Inferring Actions and Observations from Interactions

Joseph Garnier, Olivier L. Georgeon, Amélie Cordier

Inferring Actions and Observations from Interactions

Joseph P. Garnier JOSEPH.GARNIER@LIRIS.CNRS.FR
Olivier L. Georgeon OLIVIER.GEORGEON@LIRIS.CNRS.FR
Amélie Cordier AMELIE.CORDIER@LIRIS.CNRS.FR
Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205, F-69622, France

Abstract

This study follows the Radial Interactionism (RI) cognitive modeling paradigm introduced previously by Georgeon and Aha (2013). An RI cognitive model uses sensorimotor interactions as primitives—instead of observations and actions—to represent Piagetian (1955) sensorimotor schemes. Constructivist epistemology suggests that sensorimotor schemes precede perception and knowledge of the external world. Accordingly, this paper presents a learning algorithm for an RI agent to construct observations, actions, and knowledge of rudimentary entities, from spatio-sequential regularities observed in the stream of sensorimotor interactions. Results show that the agent learns to categorize entities on the basis of the interactions that they afford, and appropriately enact sequences of interactions adapted to categories of entities. This model explains rudimentary goal construction by the fact that entities that afford desirable interactions become desirable destinations to reach.

1. Introduction

Georgeon and Aha (2013) introduced a novel approach to cognitive modeling called Radical Interactionism (RI), which invites designers of artificial agents to consider the notion of *sensorimotor interaction* as a primitive notion, instead of perception and action. A sensorimotor interaction represents an indivisible cognitive cycle, consisting of sensing, attending, and acting. Within constructivist epistemology, it corresponds to a Piagetian (1955) sensorimotor scheme from which the subject constructs knowledge of reality. RI suggests a conceptual inversion of the learning process as compared to traditional cognitive models: instead of learning sensorimotor interactions from patterns of observations and actions, RI recommends constructing observations and actions as secondary objects. This construction process rests upon regularities observed in sensorimotor experience, and happens concurrently with the construction of knowledge of the environment. Figure 1 illustrates the RI cognitive modeling paradigm.

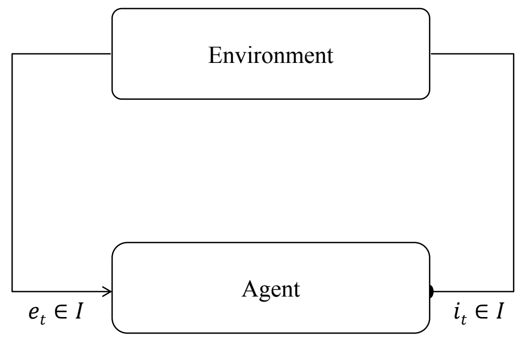

Figure 1. The Radical Interactionism modeling paradigm (adapted from Georgeon & Aha 2013). At time t, the agent chooses an *intended primitive interaction* i_t from among the set of interactions I. The attempt to enact i_t may change the environment. The agent then receives the *enacted primitive interaction* e_t. If $e_t = i_t$ then the attempt to enact i_t is considered a *success*, otherwise, a *failure*. The agent's "perception of its environment" is an internal construct rather than the input e_t.

The algorithm begins with a predefined set of sensorimotor interactions I, called *primitive interactions*. At time t, the agent chooses a primitive interaction i_t that it intends to enact, from among I. The agent ignores this enaction's meaning; that is, the agent has no rules that would exploit knowledge of how the designer programmed the primitive interactions through actuator movements and sensory feedback (such as: "if a specific interaction was enacted then perform a specific computation"). As a response from the tentative enaction of i_t, the agent receives the *enacted interaction* e_t, which may differ from i_t. The enacted interaction is the only data available to the agent that carries some information about the external world, but the agent ignores the meaning of this information.

An RI agent is programmed to learn to anticipate the enacted interactions that will result from its intentions, and to tend to select intended interactions that are expected to succeed ($e_t = i_t$). Such a behavior selection mechanism implements a type of self-motivation called autotelic motivation (the motivation of being "in control" of one's activity, Steels, 2004). Additionally, the designer associates a numerical valence with primitive interactions, which defines the agent's behavioral preferences (some primitive interactions that the agent innately *likes* or *dislikes*). Amongst sequences of interactions that are expected to succeed, an RI agent selects those that have the highest total valence, which implements an additional type of self-motivation called interactional motivation (Georgeon, Marshall, & Gay, 2012).

Our previous RI agents (Georgeon & Ritter, 2012; Georgeon, Marshall, & Manzotti, 2013) learned to organize their behaviors so as to exhibit rudimentary autotelic and interactional motivation without constructing explicit observations and actions. Here we introduce an extension to construct instances of objects (in the object-oriented programming sense of "object") that represent explicit observations and actions learned through experience. Our motivation is to design future RI agents that will use these to learn more sophisticated knowledge of their environment and develop smarter behaviors. In particular, we address the problem of autonomous goal construction by modeling how an observable entity in the environment that affords positive interactions can become a desirable destination to reach.

2. Agent

Our agent has a rudimentary visual system that generates *visual interactions* with entities present in the environment. A visual interaction is a sort of sensorimotor interaction generated by the relative displacement of an entity in the agent's visual field as the agent moves. The agent is made aware of the approximate relative direction of the enacted visual interaction e_t by being provided with the angular quadrant ρ_t in which e_t was enacted. Additionally, the agent is made aware of its displacement in space through the angle of rotation $\theta_t \in \mathbb{R}$ induced by the enaction of e_t. The information θ_t corresponds to the information of relative rotation given by the vestibular system in animals. It can be obtained through an accelerometer in robots. Figure 2 illustrates these additions to the RI model.

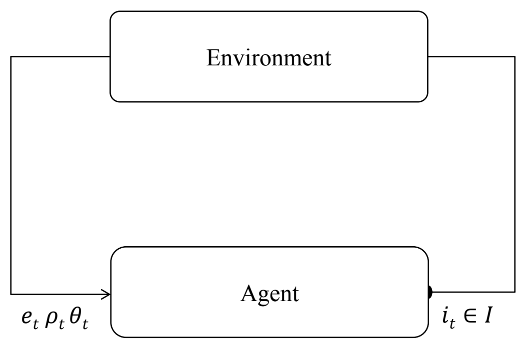

Figure 2. The Directional Radical Interactionism (DRI) model. Compared to RI (Figure 1), the DRI model provides additional directional information ρ_t and θ_t when a primitive interaction e_t is enacted at time t. ρ_t represents the directional quadrant where the interaction e_t is enacted relative to the agent, and θ_t the angle of rotation of the environment relative to the agent, generated by the enaction of e_t.

3. Experiment

We propose an implementation (see Figure 3) using the DRI model to study how agents constructs observations and actions from spatio-sequential regularities observed in its stream of sensorimotor interactions. This experiment was implemented in Java in our environment using the Enactive Cognitive Architecture (ECA). ECA is a cognitive architecture based on sensorimotor modeling, inspired by the Theory of Enaction, to control an agent that learns to fulfill its autotelic and interactional motivation. Also, ECA allows implementing self-motivation in the agent[1](Georgeon, Marshall, & Manzotti, 2013). The environment consists of a grid of *empty cells* (white squares) where the *agent* (represented by the brown arrowhead) tries to move one cell forward, turn to left or to the right. The experimenter can flip any cell from empty to wall or vice versa by clicking on it at any time. Also, the environment is composed of walls (gray squares) where the agent could bump if it tries to move through them.

1. http://e-ernest.blogspot.fr/2013/09/ernest-12.html

The agent has a rudimentary distal sensory system was inspired by the visual system of an arachaic arthropod, the limulus: the limulus's eyes responds to movement, and the limulus has to move to "see" immobile things. The agent "likes" to eat blue fish (called target). When the agent reaches a target, the target disappears as if the agent had eaten it. The experimenter can introduce other targets by clicking on the grid. The agent's visual system consists of one simple detector (violet half-circle on the agent) for detecting target. His detector covers a $180°$ span. This visual system is not sensitive to static elements of the visual field (such as the presence and the position of the target) but to changes in the visual field as the agent moves: closer, appears, unchanged and disappeared. Moreover, the agent divides his visual field in three area: A, B and C. These area inform the agent in which directional quadrant the entity is detected.

The designer can also specify the numerical valence associated with primitive interactions before running the simulation. The values chosen implement a behavioral proclivity to move towards targets because the agent has positive satisfaction when the targets appears or closer, and negative satisfaction when the target disappears.

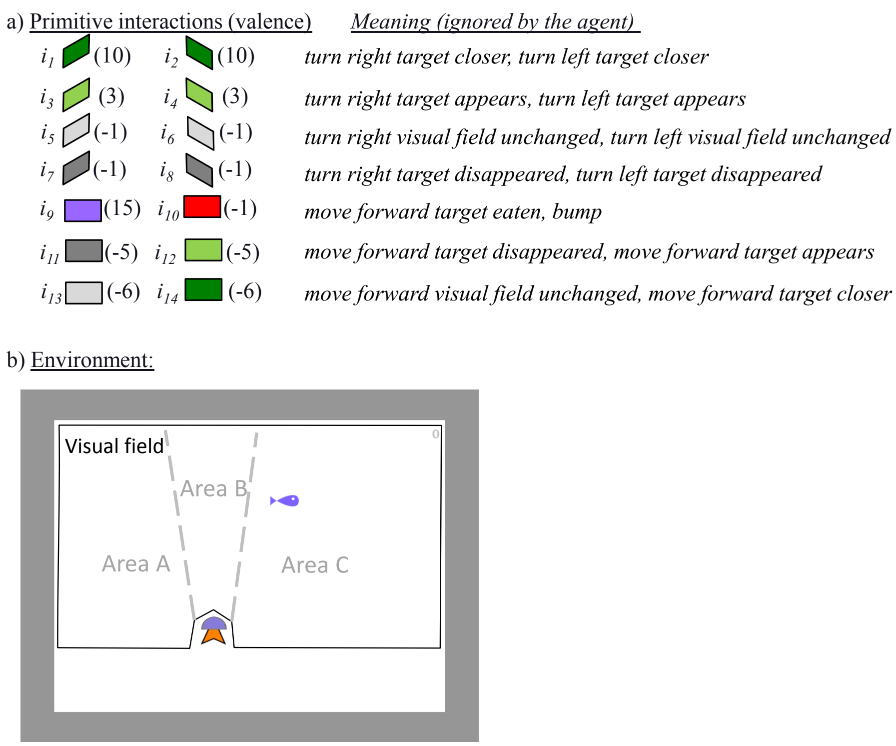

Figure 3. a) The 14 primitive interactions available to the agent with their numerical valence in parentheses, set by the experimenter. This valence system implements the motivation to move towards targets because the valence is positive when the target appears or approaches, and negative when the target disappears. b) The agent in the environment with the agent's visual field overprinted. There are three directional quadrants in which visual interactions can be localized: $\rho_t \in \{A, B, C\}$. Non-visual interactions are localized in a fourth abstract quadrant labeled "O".

To understand how the agent, during interactions with the environment, constructs its actions and its observations, we propose a simplified UML model and an example in Figure 4, and finally the algorithm.

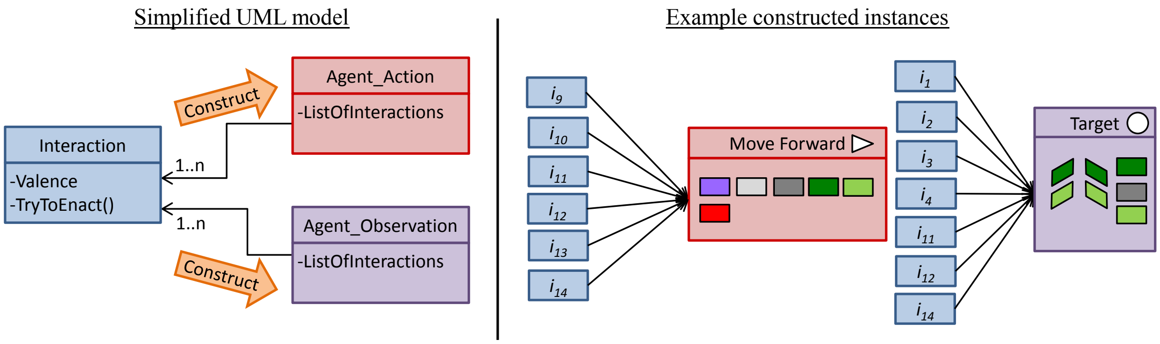

Figure 4. Simplified UML model (left): the modeler defines primitive interactions as instances of subclasses of the *Interaction* class (left) and programs their effects in the *TryToEnact()* method. The agent constructs actions as instances of the *Agent_Action* class (top-right) and observations as instances of the *Agent_Observation* class (bottom-right) from sequential and spatial regularities observed while enacting interactions. Example constructed instances (right): the action *Move Forward* can be enacted through the interactions $i_9, i_{10}, i_{11}, i_{12}, i_{13}, i_{14}$. The observation *Target* affords interactions $i_1, i_2, i_3, i_4, i_{11}, i_{12}, i_{14}$.

To interact with the environment, the agent utilizes a set of primitive interactions defined by the designer. The designer programs primitive interactions in a way that involves both commending motors and reading sensors. But, the agent originally ignores this distinction and must learn that some interactions inform it about the presence of an entity in its surrounding space, while simultaneously learning to categorize these entities. Each interaction can be afforded by a specific type of entity. In using the model DRI, see section 2, at decision step t, the agent tries to enact an *intended interaction* i_t and get the actually enacted interaction, *enacted interaction*, e_t at the end of step t. If the enacted interaction differs from the intended interaction ($e_t \neq i_t$) then the agent considers that these interactions produce two different actions a_1, a_2. Thus, a first action is represented by interaction e_t and a second action is represented by interaction i_t ($a_1 = \{e_t\}$ and $a_2 = \{i_t\}$). In case of $e_t = i_t$, the agent considers that these interactions produce the same action, which can be represented by the set of these interactions ($a_1 = a_2 = \{e_t, i_t\}$).

A type of entity present in the world affords a collection of interactions. When a set of interactions consistently overlaps in space, the agent infers the existence of a kind of entity that affords these interactions. To be concrete, a physical object would be an entity that is solid and persistent. The agent uses spatial information from DRI model to learn to categorize the entity with it can interact, according to the collection of interactions that this entity affords. At decision step t the agent tries to enact an *intended interaction* i_t and get the interaction effectively enacted, *enacted interaction*, e_t at the end of step t. In each *enacted interaction* there is the directional quadrants (A, B, C or O) where it enacted. If the enacted interaction e_t is in the same area that enacted interaction e_{t-1} then the agent considers that these interactions are afforded by the same entity

($entity_1 = entity_2 = \{e_t, e_{t-1}\}$). In case of these interactions are enacted in two different area, the agent infer it exists two kind of entity ($entity_1 = \{e_t\}$ and $entity_2 = \{e_{t-1}\}$).

4. Result

During the learning phase, the agent learns a behavior that it then uses to reach subsequent targets introduced by the experimenter. Different instances of agents may learn different behaviors as a result of having different learning experiences. Figure 5 and 6 show traces of two behaviors learned by two different agents. Once a behavior has been learned, the agent keeps using it indefinitely to reach subsequent targets.

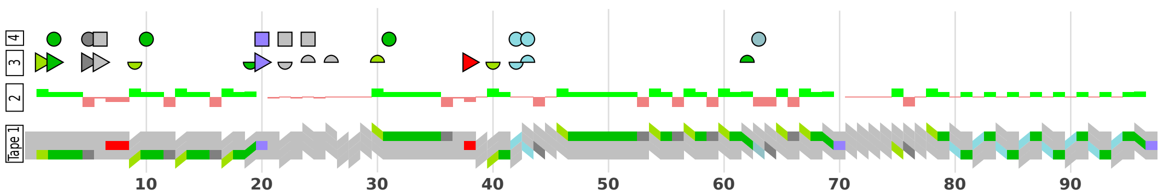

Figure 5. First 97 steps in Example 1. Tape 1 represents the primitive interactions enacted, in directional quadrant A (top), B (center), C (bottom), with the same symbols as in Figure 3. Tape 2 represents the valence of the enacted primitive interactions as a bar graph (green when positive, red when negative). Tape 3 represents the progressive aggregations of interactions to form actions. The shape represents the action and the color is the color of the enacted interaction aggregated to this action at a particular time step. The triangles correspond to the *move forward* action, the inferior half-circles to the *turn right* action, and the superior half-circles to the *turn left* action. Tape 4 represents the progressive aggregation of interactions to form observations. The shape represents the category of observation and the color is the color of the enacted interaction aggregated to this category of observation at a particular time step. The circles represent the observation of a *target*, and the squares the observation of *void*. The agent also constructs a third category of observation: the observation of *walls*. However, since walls are only observable through a single interaction (i_{10}, red rectangles), there is no aggregation of other interactions to the wall observation. In this example, the agent ate the first target on step 20 (blue rectangle in Tape 1). The experimenter introduced the second target on step 30, and the agent ate it on step 70. The third target was introduced on step 74 and eaten on step 97. The agent learned to reach the target through a "stair step" behavior consisting of repeating the sequence turn left - move forward - turn right - move forward, until it aligns itself with the target and then keeps moving forward until it reaches the target (steps 78 to 97).

A different choice of valence or modification of the environment by the experimenter at different times shows that behavior depends on the motivation that drives the agent and environment configuration. For example, if the experimenter add a target earlier than in Example 1, the agent acts differently. This behavior has been observed in Experiment 2 illustrated by the example trace in Figure 6.

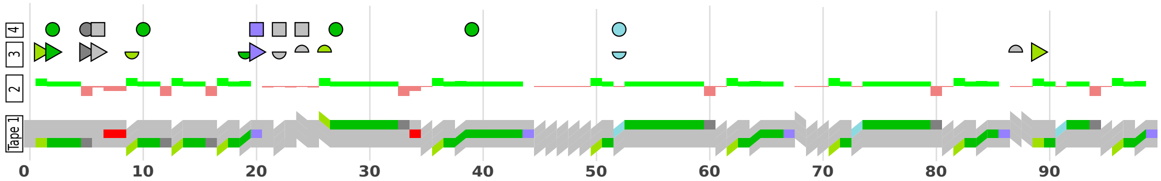

Figure 6. First 99 steps in Example 2. The behavior is the same as in Example 1 up to step 25. The experimenter introduced the second target on step 26 rather than 30 in Example 1. This difference caused the agent to learn a different behavior to reach the target, consisting in moving in a straight line until the target disappears from the visual field, then getting aligned with the target by enacting the sequence turn right – turn right – move forward – turn right, then keeping moving forward until it reaches the target (episodes 26 to 44, 50 to 67, 71 to 86 and 89 to 99).

This experiment also demonstrates the interesting property of individuation: different instances of agents with the same algorithm may learn different behaviors due to the specific learning experiences that they had. From step 26, behaviors are different. Such individuation occurs through "en habitus deposition" as conceptualized by the theory of enaction.

5. Conclusion

This work addresses the problem of implementing agents that learn to *master the sensorimotor contingencies* afforded by their coupling with their environment (O'Regan & Noë, 2001). In our approach, the modeler specifies the low-level sensorimotor contingencies through a set of sensorimotor interactions, which corresponds to what Buhrmann, Di Paolo, and Barandiaran (2013) have called the *sensorimotor environment*. The learning consists for the agent to simultaneously learn actions and categories of observable entities as second-order constructs. Here, we use the concept of action in its cognitive sense of "intentional action" (Engel et al., 2013). Our algorithm offers a solution to implements Engel et al.'s (2013, p203) view that "agents first exercise sensorimotor contingencies, that is, they learn to associate movements with their outcomes, such as ensuing sensory changes. Subsequently, the learned patterns can be used for action selection and eventually enable the deployment of intentional action".

Our agent has no pre-implemented strategy to fulfill his inborn motivation (approaching the target). We show two examples in which the agent learns two different deployments of actions to fulfill this motivation (Figure 5 and 6). These deployments of actions can be considered intentional because the agent anticipates the consequences of actions and use anticipation to select actions. In future studies, we plan on designing agents capable of reasoning upon their intentionality to learn to explicitly consider observable entities as possible goals to reach. We expect that emergent intentionality associated with explicit goal construction will make the agents capable of exhibiting more sophisticated behaviors in more complex environments, and contribute more broadly to the research effort on goal reasoning.

Acknowledgements

This work was supported by the French Agence Nationale de la Recherche (ANR) contract ANR-10-PDOC-007-01.

References

Buhrmann, T., Di Paolo, E. A., & Barandiaran, X. (2013). A dynamical systems account of sensorimotor contingencies. *Frontiers in psychology*, *4*.

Engel, A. K., Maye, A., Kurthen, M., & König, P. (2013). Where's the action? The pragmatic turn in cognitive science. *Trends in Cognitive Sciences*, *17*, 202–209.

Georgeon, O. L., & Aha, D. (2013). The radical interactionism conceptual commitment. *Journal of Artificial General Intelligence*, In press.

Georgeon, O. L., Marshall, J. B., & Gay, S. (2012). Interactional Motivation in Artificial Systems: Between Extrinsic and Intrinsic Motivation. *International Conference on Development and Learning (ICDL-Epirob)*.

Georgeon, O. L., Marshall, J. B., & Manzotti, R. (2013). ECA: An enactivist cognitive architecture based on sensorimotor modeling. *Biologically Inspired Cognitive Architectures*, *6*, 46–57.

Georgeon, O. L., & Ritter, F. E. (2012). An Intrinsically-Motivated Schema Mechanism to Model and Simulate Emergent Cognition. *Cognitive Systems Research*, *15-16*, 73–92.

O'Regan, J. K., & Noë, A. (2001). A sensorimotor account of vision and visual consciousness. *Behavioral and Brain Sciences*, *24*, 939–972.

Piaget, J. (1955). *The construction of reality in the child*. London: Routledge and Kegan Paul.

Steels, L. (2004). The Autotelic Principle. In F. Iida, R. Pfeifer, L. Steels, & Y. Kuniyoshi (Eds.), *Embodied artificial intelligence*, Vol. 3139 of *Lecture Notes in Computer Science*, 231–242. Springer Berlin Heidelberg.