Demonstrating sensemaking emergence in artificial agents: A method and an example

Olivier L. Georgeon, James Marshall

International Journal of Machine Consciousness
© World Scientific Publishing Company

DEMONSTRATING SENSEMAKING EMERGENCE IN ARTIFICIAL AGENTS: A METHOD AND AN EXAMPLE

OLIVIER L. GEORGEON

Université de Lyon,
Université Lyon 1, LIRIS, UMR5205, F-69622, France
olivier.georgeon@liris.cnrs.fr

JAMES B. MARSHALL

Sarah Lawrence College,
Bronxville, NY 10708, USA
jmarshall@sarahlawrence.edu

Received 11 February 2013
Revised 11 February 2013

We propose an experimental method to study the possible emergence of sensemaking in artificial agents. This method involves analyzing the agent's behavior in a test bed environment that presents regularities in the possibilities of interaction afforded to the agent, while the agent has no presuppositions about the underlying functioning of the environment that explains such regularities. We propose a particular environment that permits such an experiment, called the Small Loop Problem. We argue that the agent's behavior demonstrates sensemaking if the agent learns to exploit regularities of interaction to fulfill its self-motivation as if it understood (at least partially) the underlying functioning of the environment. As a corollary, we argue that sensemaking and self-motivation come together. We propose a new method to generate self-motivation in an artificial agent called interactional motivation. An interactionally motivated agent seeks to perform interactions with predefined positive values and to avoid interactions with predefined negative values. We applied the proposed sensemaking emergence demonstration method to an agent implemented previously, and produced example reports that suggest that this agent is capable of a rudimentary form of sensemaking.

Keywords: sensemaking; self-motivation; emergence; behavioral analysis.

1. Introduction

We address the problem of demonstrating the potential capability of an agent to *make sense* of information when it interacts with its environment. We refer to this problem as the *sensemaking demonstration problem*. The term *sensemaking* is used here according to its common definition of *giving meaning to experience* or *understanding experience*. We consider the problem of machine sensemaking as a particular aspect of the broader problem of machine consciousness because being able to make sense of information is an important part of what it is to be conscious.

Here we address the issue of *rudimentary sensemaking* as it may be exhibited by rudimentary systems as opposed to higher-level sensemaking such as understanding human language. Rudimentary sensemaking refers to the capability of an agent to make sense of basic elements of its experience interacting with the environment. Because the study of rudimentary sensemaking focuses on the study of cognition through the agent's experience of interaction, it falls within the field of situated cognition [e.g., Suchman, 1987] and embodied cognition [e.g., Pfeifer & Bongard, 2006].

The *sensemaking demonstration problem* is different from the problem of designing an agent that has sensemaking capabilities; it is the problem of assessing the capabilities of an existing agent. To address this problem, we acknowledge the fact that sensemaking is an emergent property of an agent interacting with an environment. As an emergent property, sensemaking can potentially be observed in the agent as a whole but cannot be traced to the agent's components. Because the demonstration of sensemaking is based on the observation of the whole agent rather than on the analysis of its components, it belongs to the experimental sciences. Our approach consists of taking an already-designed agent as an object of study and empirically investigating whether or not this agent is capable of some form of sensemaking.

In essence, we seek to produce evidence to support or contradict the claim that a given agent is capable of giving meaning to its experience; and, if it is, to clarify the form and the extent of this capability on the basis of explicit arguments. We argue that addressing this problem entails three aspects: one aspect consists of defining what may constitute acceptable evidence of sensemaking for the scientific community—typically, the cognitive science, artificial intelligence, and machine consciousness communities. Another aspect consists of designing an experiment that offers opportunities for the agent to demonstrate its sensemaking capabilities. A third aspect consists of generating acceptable evidence that the agent did indeed exhibit sensemaking during the experiment. These three aspects are intertwined: designing experiments will help clarify what we expect from a sensemaking agent, and reporting evidence will help clarify what kind of evidence is acceptable to the scientific community.

Applying experimental science to an object that was designed by humans raises the difficulty of separating design from experiment. This difficulty was captured a long time ago by John von Neumann: "if you will tell me precisely what it is that a machine cannot do, then I can always make a machine that will do just that" [e.g., cited by Samsonovich, 2012, p105]. This remark applies to the sensemaking demonstration problem: if we define precisely an experiment and specify precisely the results that we seek to observe, then one can always design an algorithm that will generate just these results. In this case, these results will not constitute evidence of sensemaking because they will be generated by a specifically designed algorithm for this precise experiment rather than arising out of the algorithm's execution in an initially unknown experiment. This remark suggests that emergence is not only a property but also a requirement of sensemaking: the experimental results should be emergent in the sense that it should not be possible to track their causes down to the algorithm's logic. Evidence that the results are emergent can be given by a description of the algorithm provided by the designer. This description must show that the algorithm was not specifically designed to generate these results in this specific experiment but rather that the algorithm has no direct causal efficacy to explain the results.

Philosophers of mind have been discussing sensemaking for a long time and some of their results can help specify the type of observations that may constitute evidence of sensemaking. Specifically, we subscribe to pragmatic epistemology [e.g., James, 1907; Wittgenstein, 1953] that suggests that meaning is given by usage. This idea has been summarized by Wittgenstein's famous quote "meaning is use". Following this idea, we acknowledge that the possibility for an observer to consider that an agent makes sense of an experience comes together with the possibility for the observer to consider that the agent has its own motivation. Based on this conception of sensemaking coming from pragmatic epistemology, we address the problem of the emergence of sensemaking jointly with the problem of self-motivation in artificial systems. We seek to produce evidence that the agent understands experiences based on evidence that the agent has a form of self-motivation, and on evidence that the agent is capable of using such experiences for the purpose of fulfilling its self-motivation.

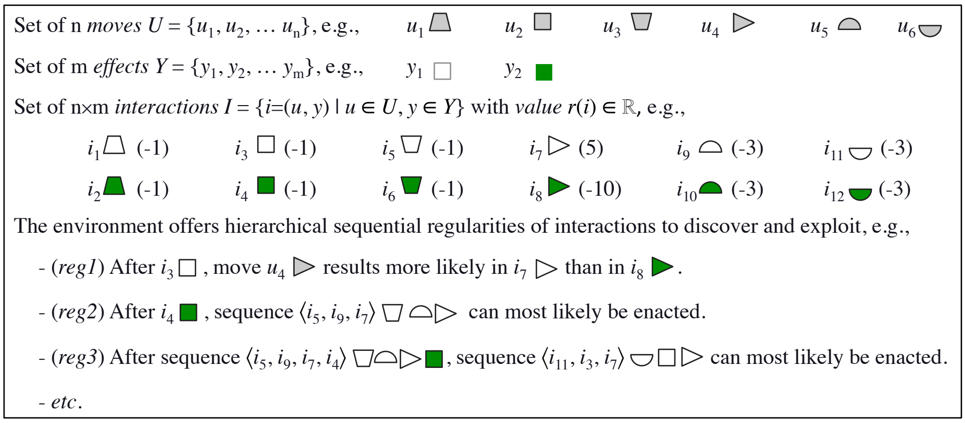
Several solutions for implementing self-motivation in artificial systems have been proposed. For example, Oudeyer, Kaplan, and Hafner [2007] proposed a motivational system based on curiosity; Blank, Lewis, and Marshall [2005] proposed the search for predictability and control; Schmidhuber [2010] proposed the search for simplification of knowledge and compressibility of data. These authors have contributed to defining the term *intrinsic motivation* as opposed to *extrinsic motivation*, which comes from an external goal defined by the designer. Indeed, specifying a final goal to reach in the environment makes the agent's motivation appear (from the observer's perspective) as if it were specified by the designer rather than coming from the agent itself. To provide the opportunity for the agent to exhibit self-motivation, we designed an experiment in which the agent's behavior is driven by preferences of interactions rather than by a final goal to reach.

For the agent to exhibit sensemaking in addition to self-motivation, we expect the agent to demonstrate a form of progressive organization of behavior through its experience of interacting with the environment. Indeed, demonstrating that the agent makes sense of its experience implies that the agent initially doesn't understand its experience but actively engages in the process of understanding it. Offering the opportunity for sensemaking implies providing the agent with an environment that affords regularities of interaction that the agent can discover, memorize, and exploit to fulfill its self-motivation.

We make the hypothesis that providing the agent with an environment that affords hierarchies of sequential regularities of interaction for the agent to discover and exploit is suitable to demonstrate a rudimentary form of sensemaking. This hypothesis is supported by studies in artificial cognition that suggest that hierarchical sequence learning is a key feature of cognition [e.g., Sun & Giles, 2000], and, more broadly, by constructivist epistemology that suggests that knowledge arises from the discovery and exploitation of sensorimotor regularities of interaction with the world [e.g., Piaget, 1951]. Section 2 presents the general formalism of such experiments, and Sections 3 and 4 present an experiment involving a specific agent as an example.

2. Hierarchical sequential regularity learning based on interactional motivation

To study rudimentary sensemaking, we place the agent in the situation of interacting with an environment that affords hierarchical sequential regularities of interaction to discover and exploit. Additionally, we offer the possibility for the experimenter to attribute scalar values to the possible interactions afforded to the agent by the environment. Values associated with interaction patterns allow the experimenter to define innate preferences: seeking interactions with positive values and avoiding interactions with negative values, without defining a final goal state to reach. We call this form of self-motivation interactional motivation [Georgeon *et al.*, 2012] because it is neither purely internal to the agent (as is curiosity) nor external (as is a goal state to reach in the environment). Fig. 1 illustrates the Hierarchical Sequential Interactional Motivation Learning problem (HSIML) with an example.



Set of n *moves* $U = \{u_1, u_2, \dots u_n\}$, e.g., u_1 ⏢ u_2 ■ u_3 ⏷ u_4 ▷ u_5 ◠ u_6 ◡

Set of m *effects* $Y = \{y_1, y_2, \dots y_m\}$, e.g., y_1 □ y_2 ■

Set of n×m *interactions* $I = \{i=(u, y) \mid u \in U, y \in Y\}$ with *value* $r(i) \in \mathbb{R}$, e.g.,

i_1 ⏢ (-1) i_3 □ (-1) i_5 ⏷ (-1) i_7 ▷ (5) i_9 ◠ (-3) i_{11} ◡ (-3)

i_2 ⏢ (-1) i_4 ■ (-1) i_6 ⏷ (-1) i_8 ▶ (-10) i_{10} ◠ (-3) i_{12} ◡ (-3)

The environment offers hierarchical sequential regularities of interactions to discover and exploit, e.g.,

- (*reg1*) After i_3 □ , move u_4 ▷ results more likely in i_7 ▷ than in i_8 ▶.
- (*reg2*) After i_4 ■ , sequence $\langle i_5, i_9, i_7 \rangle$ ⏷◠▷ can most likely be enacted.
- (*reg3*) After sequence $\langle i_5, i_9, i_7, i_4 \rangle$ ⏷◠▷■, sequence $\langle i_{11}, i_3, i_7 \rangle$ ◡□▷ can most likely be enacted.
- *etc*.

Fig. 1: Example of Hierarchical Sequential Interactional Motivation Learning (HSIML) problem. On each decision cycle t, the agent chooses a move u_t from amongst the n possible moves in the set U, represented by grey symbols. As a result of the chosen move u_t and the state of the environment, the agent receives an effect y_t from amongst the m possible effects in the set Y, represented by a color. Each pair $i = (u,y)$ forms a *primitive interaction* ($i \in I = U \times Y$), represented by a symbol whose shape represents the move u and whose color represents the effect y. A scalar value $r(i)$ is attached to each primitive interaction, represented by values in parentheses next to the interactions. The agent seeks to enact interactions with positive values (in this example: only i_7 (5)) and to avoid interactions with negative values as much as possible. The environment offers hierarchies of sequential regularities to which the agent is sensitive. We expect the agent to discover, memorize, and exploit these regularities so that it can learn to enact interactions with positive values, or slightly negative values that subsequently lead to more positive interactions, and to avoid interactions with larger negative values. For example, we expect the agent, in discovering and exploiting (*reg1*), to choose move u_2, and, if this results in interaction i_3, to subsequently choose move u_4 so as to safely enact i_7 which has a positive value, thus avoiding i_8 that has a very negative value. Regularities have a hierarchical structure: for example, the learned sequence $\langle i_3, i_7 \rangle$ used to exploit (*reg1*) is a subsequence of the (*reg3*) sequence $\langle i_{11}, i_3, i_7 \rangle$, and the sequence $\langle i_5, i_9, i_7 \rangle$ in (*reg2*) is a subsequence of the (*reg3*) sequence $\langle i_5, i_9, i_{7,} i_4 \rangle$. In turn, sequences in (*reg3*) can constitute subsequences involved in higher-level regularities. The agent can begin by discovering and exploiting lower-level regularities, then learn higher-level regularities from sequences of lower-level regularities.

The diagram in Fig. 2 represents the formalism for the HSIML problem in a form similar to Partially Observable Markov Decision Processes (POMDPs) [e.g., Aström, 1995] to allow for comparison.

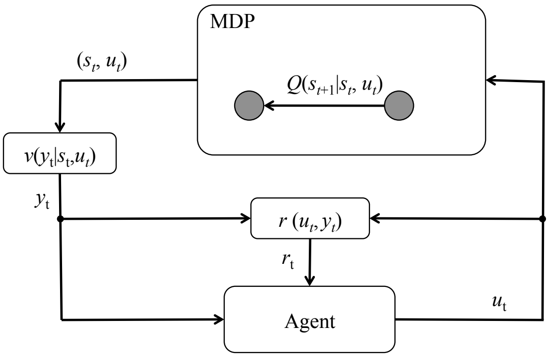


Fig. 2: Formal diagram of the Hierarchical Sequential Interactional Motivation Learning (HSIML) problem. In contrast to the traditional reinforcement learning cycle, the interaction cycle starts with the agent rather than the environment. At time t, the agent chooses a move u_t, then receives an effect y_t that depends on the state of the environment s_t and on the move u_t, according to the probability distribution $v(y_t|s_t,u_t)$. The Markov Decision Process (MDP) represents the environment, which is partially hidden to the agent by the probability distribution v, as it is in a Partially Observable Markov Decision Process (POMDP) problem. A crucial difference, however, is that y_t does not depend only on s_t but also on u_t. The pair (u_t,y_t) is called a primitive interaction. A value function $r(u_t,y_t)$ is associated with the interaction (u_t,y_t) rather than with the environment state as in the POMDP problem. The move u_t changes the environment state from s_t to s_{t+1} according to the distribution $Q(s_{t+1}|s_t,u_t)$.

Formally, the HSIML is defined as follows:
- $|S|$ states $S = \{s_1, ... ,s_{|S|}\}$ of the environment;
- n moves $U = \{u_1, \dots , u_n\}$ available to the agent;
- m effects $Y = \{y_1, ... ,y_m\}$;
- A value function $r(u,y) \in \mathbb{R}$ for each primitive interaction $(u,y) \in U \times Y$.

An HSIML evolves in a similar way as a POMDP, but we start the description from the agent's move because the effect conceptually results from the move, as opposed to a POMDP where the action conceptually results from the observation. Notably, this conceptual inversion finds some resonance with Pfeifer and Scheier's [1994] questioning of the traditional perception-action cycle. An HSIML evolves as follows:

1. Let $t = 0$ and let s_0 denote the initial environment state.
2. At time step t, generate move u_t according to the agent's policy μ.
3. Generate a new effect y_t with probability $v(y_t|s_t,u_t)$. The interaction i_t=(u_t,y_t) is called the *enacted interaction* at time t.
4. Generate the satisfaction value $r_t = r(u_t,y_t)$ associated with the enacted interaction.
5. Generate a new environment state s_{t+1} with probability $Q(s_{t+1}|s_t,u_t)$.
6. $t = t + 1$.
7. Goto 2.

The fact that the environment affords sequential regularities of interaction means that the probability distributions Q and v are such that pairs of sequences of interactions $\langle\sigma_1,\sigma_2\rangle$ exist such that, when σ_1 is enacted, it is probable that σ_2 can subsequently be enacted. σ_1 and σ_2 are sequences of primitive interactions (e.g., $\sigma_1 = \langle i_1,i_2,\ldots,\mathrm{i}_k\rangle$). Enacting a sequence of interactions means sequentially enacting all of its primitive interactions. Enacting a primitive interaction i=(u,y) means performing move u and obtaining effect y. Notably, the agent may intend to enact primitive interaction i but end up enacting a different primitive interaction if the resulting effect differs from the expected effect y. The fact that sequential regularities have a hierarchical structure means that a regularity $\sigma = \langle\sigma_1,\sigma_2\rangle$ can be part of a higher-level regularity (e.g., $\sigma_5 = \langle\sigma_3,\sigma_4\rangle$ with $\sigma_3 = \langle i_1,\ldots, \sigma,\ldots,\mathrm{i}_k\rangle$). We define HSIML problems as any problem that affords such hierarchical sequential regularities of interactions and that can be represented by the formalism presented above (Fig. 2). Section 3 presents an agent that we developed to address this class of problems.

3. A sensorimotor self-programming agent

We designed an agent called Ernest that addresses HSIML problems as presented in Section 2. A detailed description of Ernest's algorithm is provided elsewhere [Georgeon & Ritter, 2012]. In accordance with the principles of the *sensemaking demonstration problem* presented in the introduction, we present here the principle of Ernest's algorithm in order to show that Ernest is, by design, capable of addressing HSIML problems without presuppositions about the underlying functioning of the environment that causes the regularities of interaction.

Conceptually, Ernest relies on a model of cognition based on sensorimotor patterns rather than on the traditional "cognitive loop" that separates perception, cognition, and action. This model offers a framework for conceptualizing the agent's ability to perform a form of *sensorimotor self-programming*. We use the term sensorimotor self-programming because Ernest records sequences of sensorimotor patterns, and learns to recognize the context in which these sequence can subsequently be *executed* as programs. The context is encoded in the form of sensorimotor patterns, which requires no assumption about the underlying structure of the environment. This learning is bottom-up, meaning that Ernest initially learns short sequences of behaviors adapted to short sequential regularities afforded by the environment. Once Ernest masters the short-term regularities, it can discover and learn higher-level regularities that consist of sequences of shorter-term regularities. This hierarchical organization helps Ernest deal with the combinatorial complexity of the particular HSIML problem with which it is confronted. Fig. 3 illustrates the sensorimotor self-programming model.

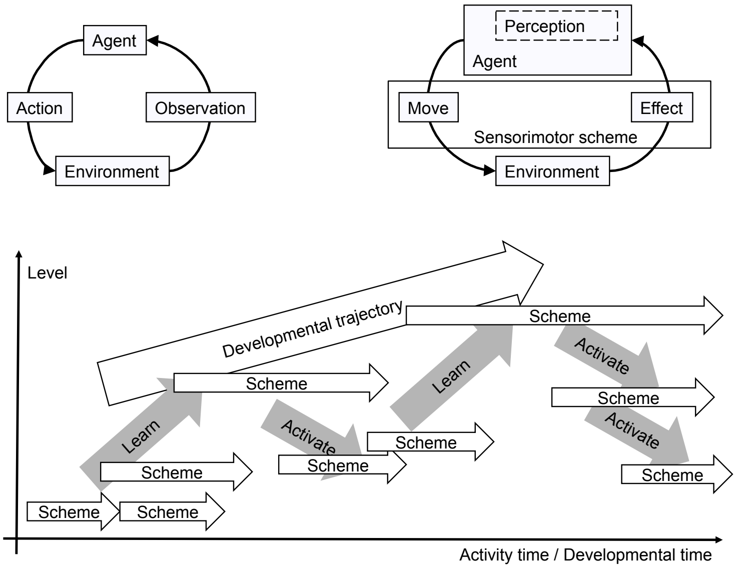
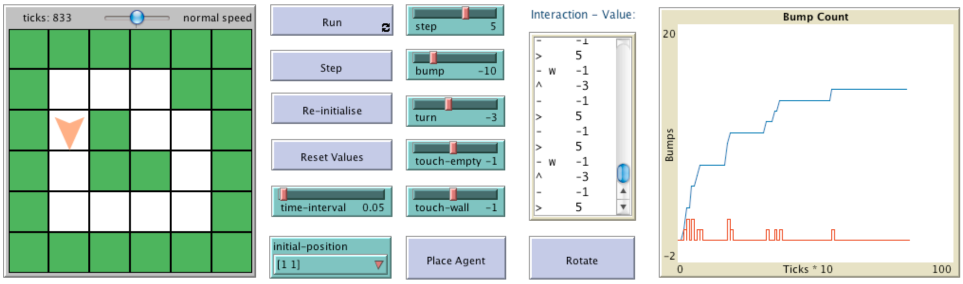


Fig. 3: From the *Cognitive Loop Model* to the *Sensorimotor Self-Programming Model*. Top left: the *Cognitive Loop Model*: the input to the agent is considered the agent's "perception"; the agent interprets its perception in order to choose its next action. Top right: the input to a sensorimotor self-programming agent is an "effect" produced by performing a "move" in the environment (the effect depends not only on the state of the environment but also on the actuator that was moved). The pair (move, effect) is called a *primitive sensorimotor scheme*. The agent's perception is an internal structure that is gradually constructed as sensorimotor schemes are enacted. The agent represents its current situation in terms of schemes, that is, in terms of possibilities of interaction. The agent chooses the next sensorimotor schemes to enact based on its anticipation of the resulting effects. Bottom: the *Sensorimotor Self-Programming Model*: the sensorimotor loop is "unfolded" to display two time scales: the activity time and the developmental time. Primitive schemes, i.e., (move, effect) pairs, that are enacted during the activity time are represented by short arrows at the bottom of the diagram. As the developmental time goes on, the agent learns higher-level schemes (longer arrows, higher in the diagram). Higher-level schemes are hierarchical compositions of lower-level schemes matched to regularities discovered along the activity time. As the agent develops, it engages in increasingly complex activities made possible by the high-level schemes that were previously learned. Enacting a high-level scheme consists of sequentially activating and enacting its sub-schemes, all the way down to predefined primitive schemes. Therefore, schemes work as executable programs that are autonomously learned by the agent.

Ernest's algorithm implements two motivational principles. The first principle is to record and test sequences of interactions, in search of those that capture regularities afforded by the environment. This motivational principle can be understood as a tendency to be *in control* of its activity [Steels, 2004]. The second principle is to seek interactions with positive values and to avoid interactions with negative values. This principle implements interactional motivation as presented in Section 2. Because of this dual motivational principle, Ernest does not learn optimal behaviors in terms of the value of the interactions. Nonetheless, it learns satisfying behaviors adapted to the regularities it discovers through its singular experience of interacting with the environment. For a complete description of Ernest's algorithm, we refer the reader to our paper [Georgeon & Ritter, 2012]. The present paper focuses on analyzing Ernest's behavior from an observer's perspective, which we do next.

4. The Small Loop Problem (SLP)

We designed a particular instance of an HSIML problem that allows an external observer to easily study the agent's behavior. We call it the Small Loop Problem (SLP) because the environment can be visualized as a small loop around which the agent can move [Georgeon & Marshall, 2012]. We implemented the Small Loop environment in NetLogo [Wilensky, 1999] to provide the interface shown in Fig. 4.



ticks: 833

step 5

bump -10

turn -3

touch-empty -1

time-interval 0.05

touch-wall -1

initial-position [1 1]

Interaction - Value:

```
-      -1
>      5
- w    -1
^      -3
-      -1
>      5
-      -1
>      5
- w    -1
^      -3
-      -1
>      5
```

Fig. 4: The Small Loop platform in NetLogo. The environment (left) is the loop of white squares surrounded by green walls. The brown arrowhead represents the agent. The experimenter can preset the values of the primitive interactions using the slider controls (center). The *Interaction-Value* window shows the sequence of the primitive interactions enacted by the agent, with ASCII characters encoding the *moves* and *effects*. The *Bump Count* graph (right) displays the number of times the agent bumps into a wall (cumulative total in blue), showing that the agent gradually learns to avoid bumping into walls. When the agent touches/feels a cell, the cell flashes yellow, and when the agent bumps into a wall, the wall flashes red, making the agent's behavior intelligible to the experimenter.

In the SLP, the set of possible moves U contains the 6 following moves: feel to the right (u_1), feel to the front (u_2), feel to the left (u_3), try to move one cell forward (u_4), turn 90° left (u_5), turn 90° right (u_6). The set of possible effects Y contains two elements, making 10 possible primitive interactions: feel an empty cell to the right/front/left i_1=(u_1,y_1), i_3=(u_2,y_1), i_5=(u_3,y_1); feel a wall to the right/front/left i_2=(u_1,y_2), i_4=(u_2,y_2), i_6=(u_3,y_2); step forward i_7=(u_4,y_1); bump i_8=(u_4,y_2); turn left i_9=(u_5,y_1); turn right i_{11}=(u_6,y_1). This environment is deterministic, meaning that, in this particular HSIML, the corresponding probability distributions Q and v (presented in Fig. 2) implement no stochasticity.

The experimenter presets the values of primitive interactions before running the agent (using the controls shown in Fig. 4). We specify the following reference values: step: 5; bump: -10; turn: -3; feel (empty or wall): -1. Note that these primitive interactions and values correspond to the example HSIML shown in Fig. 1, with the exception that turns always produce effect y_1, which makes 10 interactions rather than 12 (i.e., the interactions i_{10} and i_{12} presented in Fig. 1 do not exist in the SLP). The agent, however, ignores the meaning of these interactions, and even ignores the spatial organization of the environment.

Note that the SLP affords the regularities (*reg1*, *reg2*, *reg3*) presented in Fig. 1. Although the agent ignores their meaning, (*reg1*) means that feeling an empty cell in front indicates that it is possible to step forward. (*reg2*) means that, when a wall is felt in front, it is often possible to feel an empty cell on the left and to turn left and step forward. This regularity can only be discovered after the agent has started circling the loop in the counterclockwise direction. If the agent starts circling the loop in the clockwise direction, then it can discover the counterpart regularity based on right turns. (*reg3*) characterizes the passage in the upper right area of the loop when circling the loop in the counterclockwise direction.

Because Ernest can learn regularities of interactions, we expect it to learn a behavior that suggests that it progressively understands the possible interactions that are afforded by the environment, jointly with understanding the structure of the environment. Typically, we expect Ernest to appropriately use i_1, i_2, i_3, i_4, i_5, and i_6 to effectively feel its surrounding environment, and to use i_9 and i_{11} to effectively turn towards empty cells. This behavior should emerge as if Ernest enjoyed moving forward and disliked bumping, and progressively understood what the various available interactions correspond to.

5. Behavioral analysis

Ernest's behavior can be observed in the NetLogo platform available online[1]. One can choose different values for primitive interactions and reconfigure the environment by clicking on the grid in order to observe Ernest's adaptation to various situations. The NetLogo source code can also be downloaded so developers can change the ASCII characters that encode the different moves and effects generated by the environment without changing Ernest's algorithm, to verify that the algorithm does not exploit predefined semantics associated with these codes.

The experiment shows that, with the values defined in Section 4, Ernest learns to use *feel interactions* to avoid bumping into walls and to turn towards empty cells. First, Ernest learns to feel in front and to move forward if it feels an empty cell, and to not move forward if it feels a wall. Then, after a while, when it feels a wall in front, it learns to feel to the sides and turn towards empty cells so as to subsequently move forward. This behavior makes Ernest usually learn to avoid bumping into walls after approximately 300 steps. When engaging in circling the loop, Ernest often has difficulties in the upper right area because of the inverted corner. This is because Ernest learns only sequential regularities, which do not reflect the two-dimensional spatial organization of the environment. Yet, after a while, Ernest discovers the sequential regularities that characterize the upper left area (*reg3*), which allows it to reach a stable satisfying behavior that most often consists of circling the loop after approximately 600 steps. Notably, in some instances, Ernest learns suboptimal behaviors such as moving back and forth along a single edge of the loop, repeatedly making U-turns at each end of the edge.

Fig. 5 shows the trace of a representative run. This trace represents the primitive interactions enacted by Ernest over time with symbols similar to those used in Fig. 1. For easier reading, however, a few differences were introduced: i_8 is represented as a red triangle rather than a green triangle to more intuitively evoke bumping into walls. Feel-

[1] http://liris.cnrs.fr/ideal/demo/small-loop/

to-the-sides interactions (i_1, i_2, i_5, i_6) are not represented by trapezoids but by squares that are placed above or below the line to represent, respectively, feeling to the left and to the right.

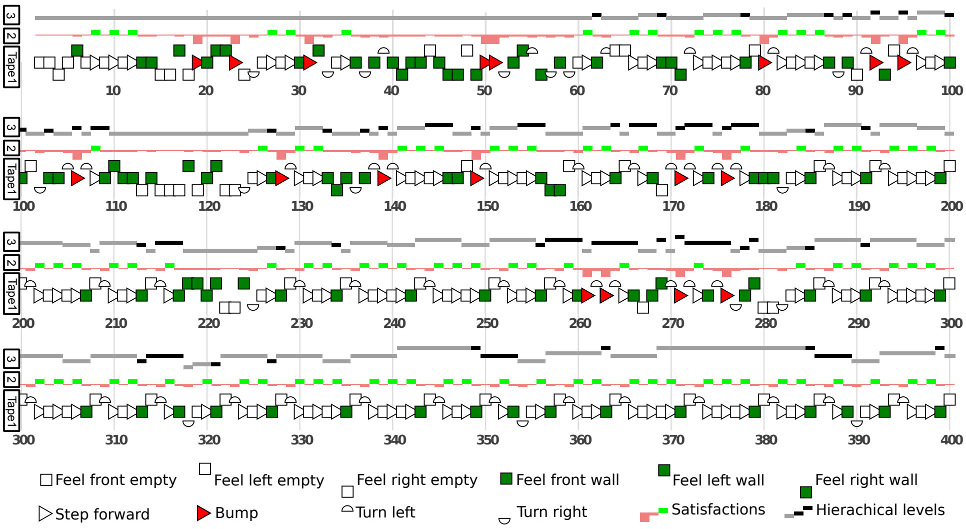


Fig. 5: Example trace of Ernest in Experiment 1. Tape 1 represents the primitive interactions enacted over time: *feel empty* (white squares), *feel wall* (green squares), *turn* (half-circles), *step forward* (white triangles), *bump* (red triangles); the upper part represents interactions to the left, the lower part interactions to the right. Tape 1 shows that Ernest learned to feel in front before trying to move forward from step 190 on, but still got puzzled in the upper right area of the loop around step 220 and 270. Ernest finally learned to avoid bumping after step 276. Tape 2 represents the interaction values as a bar graph (green when positive, red when negative); it shows that Ernest got more consistently positive interactions from step 290 on, meaning that it learned to fulfill its interactional motivation. Tape 3 represents the level of the enacted sequence in the hierarchy of learned sequences; it shows that Ernest gradually exploited higher-level sequences; black segments show sequences of interactions that are interrupted due to an unexpected effect when trying to enact a constituent primitive interaction. For example, on steps 62 and 69, Ernest expected to enact the sequence "*feel front empty – move forward"*, but this sequence was interrupted due to the unexpected feeling of a wall in front. Together, the three tapes show that Ernest learned to characterize left corners (situations with walls in front and on the left) by the sequence that leads to them when circling the loop counterclockwise: "*feel empty to the left, turn left, move forward, feel wall in front*". In this left corner context, Ernest learned to choose *turn right* (steps 318, 354, 390), which allowed it to engage in full tours of the loop. The average value obtained after behavior stabilization was of 5×12 (move forward) – 3×6 (turn) – 1×17 (touch), corresponding to 0.71 points/step.

A different choice of motivational values shows that sensemaking depends on the motivation that drives the agent. For example, if the experimenter gives positive values to the *turn interactions*, then the agent leans to spin in place, and exhibits poor sensemaking. This behavior has been observed in Experiment 2 illustrated by the example trace in Fig. 6.

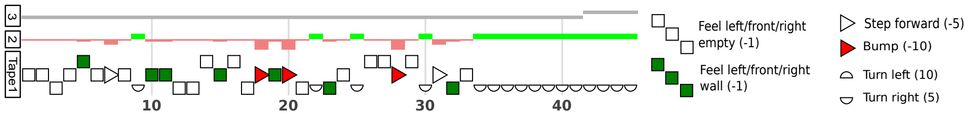


Fig. 6: Example trace of Ernest in Experiment 2, with positive values for turning and negative values for all other interactions (turn right: 5, turn left: 10, step forward -5). This trace uses the same symbols as Fig. 5. Tape 1 shows that Ernest tries different interactions before finding that it can keep turning to the right to constantly receive positive values. Tape 2 shows that Ernest receives constant positive values from step 34 on. Tape 3 shows that Ernest gradually learns sequences that consist of enacting a series of right turns (a series of 2 right turns from step 42 on, then a longer series not represented in the figure). In this run, Ernest never tried to turn to the left, so it never learned that it could receive even higher values (10) by turning to the left. This shows again that Ernest's learning is suboptimal (in terms of interactional motivation). Higher probability of finding optimal behaviors could be obtained by adding other forms of intrinsic motivation such as boredom avoidance, curiosity, or variable interactional motivation depending on internal states.

We propose these two behavior analyses as two complementary pieces of evidence supporting the claim that Ernest is capable of some form of sensemaking and self-motivation, while acknowledging that such sensemaking and self-motivation are still rudimentary.

Experiment 1 (Fig. 5) shows that Ernest appears to be driven by the motivation of moving forward and avoiding bumping, as if it enjoyed moving forward and disliked bumping into walls. Ernest learns to appropriately use the available possibilities of interaction to gain information to fulfill this motivational principle. It learns to engage in active perception (feeling) even though *feel interactions* were not initially defined as perception, and to use the resulting information to engage in appropriate subsequent behaviors (turning to the appropriate direction and moving forward) that fulfills its motivational principles.

Experiment 2 (Fig. 6) shows an example in which Ernest is motivated to turn, as if it enjoyed turning. This second example shows that Ernest's algorithm was not specifically designed to generate the sensemaking behavior demonstrated in Experiment 1 but rather to fulfill its interactional motivation whatever it may be. Therefore, Experiment 2 supports the premise that the sensemaking exhibited in Experiment 1 emerged out of a particular configuration of the agent's motivation and of the environment rather than being predefined in the algorithm.

Notably, the sensemaking exhibited in Experiment 1 is still rudimentary. For example, the agent does not understand that the environment contains other useful concepts such as "*long edge of the loop*" which would allow the agent to safely move three times forward without the need to feel in front before each step. Experiment 2 suggests that other forms of intrinsic motivation, such as boredom avoidance or curiosity, may help achieve higher-level sensemaking. We expect future versions of the agent to be able to engage in such higher forms of sensemaking and to exhibit more complex forms of self-motivation.

6. Conclusion

We propose an experimental method to study the sensemaking capabilities of an artificial agent. This method aims at producing evidence to support or contradict the claim that a particular agent is capable of some form of sensemaking. We present an application of this method that suggests that evidence of an agent's possible sensemaking capabilities can be found in the agent's behavior, through behavioral analysis. Our results also support the notion that observing sensemaking in an agent's behavior is linked to observing a form of self-motivation.

We propose a new class of problems that offers the opportunity for the agent to simultaneously exhibit rudimentary sensemaking and self-motivation, called Hierarchical Sequential Interactional Motivation Learning problems (HSIML). HSIML problems make it possible to investigate a form of self-motivation called interactional motivation. An interactionally motivated agent is an agent that seeks to enact interactions with positive values and to avoid interactions with negative values. Notably, interactional motivation can coexist with intrinsic motivation. In principle, an HSMIL problem in which all the values of interactions are set to zero can be used to demonstrate intrinsic motivation (such as curiosity), while setting interactions to nonzero values can generate supplementary interactional motivation. HSIML problems, however, contrast with most existing benchmarks in unsupervised learning [e.g. Rohrer, 2010] in that HSIML problems do not involve extrinsic motivation defined as a final goal to reach. The purpose of an HSIML problem is to offer the opportunity for the agent to learn to organize its behaviors appropriately, rather that to learn to reach a final goal.

We defined a particular instance of an HSIML problem called the Small Loop Problem (SLP) as a practical benchmark to study self-motivation and sensemaking in artificial agents. The SLP is available as a NetLogo project to other researchers interested in studying the sensemaking capabilities of their own agents.

We presented an application of the SLP to study the behavior of an agent that we developed in a previous study, and performed a behavioral analysis of this agent that suggests that it exhibits some form of rudimentary sensemaking and self-motivation. In summary, the behavioral analysis shows that the agent actively engaged in seeking information that it subsequently used to perform appropriate sequences of behaviors that fulfilled its interactional motivation, even though the semantics of such information and behaviors was not encoded in the algorithm. However, the agent was not able to find the optimal behaviors in terms of interactional motivation values, nor did it appear to seek such optimal behavior.

Overall, this study suggests that the demonstration of sensemaking relies on three types of evidence: (a) a description of the agent's algorithm that shows that the agent does not exploit a predefined semantics adapted to the experiment and encoded by the programmer; (b) an interactive demonstration of the agent's behavior in an experiment in which the experimenter can witness the gradual organization of behavior and modify the parameters to ensure that the experiment was not set up to work only in a predefined case; and (c) a set of behavioral reports that describe the kinds of behaviors that the agent is capable of.

Rather than providing absolute proof of sensemaking in an agent, the proposed method produces a set of detailed pro and con arguments for consideration by the research community. Continued work on refining this method and these arguments will provide a pathway toward more effectively evaluating the ability of autonomous agents to understand their environment. In turn, this will help us design agents that have better sensemaking capabilities and that exhibit more interesting forms of self-motivation.

Acknowledgments

This work was supported by the French Agence Nationale de la Recherche (ANR) contract ANR-10-PDOC-007-01 and by a research fellowship from the Collegium de Lyon.

References

Aström, K. [1965] Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*. (10), 174-205.

Blank, D., Lewis, J. and Marshall, J. [2005] "The multiple roles of anticipation in developmental robotics" in *proc. AAAI 2005 Fall Symposium: From Reactive to Anticipatory Cognitive Embodied Systems*, pp. 8-14.

Georgeon, O. L. and Marshall, J. B. [2012] "The Small Loop Problem: A Challenge for artificial emergent cognition" in *proc. Biologically Inspired Cognitive Architectures*, pp. 137-144.

Georgeon, O. L., Marshall, J. B. and Gay, S. [2012] "Interactional motivation in artificial systems: Between extrinsic and intrinsic motivation" in *proc. International Conference on Development and Learning (EPIROB'2012)*, 2p,

Georgeon, O. L. and Ritter, F. E. [2012] An intrinsically-motivated schema mechanism to model and simulate emergent cognition. *Cognitive Systems Research*, 15-16, 73-92.

James, W. [1907]. *Pragmatism* (Dover Publication, 1995).

Oudeyer, P.-Y., Kaplan, F. and Hafner, V. [2007] Intrinsic motivation systems for autonomous mental development, *IEEE Transactions on Evolutionary Computation*. **11**(2), 265-286.

Pfeifer, R. and Bongard, S. [2006] *How the body shapes the way we think: A new view of intelligence* (Cambridge, MA).

Pfeifer, R. and Scheier, C. [1994] From perception to action: The right direction?, in P. Gaussier & J.-D. Nicoud (eds.), *From Perception to Action* (IEEE Computer Society Press), pp. 1-11.

Piaget, J. [1951] *The psychology of intelligence* (Routledge and Kegan Paul, London).

Rohrer, B. [2010] Accelerating progress in Artificial General Intelligence: Choosing a benchmark for natural world interaction. *Journal of Artificial General Intelligence*. 2, 1-28.

Samsonovich, A. [2012] On a roadmap for the BICA Challenge. *Biologically Inspired Cognitive Architecture*. 1, 100-107.

Schmidhuber, J. [2010] Formal theory of creativity, fun, and intrinsic motivation. *IEEE Transactions on Autonomous Mental Development*. **2**(3), 230-247.

Steels, L. [2004] The autotelic Principle. In I. Fumiya, R. Pfeifer, L. Steels & K. Kunyoshi (eds.), *Embodied Artificial Intelligence* (Springer Verlag, Berlin), pp. 231-242.

Suchman, L. A. [1987] *Plans and situated actions* (Cambridge University Press, Cambridge).

Sun, R. and Giles, C. L. [2000] *Sequence Learning - Paradigms, Algorithms, and Applications.* Vol. 1828 (Springer, Berlin Heidelberg).

Wilensky, U. [1999] NetLogo. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University.

Wittgenstein, L. [1953]. *Philosophical Investigations* (Blackwell Publishing, Malden MA, 2001).