Designing Environment-Agnostic Agents

Olivier L. Georgeon, Ilias Sakellariou

Designing Environment-Agnostic Agents

Olivier L. Georgeon
Université de Lyon
CNRS, LIRIS, UMR5205
F-69622, France
+33 4 72 04 63 30
olivier.georgeon@liris.cnrs.fr

Ilias Sakellariou
Department of Applied Informatics
University of Macedonia of Economic and
Social Sciences, Thessaloniki, Greece
+30 2310-891-858
iliass@uom.gr

ABSTRACT
We present autonomous agents that are designed without encoding strategies or knowledge of the environment in the agent. The design approach focuses on the notion of *sensorimotor patterns of interaction* between the agent and the environment rather than separating *perception* from *action*. The agent's motivational system is also interaction-centered in that the agent has inborn proclivities to enact certain sensorimotor patterns and to avoid others. Such motivations result in the agent autonomously discovering, learning, and exploiting regularities of interaction afforded by the environment, and constructing operative knowledge of the environment. Because such agents have no predefined goals, we propose a set of behavioral criteria to both judge and demonstrate the agents' capacities, rather than performance measurement. A design platform based on NetLogo is presented. Results show that these agents demonstrate interesting behavioral properties such as hedonistic temperance, active perception, and individuation.

Categories and Subject Descriptors
I.2.6 [**Artificial Intelligence**]: Learning. I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *Intelligent agents.*

General Terms
Algorithms, Measurement, Design, Experimentation.

Keywords
Intrinsic motivation; Autonomous learning; Cognitive development; Enactive cognition; Affordances; Constructivism.

1. INTRODUCTION
This paper presents an argumentation in favor of the concept of *environmental agnosticism* as an original and useful concept to think about autonomous agents. We introduce this concept to capture the idea that we do not encode our knowledge of the environment in the agent's decisional mechanism. A typical example of such encoded knowledge would consist of a set of logical rules that specify the agent's behavior in response to specific information received from the environment (sensory input). Another example would consist of modeling the agent's environment as a predefined *problem space* in which reward values would be associated with predefined *problem states* and such values propagated across states [15][1]. Instead, we expect environment-agnostic agents to learn the semantics of sensorimotor information and the ontological structure of their world by themselves.

Our motivation for studying environment-agnostic agents is both theoretical and practical. On the theoretical level, we believe that this study can shed some light on how knowledge emerges in artificial agents and becomes meaningful to them. This question relates to the symbol grounding problem [13] and the study of developmental cognition (e.g., [28]). On the practical level, environment-agnostic agents will facilitate the development of agent-based simulations by unburdening the modeler from encoding knowledge in the agent.

As we intend to show, designing an environment-agnostic agent raises the crucial question of defining the agent's *drives*. We employ the term drive to emphasize the difference from traditional approaches that employ the terms task or goal. Indeed, we argue that programming an agent to perform a given task or to reach a given goal generally implies specifying how the agent interprets its world. For example, behavioral rules presuppose the semantics of input data, and traditional reward values presuppose goal assessment criteria. Such presuppositions conflict with the environment-agnosticism principle because environment-agnostic agents are precisely expected to learn by themselves how to interpret their world.

An intuitive distinction between drives and goals may be expressed by the fact that drives enforce a bottom-up approach starting from inborn behavioral tendencies toward the possible construction of higher-level goals, while goals follow a top-down approach through the decomposition of a problem into sub-goals. This conception of drives can also be related to the concept of *intrinsic motivation* (e.g., [2, 19]) in that the motivation does not come from an external reward. Yet, we acknowledge that the nuance may still seem vague and better pertaining to the domain of philosophy than computer science. At a philosophical level, let us only note here that we find some resonance with Dennett's *inversion of reasoning* argument [6]. The purpose of this paper is not to pursue this philosophical discussion any further but to show that this shift of viewpoint is not mere jargon and rhetoric but can have a strong impact on the way we develop autonomous agents.

Our intuition in developing environment-agnostic agents is to put the focus on the interaction between the agent and the environment rather than only on the agent. When designing

[1] The Soar architecture offers an emblematic illustration of these two types of examples as a rule-based system extended with reinforcement learning (Soar-RL). We credit the Soar team for acknowledging this knowledge-oriented bias in both cases.

autonomous agents, we, indeed, must presuppose the possible range of interactions between the agent and the environment. For example, in the case of robots, engineers define the robot's interactions when designing sensors and effectors. In the case of natural organisms, phylogenetic evolution selected the organism's sensorimotor system. In a similar manner, we predefine environment-agnostic agents' interactions in a given environment. We, however, neither presuppose nor specify how the agent should interpret such interactions. Instead, the agent has behavioral drives that define preferred courses of action in the world. The agent learns regularities of interactions and exploits such regularities in turn to better fulfill its drives. This mechanism is explained in more detail in Section 2.

The fact that the agent has no predefined goals raises the question of how to assess its behavior. This question is discussed in Section 3, and illustrated by experiments in Section 4. Section 5 presents the design/simulation platform employed for implementing environment-agnostic agents. Finally, the conclusion discusses the implications and limitations of the concept of environmental agnosticism for future work.

2. ENVIRONMENTAL AGNOSTICISM

Fundamentally, we believe designing environment-agnostic agents entails considering individual *sensorimotor patterns* as the atomic elements of cognition, without making an initial distinction between perception and action. This assumption is supported by many theories in cognitive science that argue that perception, cognition, and motion are entangled (e.g., [5, 14, 18, 21]). Specifically, Piaget [22], proposed the term *scheme* to refer to sensorimotor patterns. In the rest of this paper, we simply refer to sensorimotor patterns by the term elementary *interactions*.

Figure 1 illustrates the elementary interactions along the *interaction timeline*. Elementary interactions are represented by letters (A, B, C, A, D, A …). As introduced in Section 1, these interactions have no associated semantics implemented in the agent. Because of the absence of implemented semantics, these interactions can correspond to anything in the environment, which is why we characterize the agent as environmentally agnostic. As opposed to traditional artificial agents, this agent has no channel by which it would directly "perceive" its environment, but can only discover the structure of the environment through the regularities experienced while enacting these interactions.

2.1 Intrinsic drives

In addition to focusing on elementary interactions, we propose implementing inborn *proclivity values* associated with such interactions (in parenthesis in Figure 1). Accordingly, we provide the agent with a mechanism that tends to seek interactions with positive proclivity values, and to avoid interactions with negative proclivity values. Such a mechanism results in the implementation of primitive intrinsic drives because, subjectively, the agent seems to simply enjoy interactions that have positive proclivity values, and to dislike interactions that have negative proclivity values. As we present later, the difficulty resides in that the agent needs to learn what interactions will result from its choices.

Proclivity values are related to the notions of intrinsic reward (e.g., [25]) and value systems (e.g., [21, 26, 27]) in reinforcement learning. Because of this relation, our approach can be considered a form of *discrete time decision process* consisting of learning a *policy function* that tends to maximize a *reward function* over time. Traditional algorithms of discrete time decision processes, however, require assumptions incompatible with the agnosticism principle. For example, Markov Decision Process (MDP) algorithms require that the temporal dependency be known a priori, and that the environment be modeled in the form of states that the agent can directly recognize. Partially Observable Markov Decision Process (POMDP) [1] algorithms offer a way toward eliminating these hypotheses but they require a *state evaluation function* to assesses a *believed state* from observational data. The

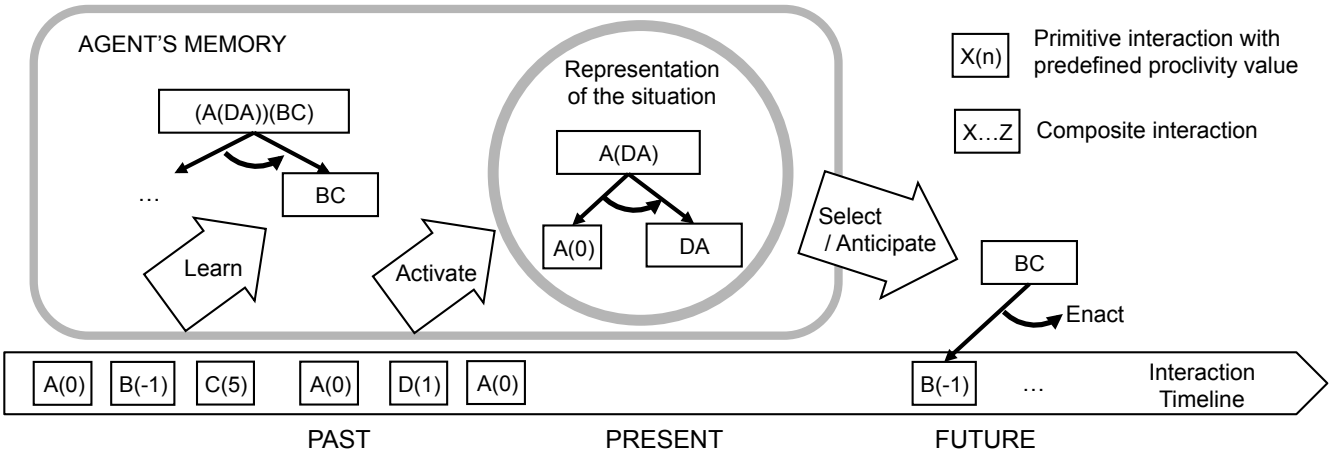


Figure 1. Learning and decisional mechanism of an environment-agnostic agent.

The agent's activity is represented along the *interaction timeline*. Letters (A, B, C, etc.) represent *primitive interactions*. Primitive interactions are associated with proclivity values pre-defined by the designer (e.g., (0), (-1), (5), etc.). Through its activity, the agent learns hierarchical sequences of interactions (*composite interactions*) that capture hierarchical regularities of interaction with a given environment (e.g., (A(DA))(BC)). The agent represents its current situation in the form of these hierarchical composite interactions. Over time, learned composite interactions allow the agent to predict the consequences of its choices depending on the current situation, and, therefore to select interactions that have the best chance to maximize the agent's proclivity in a given environment (e.g., BC). The agent can learn to enact unsatisfying interactions (e.g., B(-1)) to reach situations where it can enact even more satisfying interactions (e.g., C(5)).

state evaluation function needs to be known a priori, which remains incompatible with the agnosticism principle.

Here, we propose the term proclivity because this term conveys the idea that interactions are enacted for their own sake, while the term reward suggests that interactions are enacted for the sake of their outcome. Enacting interactions for their own sake removes the need for a pre-assumed model of the world. With proclivity, knowledge follows from action, while, with reward, action follows from knowledge. Our approach based on proclivity values relates more to continuous case-based reasoning [23] or trace-based reasoning [17] (but unsupervised, with no presupposed knowledge) than to traditional reinforcement learning, as we will develop in Section 2.2.

Notably, we consider this proclivity value mechanism only as an initial value system. In the future, we imagine implementing more elaborated drives, for example drives that may vary according to the agent's internal status (e.g., simulating a form of hunger varying over time).

2.2 Knowledge representation

An agent's stream of interaction with a given environment depends both on its decisions and on the unfolding of the environment. That is, the agent may well decide to try to enact an interaction with a high value, but this attempt may result in an actual interaction with a low value, due to unexpected environmental conditions (e.g., the agent may try to enact an interaction consisting of moving forward, but this attempt may result in bumping into a wall). This effect will be further explained in Section 4 and Figure 2. Because of this, the agent needs to learn to predict the interactions that result from its decisions. This raises the question of how to encode the knowledge that the agent learns.

We recommend representing the agent's knowledge as sequences of interactions. This view conforms with Gibson's [11] notion of *affordance*. Gibson suggests that the world is not known "objectively" but is rather known in terms of possibilities of interaction, called affordances. Encoding the agent's knowledge as sequences of interactions also follows from the fact that our agent has no other source of information about the environment anyway.

We devised an algorithm in compliance with these principles, called the *Intrinsically Motivated Schema mechanism* (IMOS) [10]. A significant difference from most existing decision process algorithms is that IMOS does not require Markov's hypothesis that the duration of the temporal dependency be known a priori. It can learn arbitrarily long episodes of interest by autonomously finding their beginning and end points. As opposed to partially observable Markov decision process (POMDP), IMOS does not require that the set of possible hidden states be defined a priori. IMOS recursively organizes episodes in a hierarchy of sub-episodes, a higher-level episode being a sequence of lower-level episodes. The proclivity value of a higher-level episode is set equal to the sum of the proclivity values of its sub-episodes, all the way down to predefined primitive proclivity values. The ability to perform such learning stems from the fact that, in the environmentally agnostic approach, the criteria for selecting interesting episodes are incorporated within the learning mechanism in the form of proclivity values. The agent can test hypothetical episodes and progressively select the most satisfying episodes with regard to their value.

Figure 1 illustrates this principle by representing learned hierarchical episodes of interaction in the agent's memory ((A(DA))(BC)). The activity at hand reactivates previously learned episodes that match the current situation (A(DA)) which in turn, triggers the subsequent interactions that are the most likely to result in satisfying interactions (as far as the agent can tell at this point in its development). This matching is possible because of the consistency between the representation of the situation and the representation of agent's procedural experience (a form of *homoiconicity*).

Again, we consider this mode of knowledge representation only as a starting point from which more elaborated representational structures can be derived. In particular, this mode of representation can be coupled with additional structures proposed by other researchers in Piagetian mechanisms, such as synthetic elements (e.g., [20]) or bare schemas (e.g., [12]). When implementing such structures, however, the assumptions these structures make about the environment should be explicitly stated. Notably, the purely sequential representations that we suggest here have the advantage of remaining compliant with the principle of environmental agnosticism because the agent has no *a priori* knowledge of how to represent the world "as such". The agent only knows and learns interactions.

3. EXPERIMENTAL PARADIGM

As introduced in Section 1, environment-agnostic agents are not designed to improve their performance over time with regard to a predefined problem set, task, or goal. Performance, as traditionally defined, is, therefore, not a property that appropriately accounts for the expected qualities of such agents. Instead, environment-agnostic agents are developed for their qualitative behavioral properties or for their theoretical implications on the study of developmental cognition. To study these agents, researchers need to agree on such expected properties.

This section proposes an initial list of expected properties based on our experience implementing environment-agnostic agents and on existing literature. In particular, Oudeyer, Kaplan, and Hafner [19] noted similar needs to characterize the properties of intrinsically motivated robots. These authors distinguished between three categories of criteria: (a) evolution of internal variables that account for the robot's learning (e.g., accuracy of anticipation or level of detail of learned categories); (b) evolution of external variables that characterize the robot's behavior (e.g., efficiency in the interaction with the environment); (c) evidence of reaching certain well-known developmental stages with regard to psychological or ethological theories.

3.1 Internal evaluation criteria

Category (a), the evolution of internal variables, has the advantage of objectivity because these criteria are based on variables implemented in the system. The drawback, however, is that each system has its specific variables, which complicates the comparison across systems. With these criteria, the authors need to clearly explain the significance of the variables.

A typical internal criterion is the growth of the variable that represents the system's *satisfaction*, as measured by its value system. We can formulate this criterion as:

a.1 Principle of objective hedonism.

For example, with the value system introduced in Section 2.1, the agent's objective hedonism is demonstrated by the agent's increasing ability to perform interactions with high values, and avoid interactions with negative values.

Notably, the principle of objective hedonism does not require the agent to reach the optimum value but only good-enough values (notion of bounded rationality [24]). Because good-enough values cannot be precisely defined, this principle should be complemented with qualitative principles that reflect the agent's decisional mechanisms more precisely. In particular, the agent should not simply react towards the highest immediate value. This can be expressed in the form of the corollary principle:

a.2 Principle of hedonistic temperance.

The agent should learn to enact negative interactions when such interactions can lead to even more positive interactions. Conversely, the agent should learn to refrain from enacting positive interactions when such interactions would lead to more negative interactions.

3.2 Behavioral criteria

Category (b), behavioral criteria, has the advantage of supporting comparisons across systems, because these criteria are based on the external observation of the system's behavior. The expected behavior can, however, vary across studies, raising the need for defining general principles. Assessing the agent's development with regard to general principles is the point of the third category listed by Oudeyer and coauthors (c). Yet, principles proposed by theories in psychology and ethology appear too vague, and out of reach for current artificial systems [12]. In the current state of the art, we need precise behavioral principles that account for the very beginning of the developmental process.

Surveys in developmental robotics (e.g. [2, 16, 28]) suggest three widely acknowledged principles:

b.1 Principle of situational categorization.

The agent should exhibit the capacity to categorize aspects of its situation and to adjust its behavior according to such categories.

b.2 Principle of situational disambiguation.

The agent should distinguish between different situations that generate the same sensory stimuli (perceptual aliasing, [3]).

b.3 Principle of graceful readaptation.

The agent should readapt gracefully to novel situations rather than experiencing catastrophic forgetting [7].

Moreover, the interaction-centered approach inspires two additional principles:

b.4 Principle of active perception.

The agent should learn to enact interactions not only because of their direct proclivity but also to update its representation of the current situation so as to take better decisions.

b.5 Principle of individuation.

Different executions of the same system should possibly lead to individualized instances exhibiting different habits. This principle accounts for the intuitive distinction between drives and goals according to which drives should leave room to individual *choices*. Individuation can occur through an "en habitus deposition" (De Loor [4] citing Husserl).

In summary, the criteria to assess environment-agnostic agents generally involve temporal analysis—either quantitative (category a) or qualitative (category b). This indicates a need for implementation platforms that generate activity traces and support activity trace analysis. The next section illustrates these principles by presenting example experiments.

4. EXAMPLE IMPLEMENTATIONS

When implementing an environment-agnostic agent in a specific environment, the designer chooses the meaning that he or she assigns to the primitive interactions. For example, he or she may design a two-dimensional grid where interactions may consist of moving, turning, bumping into obstacles, touching objects, etc. An agnostic agent, however, may even ignore that its world has two dimensions. The designer implements the execution of the interactions in an interface layer, as depicted in Figure 2.

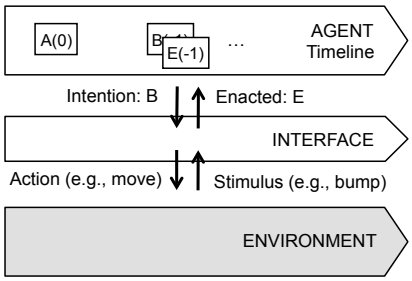


Figure 2. The interface agnostic agent/environment.

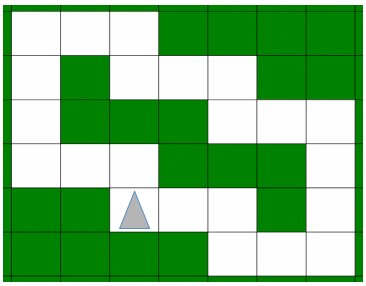
The interface between the agent and the environment generates the actions in the environment (e.g., try to move) from the intended interactions (e.g., B: move), and either confirms the intended interaction (B) or generates a contradictory interaction (e.g., E: bump). The meaning of the interaction is not implemented in the agent's decisional algorithm.

The designer also chooses the primitive interactions' proclivity values to generate interesting behaviors. For instance, if a negative value is associated with the interaction *turn*, and a positive value with the interaction *move forward*, then the agent tends to move forward, and turns only to ensure even more moving forward or to avoid subsequent even more negative interactions (such as bumping a wall). Conversely, with positive values associated with turning, the agent would learn to spin in place.

Interesting behaviors come from that an agent's possibilities of interaction and their proclivity values are adapted to a specific environment. In the case of natural organism, we assume that such adaptation results from phylogenetic evolution. Notably, this motivational mechanism is neither purely extrinsic (as would be a reward associated with an object in the environment), neither purely intrinsic (as *curiosity* in [2, 19]). Instead, it is *interaction-centered* and results from the pairing of an agent with an environment.

4.1 Simple sequential environment

We first demonstrated the intrinsically motivated schema mechanism in the experiment shown in Figure 3 [10]. In this case,



Primitive interactions		Val.
Move forward	→	10
Bump wall	→	-10
Turn 90° right toward empty square	↘	0
Turn 90° right toward wall	↘	-5
Turn 90° left toward empty square	↗	0
Turn 90° left toward wall	↗	-5
Touch wall ahead	–	0
Touch empty square ahead	–	-1
Touch wall on the right	\	0
Touch empty square on the right	\	-1
Touch wall on the left	/	0
Touch empty square on the left	/	-1

Trace:
1→ 2\ 3→! 4\ 5\ 6/! 7\ 8/ 9/! 10↘! 11↗! 12→! 13/! 14/ 15→! 16↗ 17↘! 18↘! 19→! 20/! 21/ 22↘ 23→ 24↘! 25\! 26/ 27→! 28↗! 29- 30→! 31→! 32- 33↘! 34- 35↗! 36\ 37↗ 38-! 39→ 40/ 41↘! 42/! 43/ 44- 45/! 46/ 47/ 48/ 49↗ 50\ 51-! 52→ 53- 54/! 55\! 56↗! 57↗ 58→ 59↗! 60/! 61/ 62(//) 63/ 64(//) 65↗ 66→ 67→! 68↘ 69→ 70- 71- 72(//) 73↗ 74→ 75/ 76\! 77-! 78→ 79\ 80- 81\ 82- 83(//) 84↗ 85→ 86/ 87↗! 88/ 89↗ 90→ 91\! 92- 93/! 94- 95\! 96↘ 97→ 98\ 99-! 100→ 101- 102- 103/ 104\! 105↘ 106→ 107\ 108- 109(//) 110↗ 111→ 112-! 113→ 114- 115/! 116/ 117- 118\! 119↘ 120→ …

Figure 3. Example environment-agnostic agent in a sequential environment (adapted from [10]).

Left: The agent (triangle) in the environment. Filled cells are walls which the agent can bump into. Center: List of the 12 primitive interactions with their proclivity values. Right: Activity trace of an example run. Steps 116 through 120: the agent has learned how to recognize and deal with a corner: touch the wall on the left – touch the wall in front – touch the empty square on the right – turn right – move forward.

The agent had 6 possible choices: try to move forward, turn left, turn right, touch in front, touch left, and touch right. The environment generated a single bit in return, which resulted in the 12 possible primitive interactions listed in Figure 3 (center). Again, the agent had no other way of apprehending its environment than interaction (no traditional "perceptual system"). This experiment can be seen online[2].

An analysis of this experiment based on the criteria listed in Section 3 leads to the following findings. This agent met the principle of objective hedonism (a.1) by learning to enact interactions of higher value, in particular by learning to avoid bumping into walls. Yet, the agent demonstrated temperance (a.2) because it learned to enact turn and touch interactions to ensure safer moves. The agent also demonstrated its capacity to identify and discriminate between situations (b.1, b.2) by representing the situation in the form of sequences rather than with the current feedback received from the environment (a single bit at each single point in time). The agent showed active perception (b.4) by adopting the habit of touching ahead before moving forward, and not moving forward if it touched a wall. This result is original because nothing initially differentiated perceptive interactions from motion interaction from the agent's viewpoint, except their cost (value). In essence, the agent learned to use cheap interactions to gain a better representation of the situation to ensure safer high-value interactions, which grounded the meaning of the constructed perception in the agent's activity.

This agent, however, had difficulties when the environment was not shaped as a linear route but was rather an open space, because the agent's sequential mechanism had trouble capturing spatial regularities. To address this difficulty, we implemented the experiment reported next.

4.2 Simple open space environment

In the experiment reported in Figure 4, we implemented interactions that were sensitive to remote properties of the environment [8]. This implementation was inspired by the visual system of an archaic arthropod, the limulus (horseshoe crab). In particular, two notable properties of the limulus were reproduced: (a) sensitivity to movement: the limulus's eye responds to movement, and the limulus has to move to "see" immobile things; (b) behavioral proclivity toward targets: male limulus move toward females, based on vision.

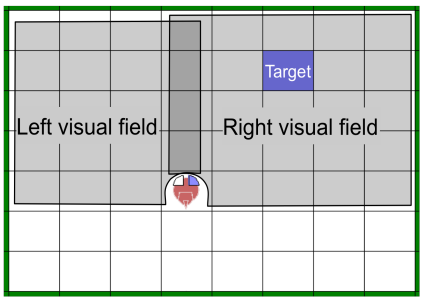
To replicate these behaviors, we simply implemented an interface that sent dynamic visual features to the agent. In this case, the primitive interactions were generated from the agent's action in the environment (move and turn), associated with the dynamic features resulting from the changes in the agent's visual field (target appear, closer, reached, and disappear). The behavioral proclivity toward targets was implemented by giving positive values to interactions in which the target appeared or enlarged in the visual field, and negative values when the target disappeared from the visual field (Figure 4, center). Note that the agent had no way to accurately predict when the target would disappear from its visual field as it moved forward.

This experiment demonstrated hedonist temperance (a.2) and active vision (b.4). In some instances, the agent learned to move forward until the target disappeared from its visual field as the agent passed it (active vision, b.4), then made a U-turn (hedonist temperance, a.2) and turned to realign itself with the target (in bold in Figure 4, right). In other instances, the agent learned to move toward the target in stair steps until it aligned itself with the target (trail shown in Figure 5). Once learned, an agent instance kept the same strategy when new targets were introduced in the board. This demonstrates the principle of individuation (b.5). Videos of these behaviors are available online[3]. These behaviors can also be reproduced in the simulation platform described in Section 5.

In this experiment, the agent exhibited adaptation to a spatial environment by simply capturing sequential regularities, but without capturing spatial regularities. For example, the agent did not learn the persistence of objects in space: it stopped pursuing a target in configurations where the target became hidden behind walls. Additionally, this agent was unable to adapt its behavior to different objects in the environment, for example, seeking food when hungry and water when thirsty. Such issues of spatial regularity learning and object persistence learning constitute a topic of research that we are currently exploring [9].

[2] http://e-ernest.blogspot.com/2010/12/java-ernest-72-in-vacuum.html

[3] http://e-ernest.blogspot.com/2011/01/tengential-strategy.html



Primitive interactions		Values
Move forward	(>)	0
Bump wall	[>]	-8
Turn 90° right to empty square	(v)	0
Turn 90° right to adjacent wall	[v]	-5
Turn 90° left to empty square	(^)	0
Turn 90° left to adjacent wall	[^]	-5
Additional value for each eye		
Appear	*	15
Closer	+	10
Reached	x	15
Disappear	o	-15

Trace:
1 2(> |+) 3(> |+) 4(> |+) 5(> |+) 6(> |+) 7(> |o) 8(v |*) 9(v*|o) 10(>+|) 11(^ |*) 12(^o|) 13(^ |o) 14(^*|) 15(>o|) 16(>) 17(>) 18(^*|) 19(vo|) 20(v) 21(^) 22(>) 23(^*|) 24(^o|*) 25(^ |o) 26(^) 27(v) 28(v |*) 29(^ |o) 30(^) 31(>) 32(^*|) 33(^o|*) 34(v*|o) 35(>+|) 36(^o|*) 37(v*|o) 38(>+|) 39(^ |*) 40(v |o) 41(>o|) 42(>) 43(^*|) 44(^o|*) 45(> |+) 46(> |+) 47(> |o) 48(v |*) 49(v*|o) 50(>+|) 51(^ |*) 52(>+|+) 53(v |o) 54(vo|) 55(v |*) 56(v*|) 57(v |o) 58(^ |*) 59(>+|+) 60(>+|+) 61(>+|+) 62(>x|x) 63(>o|o) 64(v) 65(v) 66(v) 67[v] 68(^) 69[v] 70(v) 71(v) 72(v) 73[v] 74[>] 75(^*|) 76(^o|*) **77(> |+) 78(> |+) 79(> |+) 80(> |+) 81(> |+) 82(> |o) 83(v |*) 84(v*|o) 85(>+|) 86(^ |*) 87(>+|+) 88(>x|x)** 89(^o|o) ...

Figure 4. Example environment-agnostic agent in an open space environment (adapted from [8]).

Left: The agent in the environment. The agent's visual system is made of two pixels that can only detect blue cells (targets). Each pixel covers a 90° span. Center: primitive interactions: move, or turn 90°, plus dynamic features possibly returned by each eye: appear, closer, reached, disappear. Right: Activity trace. An interaction consists of associating the agent's action with the signal sent by the eyes, separated by the symbol "|". Step 62: the agent finds and "eats" the first target. Step 75: a second target is inserted in the environment. Steps 77 to 88 (bold) demonstrate the learned behavior: Steps 77-81: the agent goes on a straight line with the target enlarging in its right eye's field. Step 82: the target disappears from the agent's right eye's field as the agent passes it. Steps 83-86: the agent makes a U-turn, returns back one step, and turns left towards the target. Step 87: the agent is aligned with the target and moves forward. Step 88: the agent reaches the target. Once learned, this "strategy" is repeated to reach other targets that the experimenter randomly introduces in the environment.

5. DESIGN/SIMULATION PLATFORM

Evaluating agnostic agents using the principles and criteria described in Sections 2 and 3 demands a flexible agent simulation platform to easily set up and run experiments. Such a platform should provide the means to parameterize experiments as well as offer mechanisms to detect patterns of agent behaviors in order to assess the agent's performance. We decided to use NetLogo [29] because we find it one of the simplest, most powerful, and most widely used agent simulation platforms available.

In a typical NetLogo simulation, the world consists of patches, i.e. components of a grid, and turtles that are agents that "live" and interact in that world. Modeling complex agents and environments is greatly facilitated by the fact that each entity participating in the simulation can carry its own state, stored in a set of system and/or

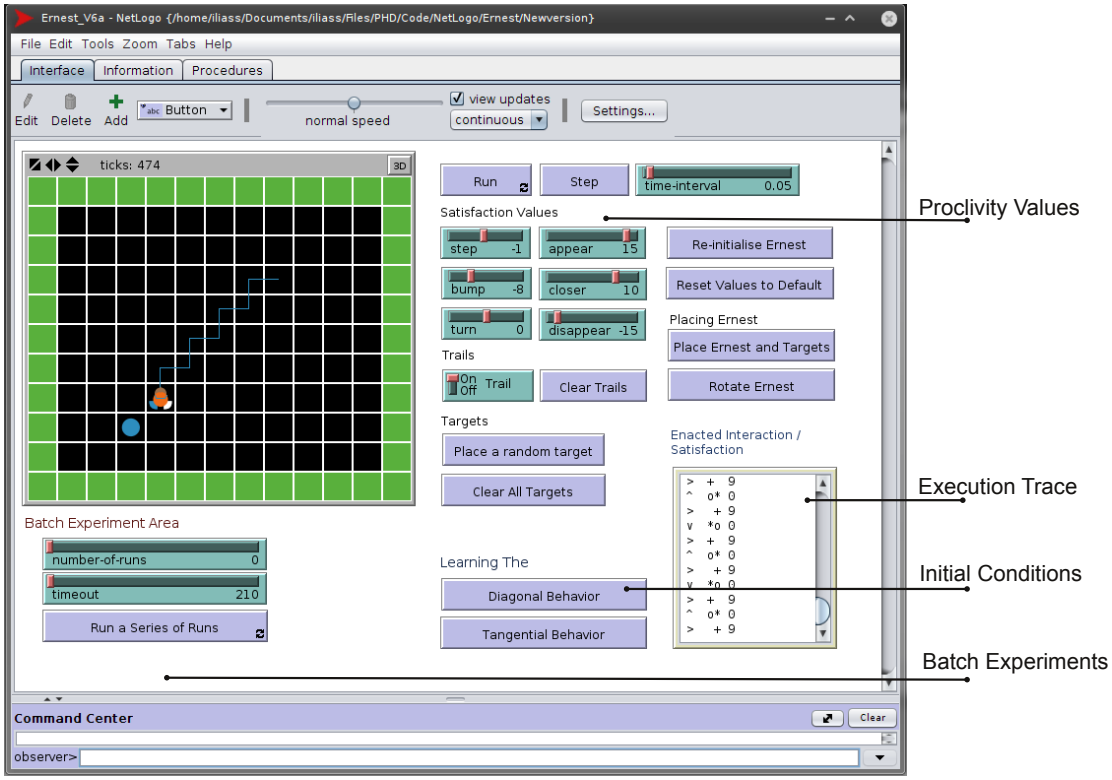


Figure 5: The NetLogo implementation of the IMOS agnostic agent. The platform allows the user to easily draw qualitative results. In the image, the diagonal strategy is shown by the trail left behind by the agent. The user can also set up a series of experimental runs (batch experiments) to observe the agent's long-term behavior.

user defined variables, allowing great flexibility in the simulations that can be implemented. Moreover, a simulation domain specific programming language permits implementation of any "sensors" or "actuators" that the experiment requires. Thus, a large range of worlds and a variety of agents can be simulated in order to evaluate design choices and test ideas in agent systems.

The intrinsically motivated schema mechanism [10] was integrated in NetLogo through an appropriate Java module, that we called the IMOS extension (Interaction MOtivation System). Thus, the environment and the interface (as defined in Section 4 and illustrated in Figure 2) are implemented in standard NetLogo, whereas the agents' decision/learning mechanism is handled by the IMOS extension. This architecture facilitates implementation of an unlimited number of different experimental settings by simply encoding the agent/environment related models, as well as effortlessly changing the agent parameters (e.g. proclivity values) in order to see how these parameters affect the agent's behavior. Finally, the NetLogo primitives offer an easy visualization of the agent's motion in the world (trails) that leads to a clear detection of patterns of interaction in the agent's behavior (Figure 5). Although, all the above could be implemented in any programming language, the simulation Domain Specific Language (DSL) that NetLogo offers makes the task of encoding simulation environments including controls, monitors etc., simple.

The implementation of the open space environment described in Section 4.2 is depicted in figure 5. Through the interface, the user can change the full set of experimental parameters in order to investigate their impact on the learned behavior. Such experimentation can lead to rather interesting observations. For example, using the platform, we were able to investigate how proclivity values affect the agent's interaction with the environment and also to investigate the impact of initial conditions, i.e. the initial locations of the agent and the target, on the different "strategies" learned by the agent in order to approach the target.

In the experiment in Section 4.2, the agent had a rather limited way of navigating through the environment, simply by moving between grid positions and turning by 90 degrees each time. In order to assess the algorithm's robustness and performance in a higher resolution environment, we simply changed some of the sensors and effectors (turning angle, cone of vision and movement capabilities) of the agent in the NetLogo part of the simulation, with proclivity values and the learning algorithm remaining the same as in the original experiment. As seen in figure 6, the agent managed to learn a similar strategy to approach the target, supporting further the argument regarding the robustness of the learning algorithm and the approach.

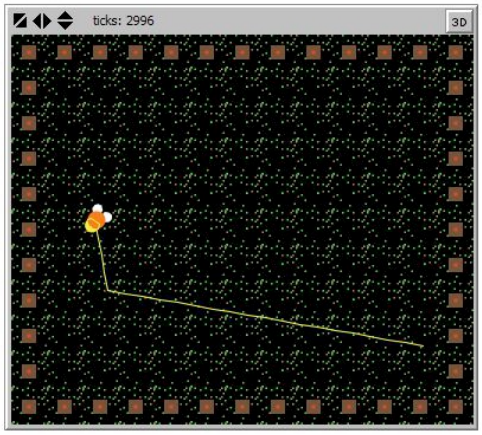


Figure 6: Learning to reach the target in a continuous environment

Thus, in more than one respect, NetLogo proved to be an excellent platform for evaluation and testing of the agnostic agent approach presented in this paper. The whole experimental platform, together with the Java library module, is available online[4].

6. CONCLUSION

We propose an approach to designing autonomous agents with minimal preconception of their environment. We characterize such agents as environmentally agnostic. This approach focuses on the notion of *sensorimotor patterns of interaction* between the agent and the environment rather than on the usual notions of *perception* and *action*.

The sensorimotor approach allows the implementation of an *interaction-centered* motivational mechanism. With this mechanism, the agent has inborn proclivities to enact sensorimotor patterns with high values and to avoid sensorimotor patterns with negative values. To do so, the agent needs to discover, learn, and exploit regularities of interaction afforded by the environment, which results in the autonomous construction of operative knowledge.

The interaction-centered motivational mechanism contrasts with traditional problem-solving or reinforcement-learning approaches in that it implements *drives* rather than *goals*. We argue that the fulfillment of drives should be assessed through activity analysis rather than performance measurement. We propose a list of developmental principles that should be observed in the agent's activity, in particular: objective hedonism, hedonistic temperance, active perception, and individuation.

Experiments show agents that meet the developmental principles in rudimentary settings. We provide the algorithm as a NetLogo extension to demonstrate that the agent's decisional process is independent from the environment that the designer chooses to implement. These experiments constitute an initial investigation of the principles of environment-agnosticism but the resulting behaviors are still rudimentary and many issues remain. In particular, we are now studying how environment-agnostic agents can learn spatial regularities and knowledge of persistent objects in the environment.

7. ACKNOWLEDGMENTS

This work was supported by the *Agence Nationale de la Recherche* (ANR) contract ANR-10-PDOC-007-01. We gratefully thank Jonathan Morgan and James Marshall for their review of this article.

8. REFERENCES

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

[4] http://users.uom.gr/~iliass/projects/NetLogo/Ernest/index.html

[2] Blank, D.S., Kumar, D., Meeden, L. and Marshall, J. 2005. "Bringing up robot: Fundamental mechanisms for creating a self-motivated, self-organizing architecture". Cybernetics and Systems, 32 (2). 125-150.

[3] Crook, P. and Hayes, G. 2003. Learning in a state of confusion: Perceptual aliasing in grid world navigation. In Proceedings of Towards Intelligent Mobile Robots (Bristol), UWE.

[4] De Loor, P., Manac'h, K. and Tisseau, J. 2010. "Enaction-based artificial intelligence: Toward co-evolution with humans in the loop". Minds and Machine, 19. 319-343.

[5] Dennett, D. 1991. Consciousness explained. Penguin.

[6] Dennett, D. 2009. "Darwin's ''strange inversion of reasoning''". PNAS, 106. 10061-10065.

[7] French, R. 1999. "Catastrophic forgetting in connectionist networks: Causes, consequences and solutions". Trends in Cognitive Sciences, 3 (4). 138-135.

[8] Georgeon, O.L., Cohen, M. and Cordier, A. 2011. A Model and simulation of Early-Stage Vision as a Developmental Sensorimotor Process. In Proceedings of Artificial Intelligence Applications and Innovations (Corfu, Greece, 2011), 11-16.

[9] Georgeon, O.L., Marshall, J. and Ronot, P.-Y. 2011. Early-Stage Vision of Composite Scenes for Spatial Learning and Navigation. In Proceedings of First Joint IEEE Conference on Development and Learning and on Epigenetic Robotics (Frankfurt, Germany, 2011), 224-229.

[10] 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.

[11] Gibson, J.J. 1979. The ecological approach to visual perception. Houghton-Mifflin, Boston.

[12] Guerin, F. and McKenzie, D. 2008. A Piagetian model of early sensorimotor development. In Proceedings of Eighth International Conference on Epigenetic Robotics (Brighton, UK).

[13] Harnad, S. 1990. "The symbol grounding problem". Physica D, 42. 335-346.

[14] Hurley, S. 1998. Consciousness in action. Harvard University Press, Cambridge, MA.

[15] Laird, J.E. and Congdon, C.B. The Soar User's Manual Version 9.1, University of Michigan, 2009.

[16] Lungarella, M., Metta, G., Pfeifer, R. and Sandini, G. 2003. "Developmental robotics: a survey". Connection Science, 15 (4). 151-190.

[17] Mille, A. 2006. "From case-based reasoning to traces-based reasoning". Annual Reviews in Control, 30 (2). 223-232.

[18] O'Regan, J.K. and Noë, A. 2001. "A sensorimotor account of vision and visual consciousness". Behavioral and Brain Sciences, 24 (5). 939–1031.

[19] 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.

[20] Perotto, F., Buisson, J. and Alvares, L. 2007. Constructivist anticipatory learning mechanism (CALM): Dealing with partially deterministic and partially observable environments. In Proceedings of Seventh International Conference on Epigenetic Robotics (Rutgers, NJ, 2007).

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

[22] Piaget, J. 1970. L'épistémologie génétique. PUF, Paris.

[23] Ram, A. and Santamaria, J.C. 1997. "Continuous case-based reasoning". Artificial Intelligence, 90 (1-2). 25-77.

[24] Simon, H. 1955. "A behavioral model of rational choice". Quaterly Journal of Economics, 69. 99-118.

[25] Singh, S., Barto, A.G. and Chentanez, N. 2005. Intrinsically motivated reinforcement learning. In Advances in Neural Information Processing Systems, Saul, L.K., Weiss, Y. and Bottou, L. eds. MIT Press, Cambridge, MA, 1281-1288.

[26] Sporns, O. 2003. Embodied cognition. In MIT Handbook of Brain Theory and Neural Networks, Arbib, M. ed. MIT Press, Cambridge, MA.

[27] Sutton, R.S., Modayil, J., Delp, M., Degris, T., Pilarski, P.M., White, A. and Precup, D. 2011. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In Proceedings of Tenth International Conference on Autonomous Agents and Multiagent Systems (Taipei, Taiwan).

[28] Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M. and Thelen, E. 2001. "Artificial intelligence - Autonomous mental development by robots and animals". Science, 291 (5504). 599-600.

[29] Wilensky, U. 1999. NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL. http://ccl.northwestern.edu/netlogo/.