have consistently been proponents of grounded representations throughout LIDA's development (Ramamurthy et al., 2006; McCall et al., 2010a; Franklin et al. 2014; Agrawal et al., 2018), and have advocated for the use of modal (perceptual) symbols (Barsalou, 1999). Apart from its commitment to symbol grounding, LIDA is representation agnostic.

Traditionally, we have conceptualized (and implemented) LIDA's "cognitive content" (i.e., knowledge representations) as directed graphs called *node structures*. Node structures contain one or more *nodes* and zero or more *links*. Nodes refer to objects, entities, events, concepts, etc., and links indicate relationships between the nodes' referents (e.g., "X is a Y", "X has a Y", "X caused Y").

Node structures are conceptually appealing because they are easily visualized and new associations are trivial to create; however, they suffer from many computational limitations (Snaider & Franklin, 2014b). These limitations have inspired exploration into more scalable representational options, such as, Modular Composite Representation (MCR) vectors (Snaider

& Franklin, 2014a; 2014b) and sparse distributed representations from Hierarchical Temporal Memory (HTM) regions (Agrawal et al., 2018). For the remainder of this paper, we will generally assume node structures, as they are conceptually simpler to describe. Any deviations from this convention will be called out in the text.

4.3 The LIDA Conceptual Model and Its Implementations

In Sections 2 and 3, we described the LIDA conceptual model and its commitments. The conceptual model defines LIDA's high-level components and their interactions, and the conceptual commitments provide a theoretical framework within which these components *must* operate. LIDA's computational instantiations (software implementations) are constrained to abide by both the conceptual model and its commitments; however, within those confines, there are many details that are *necessarily* unspecified (or underspecified). Intrepid creators of artificial minds are empowered to explore within this uncharted territory and choose appropriate algorithms and module implementations that facilitate their efforts.

We say that some details are necessarily unspecified for at least two reasons. First, an agent may be situated in any of an enormous variety of environments, each with its own character and relative complexity. This environmental sensitivity will likely flavor all module implementations, and can be interpreted as a direct consequence of embodiment principles: *all aspects of cognition are shaped by the agent's situated relationship with its environment*. This can be illustrated by considering Sensory Memory: modality-specific representations and feature detectors that work well for a text-based environment (for example, a "web crawling LIDA agent") would be abysmal for processing visual images from a camera sensor. This is equally true of LIDA's learning mechanisms. Second, we must acknowledge that our knowledge of minds is imperfect. Biological minds provide our only known examples – the proof of concept – that autonomous agents with "general intelligence" are constructible; however, our knowledge of how they accomplish this feat is woefully incomplete. The LIDA conceptual model and its commitments are a testament to what we believe we know about minds based on all available evidence; however, the computational implementations of those conceptual constructs are subject to change without invalidating the LIDA conceptual model. Therefore, we must stress that any computational mechanism of a module should be considered *an* implementation not *the* implementation of a module, and it is entirely consistent (and eminently useful) to admit many implementations, so long as they are consistent with LIDA's conceptual foundations.

5 Learning Mechanisms

Every LIDA long-term memory (LTM) module supports both *instructionalist* and *selectionist* learning (Edelman, 1987). Instructionalist learning results in new representations in LTM modules. Selectionist learning results in adjustments to parameters associated with preexisting LTM representations. Each LTM module may support different data structures (i.e., representations) with different functional purposes and associated parameters; therefore, there are many varieties of instructionalist and selectionist learning in LIDA. Table 1 presents a partial list of types of learning supported in LIDA including their associated LTM modules and their instructionalist and selectionist flavors.

The remainder of this section will be devoted to elaborating on each learning type and their associated learning mechanisms. For the purposes of this paper, we do not consider the storage or updating of representations in short-term memory (STM) modules as *significant* learning; however, STM modules play a critical role in supporting all LTM learning mechanisms.

Learning Type	**Associated LTM module(s)**	**Instructionalist Learning**	**Selectionist Learning**
Attentional	Attention Codelet Memory (Attentional Memory)	New Attention Codelets	Update Attention Codelet Base-Level Activation
Declarative	Declarative Memory	New (i.e., consolidated) Autobiographical Memories (episodes)	Update Base-Level Activation of Episodes in Declarative Memory (autobiographical and semantic memories)
Motivational	Perceptual Associative Memory (PAM)	New Incentive Salience Links	Update Base-Level Incentive Salience of PAM Nodes
Perceptual	Perceptual Associative Memory (PAM)	New object, event, entity, concept, etc., node structures (PAM)	Update Base-Level Activation of PAM Nodes
Procedural	Procedural Memory	New Schemes	Update Scheme Base-Level Activation
Sensorimotor	Sensory Motor Memory	New Motor Plan Templates (MPTs)	Update MPT Rewards and Errors databases. Update Rewards Learning Rate
Spatial	Spatial Memory	New Cognitive Maps	Update Base-Level Activations of Cognitive Maps

Table 1 - Instructionalist and selectionist learning by type with associated modules.

5.1 Attentional Learning

Attentional learning involves learning how to identify the most salient portions of the CSM and ensure that the likelihood of that content becoming conscious is proportional to the current saliency of the content.

Instructionalist learning occurs in at least two scenarios. In the first scenario, the Attention Codelet Memory module creates a new *specific attention codelet* whenever content advocated for by the *default attention codelet* (see Section 2.2.4) becomes part of a winning coalition. The "concern" of this new *specific attention codelet* will be to advocate for content that is similar to the winning conscious content, and its base-level activation will be a function of the base-level activation of the *default attention codelet* and the activation of the coalition containing the content. In the second scenario, the Attentional Memory module creates an *expectation codelet* (D'Mello et al., 2006a) whenever a selected behavior is received from the Action Selection module. The "concern" of this new *expectation codelet* will be to advocate for content based on the expected result of the selected behavior; for example, surprising, or unpredicted, outcomes that present opportunities to learn something new. Its base-level activation will be a function of the base-level activation and base-level incentive salience of this expected result.

Selectionist learning occurs when any attention codelet successfully advocates for cognitive content that becomes conscious (i.e., is part of a winning coalition). The Attention Codelet Memory module will increase the base-level activation of any attention codelet that was responsible for suggesting content in the winning coalition. The base-level activation of attention codelets is a factor that influences (positively biases) the activation of coalitions in the Global Workspace.

Due to LIDA's adherence to the conscious learning hypothesis, all learning mechanisms are heavily dependent on the efficacy of attentional learning; no significant learning occurs unless the agent is conscious of that content.

For more information on attentional learning, including computational experiments, see Faghihi et al. (2012).

5.2 Declarative Learning

Declarative learning involves the creation and reinforcement of autobiographical memories (episodes) and semantic memories (facts and rules). It requires the involvement of Transient Episodic Memory (TEM) as a temporary store of recent episodes, and an offline consolidation process that is responsible for the creation and reinforcement of episodes within Declarative Memory.

New episodes are added to Transient Episodic Memory (TEM) from content in the conscious broadcast. If content from the conscious broadcast already exists in TEM, then its base-level activation will be reinforced. Content that is not reinforced in TEM will decay and eventually be pruned from TEM when it crosses below a minimum base-level activation threshold. Any episodes remaining in TEM when an offline consolidation event occurs will be "consolidated" into Declarative Memory. During consolidation, any content from these episodes that did not exist in Declarative Memory will be added (instructionalist learning). Content that did exist will have its base-level activation reinforced (selectionist learning).

Note that *portions* of an episode can be reinforced. For example, a LIDA agent may hear that "Paris is the capital of France" in dozens of different episodes, each occurring in different places, at different times, and involving different people. The only content being reinforced is at the intersection of these episodes: “Paris is the capital of France”. All autobiographical context (place, time, feelings) from the individual episodes is interfered with and decays away, eventually leaving a context-agnostic, informational nugget; that is, a semantic memory.

5.3 Motivational Learning

Recall from Section 4.1 that a LIDA agent starts life with a set of built-in, intrinsic motivations in the form of feeling nodes (PAM nodes with affective valence). These feeling nodes specify an agent’s drives and hedonic interpretations of its environment. *LIDA agents do not learn new feeling nodes*; however, from this built-in set of feeling nodes, LIDA agents can learn a rich set of *wants* and *dreads* (see Section 3.2) that direct an agent’s attention, influence action selection, and modulate all forms of learning. Wanting and dreading is quantified in LIDA by a node’s total incentive salience, which is a function of its base-level incentive salience and its current incentive salience. Motivational learning involves updating base-level incentive saliences, and creating incentive salience links that serve as conduits for current incentive salience.

Updating Base-Level Incentive Salience

Base-level incentive salience is updated by PAM when a conscious broadcast contains event nodes with associated feeling nodes. For example, suppose that our agent is eating vanilla ice cream. The agent’s “tongue” (i.e., gustatory sensor) receives incoming sensory stimuli corresponding to the ice cream. These sensory stimuli may result in the activation of a “sweet” feeling node in PAM, which PAM instantiates into the CSM. Structure building codelets (SBCs) monitoring the CSM may create an event node structure for this “eating ice cream” event, and an “activation” link between the instantiated “sweet” feeling node and the “eating ice cream” event node. If this event node structure (which is now augmented with an activated “sweet” feeling node) comes to consciousness, PAM will receive it and update the base-level incentive salience associated with the “eating ice cream” event node. In this case, only one feeling node is associated with the event; however, in general, base-level incentive salience updates will be a function of the affective valences of all feeling nodes associated with an event.

Creating Incentive Salience Links

Current incentive salience, which quantifies the context-sensitive attraction or repulsion associated with an event in a given situation, is transmitted from drive feeling nodes to event nodes over incentive salience links (McCall et al., 2020, p.17). Recall from Section 3.2 that each feeling node has an affective valence, $v = a_t \operatorname{sgn}(v)$, where a_t is the total activation of the feeling node at time t and $\operatorname{sgn}(v)$ denotes its valence sign (i.e., the “direction” of like or dislike). Also recall that each incentive salience link has an associated weight w. The current incentive salience contributed by a feeling node to a linked event (over an incentive salience link) is defined as

$$\text{current incentive salience} \equiv vw = a_t \operatorname{sgn}(v) w$$

where

$$\operatorname{sgn}(v) = \begin{cases} -1 \text{ if } v < 0, \\ 0 \text{ if } v = 0, \\ 1 \text{ if } v > 0. \end{cases}$$

SBCs create incentive salience links based on changes in a feeling node's activation *between conscious broadcasts*. Recall that conscious broadcasts are stored temporarily in the CCQ (see Section 2.2.2), and are accessible to SBCs. If an SBC (that is interested in such things) notices an increase in the activation of a feeling node between conscious broadcasts, then it will create an incentive salience link with a positive weight; otherwise, if the activation decreases between broadcasts, the link will have a negative weight. In general, for any incentive salience link, the weight w will be given by

$$w = \operatorname{sgn}(v) f(\Delta a),$$

where $\Delta a = a_{t+1} - a_t$ is the change in activation between conscious broadcasts at time t and $t + 1$, and $f: \mathbb{R} \to [0,1]$ is a function that scales Δa (for example, a sigmoid function). Intuitively, the weights will be positive for events that lead to beneficial changes in an agent's homeostatic state, and negative if they lead to detrimental changes.

For more information on motivational learning and LIDA's motivational system, including computational experiments, see McCall et al. (2020).

5.4 Perceptual Learning

The primary goal of perceptual learning is to learn grounded, potentially multi-modal and hierarchically-organized, representations in PAM (see Section 2.2.3). These representations support an agent's ability to recognize objects, events, entities, and concepts, as well as, the relationships between them. Additional conscious experiences with previously learned PAM representations result in an increase in the base-level activations of those representations. This serves the dual purposes of persisting those representations in PAM (i.e., reducing their likelihood of being pruned) and providing an empirical measure of their relative frequencies of occurrence.

LIDA's perceptual learning mechanisms were heavily inspired by Perceptual Symbol Systems (Barsalou, 1999). Perceptual symbols, which capture a subset of an agent's sensory and perceptual state while sensing objects and events, are formed in LIDA's *perceptual scene*: a multi-modal data structure within the CSM. Modality-specific representations from Sensory Memory's *sensory scene* are integrated into the perceptual scene along with any percepts from PAM that may have been sufficiently activated by the incoming sensory content. "Virtual" content created by structure building codelets (e.g., preconscious mental simulations) may also be present in the perceptual scene. During the perceptual symbol formation process, structure building codelets create new nodes (referred to in McCall et al. (2010b) as "blank nodes") that serve as anchoring points around which representations comprising the new perceptual symbols will organize. Incoming sensory content for a given object or event, as well as, percepts and cued long-term memories (e.g., episodes) may be linked to these new ("blank") nodes. (See Figure 8.)

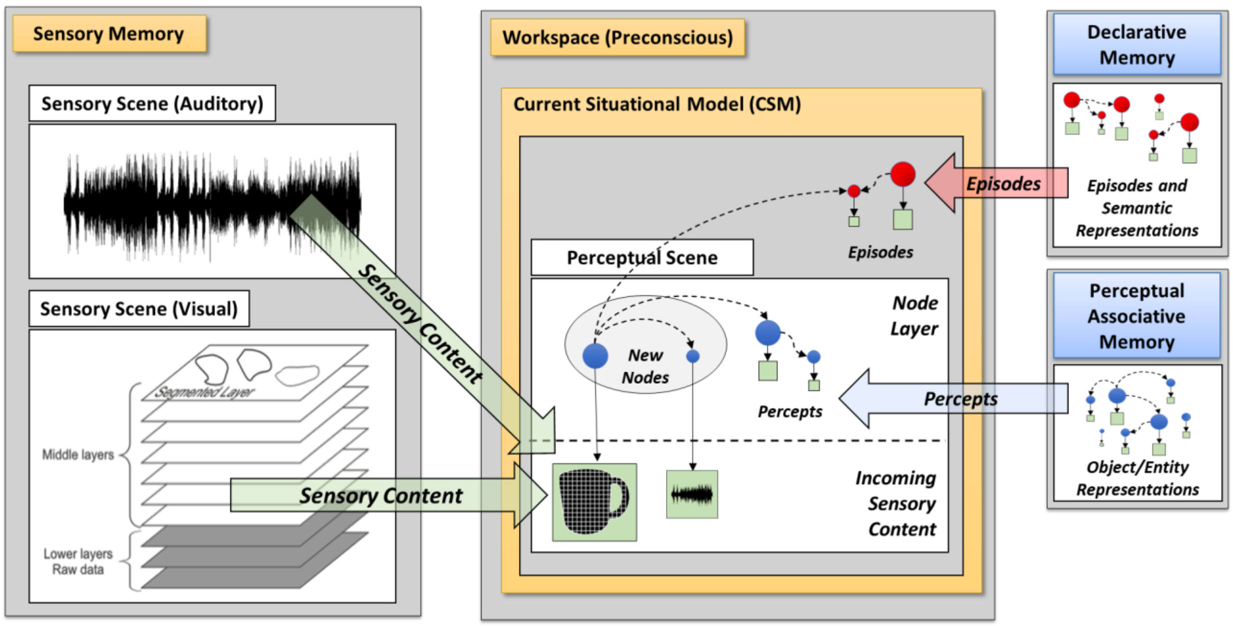

Figure 8 – The sensory and perceptual scenes and the creation of new perceptual symbols. Nodes are depicted as circular shapes. The sensory content associated with each node (via links) is depicted using square shapes. Image of the Sensory Scene (Visual) adapted from McCall et al. (2010b).

To illustrate this process, suppose our agent is enjoying a cup of coffee in her favorite coffee shop. Her visual sensors capture a pixel map of her coffee mug, as well as, other elements from the coffee shop environment, which Sensory Memory receives and integrates into its modality-specific sensory scenes. A "segmented layer" in her sensory scene (visual) is updated based on predicted object boundaries in the pixel map. In this case, one of the pixel regions in the segmented layer corresponds to the agent's coffee mug. Other (non-visual) sensory modalities may also exist with their own sensory scenes, for example, an auditory modality that transduces waveforms corresponding to speech sounds and ambient noise in the coffee shop. This visual and auditory sensory content, which was temporarily stored in Sensory Memory, is integrated into the CSM. Additionally, this content may activate percepts (e.g., feeling nodes with positive affective valence) from PAM or cue episodes of prior experiences in the same coffee shop from Declarative Memory. Structure building codelets create one or more new perceptual symbols in the perceptual scene by first creating new ("blank") nodes and then associating the incoming sensory content (e.g., coffee mug pixel region and ambient speech sounds) with those nodes. Relevant percepts, such as feeling nodes or recognized words, as well as, other cued long-term memories may also be linked to these developing perceptual symbols. If one or more attention codelets advocate for these new perceptual symbols, they may be included in a subsequent conscious broadcast, and learned into PAM. (Figure 8 depicts the sensory and perceptual scenes, and the intra-module interactions supporting the creation of the perceptual symbols described in this coffee shop example.)

Perceptual learning encompasses far more than a simple recording mechanism for saving static snapshots of sensory and perceptual states. Generalization (e.g., categorization) processes continually operate on perceptual symbols, consolidating isolated experiences into representations that support viewpoint invariant recognition of distinct objects. These object representations are further generalized into more abstract representations that can encapsulate the essence of entire classes of objects, classes of classes, and so on. In this way, hierarchies of representations (of indefinite depth) can be learned. Structure building codelets will be integral to both the formation of new categories and the assignment of category membership; however, the specifics remain an open research question.

Let's illustrate this by returning to our agent at the coffee shop. As our agent sits in her favorite coffee shop, drinking a cup of premium dark roast coffee, she glances at her coffee mug from many different angles. She quickly learns to recognize that many of these distinct experiences correspond to the same object: a rounded, ceramic, coffee mug. On her next visit to the same coffee shop, she may get a very different coffee mug – an octagonal, glass, coffee mug – filled with a delicious caramel macchiato. Again, she has numerous, distinct, multi-modal experiences, and, she quickly learns to recognize that many of these correspond to the same octagonal mug. By this time, our agent has also learned to generalize her experiences into the abstract concept of a "cup of coffee" that transcends not only the cup, but the type of coffee. Figure 9 illustrates this process of generalization.

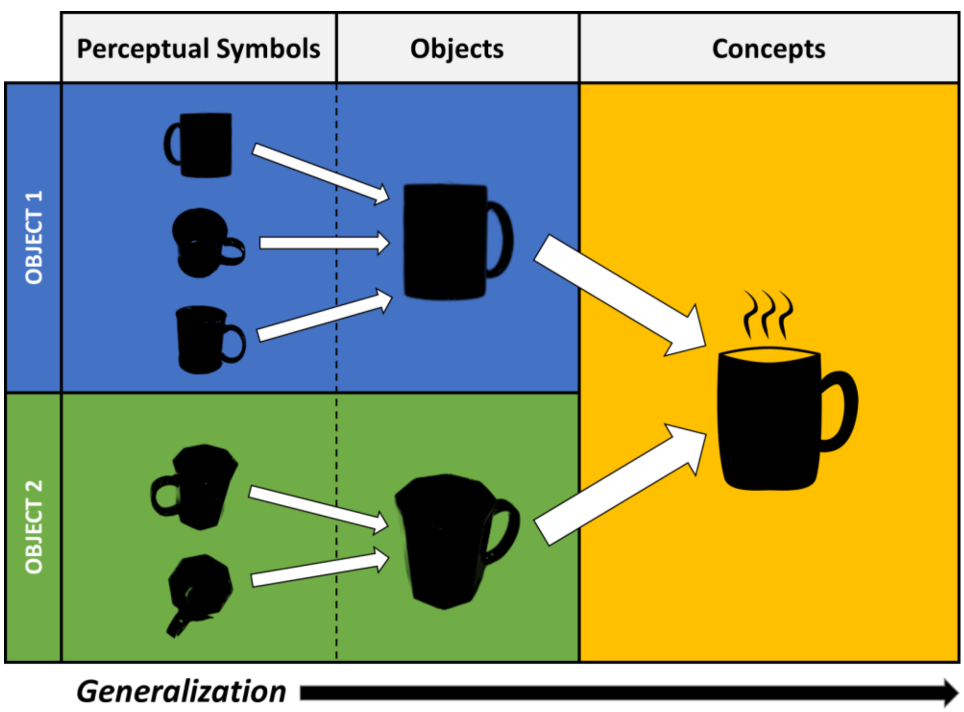

Figure 9 – A high-level illustration of the generalization processes from perceptual symbols, to viewpoint invariant object representations and more abstract concepts.

For more information on perceptual learning in general see D'Mello et al. (2006b). For more information on the sensory and perceptual scenes see McCall et al. (2010b).

5.5 Procedural Learning

The goal of procedural learning is to learn a comprehensive set of *relevant* and *reliable* schemes that enable the instantiation of useful, situation-specific, candidate behaviors (see Section 2.2.3). In this section, we will describe how Procedural Memory's instructionalist and selectionist learning mechanisms support this goal. Recall from Section 2.2.3 that the relevance and reliability of schemes are quantified by the current and base-level activations of the schemes, respectively.

LIDA agents typically begin life with a set of primitive (action-only) schemes. Upon receiving a conscious broadcast, Procedural Memory may instantiate one or more these primitive schemes as candidate behaviors and send them to Action Selection. The relevance (i.e., current activation) of primitive schemes will be minimal since they lack contexts and results; however, in the absence schemes with greater relevance, primitive schemes can, and eventually will, be instantiated as candidate behaviors. If Action Selection subsequently chooses one of these candidate behaviors for execution, and the result of the executed action comes to consciousness, then new, non-primitive schemes (containing a context, action, and result) will be learned via Procedural Memory's instructionalist learning mechanisms. Through an agent's continued exploration of its environment, a multitude of schemes may be learned, each relevant to different situations and pursuits.

Relevant schemes are not necessarily reliable schemes. To illustrate this, consider an agent that executes a "reaching" action from many different locations in a room. From some of these locations, our agent may discover that the gripper on the end of its arm-like effector touches an object after executing the "reaching" action, and in other cases it doesn't. Perhaps, in one of these cases, another agent is passing by (unbeknownst to our agent) as our agent begins its "reaching" action, and it collides with our agent's arm-like effector. Assuming our agent learns schemes for each of these different contexts and results, how is our agent to differentiate between the efficacy of these "reaching" schemes for realizing some result (e.g., touching an object)? The scheme that encoded the coincidental collision with another agent may be highly relevant in the future if the agent wishes to touch an object; however, since this scheme reflects a low probability event, it will be unreliable (i.e., unlikely to obtain the desired result).

In general, unreliability may be the result of many different factors including, but not limited to, inherently nondeterministic and partially observable environments (Russell & Norvig, 2016), frequent failures during action execution, or deficiencies in the specification of an agent's schemes. Procedural Memory's selectionist learning mechanisms are responsible for updating the schemes' base-level activations to reflect the empirical reliability of the scheme: Procedural Memory will increase the base-level activation of any scheme when the execution of its action produces the expected result. Over time, reliable schemes will feature greater base-level activations, and unreliable schemes will decay, and may eventually be pruned from Procedural

Memory. Box 1 illustrates the processes common to both instructionalist and selectionist procedural learning.

Box 1. Illustration of Procedural Learning

Initiation of the Procedural Learning Process (See Figure 10.)

1. **Conscious Broadcast.** Procedural Memory receives a conscious broadcast, which it uses to activate relevant schemes.
2. **Instantiation of Candidate Behaviors.** Procedural Memory instantiates *candidate behaviors* from its most activated (i.e., most relevant) schemes. These candidate behaviors are then received by Action Selection.
3. **Behavior Selected.** Action Selection chooses a single behavior from among these newly instantiated candidate behaviors and perhaps other candidate behaviors that remain in the Action Selection module from prior cycles. The selection criteria are based, in part, on the behaviors' base-level activations, which quantify their reliability. This *selected behavior* is received by Attention Codelet Memory, Procedural Memory, and Sensory Motor Memory.

4(a). New Expectation Codelet. Attention Codelet Memory may spawn a new *expectation codelet*. This expectation codelet biases the agent's attentional processes based on the currently executing behavior; for example, if an unexpected result occurs (as determined by the executing behavior's expected results), then that result may have a greater likelihood of coming to consciousness.

4(b). Update Recently Selected Behaviors. Procedural Memory adds the selected behavior to a set of *recently selected behaviors* maintained in Procedural Memory. Once added the recently selected behaviors rapidly decay from Procedural Memory.

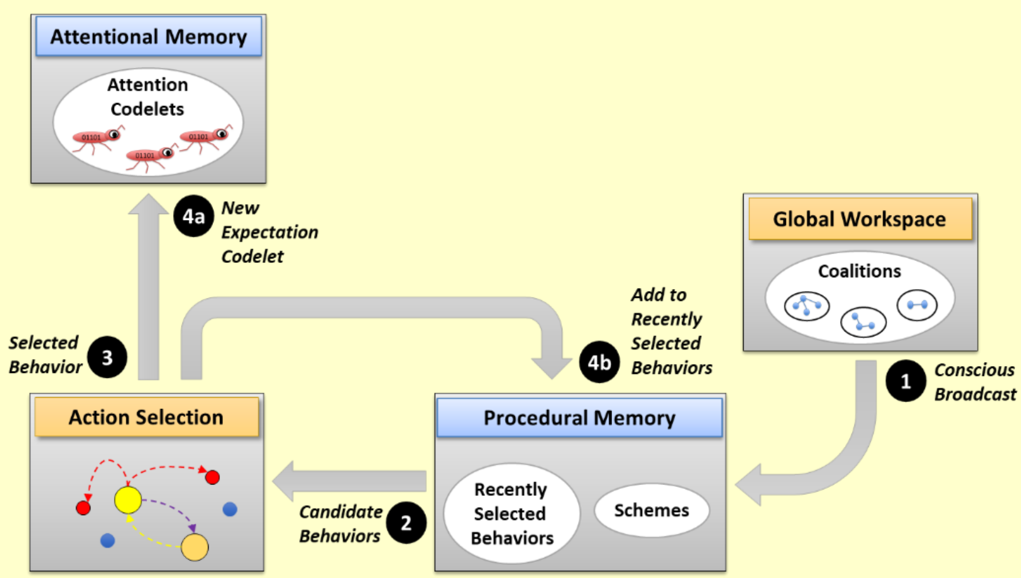

Figure 10 - Initiation of the procedural learning process.

Selectionist and Instructionalist Procedural Learning (See Figure 11.)

5. **Incoming Sensory Stimuli.** Incoming sensory stimuli from the environment[8] is transduced into modality-specific sensory content in Sensory Memory. This *sensory content* will correspond to a state of the environment that occurred after commencement of action execution for the selected behavior.

6. CSM Integration. The Workspace (preconscious) integrates the new sensory content from Sensory Memory into multi-modal representations in the perceptual scene (see McCall et al., 2010b).

7. Attention. If learning is to occur, one or more attention codelets must advocate for content in the CSM. Expectation codelets (e.g., the codelet created in Step 4(a)) help direct the agent's attention towards aspects of the perceptual scene that correspond to the expected results (or non-results) specified by previously selected behaviors. Any content selected by attention codelets will become part of a coalition and compete in the Global Workspace.

8. Conscious Broadcast. The Global Workspace selects a winning coalition and globally broadcasts its contents to all LIDA modules, including Procedural Memory.

9. Procedural Learning. Procedural Memory receives the conscious broadcast from Step 8 and initiates its selectionist and instructionalist learning mechanisms.

Selectionist learning occurs when the contents of the conscious broadcast from Step 8 match the expected results of one or more recently selected behaviors. In this case, the base-level activations of the corresponding schemes are increased.

Instructionalist learning occurs when there is no match between the conscious broadcast and a recently selected behavior. In this case, one or more schemes are created from the contexts and actions of the recently selected behaviors. The expected result for these schemes will be based on the conscious broadcast from Step 8.

---

[8] This assumes that the selected behavior contained an *external action*. The learning mechanisms for *internal actions* are identical with the exception that the expected result will be based on content that was created by structure building codelets; for example, an action corresponding to the mental rotation of an object by 90° would have an expected result that is internally generated. See Franklin et al. (2016) for more information on internal and external actions.

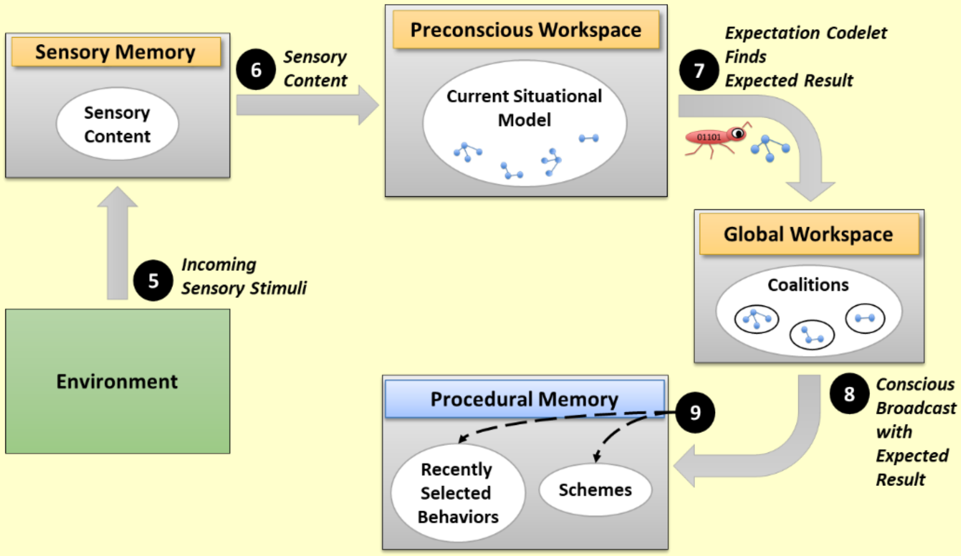

Figure 11 - Selectionist and instructionalist procedural learning. Illustrates the continuing process of selectionist and instructionalist procedural learning (continuing from Figure 10).

Procedural learning is heavily dependent on perceptual learning. An agent's schemes will become more relevant and reliable as it learns to recognize new details from its environment that more precisely discriminate what matters in a given situation. Drescher (1991) introduced *synthetic items* to his Schema Mechanism as a means of capturing "aspects of the world that the existing repertoire of representations fails to express" (Dresher, 1991, p. 11). LIDA's perceptual learning mechanisms (see Section 5.4) perform a similar function, allowing the creation of new representations that can be encoded in the context and results of schemes. For example, an agent may initially learn a general scheme for opening doors based on an action that involves turning a door knob clockwise and pulling the door handle towards the agent. This scheme may work well for hinged doors, but it will fail when the agent is confronted with sliding doors. Once the agent can perceptually discriminate between hinged doors and sliding doors, different schemes can be created for contexts with hinged doors and contexts with sliding doors featuring different actions. Figure 12 illustrates this relationship between schemes and the perceptual structures on which they depend.

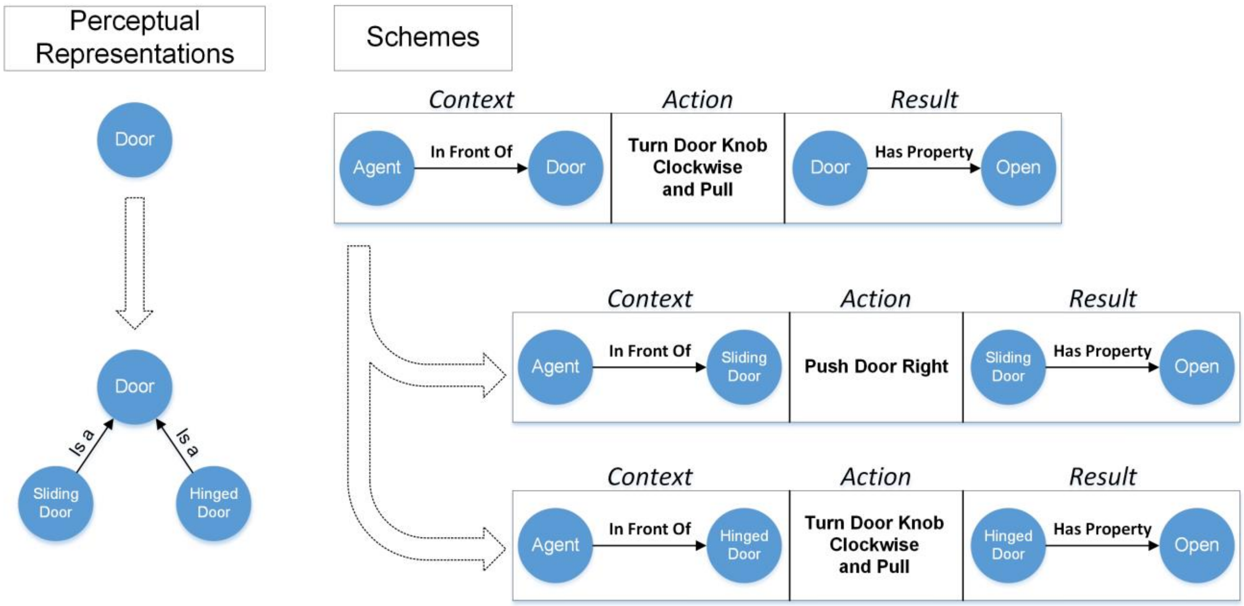

Figure 12 – Illustration of scheme dependency on perceptual representations. Initially a generic scheme for opening doors is learned. Once the agent learns to differentiate perceptually between different door sub-types, more relevant and reliable schemes can be learned for each sub-type. In this way, Perceptual Memory hierarchies naturally translate into Procedural Memory hierarchies.

Procedural learning is also heavily dependent on attentional learning and what the agent was attending to in a given situation. A scheme's context and result are based on the contents of multiple conscious broadcasts, each of which contain what was deemed most salient at those moments; however, there is no guarantee that the most salient representations at a given moment will prove useful for creating relevant and reliable schemes. To illustrate this point, we return to our earlier example of an agent with a fondness for "reaching". Suppose the agent executes its "reaching" action and succeeds in touching an object, but, in the moment its gripper touches the object, it is distracted by a flashing light detected by its camera sensor. In this case, the result encoded in the scheme may relate to the flashing light and not the fact that its gripper touched the object. These incidental distractions should be reduced over time as the agent's attentional mechanisms become attuned to its environment, leading to more relevant and reliable schemes.

For more information on procedural learning see D'Mello et al. (2006a).

5.6 Sensorimotor Learning

The primary goal of sensorimotor learning is the optimization of Motor Plan Execution through the creation and update of motor plan templates (MPTs) in Sensory Motor Memory. Wolpert et al. (2011) stated that "optimizing motor performance is achieved through three classes of control" (p.740) – predictive or feedforward control, reactive control, and biomechanical control – and that "all three of these control processes are adaptable and can contribute to motor learning" (p.740). They also defined three associated forms of sensorimotor learning: error-based learning, reinforcement learning, and use-dependent learning. A significant observation from their work is that *errors* and *rewards* drive most sensorimotor learning. Dong and Franklin (2015a; 2015b) established the foundations of sensorimotor learning in LIDA by adding errors and rewards to the Sensory Motor System (SMS): each MPT was augmented with its own errors and rewards databases (see Figure 5).

A modified Kalman filter (Kalman, 1960) was added to the Sensory Motor System in Dong et al. (2015). It implemented a predictive control process – one of the three classes of control suggested by Wolpert et al. (2011). This predictive control process was used as a (forward) internal model that predicts the results of motor command execution by simulating the dynamics of an agent's environment (including the agent's bodily state). This prediction is then combined with a reafferent sensory correction. Wolpert et al. (1995) argued, based on their experimental findings, that such a mechanism exists in the central nervous system, and Dong et al. (2015) replicated their findings in a computational experiment using a LIDA agent.

Reinforcement learning, in the form of Q-Learning (Watkins, 1989), was implemented in the SMS in Dong and Franklin (2015b). The Q-Learning algorithm was used to update *motor command-value functions*[9] (one per MPT) from immediate rewards calculated in the SMS during the process of *online control* (see Figure 3). Note that while the rewards were calculated in the SMS, the calculation was based on the contents of a conscious broadcast; therefore, this conforms to the Conscious Learning Hypothesis. Once determined, these immediate rewards were stored in the MPT's reward database (corresponding to the executing motor plan) and utilized by Q-Learning to update the MPT's motor command-value function. Since these motor command-value functions estimate the "goodness" (more precisely, the expected cumulative discounted reward) of executing a motor command from a given state, the performance of motor plan *choice functions* (see Figure 3) can be optimized by selecting the best-known motor command, as reckoned by the motor-command value function for a given state. In practice, the choice function also permits a small percentage of "non-greedy" motor command selections to allow continued exploration[10]. See Dong and Franklin (2015b) for more details.

The learning rates for both the Kalman filter and Q-Learning algorithms were also learned (i.e., adjusted) based on the idea of "memory of errors" (Herzfeld et al., 2014). This experience-based adjustment of the parameters associated with a learning mechanism (in this case the learning rate) constitutes a form of "meta-learning" (i.e., learning about learning), which is arguably the first of its kind in LIDA.

For more information on sensorimotor learning and the SMS, including computational experiments, see Dong and Franklin (2015a; 2015b) and Dong et al. (2015).

---

[9] Compare "action-value functions" (Sutton & Barto, 2018).
[10] This approach is referred to as epsilon-greedy in the RL literature (Sutton & Barto, 2018).

5.7 Spatial Learning

Spatial learning involves the creation and maintenance of cognitive maps (see Spatial Memory in Section 2.2.3). Cognitive maps encode spatial information about an agent's environment, and its orientation within that environment, enabling localization and route planning.

Madl et al. (2016) defined the processes involved in the construction and learning of cognitive maps. The constructive responsibilities are divided among five structure building codelets (SBCs)[11]. A "Map Structure SBC" creates new cognitive maps in the CSM as a regular, grid-like node structure comprised of locally connected *place nodes* (see Figure 3). A "Boundary SBC" removes links from previously constructed cognitive maps when boundaries are recognized that prevent traversal between any two locations associated with place nodes. A "Object-Place SBC" creates links from recognized objects to place nodes in the cognitive maps. A "Localization SBC" creates or updates links between the "self" PAM node and place nodes in the cognitive map based on the agent's belief about its current position in the environment. Finally, a "Map Correction SBC" updates existing cognitive maps (adding/removing links) whenever a location is revisited and updates are needed to reflect the changed environment. Spatial Memory learns new (or reinforces existing) cognitive maps whenever parts of a cognitive map are received by Spatial Memory from the conscious broadcast.

For more information on spatial learning and Spatial Memory in LIDA, including computational experiments, see Madl et al. (2016; 2018).

5.8 Language and Protolanguage Learning

Language learning can be divided into language comprehension and language production. These can be further sub-divided by communication media: oral, written, gestural, etc. To date, very little attention has been focused on language learning in LIDA. This can be explained, in part, by the fact that language learning builds on most (if not all) of the other learning mechanisms that have been described in this paper. CMattie and IDA (Franklin, 2000) – agent precursors of LIDA – communicated with humans using natural language over email; however, CMattie and IDA's language comprehension and production abilities were hand-engineered rather than learned.

Ait Khayi and Franklin (2018) represented our first attempt at modeling language learning (more specifically, "protolanguage" learning) in LIDA. The paper presents a conceptual explanation and computational simulation of how an infant vervet monkey could learn associations between distinct "alarm calls" and the meanings of those calls. The alarms studied were "vocalizations" by an adult vervet monkey (the infant's mother) in the presence of various classes of predators (eagles, leopards, and snakes). Three kinds of meaning – referential (i.e., the class of predator), affective (i.e., the mother's emotional response), and behavioral (i.e., the mother's escape action) – were learned by the infant by observing its mother's behaviors following each alarm

[11] The implementation details and low-level mechanisms underlying these SBCs have been excluded from this paper. Interested parties should review Madl et al. (2016).

call. The learning modeled in this paper is perceptual in nature, based on event structures created by custom structure building codelets. Language production was not covered in this paper.

For additional information on their implementation and results, see Ait Khayi and Franklin (2018).

6 Future Work

Our goals for this paper were (1) to convey a deeper understanding (and appreciation) of LIDA and its many learning mechanisms, and (2) to provide a roadmap for future work. In service of the first goal, we have elaborated on the LIDA conceptual model (in Section 2) and many of its learning mechanisms (in Section 5). We have explored LIDA's conceptual commitments that are pertinent to learning (in Section 3), as well as general concerns related to learning in LIDA (in Section 4). In this final section, we will fulfill the paper's second goal by outlining specific areas where opportunities may exist for additional work.

Table 2 identifies topics in learning that need additional exploration; however, it is by no means a complete list. We briefly explain each topic in the text that follows.

Topics for Future Work	**Associated Learning Type(s)**
Action Expectations in Non-Deterministic Environments	• attentional (expectation codelets) • procedural (enhanced schemes)
Affordances	• perceptual (recognizing affordances) • SBCs (affordance categorization)
Alarms	• motivational ("panic" feelings) • perceptual (alarm conditions)
Automatization	• procedural (behavior streams)
Behavior Streams	• procedural
Built-In Feelings with Learned Receptive Fields?	• motivational
Consolidation and the Emergence of Semantic Knowledge	• declarative
New Motor Plan Templates (MPTs)	• sensorimotor
Observational Learning and Imitation	• declarative • procedural • sensorimotor
Scheme Generalization	• perceptual (recognizing categories) • procedural ("parameterized" schemes) • SBCs (categorization)
Simulators, Generative Models, and Mental Imagery	• perceptual (generative models) • SBCs (generative processes)

Table 2 – Topics for future work in learning. In the right column we have speculated on the types of learning that may be involved.

6.1 Action Expectations in Non-Deterministic Environments

Recall from Section 2.2.4 that expectation codelets influence a LIDA agent's attention following action selection, which, among other things, is critical for efficient procedural learning (see Section 5.5). Ideally, expectation codelets will bias an agent's attention towards cognitive content (in the CSM) that is both relevant and informative. Relevant, in this context, means aspects of the environment that were likely influenced by the execution of the selected action. Informative refers to surprising, or unpredicted, outcomes that present opportunities for the agent to learn something new. Unfortunately, the evaluation of these criteria is much more difficult in non-deterministic environments.

Non-deterministic environments contain sources of inherent, or apparent[12], randomness, and agents faced with such environments will likely observe different outcomes after executing the same action from what appears to be the same context. LIDA agents may learn schemes for each of these outcomes, resulting in sets of schemes with the same context and action, but different expected results (and possibly base-level activations). Since expectation codelets only have access to a single expected result, they are unaware of the full breadth of possible outcomes for an action in a given situation. This leads to a myopic determination of relevance and informativeness based solely on one possible outcome and its likelihood.

To address this issue, the LIDA conceptual model may need to be enhanced. One solution would be to update the scheme data structure so that a scheme's expected result reflects all known outcomes and their associated likelihoods. In other words, all schemes with the same context and action are collapsed into a single scheme with a composite expected result. The resulting "outcome likelihood distribution" could be used by expectation codelets to make more informed decisions about which content deserves attention. Note that novel (unprecedented) outcomes could be identified by their absence from this likelihood distribution.

6.2 Affordances

Gibson (1979) defined affordances as a complementary relationship between an agent and its environment, characterized by what the objects, or situations, in that environment "offer", "provide", or "furnish" that agent. Affordances are not merely physical properties possessed by objects within the environment (e.g., rigidness, flatness, etc.), but emerge with respect to both the capabilities of an agent and its current goals. For a person, a stone could be a weapon or a paperweight depending on the situation at hand. For a lizard, that same stone may afford shelter from a storm or a hiding place from a predator. While humans and many non-human animals appear capable of effortlessly perceiving affordances (see Jamone et al., 2016), it

[12] Non-determinism may be the result of *inherent randomness* (e.g., a slot machine) or *apparent randomness* due to partial observability (see Russell & Norvig, 2016). For LIDA agents, the conscious broadcast will contribute to the partial observability of many environments, as it is a source of partial information by design.

remains an important open research question how affordances could be learned in a LIDA agent.

6.3 Alarms

Recall from Procedural Memory in Section 2.2.3 that schemes are typically activated based on the contents of a conscious broadcast. In exceptional circumstances, for example those that require an extremely rapid response, schemes can be activated directly from a percept, *bypassing consciousness*. For example, when driving a vehicle, we are occasionally faced with other drivers that “slam on their brakes” or “cut us off”, forcing us to rapidly react to avoid an accident. In such cases, the conscious realization of the danger that lead to our rapid reactions may come during, or after, the execution of those actions. We call these unconsciously selected and executed actions *alarms*, following Sloman (1998; 2001).

Mechanisms for learning alarms in LIDA are currently unknown. Minimally, both perceptual and motivational learning will be needed[13]: perceptual learning to (preconsciously) recognize imminently dangerous events that require rapid and immediate action; motivational learning to associate interpretative feeling nodes (e.g., “panic” feelings) with these dangerous events (e.g., the sudden appearance of an obstacle when driving), and to learn strong incentive saliences that facilitate the rapid recruitment of schemes and influence Action Selection. Alarms may also depend on procedural and sensorimotor learning, as both alarm schemes and MPTs must exist that allow the selection and execution of alarms.

6.4 Behavior Streams (Action Plans)

Behavior streams are associated with LIDA’s implementation of *action plans*. Ramamurthy et al. (2001) stated that a behavior stream is a “partially ordered plan which guides execution of behaviors (plan operators) so as to effect the required transition from the initial state to the goal state” (p. 7). Behavior streams are similar in kind to Drescher’s *composite actions*. Drescher (1991) stated that “a composite action is defined with respect to some goal state; it is the action of bringing about that state…. The means are given by chains of schemas that lead to the goal state from various other states*”* (p. 59). To the best of our knowledge, there have been no attempts to explain how behavior streams could be learned in IDA or LIDA; therefore, they represent a significant opportunity for much needed future work. We expect that most of the necessary enhancements will be in Procedural Memory.

6.5 Automatization

In Section 6.4 we mentioned that sequences of instantiated schemes (i.e., behaviors) can be integrated into action plans that we call behavior streams. With increased proficiency and

[13] Franklin et al. (2016) stated “Learning alarms, that is, learning to bypass attention, is a form of attentional learning” (p. 32). While it’s true that alarms bypasses consciousness, we now question whether this involves attentional learning. It may be possible to model this behavior simply by allowing the unconscious instantiation of schemes from content in PAM (or the CSM).

predictability, these behavior streams can become *automatized*, such that each subsequent behavior in a behavior stream can be selected *without conscious intervention*. This automatization supports the conservation of an agent's limited attentional resources and facilitates the parallel execution of behavior streams.

Negatu (2006) fleshed-out significant portions of the automatization process for IDA, LIDA's precursor. Unfortunately, many of the architectural components that existed in IDA do not have clear analogs in LIDA. Additional work is needed to translate IDA's implementation of automatization to LIDA, and to fill in any conceptual gaps where irreconcilable differences are discovered. We currently believe that LIDA's Action Selection module will need to support new *triggers* that will initiate the *continued selection* of behaviors in a *currently active* automatized behavior stream[14]. Procedural learning will be needed to model the "over-learning" of behavior streams that will trigger this mode of action selection.

6.6 Consolidation and the Emergence of Semantic Knowledge

Recall from Section 5.2 that declarative learning relies on an offline consolidation process that is responsible for transforming non-decayed away episodes in TEM into autobiographical memories and semantic knowledge in Declarative Memory. Modified versions of Sparse Distributed Memory (SDM) (Kanerva, 1988) have been used as the basis for most of LIDA's TEM and Declarative Memory implementations, and, by extension, LIDA's consolidation process. Ramamurthy and Franklin (2011) noted that SDM is attractive for modeling many aspects of human memory systems, such as interference and forms of "fringe" consciousness (Mangan, 1993) like "knowing that one does not know something" and the tip-of-the-tongue phenomena.

While SDM has received a great deal of attention, additional work is needed to integrate SDM-based TEM and Declarative Memory with other LIDA modules. MCR vectors (Snaider & Franklin, 2012; 2014a; Agrawal et al., 2018) are one promising direction that may reduce (or eliminate) the need to map SDM representations to, and from, other cognitive content; however, the exploration of MCR vectors and the implementation of Vector LIDA (Snaider & Franklin, 2014b) are not yet complete.

Another declarative learning topic needing more exploration is the emergence of semantic memories (over time) from episodes. While we have explored "forgetting" (D'Mello et al., 2006c) using SDM, we have not studied the emergence of semantic memories in depth. Work is needed to verify that interference and decay are sufficient to explain the genesis of semantic knowledge.

Finally, triggers for the offline consolidation process have not been explored. In animals, consolidation, though still poorly understood, appears to coincide with portions of the sleep

---

[14] By "continued selection" we mean the selection of behaviors in a behavior stream based on the earlier selection of their immediate predecessors in a stream. By "active" we mean instantiated behavior streams that have not yet decayed from Action Selection.

cycle. It is not clear what life events should trigger consolidation in LIDA-based software agents, and at what frequency of occurrence.

6.7 Built-In Feelings with Learned Receptive Fields?

In Section 5.3, we contended that a LIDA agent's feeling nodes must be built-in (never learned). In many cases, it's clear how evolutionary processes could have endowed biological agents with innate feelings and their corresponding receptive fields (i.e., sources of activation); for example, feelings of hunger and satiety could have genetically predetermined sources of activation originating from "sensors" monitoring an agent's digestive system. Similarly, it would not be difficult to engineer a LIDA-based agent with built-in "hunger" and "satiety" feeling nodes that are activated based on the energy remaining in the agent's battery. It's much more difficult to explain how a feeling like shame is innate because it appears to be activated by learned events; for example, violations of cultural norms may result in feelings of shame. This leads to the following chicken-and-egg problem: How can innate or built-in feelings receive activation and be associated with events, if they rely *solely* on those events to receive activation? It's likely that most cultural norms are learned rather than innate (since they differ across cultures), so how is shame activated initially? It's clear how the receptive field of shame could *expand* to include learned cultural norms; however, there must be something *not learned* that initially activates the feeling, providing the basis for later generalization. At present, we do not have a satisfactory answer for this question, but it merits careful study, as it is a gap in our current understanding of how motivational learning works in LIDA.

6.8 Simulators, Generative Models, and Mental Imagery

Barsalou (1999) stated that "the primary goal of human learning is to establish simulators", which he defined as "the knowledge and accompanying processes that allow an individual to represent… an entity or event" (p. 587). Simulators are *generative processes* that can construct limitless *simulations* of the concepts they represent. When simulations come to consciousness they are referred to as *mental images*.

This generative aspect of perceptual learning should be accounted for in LIDA, as there is substantial evidence that humans (and likely other animals) routinely make use of conscious (and possibly preconscious) mental simulations in support of cognitive processes such as language comprehension (Zwaan et al., 2002; Bergen et al., 2007), prediction (Moulton & Kosslyn, 2009), problem solving (Shaver et al., 1974; Qin & Simon, 1992; Clement, 1994), and mental rehearsal (Driskell et al., 1994; Keller, 2012). While we have just begun to explore LIDA's counterparts to Barsalou's simulators and the phenomena of mental imagery, we currently believe that the perceptual symbol generalization processes described in Section 5.4 may need to learn *generative models* for each object, entity, concept, etc., (in PAM) that can be utilized by SBCs to generate quasi-sensory instantiations of those objects, entities, concepts, etc.

6.9 Scheme Generalization

Recall from Section 2.2.3 that schemes are data structures in Procedural Memory that specify a context, action, and expected result. Also recall that any of these components may be parameterized with *variables* that serve as placeholders allowing numerous *instantiations* of a scheme. For example, an agent may have a scheme (uninstantiated) for picking up “*any* cup”, where “*any* cup” is represented by a {CUP} variable. Furthermore, the {CUP} could be picked up with either the agent’s left or right {HAND}. When these variables are bound by Procedural Memory during instantiation, {CUP} may be replaced by a particular “Coffee Cup” and {HAND} by the agent’s “Right Hand”. Both uninstantiated and instantiated version of this scheme are illustrated in Figure 13.

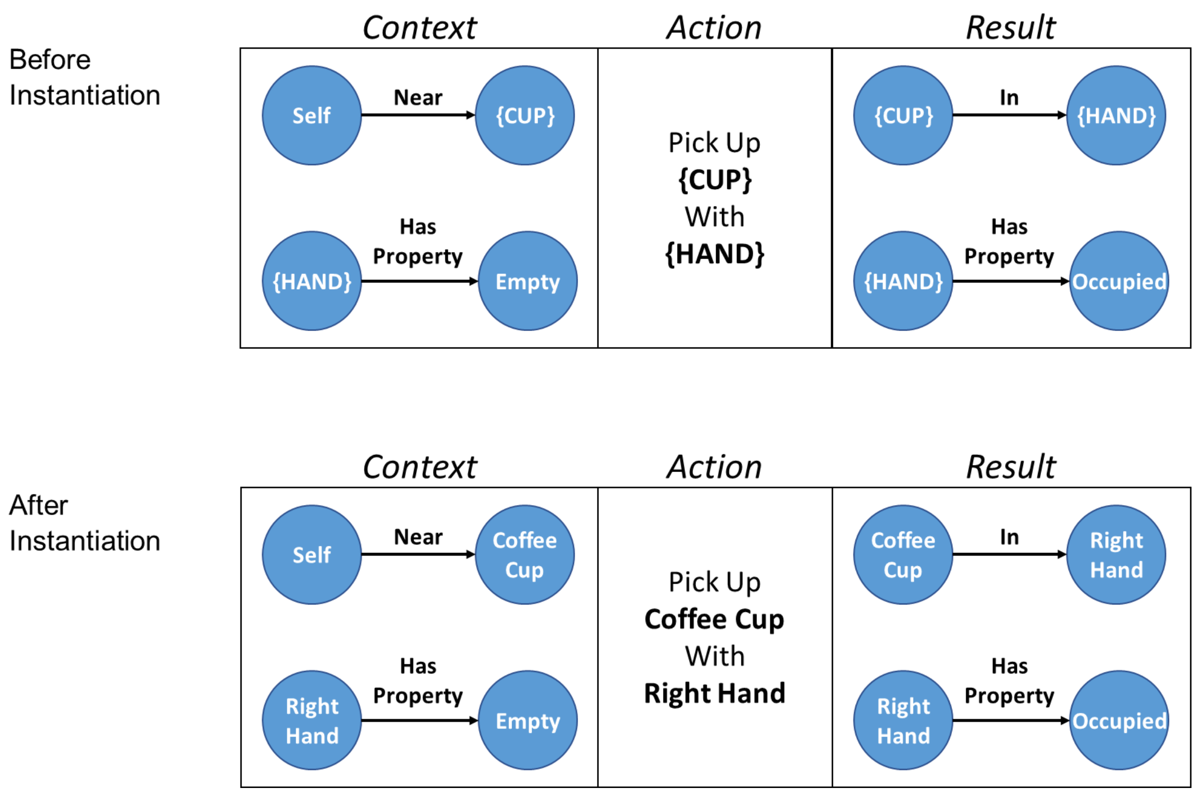

Figure 13 – Scheme Instantiation. The uninstantiated scheme (top) has unbound variables for {CUP} and {HAND}. The instantiated scheme (bottom) replaces each unbound variable with a specific value; in this case, {CUP} is replaced by “Coffee Cup” and {HAND} is replaced by “Right Hand”.

How these scheme variables are learned during procedural learning is currently unknown; however, one approach (that may or may not be viable) would be to let categories serve as the unbound variables. For example, if an agent learns an SBC that associates a “cup” category with all manner of cups, and another SBC that associates a “hand” category with cognitive content representing the agent’s left and right hands, then schemes containing these categories could be learned from the conscious broadcast. Schemes that contain category nodes in contexts, actions, and/or results could later be instantiated by binding content in the conscious broadcast, belonging to those categories, to the content in the schemes corresponding to those categories.

6.10 Observational Learning and Imitation

Wolpert et al. (2011) stated

> *An important source of information in the development of motor skills is the observation of others* [emphasis added]…, many studies have provided evidence that watching another person perform an action engages sensorimotor representations of the observed action…. It is well established that people can learn high-level information about what movements to make, and in what sequence, by observing actions. (p. 744)

Closely related to this idea of "observational learning" is learning through imitation. Meltzoff and Moore (1994) found that six-week-old human infants imitated tongue protrusions and mouth opening behaviors demonstrated by adults. Hanna and Meltzoff (1993) found that human toddlers (aged 14-18 months) imitated their peers' interactions with various objects and toys. There is also a growing body of evidence that imitation is common in a wide range of non-human animals; for example, rodents (Heyes & Dawson, 1990), chimpanzees (Myowa, 1996), birds (Akins & Zentall, 1998), and cephalopods (Fiorito & Scotto, 1992). Based on this, and other research, it appears that many animals are capable of imitation from a very early age, and they have built-in motivations to do so. In the machine learning community, there has also been a great deal of interest in the idea of learning by demonstration, especially for robotics (Argall et al., 2009), and reinforcement learning algorithms have recently been proposed (e.g., Hester et al., 2018) that use demonstrations to greatly accelerate learning and improve performance.

We have yet to flesh-out how observational learning and imitation could be realized in LIDA; however, there are several avenues of exploration. (1) Declarative memories could be learned through observation that provide agents with semantic knowledge about cultural norms and other forms of behavioral "scripts" (see Gioia & Poole, 1984). (2) The discovery of "mirror neurons" in primates, which are excited both when performing an action and observing the same action performed by others, suggests the possibility of a "simulation-based approach" (see Gallese & Goldman, 1998) to procedural and sensorimotor learning. This could be augmented with a "theory of mind" (see Friedlander and Franklin, 2008) supporting the inference of purpose and intentions from the actions of observed agents.

6.11 Learning New Motor Plan Templates (MPTs)

Dong and Franklin (2015b) described a LIDA agent that learned to push boxes in a simulated 3D environment. The agent came equipped with a *box pushing* MPT composed of subtasks for *finding*, *pushing*, and *unwedging*. Furthermore, each subtask was designed to receive rewards based on various events, such as the fulfillment of built-in "sub-goals"[15]. For example, the *finding* subtask was designed to associate a reward of $+3$ with a *forward movement* motor command if the agent perceived it was "near a box" during forward motion; otherwise, the motor command received a -1 reward.

---

[15] As described in Section 5.6, MPTs use the contents of the conscious broadcast to determine whether a reward condition has occurred; hence, this is consistent with the Conscious Learning Hypothesis.

We have yet to explore how such an MPT could be learned by a LIDA agent, and there are several significant challenges that need to be addressed. First, we must identify the processes and events responsible for initiating the creation of a new MPT. Next, we need a learning mechanism capable of decomposing a high-level goal into supporting subtasks and wiring these up appropriately in the new MPT. This learning mechanism must also be able to identify relevant MPT reward conditions (e.g., corresponding to the fulfillment of sub-goals) that can serve as the basis for optimizing motor performance. Finally, and perhaps most importantly, the interactions between the Sensory Motor System, Procedural Memory, various modes of Action Selection (e.g., deliberative and consciously-mediated), and workspace dynamics (e.g., goal contexts and motivations) need to be explored.

Acknowledgements

The authors would like to thank Daqi Dong for reviewing the article and providing thoughtful feedback on sensorimotor learning. We would also like to thank the members (past and present) of the Cognitive Computing Research Group (CCRG) for their many contributions to the content of this paper.

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

Agrawal, P., Franklin, S. & Snaider, J. (2018). Sensory Memory For Grounded Representations in a Cognitive Architecture. Sixth Annual Conference on Advances in Cognitive Systems.

Ait Khayi, N., & Franklin, S. (2018). Initiating language in LIDA: learning the meaning of vervet alarm calls. Biologically Inspired Cognitive Architectures, 23, 7-18. doi:10.1016/j.bica.2018.01.003

Akins, C. K., & Zentall, T. R. (1998). Imitation in Japanese quail: The role of reinforcement of demonstrator responding. *Psychonomic Bulletin & Review*, *5*(4), 694-697.

Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and autonomous systems*, *57*(5), 469-483.

Baars, B. J. (1988). *A Cognitive Theory of Consciousness*. Cambridge University Press.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and brain sciences*, *22*(4), 577-660.

Bergen, B. K., Lindsay, S., Matlock, T., & Narayanan, S. (2007). Spatial and linguistic aspects of visual imagery in sentence comprehension. *Cognitive science*, *31*(5), 733-764.

Born, J., & Wagner, U. (2006). Memory Consolidation during Sleep: Role of Cortisol Feedback. *Annals of the New York Academy of Sciences, 1032*, 198 - 201. doi: 10.1196/annals.1314.020

Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE journal on robotics and automation*, *2*(1), 14-23.

Clement, J. (1994) Use of physical simulation and imagistic simulation in expert problem solving. In *Implicit and Explicit Knowledge*, (Tirosh, D., ed.), 204–242, Erlbaum

D'Mello, S. K., U. Ramamurthy, A. Negatu, and S. Franklin. (2006a). A Procedural Learning Mechanism for Novel Skill Acquisition. In Workshop on Motor Development: Proceeding of Adaptation in Artificial and Biological Systems, AISB'06, vol. 1, ed. T. Kovacs, and J. A. R. Marshall. Bristol, England: Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

D’Mello, S. K., Franklin, S., Ramamurthy, U., & Baars, B. J. (2006b). A cognitive science based machine learning architecture. Paper presented at the AAAI 2006 spring symposium series sponsor: American association for artificial intelligence. Palo Alto, California, USA: Stanford University.

D'Mello S., Ramamurthy U., and Franklin S. (2006c). Realizing Forgetting in a Modified Sparse Distributed Memory System. *Proceedings of the 28th Annual Meeting of the Cognitive Science Society*, (pp. 1992-1997), Vancouver, Canada.

Dong, D., & Franklin, S. (2015a). A New Action Execution Module for the Learning Intelligent Distribution Agent (LIDA): The Sensory Motor System. Cognitive Computation. doi: 10.1007/s12559-015-9322-3.

Dong, D., & Franklin, S. (2015b). Modeling Sensorimotor Learning in LIDA Using a Dynamic Learning Rate. Biologically Inspired Cognitive Architectures, 14, 1-9. doi: 10.1016

Dong, D., Franklin, S., & Agrawal, P. (2015). Estimating Human Movements Using Memory of Errors. Procedia Computer Science, 71, 1-10. doi: 10.1016

Drescher, Gary L. (1991). *Made-Up Minds: A Constructivist Approach to Artificial Intelligence*. Cambridge, MA: MIT Press.

Driskell, J. E., Copper, C., & Moran, A. (1994). Does mental practice enhance performance?. *Journal of applied psychology*, *79*(4), 481.

Edelman, G. M. (1987). *Neural Darwinism: The theory of neuronal group selection*. New York: Basic Books.

Faghihi, U., McCall, R., & Franklin, S. (2012). A Computational Model of Attentional Learning in a Cognitive Agent. *Biologically Inspired Cognitive Architectures*, 2, 25-36.

Fillmore, C. (1968). The case for case. In E. Bach & R. T. Harms (Eds.), *Universals in linguistic Theory* (pp. 1–90). New York: Holt, Rinehart and Wilson.

Fiorito, G., & Scotto, P. (1992). Observational learning in Octopus vulgaris. *Science*, *256*(5056), 545-547.

Franklin, S., & Baars, B. (2010). Two Varieties of Unconscious Processes. In E. Perry, D. Collerton, H. Ashton & F. LeBeau (Eds.), *New Horizons in the Neuroscience of Consciousness* (pp. 91–102). Amsterdam: John Benjamin.

Franklin, S. and Graesser, A. (1997). Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, published as Intelligent Agents III, Springer-Verlag, 21-35.

Franklin, S. (2000). Building Life-Like "Conscious" Software Agents. *Artificial Intelligence Communications*, 13, 183-193.

Franklin, S. (2003). IDA: A conscious artifact? *Journal of Consciousness Studies*, *10*(4-5), 47-66.

Franklin, S., Strain, S., McCall, R., & Baars, B. (2013). Conceptual Commitments of the LIDA Model of Cognition. *Journal of Artificial General Intelligence*, 4(2), 1-22, DOI: 10.2478/jagi-2013-0002

Franklin, S., Madl, T., D'Mello, S., & Snaider, J. (2014). LIDA: A Systems-level Architecture for Cognition, Emotion, and Learning. *IEEE Transactions on Autonomous Mental Development*, 6(1), 19-41. doi: 10.1109/TAMD.2013.2277589

Franklin, S., Madl, T., Strain, S., Faghihi, U., Dong, D., Kugele, S., Snaider, J., Agrawal, P., Chen, S. (2016). A LIDA cognitive model tutorial. Biologically Inspired Cognitive Architectures, 105-130. doi: 10.1016/j.bica.2016.04.003

Friedlander, D. & Franklin, S. (2008). LIDA and a Theory of Mind. In Artificial General Intelligence (AGI-08), ed. Ben Goertzel and Pei Wang. Memphis, TN, USA: IOS Press.

Gallese, V., & Goldman, A. (1998). Mirror neurons and the simulation theory of mind-reading. *Trends in cognitive sciences*, *2*(12), 493-501.

Gibson, James J. (1979). *The Ecological Approach to Visual Perception*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Gioia, D. A., & Poole, P. P. (1984). Scripts in organizational behavior. *Academy of management review*, *9*(3), 449-459.

Hanna, E., & Meltzoff, A. N. (1993). Peer imitation by toddlers in laboratory, home, and day-care contexts: Implications for social learning and memory. *Developmental psychology*, *29*(4), 701.

Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3), 335-346.

Herzfeld, D. J., Vaswani, P. A., Marko, M. K., & Shadmehr, R. (2014). A memory of errors in sensorimotor learning. *Science*, *345*(6202), 1349-1353.

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., ... & Dulac-Arnold, G. (2018). Deep q-learning from demonstrations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Heyes, C. M., & Dawson, G. R. (1990). A demonstration of observational learning in rats using a bidirectional control. *The Quarterly Journal of Experimental Psychology Section B*, *42*(1b), 59-71.

Jamone, L., Ugur, E., Cangelosi, A., Fadiga, L., Bernardino, A., Piater, J., & Santos-Victor, J. (2016). Affordances in psychology, neuroscience, and robotics: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, *10*(1), 4-25.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, *82*(1), 35-45.

Kanerva, P. (1988). *Sparse Distributed Memory*. Cambridge MA: The MIT Press.

Keller, P. E. (2012). Mental imagery in music performance: underlying mechanisms and potential benefits. *Annals of the New York Academy of Sciences*, *1252*(1), 206-213.

Kringelbach, M. L., & Berridge, K. C. (2009). Towards a functional neuroanatomy of pleasure and happiness. *Trends in cognitive sciences*, *13*(11), 479-487.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Madl, T., Franklin, S., Chen, K., Montaldi, D., & Trappl, R. (2016). Towards real-world capable spatial memory in the LIDA cognitive architecture. Biologically Inspired Cognitive Architectures, 87-104.

Madl, T., Franklin, S., Chen, K., & Trappl, R. (2018). A computational cognitive framework of spatial memory in brains and robots. *Cognitive Systems Research*, 47, 147-172.

Mangan, B. (1993). Taking Phenomenology Seriously: The "Fringe" and Its Implications for Cognitive Research. *Consciousness and Cognition, 2*(2), 89-108. doi:https://doi.org/10.1006/ccog.1993.1008

McCall, R., Franklin, S., Friedlander, D. (2010a). Grounded Event-Based and Modal Representations for Objects, Relations, Beliefs, Etc. Paper presented at the FLAIRS-23, Daytona Beach, FL.

McCall, R., Snaider, J., & Franklin, S. (2010b). Sensory and perceptual scene representation. Journal of Cognitive Systems Research.

McCall, R. J., Franklin, S., Faghihi, U., Snaider, J., & Kugele, S. (2020). Artificial Motivation for Cognitive Software Agents. *Journal of Artificial General Intelligence*, *11*(1), 38-69.

McGaugh, J. L. (2000). Memory – a Century of Consolidation. *Science, 287*(5451), 248-251. doi: 10.1126/science.287.5451.248

Meltzoff, A. N., & Moore, M. K. (1994). Imitation, memory, and the representation of persons. *Infant behavior and development*, *17*(1), 83-99.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529

Moulton, S. T., & Kosslyn, S. M. (2009). Imagining predictions: mental imagery as mental emulation. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, *364*(1521), 1273-1280.

Myowa, M. (1996). Imitation of facial gestures by an infant chimpanzee. *Primates*, *37*(2), 207-213.

Nadel, L., Hupbach, A., Gomez, R., & Newman-Smith, K. (2012). Memory formation, consolidation and transformation. *Neuroscience & Biobehavioral Reviews*, *36*(7), 1640-164.

Negatu, A. S. (2006). *Cognitively inspired decision making for software agents: integrated mechanisms for action selection, expectation, automatization and non-routine problem solving* (Doctoral dissertation).

Newell, A. (1973). You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In W. G. Chase (Ed.), *Visual Information Processing*. New York: Academic Press.

Qin, Y., & Simon, H. A. (1992). Imagery and mental models in problem solving. In *Proc. AAAI Symposium on Reasoning with Diagrammatic Representations., Stanford, CA*.

Ramamurthy, U., Negatu, A., & Franklin, S. (2001). Learning Mechanisms for Intelligent Systems. SSGRR-2001 International Conference on Advances in Infrastructure for e-Business, e-Education and e-Science on the Internet L'Aquila Italy. August 6-12, 2001.

Ramamurthy, U., D'Mello, Sidney K., & Franklin S. (2006). LIDA: A computational model of Global Workspace Theory and developmental learning. Paper presented at the BICS 2006: Brain Inspired Cognitive Systems.

Ramamurthy, U., Franklin, S. (2011). Memory Systems for Cognitive Agents. Proceedings of Human Memory for Artificial Agents Symposium at the Artificial Intelligence and Simulation of Behavior Convention (AISB'11), University of York, UK, 2011, p 35-40.

Russell, S. J., & Norvig, P. (2016). *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall.

Shaver, P., Pierson, L., & Lang, S. (1974). Converging evidence for the functional significance of imagery in problem solving. *Cognition*, *3*(4), 359-375.

Simion, F., Regolin, L., & Bulf, H. (2008). A predisposition for biological motion in the newborn baby. *Proceedings of the National Academy of Sciences*, *105*(2), 809-813.

Sloman, A. (1998). Damasio, Descartes, Alarms and Meta-management *Proceedings Symposiumon Cognitive Agents: Modeling Human Cognition*. San Diego: IEEE.

Sloman, A. (2001). Beyond shallow models of emotion. *Cognitive Processing, 2*(1), 177-198.

Snaider, J., McCall, R., & Franklin, S. (2012). Time production and representation in a conceptual and computational cognitive model. *Cognitive Systems Research*, *13*(1), 59-71.

Snaider, J., & Franklin, S. (2012). Extended Sparse Distributed Memory and Sequence Storage. Cognitive Computation, 4(2), 172-180. doi: 10.1007/s12559-012-9125-8

Snaider, J., & Franklin, S. (2014a). Modular Composite Representation. *Cognitive Computation*, 6(3), 510-527. doi: 10.1007/s12559-013-9243-y

Snaider, J., & Franklin, S. (2014b). Vector LIDA. *Procedia Computer Science*, 41, 188-203.

Stickgold, R., & Walker, Matthew P. (2005). Memory consolidation and reconsolidation: what is the role of sleep? *Trends Neurosci., 28*, 408–415.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Turati, C., Macchi Cassia, V., Simion, F., & Leo, I. (2006). Newborns' face recognition: Role of inner and outer facial features. *Child development*, *77*(2), 297-311.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards* (Doctoral dissertation).

Wolpert, D. M., Ghahramani, Z., & Jordan, M. I. (1995). An internal model for sensorimotor integration. *Science*, *269*(5232), 1880-1882.

Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. *Neural computation*, *8*(7), 1341-1390.

Wolpert, D. M., Diedrichsen, J., & Flanagan, J. R. (2011). Principles of sensorimotor learning. *Nature Reviews Neuroscience*, *12*(12), 739.

Wood, J. N. (2013). Newborn chickens generate invariant object representations at the onset of visual object experience. *Proceedings of the National Academy of Sciences*, *110*(34), 14000-14005.

Yerkes, R. M., & Dodson, J. D. (1908). The relation of strength of stimulus to rapidity of habit-formation. *Journal of comparative neurology and psychology*, *18*(5), 459-482.

Zwaan, R. A., Stanfield, R. A., & Yaxley, R. H. (2002). Language comprehenders mentally represent the shapes of objects. *Psychological science*, *13*(2), 168-171.