

Sensory Motor System: Modeling the Process of Action Execution

Daqi Dong (DDONG@Memphis.Edu)
Stan Franklin (FRANKLIN@Memphis.Edu)

Department of Computer Science and the Institute for Intelligent Systems, University of Memphis,
Memphis TN 38152, USA

Abstract

This paper presents a cognitive model—the Sensory Motor System (SMS)—for an action execution process, as a new module of the LIDA systems-level cognitive model. Action execution refers to a situation in which a software agent or robot executes a selected goal-directed action in the real world so as to output pertinent movement. Action execution requires transforming a selected goal-directed action into lower-level executable actions, and executing them. A sensorimotor system derived from the subsumption architecture has been implemented into the SMS; and several cognitive neuroscience hypotheses have been incorporated as well, including the two visual systems and others. A computational SMS has been created inside a LIDA-based software agent in Webots to model the execution of a grip action. The grip's design is inspired by the arm controller of the robot Herbert and the current study of human's action. Simulated results are compared to human performance.

Keywords: Sensory Motor System (SMS); action execution; LIDA model; subsumption architecture; two visual systems; grip action; robot Herbert; Webots; cognitive modeling.

1. Introduction

Action presents two aspects. On the one hand, the agent (Franklin & Graesser, 1997) selects the action motivated from inside as a result of mental processes. Thus, the agent understands what it will do before the execution begins. However, this understanding of the action is not executable in the real world, because the needed low-level environmental information is not yet involved. On the other hand, the action's execution may not be understandable to the agent, because the environmental elements involved are low-level raw data without explicit meaning, while that which is understandable must have some form of meaning for the agent. As an example, the agent does not directly understand the raw stimulus data retrieved by its sensors from the environment; rather, the data must be transformed into high-level meaning by a perception process; that is, the transformation produces an understandable representation of the sensed data. Action execution performs a transformation similar to that of perception, but in reverse: converts an understandable action into low-level movements.

Milner & Goodale have proposed a hypothesis in their work on the two visual systems¹ (1992; 2008), which supports a model for how a human maintains and integrates these two facets of action: the understandable and the

executable—in other words, “what to do” and “how to do it”. They proposed two cortical systems, the ventral and dorsal streams, providing “vision for perception” and “vision for action” respectively. Regarding the roles of the two streams in the guidance of action, the perceptual mechanism in the ventral stream identifies a goal object, and helps to select an appropriate course of action, while the dorsal stream “is critical for the detailed specification and online control of the constituent movements that form the action” (Milner & Goodale, 2008, p. 775).

Following the hypothesis of the two visual systems, the dual aspects of action are represented in the LIDA Model as the distinct processes of action selection and action execution. Action selection has been described in previous work (Franklin, Madl, D'Mello, & Snaider, 2013); here we specify the action execution in the form of the Sensory Motor System (SMS) to extend LIDA. The SMS responds by transforming a desired understandable action, a selected behavior in LIDA, into an executable low-level action sequence, a sequence of motor commands, and executing them.

The next section describes the LIDA Model and its relationship to the SMS. Section 3 contains an overview of the subsumption architecture (Brooks, 1986, 1991), which is used as the SMS's prototype; and the SMS's concept is described in Section 4 in detail. Section 5 introduces the simulation of a specific action execution process, gripping. One aspect of human grip performance, the grip aperture, has been compared to the simulated results.

We are currently comparing SMS to other models of the action execution process. Also, besides the grip aperture, we plan to simulate other aspects of human grip performance in future, such as the grip force, velocity, etc.

2. The LIDA Model

The LIDA model is a systems-level cognitive model (Franklin et al., 2013). It implements and fleshes out a number of psychological and neuropsychological theories, but is primarily based on Global Workspace Theory (Baars, 1988, 2002). The model is grounded in the LIDA cognitive cycle. The simulated human mind can be viewed as functioning via a continual sequence of these cycles. Each cognitive cycle consists of three phases: 1) The LIDA agent first senses the environment, recognizes objects, and builds its understanding of the current situation; 2) By a competitive process, as specified by Global Workspace Theory (Baars, 1988), it then decides what portion of the represented situation should be attended to and broadcasted to the rest of the system; 3) Finally, the broadcasted portion

¹ In the LIDA Model, the concept of ventral and dorsal streams for the transmission of visual information has been extended to multimodal transmission.

of the situation supplies information allowing the agent to choose an appropriate action to execute, and modulates learning.

The Sensory Motor System (SMS) is proposed to complete a model for the process of action execution in LIDA. Two LIDA modules, Action Selection and Sensory Memory, provide relevant information—a selected behavior and the sensory data through a dorsal stream channel²—as inputs to the SMS separately. The selected behavior is a data structure resulting from the preceding action selection in the LIDA Model. It is comprised of three components: a context, an action³, and a result. With some reliability, the result is expected to occur when the action is taken in its context. The SMS sends out motor commands as its output to the environment.

3. The Subsumption Architecture

The subsumption architecture is a parallel and distributed computation formalism for controlling a robot using a type of reactive structure connecting sensors to actuators (Brooks, 1986, 1991). Its capabilities match many required features of action execution as we plan to model it. First, it fulfills the requirements for modeling online control of action execution because the sensor is directly linked to the motor that drives the actuators.

Second, in the subsumption architecture, specific behaviors are merged into a comprehensive classification, organized in multiple layers; it fulfills the need for transforming an understandable action into executable motor commands. Marc Jeannerod built upon the concept that covert action representation is followed by overt real executed action. In detail, “the conceptual content, when it exists (i.e. when an explicit desire to perform the action is formed), is present first. Then, at the time of execution, a different mechanism comes into play where the representation loses its explicit character and runs automatically to reach the desired goal” (2006). We equate the concepts used in the SMS with Jeannerod’s, although our terminologies differ.⁴

The subsumption architecture supports the decomposition of the desired goal into separate sub-goals to be accomplished with low-level tasks, so as to run automatically to reach the desired goal without “its explicit character”. “It’s a method of decomposing a robot’s control architecture into a set of task-achieving behaviors or competences” (Dawson, n.d.). Competences refer to low-

² In LIDA, a dorsal stream channel directly passes sensory data from the sensory memory to the action execution process.

³ In this context, this term refers to a component of a behavior. This differs from the general usage, such as in the phrase “action execution”. In this paper, we use “action” in the general sense, while “action of a behavior” refers to a particular component of that behavior.

⁴ ‘An explicit desire to perform the action’ refers to a selected behavior; ‘a different mechanism’ is our SMS; and ‘the representation loses its explicit character’ indicates that the action’s execution may not be understandable to the agent.

level tasks; they play a role in connecting an understandable action to executable motor commands.

Furthermore, the subsumption architecture typically has no central control, and the environment is used as the communication medium because “[the] world is its own best model” (Brooks, 1990, p. 3). This fact is consistent with our design requirement, as Jeannerod proposed above, for the absence of an understandable action’s “explicit character” in the action execution process. This explains why action execution remains outside the awareness of the agent, although it could become aware of the execution indirectly; we will introduce an example regarding this indirect awareness later in Section 5.4.

4. The Sensory Motor System (SMS)

4.1 The Motor Plan and Online Control

The output of the SMS, a sequence of motor commands, is sent out in a certain order; however, this “ordering” effect is not a plan working inside the SMS to determine when each motor command will be sent out. Since the action execution process is running in a real world with unlimited data available, much of this heavily affects the order of the motor commands in real time. It is hard to anticipate such environmental situations fully enough to prepare a specific sequence of motor commands before the execution begins.

Citing the work of Herbert Simon (1969), Rodney Brooks built upon the concept that complex behavior need not necessarily be a product of an extremely complex control system; rather, it may simply be the reflection of a complex environment (Brooks, 1986). Therefore, a reactive structure is introduced to model the source of ordered motor commands (Figure 4-1). Inside the SMS, first a set of motor commands are built in; each of them is represented by a \odot , which is independent of any timestamp. Next is a set of triggers, represented by T_x ; a trigger activates a specific command in order to send it out as a part of the SMS output when the input sensory data matches one or more of the trigger’s conditions. The subscript x stands for the number of conditions the trigger contains. Third, before sending out the commands, a choice function chooses a command from possibly multiple candidates as the final output at each moment. The set of motor commands, the triggers, and the choice function are referred to as a Motor Plan (MP), which specifies what to do in a particular situation, independently of time.

An environment module located outside the SMS is shown in Figure 4-1 as well; it provides environmental data to the SMS through the dorsal stream. These sensory data are classified based on different modalities, such as visual, tactile, etc., and sent to the triggers. The output of the SMS, a sequence of motor commands, executes using the agent’s actuators, and thereby acts on the environment. These processes occur cyclically between the environment module and the SMS, which models the hypothesis regarding one of the dorsal stream’s roles, online control.

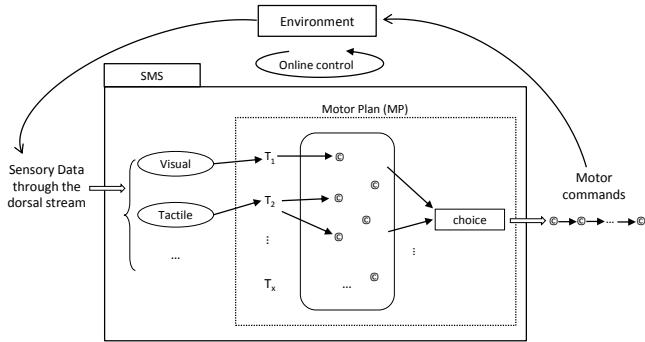


Figure 4-1: SMS with a MP and the online control process

As shown in Figure 4-1, the SMS resembles a wrapper for the MP, supporting pre-processed sensory data, and passing the MP's output to the agent's actuators.

4.2 Motor Commands

A motor command (MC) is applied to an actuator of an agent. Since they are the output of the SMS, their general format has been defined. Every MC has two components: a motor name, and a command value. The motor name indicates to which motor of an actuator the MC specifically controls, while the command value of a MC encodes the extent of the command applied to the motor.

4.3 Specification: From a Motor Plan Template to a Motor Plan

A set of motor commands (MCs) is prepared inside a Motor Plan (MP) and bound with fixed command values. In order to specify a MC's command value before the execution begins—thus modeling one of the dorsal stream's hypothesized roles, specification—a Motor Plan Template (MPT) is proposed and a specification process is created in the SMS as depicted in Figure 4-2.

A MPT is an abstract motor plan whose MCs are not yet bound with specific values. After a specification process, the motor commands inside the MPT are bound with specific command values, instantiating the MPT into a concrete MP. MPTs and MPs have very similar structures. Their major differences are 1) an MPT is persistently stored in a long-term memory, while an MP is short-term, and created anew each time it is used; and 2) most of an MP's command values have been specified, while those of an MPT have not.

Both sensory data through the dorsal stream and the selected behavior determine the specification process. As shown in the Figure 4-2, two cylinders lie under the set of motor commands (\odot s); they receive the sensed data and the context of a selected behavior separately, and provide specific command values to motor commands mainly through a *specification* process—the *update* process is another option described later. Each of these cylinders represents a set of associations; every association transforms relevant environmental features into a command value.

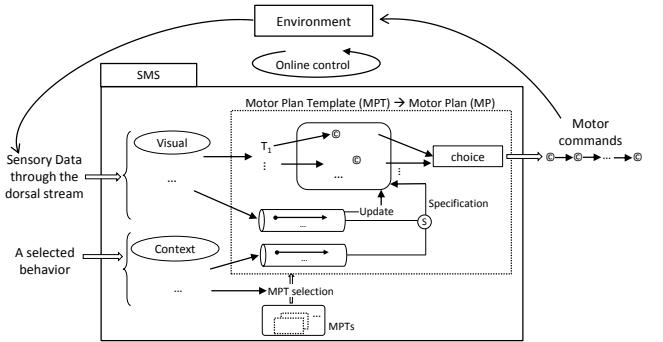


Figure 4-2: All of SMS's components. See text for details.

The data sensed through the dorsal stream provides environmental features' true value, such as a numeric value of positive five as an object's width, while the context of a selected behavior supports the semantic values “*large*” or “*small*” for the object's size. Usually, the command values specified in the motor commands are only relying on the sensory data, although the context affects the command values in a few conditions (Milner & Goodale, 2008). Accordingly, to implement the relationship of the effects of sensory data and the context, a suppress operation is represented by an encircled uppercase S in Figure 4-2.

There are some types of MCs whose command values are conceptually specified in the process of online control but not in the specification process (Grafton, 2010). To model this situation in the SMS, the pertinent command values are set with a default value in the specification process first, and are then updated—really specified—in the online control by an *update* process; it is represented in Figure 4-2 as well.

4.4 MPT Selection

A MPT awaits initiation by an incoming selected behavior before being specified as a concrete MP. From a general engineering viewpoint, a special process called *MPT selection* has been proposed. MPT selection chooses one MPT from others associated with the selected behavior.

5. Modeling the Execution of a Grip

Different actions execute variously, due to vastly different actuators, goals, or contexts. In other words, we need a Sensory Motor System (SMS) that allows the modeling of the action's distinctive characteristics in the execution process.

We have implemented a computational SMS to model the execution of a grip action inside a LIDA-based software agent. Our software robot and the experimental environment are introduced in Section 5.1. In the remaining sections, we introduce the implementations of the grip SMS's data structures (MPT and MP) and processes (Online control, Specification, and MPT selection). The implementations follow the design principle of the Herbert (Connell, 1989a) arm controller, and embody certain hypotheses and observations regarding human grip; related computational experiments are described as well.

5.1 The youBot, the LIDA Framework, and Webots

The youBot is a software robot. Its actuators are a mobile base, an arm, and two grippers. We chose this robot on the basis of its similarity to Herbert, whose arm controller serves as the prototype of the computational SMS for the execution of a grip action. As shown in Figure 5.1 (a), the youBot arm comprises multiple segments linearly connected by joints; the end segment plays the role of a hand, and two grippers are attached to it as fingers. Following the configuration of sensors in Herbert, we have extended the youBot sensors by additionally simulating two infra-red (IR) beams sensing the area in front of the hand, one IR beam between the grippers as their closing trigger, and touch sensors on the grippers (Figure 5.1 (b)).

The LIDA Framework is an underlying computational software framework. We use it to create a simulated human mind as the controller of our software robot, youBot. “[The LIDA Framework] allows the creation of new intelligent software agents and experiments based [on] the LIDA model. Its design and implementation aim to simplify this process and to permit the user to concentrate [on] the specifics of the application” (Snaider, McCall, & Franklin, 2011). The computational SMS is embedded into the Framework as a submodule for the model of a grip.

Webots is a mobile robot simulation software package. It offers an environment for rapid prototyping a 3D virtual world, an array of ready-made sensors and actuators, and programmable controllers controlling robots living in the virtual world (www.cyberbotics.com). We use Webots as an experimental environment in which to manipulate the youBot, controlled by the LIDA Framework, in order to run a computational SMS for gripping.

5.2 The Simulation of Herbert’s Arm Controller

Herbert’s arm controller drives the robot to pick up a soda can, and bring it back to a home location (Connell, 1989a).

We simulated Herbert’s arm controller⁵ in a newly created SMS embedded in a LIDA-based software agent in Webots to execute the grip. This simulation implements the *online control* process and a Motor Plan (MP) for gripping in the SMS, as designed in Section 4.1.

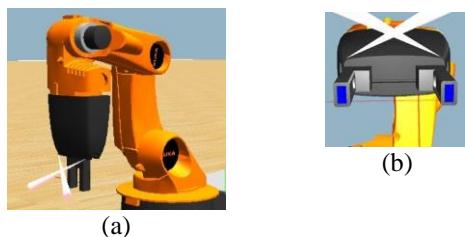


Figure 5-1: (a) A snapshot of the youBot’s arm, and (b) the touch sensors on the tip of grippers (bottom view).

⁵ In comparison with the original design (Connell, 1989b), the cradle level, the back module, and the edge module were removed in the simulation because either their function is substituted for by the Webots simulated environment, or they are irrelevant to the hand and arm actuators.

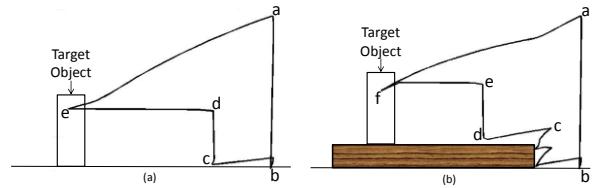


Figure 5-2: The trajectories produced by the simulated Herbert arm controller in different domains.

Two of Herbert’s grip experiments have been duplicated. Figure 5-2 (a) represents a grip retrieving an object on the ground surface. The lines show the path followed by the tips of the grippers. The grip starts from point a, going through the rest of the points exploring for the target object, and finally carrying the object back. In Figure 5-2 (b), the same controller retrieves the object from a pedestal. These simulations successfully replicate the execution of a grip action driven by the simulated Herbert arm controller, lending support to the idea of utilizing the subsumption architecture as a prototype for an SMS model of the action execution process.

5.3 Biologically Inspired Modification

The computational SMS implemented in Section 5.2, a simulated Herbert arm controller, is modified in order to implement the *specification* process and a grip Motor Plan Template (MPT) in the SMS, as designed in Section 4.3. Here, two sets of associations are created. In each of them, a single type of association is implemented, transforming the object’s width into a value for a particular motor command known as the grip aperture of the grippers.

As represented in Figure 5-3, the simulated grip aperture is sampled at unit intervals in Webots virtual time during a grip execution that is as same to the execution described by Figure 5-2 (a); these simulated grip apertures are analyzed and compared in detail below with hypotheses and observations of human gripping.

First, the grip action is executed using the unmodified arm controller as an experimental control. As shown in Figure 5-3 (a), whatever its starting value, the grip aperture almost always reaches 0.0656m (the maximum grip aperture, or MGA) before the grip closes around the target object. The grippers squeeze the target object, and thus the resulting grip aperture is smaller than the original target object width.

Second, an association (the upper cylinder in Figure 4-2) has been implemented by connecting the object’s width, as sensed through the dorsal stream, to the value of the grip aperture. As shown in Figure 5-3 (b), the grip aperture reaches the specified value of 0.03m before the value falls as the grippers close. Compared to the maximum grip aperture (MGA), the value specified here is much closer to the target object width. This simulated calibration is qualitatively the same as saying that “the dorsal stream plays a central role in the programming of actions (i.e. the pre-specification of movement parameters)” (Milner & Goodale, 2008, p. 776), because currently it is the sensory

data through the dorsal stream which affects the grip aperture's value during the specification process.

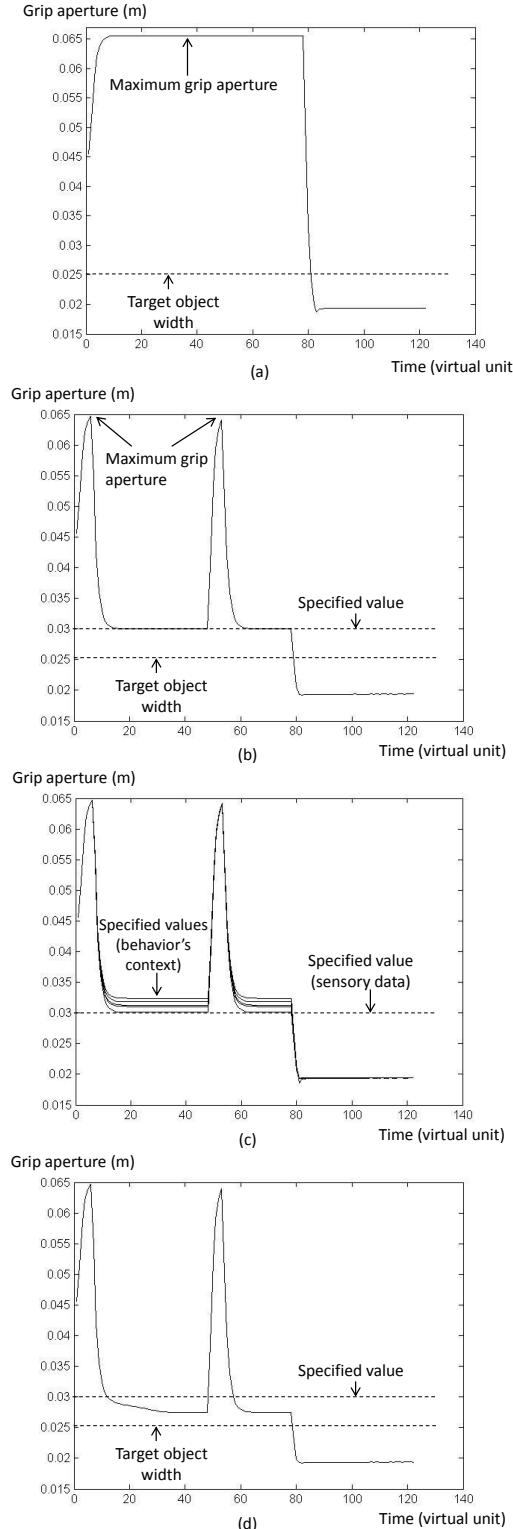


Figure 5-3: The simulated grip aperture sampled in Webots virtual time during certain grip executions.

The specified value in the simulation is larger than the object width: $0.03\text{m} > 0.025\text{m}$, since experimentally, “the [human] finger grip opens more than required by the size of the object” (Jeannerod, 1981, 2006). The first MGA peak is modeled by setting a fixed MGA value to the grip aperture for a short while when the execution starts, in keeping with the observed human behavior that people open their fingers maximum when starting a grip (Farnè, Pavani, Meneghelli, & Làdavas, 2000; Jeannerod, 2006). The second MGA peak occurs because the grippers touch the surface; this behavior both tracks the object's width value and adapts to an unpredicted collision.

Third, Instead of the data being sensed through the dorsal stream, the selected behavior's context may affects the relevant command values in several conditions (Milner & Goodale, 2008). We simulated two of these conditions: 1) Deleting the association that connects the object's width through the dorsal stream to the grip aperture, in effect rendering the skill unfamiliar to the agent, or 2) Terminating the relevant data received from the dorsal stream, so as to simulate a delay in the dorsal stream.

Additionally, another association (the bottom cylinder in Figure 4-2) has been implemented by connecting the object width represented in the context component of a selected behavior to the grip aperture value. Since the object width represented in a behavior's context is a semantic value, such as “large” or “small,” which are not precise, its value is designed to be distributed in a range, so that the represented object width approximates its true value. As shown in Figure 5-3 (c), five executions of the same grip produced a range of context-specified values rather than a precise value. We argue that these imprecise movements result from an association connecting the selected behavior's context to a command value. This interpretation of these imprecise results agrees with the conclusion we reached above that the dorsal stream plays a central role in specification process. Additional evidence is found in patients suffering from bilateral optic ataxia caused by damage to the dorsal stream—these patients show deficits in calibrating their grip aperture (Jakobson, Archibald, Carey, & Goodale, 1991; Jeannerod, Decety, & Michel, 1994; Milner & Goodale, 2008).

Fourth, an *update* process is implemented to specify the grip aperture during the execution. As shown in Figure 5-3 (d), the grip aperture value comes closer to the object width than the specified value mentioned previously in Figure 5-3 (b) and (c); it follows that the sensory data provided through the update process are more precise than the context of the specification process, because the situation becomes clearer to the agent as it executes the action.

5.4 Linking the Modified Simulation to LIDA

The grip Motor Plan Template implemented in Section 5.3 is mapped one-to-one onto the action component of a selected grip behavior. This is a simple implementation of MPT selection following the SMS concept introduced in Section 4.4.

As discussed in Section 2 and shown in Figure 4-2, both the data sensed through a dorsal stream channel and a selected behavior corresponding to a goal-directed action are input to the SMS, and the SMS's output is sent out to the LIDA Environment module. These I/Os are implemented in the LIDA-based agent including the SMS.

Additionally, in order to let the agent monitor the execution status, an expectation codelet (Faghihi, McCall, & Franklin, 2012) is created when the grip behavior is selected⁶; this codelet—a small and special purpose computational process—contains the expected result component of the currently selected behavior. It checks whether this result has been reached (sensed and recognized by the agent) at run time. The checking result is sent to LIDA's Global Workspace module, where it competes for the agent's attention (Baars, 1988). In this way, the agent's awareness of its own action execution is indirectly achieved.

6. Concluding Comments

The Sensory Motor System (SMS) is proposed to model the human action execution process. It is based on the LIDA Model, the subsumption architecture, the two visual systems, as well certain other cognitive neuroscience hypotheses. Furthermore, a computational SMS has been implemented for the execution of a grip behavior, and its simulated results have been compared to the values of the grip aperture in human gripping experiments. This biologically inspired design, together with a computational verification by comparing the model with human behaviors, supports that the SMS is a qualitatively reasonable cognitive model for the execution of a human action.

References

Baars, B. J. (1988). *A cognitive theory of consciousness*. New York: Cambridge University Press.

Baars, B. J. (2002). The conscious access hypothesis: origins and recent evidence. *Trends in cognitive sciences*, 6(1), 47-52.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, 2(1), 14-23.

Brooks, R. A. (1990). Elephants don't play chess. *Robotics and autonomous systems*, 6(1), 3-15.

Brooks, R. A. (1991). How to build complete creatures rather than isolated cognitive simulators. *Architectures for intelligence*, 225-239.

Connell, J. H. (1989a). A behavior-based arm controller. *Robotics and Automation, IEEE Transactions on*, 5(6), 784-791.

Connell, J. H. (1989b). A colony architecture for an artificial creature: DTIC Document.

Dawson, M. R. W. (n.d.). Dawson's Margin Notes On "Understanding Intelligence" (Rolf Pfeifer, Christian Scheier, 2001). from http://www.bcp.psych.ualberta.ca/~mike/Pearl_Street/Margin/Pfeifer/chap7.html

Faghihi, U., McCall, R., & Franklin, S. (2012). A computational model of attentional learning in a cognitive agent. *Biologically Inspired Cognitive Architectures*.

Farnè, A., Pavani, F., Meneghelli, F., & Làdavas, E. (2000). Left tactile extinction following visual stimulation of a rubber hand. *Brain*, 123(11), 2350-2360.

Franklin, S., & Graesser, A. (1997). Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents *Intelligent agents III agent theories, architectures, and languages* (pp. 21-35): Springer Berlin Heidelberg.

Franklin, S., Madl, T., D'Mello, S., & Snaider, J. (2013). LIDA: A Systems-level Architecture for Cognition, Emotion, and Learning. *IEEE Transactions on Autonomous Mental Development*.

Goodale, M. A., & Milner, A. D. (1992). Separate visual pathways for perception and action. *Trends in neurosciences*, 15(1), 20-25.

Grafton, S. T. (2010). The cognitive neuroscience of prehension: recent developments. *Experimental brain research*, 204(4), 475-491.

Jakobson, L., Archibald, Y., Carey, D., & Goodale, M. A. (1991). A kinematic analysis of reaching and grasping movements in a patient recovering from optic ataxia. *Neuropsychologia*, 29(8), 803-809.

James, T. W., Culham, J., Humphrey, G. K., Milner, A. D., & Goodale, M. A. (2003). Ventral occipital lesions impair object recognition but not object-directed grasping: an fMRI study. *Brain*, 126(11), 2463-2475.

Jeannerod, M. (1981). Intersegmental coordination during reaching at natural visual objects. *Attention and performance IX*, 9, 153-168.

Jeannerod, M. (2006). *Motor cognition: What actions tell the self*: Oxford University Press.

Jeannerod, M., Decety, J., & Michel, F. (1994). Impairment of grasping movements following a bilateral posterior parietal lesion. *Neuropsychologia*, 32(4), 369-380.

Milner, D., & Goodale, M. A. (2008). Two visual systems re-reviewed. *Neuropsychologia*, 46(3), 774-785.

Milner, D., Perrett, D., Johnston, R., Benson, P., Jordan, T., Heeley, D., . . . Terazzi, E. (1991). Perception and action in 'visual form agnosia'. *Brain*, 114(1), 405-428.

Simon, H. A. (1969). The sciences of the artificial. Cambridge, MA.

Snaider, J., McCall, R., & Franklin, S. (2011). The LIDA framework as a general tool for AGI *Artificial General Intelligence* (pp. 133-142): Springer Berlin Heidelberg.

www.cyberbotics.com. Webots, a commercial mobile robot simulation software developed by Cyberbotics Ltd.

⁶ At the time of submission of the paper, this work is still in process. Currently, instead of dynamically creating an expectation codelet when a behavior is selected, the codelet is simply built into the LIDA agent.