Autonomous affordance construction without planning for environment-agnostic agents

Simon L. Gay
Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205
Villeurbanne, F-69622, France
Email: simon.gay@liris.cnrs.fr

Alain Mille
Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205
Villeurbanne, F-69622, France
Email: alain.mille@liris.cnrs.fr

Amélie Cordier
Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205
Villeurbanne, F-69622, France
Email: amelie.cordier@liris.cnrs.fr

Abstract—We present an architecture for self-motivated agents to organize their behaviors according to possibilities of interactions proposed by the environment, and to modify the environment to construct new possibilities of interactions. The long-term goal is to design agents that construct their own knowledge of objects through experience, rather than exploiting pre-coded knowledge, and exploit this knowledge to generate complex behaviors that satisfy their intrinsic motivation principles. Self-motivation is defined here as a tendency, based on inborn behavioral preferences, to experiment and to respond to behavioral opportunities afforded by the environment. Over time, the agent integrates, through its experience, relations between interactions and object affording them in the form of data structures, called signatures of interaction, which encode the minimal spatial configurations affording an interaction. The agent then exploits these signatures to recognize distant possibilities of interactions (or affordances), but also incomplete affordances. These structures help the agent defining behaviors that can construct affordances from separated elements. Experiments with a simulated agent show that they learn to navigate in their environment, reaching, avoiding and constructing objects according to the valence of the interactions that they afford.

I. Introduction

In this paper, we address the problem of generating behaviors that can modify the environment in the purpose of letting affordances emerge, by an artificial agent that initially ignores elements that compose its environment and geometrical properties of its environment. Such an agent can be defined as environment-agnostic [6]. We base our work on a design principle introduced by Georgeon and Aha, called *Radical Interactionism* [4], that intends to account for cognitive theories suggesting that perception and action are inseparable (i.e. O'Regan [10], Piaget [12]). Specifically, interactions are used to model Piaget's notion of *sensorimotor scheme*. In this approach, the agent is given a predefined set of uninterpreted interactions associated with predefined valences, and seeks to *enact* interactions with positive valences and avoids interactions with negative valences. This motivation principle is called *interaction motivation* [5], and relates to the problem of intrinsic motivation [11]. The agent *perceives* its environment by identifying affordances proposed by the environment rather than recognizing objects on the basis of predefined features. This approach addresses the knowledge grounding problem [8] by letting knowledge of objects arise from experience, and introduces no disconnection between agent's experience and representation of objects.

In previous works, we implemented agents that can integrate and exploit spatial properties and elements of their environments (static [1] or dynamic [3]) discovered through experience, enabling emergence of behaviors satisfying agent's motivational principles. The mechanisms were robust enough to be implemented on a robot [2]. However these agents can only consider affordances that are actually present in their environment. In this paper, we propose additional structures and decisional mechanisms to generate behaviors that enable constructing objects affording interactions without any pre-conception about objects and spatial properties of space. Such abilities relate to the problem of tool manipulation based on affordances [15]. Our approach consists in learning properties of interactions and relations that emerge between interactions, without using other structures than interactions: a tool is recognized as such because it can complete an affordance. Moreover, our goal is to make an agent able to generate behaviors satisfying its motivational principle according to offered possibilities of interaction. We tested our mechanism in an environment proposing objects that can be moved by the agent to let new affordances emerge.

II. Formalization of Radical Interactionism

A Radical Interactionism (RI) algorithm [4] begins with a set I of primitive interactions. Each primitive interaction i is attributed a *valence* v_i that defines the agent's behavioral preferences. At step t, the agent selects an *intended interaction* i_t, and is informed, at the end of step t of the interaction e_t that was actually enacted. The *enaction* is a success if $i_t = e_t$, and a failure otherwise. A RI agent learns to anticipate the results of its interactions, and tries to *enact* interactions with high valences.

However, it is difficult to discover spatial properties of the environment with a unique enacted interaction. We thus proposed an extension of the RI model, we called *Parallel Radical Interactionism* (PRI) [1][3]. The PRI model differs from the RI model as it allows to experience simultaneously more than one enacted interaction as the result of an intended interaction. The intuition comes from living beings who receive multiple sensory stimuli while they are acting.
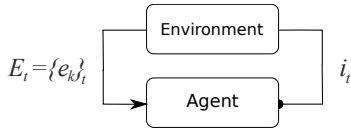
$$E_t = \{e_k\}_t$$

Fig. 1. Diagram of the Parallel Radical Interactionism model. At time t, the agent tries to enact an intended interaction i_t, and receives a set of enacted interactions $\{e_k\}_t$, called *enacted context* E_t. The environment is *opaque* and the agent can only experience it through interactions.

For example, an animal can move forward, and experience the optical flow resulting from this movement. We thus propose that the agent can experience additional *stimuli*, in addition to the enacted interaction. However, these *stimuli* cannot be considered without the movement produced by the enacted interaction. As an example, the optic flow on a retina can only carry a spatial information if it is considered with the movement that generates it. We thus propose to construct new interactions by associating an interaction and an additional stimulus. We call *primary interaction* an indivisible association between an action and a perception, and *secondary interaction* an indivisible association between an interaction and an additional perception. A primary interaction thus consists of a couple $i_p = (action, perception)$, and a secondary interaction, a couple $i_s = (i_{\{p,s\}}, perception)$, with $i_{\{p,s\}}$ the *associated* interaction of i_s.

Formally, the parallel RI model is similar to the RI model. The difference is that, at the end of step t, the agent experiences a set of enacted interactions $\{e_k\}_t$, containing a unique primary interaction and a set of enacted secondary interactions associated with this primary interaction. A secondary interaction i, like a primary interaction, successes when $i \in E_t$. We however consider that a secondary interaction i fails when the associated movement is produced (i.e. the associated interaction of i is in E_t), but the additional perception is not observed (i.e. $i \notin E_t$). Figure 1 illustrates this formalism.

III. THE SPACE MEMORY

The space memory is a structure dedicated to the integration and exploitation of the environmental properties observed through enaction of interactions. This section formalizes concepts and principles used to implement the space memory: signatures of interactions and object instances, that were introduced in previous work [1][3], and proto-objects and mobile objects defined to address the problem of affordance construction.

A. Signatures of Interactions

This structure is based on the assumption that the result of enacting an interaction depends on a limited spatial context of elements in the environment. We expect such contexts to define *objects* with which the agent can interact. This definition of objects relates to the concept of *affordances* proposed by J.J. Gibson [7]. An *object* is thus defined as a specific spatial configuration of elements affording an interaction and does not require *a priori* knowledge.

A RI agent can only perceive its environment by *experiencing* it through interactions. We formalize a *signature* S_i of an interaction i as a function $S_i : \mathcal{P}(I) \to [-1; 1]$, where $\mathcal{P}(I)$ denotes the partition of I (possible interactionnal contexts) that gives a numerical value in $[-1, 1]$ that reflects the possibility of successfully enacting i in an interactional context E. $S_i(E) = 1$ means an absolute certainty of success and $S_i(E) = -1$ an absolute certainty of failure. S_i is learned and reinforced when i succeeds or fails to generate accurate predictions. A signature must be reversible: it must be possible to define a function $\hat{S}_i : \{1; -1\} \to \mathcal{P}(\mathcal{P}(I))$ that can provide *minimum contexts* (i.e. $\nexists E_1, E_2 \in \hat{S}_i(x), x \in \{1; -1\}/E1 \subset E2$) affording i ($\hat{S}_i(1)$) and preventing enaction of i ($\hat{S}_i(-1)$). We use signature S_i to predict the enaction result of i and $\hat{S}_i$ to extract and exploit information about the object affording i.

Defining objects by learning to recognize affordances they provide is abundant in literature [9][14]. Signatures differs by the use of interactions, which allows implicit relations between interactions to be discovered, and recognition and localization of distant affordances in space in terms of interactions. See [1] and [3] for more details and examples of implementations.

B. Object Instances

A signature S_i of an interaction i characterizes a context at a certain position relative to the agent, in the form of sets of interactions $\{j_k\} \in \hat{S}_i(1)$. However, each interaction j_k has its own signature, and each context $E_l = \{j_k\}$ is composed of interactions related to the same primary interaction j. We thus propose to *backmove* a signature S_i through a primary interaction j using the following procedure: we note $\hat{S}_i^{\sigma_0} = \hat{S}_i(1)$, where σ_0 is an empty sequence of interactions, and construct: $\hat{S}_i^{[j,\sigma_0]} = \bigcup_{\forall E_l \in \hat{S}_i^{\sigma_0}/j \in E_l} \{E \in \mathcal{P}(I)/\forall j_k \in E_l, S_{j_k}(E) > 0\}$, which characterizes contexts that can afford i after enacting j. As this process can be repeated by considering $\sigma_{a+1} = [j, \sigma_a]$, we can *backmove* a signature S_i by a sequence of interactions σ, to obtain a *predecessor* of i, noted S_i^σ. A predecessor characterizes a context that, if *moved* through the enaction of the sequence of interactions σ, affords i. We consider that an *instance* of the object affording i is present at *position* σ with a certitude of $S_i^\sigma(E_t) > 0$. We thus characterize a position in space as sequences of interaction, which relates to the notion of Representative Space of Poincaré, for whom localizing an object in space means considering the movement needed to reach it [13].

C. Proto-object

We define a *proto-object* of an interaction i as a part of the context affording i. We define a partial backmoved signature $S_i^{\sigma*}$ of i as a structure $\hat{S}_i^{\sigma*} \subset \hat{S}_i^\sigma, \hat{S}_i^{\sigma*} \neq \emptyset$. A proto-object of i is detected at *position* σ when $\exists S_i^{\sigma*}/S_i^{\sigma*}(E_t) > 0 \wedge S_i^\sigma(E_t) \leq 0$.

D. Mobile object

We consider that an object is *mobile* when the same instance of this object, experienced through a set of interaction $E \in E_t$, is detected at positions $[\sigma_0]$ and $[\sigma_0, i, \sigma_m]$, with σ_m the movement of the object relative to the agent (possibly an empty

sequence), i.e. $\exists t \in \mathbb{N}, E \subset E_t, \sigma_0, \sigma_m / E \in \hat{S}_i^{\sigma_0} \wedge E \in \hat{S}_i^{[\sigma_0, i, \sigma_m]}$. This means that the agent can move toward the object by enacting sequence σ_0, interacting with the object (i), then moving according to σ_m and interacting again with the object. When the object affording i is considered as mobile, the space memory gathers properties related to the manipulation of this object. Our model requires two types of properties:

- positions that allow to interact with the same mobile object (i.e. defined by the same context). Indeed, changing the position from which the agent will interact with an object does not change the distance from this object to other proto-objects. A sequence σ_1 is integrated as a *manipulation sequence* of i if $\exists t \in \mathbb{N}, E \subset E_t, \sigma_0 / E \in \hat{S}_i^{\sigma_0} \wedge E \in \hat{S}_i^{[\sigma_0, \sigma_1]}$, with σ_1 not containing i.

- sequences of interactions that allow to interact with the constructed object. Sequence σ_2 is a *post-construction sequence* of mobile object affording i that enables enaction of j once object is constructed if : $\exists t \in \mathbb{N}, E \subset E_t, \sigma_0 / E \in \hat{S}_i^{\sigma_0} \wedge E \in \hat{S}_j^{[\sigma_0, \sigma_2]*}$.

IV. Selection Mechanism

We propose three decisional mechanisms that can work in parallel to select the next *intended interaction* i_{t+1}. Each of these mechanisms adds a *utility* value to the valence of interactions, which influences the selection of the next intended interaction. The two first mechanisms were introduced in previous work [1][2][3] while the third, called *Construction Mechanism* was developed to address the problem defined in this paper.

The *exploration mechanism* allows testing and reinforcing signatures when the certainty of prediction of an interaction or the reliability of the signature are low. Defining the utility value relies on the implementation of signatures. Section V-B gives rules used in the current implementation of the space memory. The exploration utility value u_i^λ is computed for each primary interaction i. Utility values of secondary interactions are added to the utility values of their associated primary interaction.

The *exploitation mechanism* helps to generate behaviors that satisfy the agent's motivational principles at the short and medium terms. This mechanism adds a positive utility value to interactions that enable moving closer to object instances affording interactions with high valence, and a negative value when the object instances afford interactions with low valences. The utility value is weighted by the distance of object instances so that far object instances have a lower influence. As we define positions with sequences of interaction, the distance is given by the length of sequences and the interaction that allows to move closer is the first element of sequences. The *exploitation utility* value $u_{i_c}^e$ is computed for each candidate (i.e. predicted as a success) primary interaction i_c as:

$$u_{i_c}^e = \sum_{o_j \in O^{i_c}} \nu_j \times f(d_{o_j}) \quad (1)$$

Where O^{i_c} is the set of object instances that can be moved closer by enacting i_c, o_j is an object instance affording

interaction j, ν_j is the valence of j, d_{o_j} is the distance of o_j, and $f : \mathbf{R}^+ \to]0; 1]$ is a function that characterizes object influence according to their relative distance. In our implementations, we use the function $f : x \to e^{-\gamma \times x}$ where γ is a coefficient that characterizes the decreasing of object influence depending on their distance.

The *construction mechanism* measures variation of distance between proto-objects composing the same object that the enaction of an interaction afforded by a mobile object can produce. The utility values are computed as:

- for each detected mobile object instance o_{i_m} affording i_m including a proto-object p_j affording j (i.e. $\exists S_j^* / \hat{S}_j^* \subset \hat{S}_i$), we define a list of couples of possible sequences $\sigma' = [\sigma_0, \sigma_1]$ and $\sigma'' = [\sigma_0, i_m, \sigma_m, \sigma_1]$, where σ_0 is the position of o_{i_m}, σ_1 is a *manipulation sequences* (which can be an empty sequence) and σ_m the *movement* of the mobile object when interacted (Section III-D). σ' characterizes a position of o_{i_m} before moving it, and σ'' a position of o_{i_m} after moving it.

- for each proto-object p_j included in the mobile object instance o_{i_m}, we detect positions of complementary proto-objects $\overline{p}_{j,k}$ of p_j, defined as $\{\overline{p}_{j,k}\}_k = o_j - p_j$. We consider proto-objects for which the position can be considered both with sequences under the form $[\sigma', \sigma_{\Delta_1}, \sigma_3]$ and $[\sigma'', \sigma_{\Delta_2}, \sigma_4]$, where σ_3 and σ_4 are post-construction sequences (possibly the same) of i_m that allow to enact j. Thus, σ_{Δ_1} characterize the distance between proto-objects p_j and $\overline{p}_{j,k}$ before moving the object instance o_{i_m} and σ_{Δ_2} the distance after moving o_{i_m}. We can then estimate the variation of distance between proto-objects produced by enacting i_m from position σ_0 by comparing length l_{Δ_1} of σ_{Δ_1} and length l_{Δ_2} of σ_{Δ_2}.

- The construction utility $u_{i_c}^c$ of each candidate primary interaction i_c considers the maximum utility for each interaction i_m afforded by a mobile object that can construct an object affording an interaction j:

$$u_{i_c}^c = \sum_{i_m} \max_{\sigma', \sigma'', p_j, \overline{p}_{j,k}} \nu_j \times (l_{\Delta_1} - l_{\Delta_2}) \times f'(l_{\Delta_1}) \times f(d_{\sigma_0}) \quad (2)$$

Where f' is a function that characterizes the influence of the distance between proto-objects and f is the same function than for (1). In our implementations, we use the function $f' : x \to e^{-\gamma' \times x}$ where γ' is a coefficient that characterizes the decreasing of object influence depending on distance between proto-objects.

The mechanism then selects, among candidates i_c, the interaction with the greatest global valence v'_{i_c} defined as :

$$v'_{i_c} = v_{i_c} + \lambda \times u_{i_c}^\lambda + \beta \times u_{i_c}^e + \delta \times u_{i_c}^c \quad (3)$$

where $\lambda, \beta, \delta \in \mathbb{R}$ are *influence coefficients* of the memory. There are no separated learning and exploitation periods.

V. Implementation on Artificial Agents

We designed a minimalist experiment to test our mechanisms. This experiment is inspired by the *sokoban game*, that consists in moving boxes on specific tiles without being stucked, in a discrete 2D top-view environment. We simplified
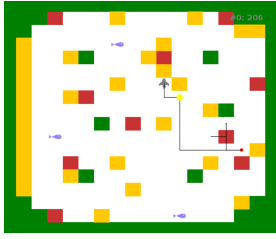
Fig. 2. Environment of the simulated agent. The trace (black line) shows the last 30 steps. The agent is represented as a grey shark, boxes as yellow squares, tiles as red squares and food as blue preys. Enaction of interaction push and bump are represented with yellow and red discs.

this principle: when the agent moves a box on a specific tile, the box *opens* and releases a piece of food. Such an environment seems relevant to test our mechanisms as it requires to move elements in order to make objects affording high valence interaction appear. The simplicity of this environment allows to quickly obtain accurate signatures of interactions from which the agent can extract information required by the construction mechanism.

These agents have a predefined list of six primary interactions, listed below (valences are in parentheses):
- ▷ move forward by one step (1)
- ▷ push a mobile element (5)
- ▶ bump in a solid element (-5)
- ▶ eat something edible (200)
- ▽ turn right by 90° (-1)
- △ turn left by 90° (-1)

We add a set of secondary interactions provided by the agent's visual system, that can detect colors among {red, green, blue, yellow}, and measure distances. Like a *sokoban* player, the agent can perceive its whole environment (although the visual system is ego-centric rather than geo-centric). This simplification dispenses the use of structures learning object permanence [1] that would require a long learning period while not being related to the problems addressed in this paper. Visual interactions consist in seeing a red, green, blue or yellow element while enacting a primary interaction, at a certain position of space. We discretize the visual field as a regular grid of 25×25 positions centered on the agents that matches the grid of the environment. These interactions have a predefined valence of zero. We thus define $6 \times 4 \times 25 \times 25 = 15000$ possible secondary interactions.

A. Environment

The environment is designed to afford spatial regularities that the agent can discover through its interactions. We defined four types of elements characterized by a color that makes them recognizable with visual interactions:
- wall (green), affording *bump*.
- box (yellow), affording *push*. If a box is in front of a wall preventing it from being pushed, it affords *bump*.
- food (blue), affording *eat*. When the agent eats a piece of food, it becomes a tile.
- tiles (red), affording *move forward*, as well as empty spaces. However, boxes and tiles can be *combined* to get a

piece of food.

The environment's contents can be edited during the experimental run. Figure 2 gives an example of environment.

B. Implementation of signatures

We propose a new implementation of signatures of interactions that is more efficient in discrete environment than an implementation based on formal neurons [2][3] (although it is less robust), as it enables obtaining reliable and accurate signatures after a shorter period. This implementation is based on the construction of minimal *contexts* that characterize objects affording interactions. We define a context c as a set of interactions that can be observed simultaneously. c is considered as *active* when $c \subset E_t$. Each interaction is attributed two nodes, the first predicting a success and the second predicting a failure. When the prediction of a node is observed as wrong, the signature integrates the context E_t as a context that would have inhibited the node. When adding the new context, the signature mechanism first looks for existing contexts c_k for which $E_t \cap c_k \neq \emptyset$. If such contexts exist, then the context for which $Card(E_t \cap c_k)$ is maximum is removed and replaced by $c'_k = E_t \cap c_k$. This principle ensures that contexts are quickly reduced to the minimum context affording or forbidding an interaction (basically 3 to 8 tests). While each node collects contexts that inhibit its prediction, it produces less errors. We propose that when a node gives 100 consecutive correct predictions, it can inhibit the other node. If the node predicting a success is inhibited, the signature can be interpreted as "interaction will fail except if a context inhibiting node predicting failure is active", and vice-versa. In case of complex objects, a context c can also make errors in prediction. When a context makes an error, it constructs contexts that can inhibit it. Contexts are considered as wrong and removed after 10 errors.

We define the reliability of a context c in interactional context E_t as the number of consecutive correct predictions (bounded by 10). The reliability of a signature is defined as the maximum number of consecutive correct predictions among the two nodes, added with the minimum reliability among contexts c_k for which $E_t \cap c_k \neq \emptyset$.

VI. EXPERIMENTS AND OBSERVATIONS

As the construction mechanism is very CPU consuming, we first tested the exploration and exploitation mechanisms to obtain accurate signatures. We then tested the exploitation and construction mechanisms, using signatures obtained in the first part of the experiment. Testing the construction mechanism separately is not problematic as this mechanism is not functional until a large amount of signatures become reliable. Results described in this section use the following coefficients: $\lambda = 1$, $\beta = 1$, $\delta = 2$, $\gamma = 0.5$, $\gamma' = 0.001$.

A. Signature Learning

We let the agent move in its environment and observe evolution of signatures. Signatures of primary interactions stabilize after 4000 to 8000 simulation steps, depending on interactional opportunities offered by the environment. These
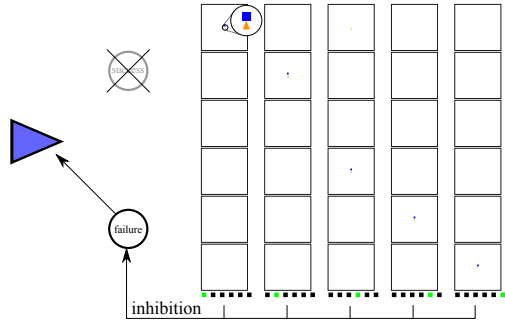
Fig. 3. Signature of interaction *eat* at decision cycle 8000. The interaction is considered as a failure, except when one of the above context (each column) is experienced. To make context easier to read by an external observer, secondary interactions are represented, for each context of $\hat{S}_{eat}$, with colored squares for which color and position correspond to properties of the considered interaction, and primary interaction with a green square. Interactions are gathered according to their associated primary interaction. All contexts are similar and characterize the presence of a blue object in front of the agent (position of the agent is displayed with a orange triangle). We can however observe that there is no contexts related to *bump* (third group). Indeed, it is not possible to enact *eat* after bumping as the agent stays in front of a wall.
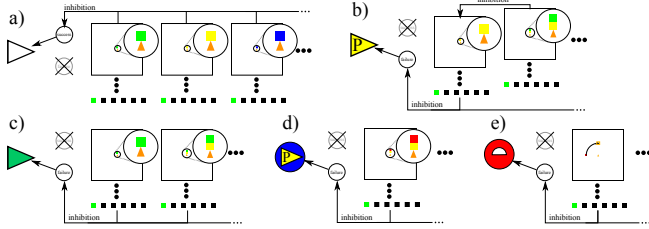

Fig. 4. Sample of signatures. We only represent contexts related to primary interaction *move forward*, as other context are similar (as in Figure 3). a) *Move forward* is predicted as a success except if a yellow, green or blue object is in front of the agent. b) *Push* is predicted as a failure except if a yellow element is present in front of the agent. These contexts can be inhibited if a green object is present behind the yellow element. c) *Bump* is predicted as a failure except if a green element is present in front of the agent, or if a yellow element with a green element behind are present. d) The visual interaction consisting of *seeing a blue element in front of the agent while pushing* is afforded by the presence of a yellow element in front of the agent and a red element behind the yellow element. This signature thus characterizes the fact that pushing in such a configuration generates a blue element. e) A visual interaction that consist in *seeing a red element ahead the agent* (position indicated with a orange square) *while turning left*. We observe that this signature designates a red element at the left side of the agent and thus characterizes a geometrical property of space.

signatures successfully integrated contexts that afford primary interactions, as shown in Figures 3 and 4.

Signatures of visual interactions show an interesting structure: they designate contexts containing visual interactions related to seeing an element of the same color at a position that characterizes the movement produced by the associated primary interaction, characterizing spatial properties of the environment (Figure 4). Signature of *seeing a blue element in front of the agent while pushing* is also interesting: it designates a context composed of a yellow box in front of the agent and a red tile behind the box. Indeed, enacting *push* in this context will *construct* a *food* element. This signature thus integrated a way to construct the affordance of *eat*.

While signatures of interactions becomes reliable, influence of the exploration mechanism decreases, while influence of
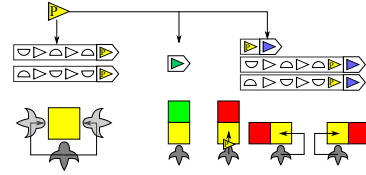

Fig. 5. Properties discovered about mobile object affording *push* (we do not display redundant sequences). Top left: two *manipulation sequences*. Enacting one of these sequences does not change the distance between proto-objects. Top middle and top right: post-construction sequences. Interaction *bump* can be directly enacted after constructing the affordance, and three sequences lead to the enaction of *eat* after constructing the affordance. Bottom: interpretation of above sequences by an external observer.

exploitation mechanism increases as the agent becomes able to detect distant object instances. The agent thus moves toward objects affording interactions with high valence (food and boxes), but neglects red tiles.

B. Construction of affordances

We tested the agent equipped with the exploitation and construction mechanism, using signatures obtained previously (Section VI-A) after 42000 decision cycles. We observed the behavior of the agent in several environmental configurations where a box and a tile are present. When the box and the tile are close enough to make influence of construction mechanism greater than other mechanisms (eq. 3) (typically less than 4-5 grid units with current parameters), the agent moves around the box and pushes it toward the tile, changing direction when needed, until the box is in front of the tile. Then, the agent pushes the box to *construct* the food and eats it. Otherwise, the agent only pushes the box, as *push* has a positive valence. Figure 5 lists discovered properties of mobile object affording *push*: the same object instance can be interacted from two additional positions, after pushing a box toward a green wall, the interaction *bump* can be immediately enacted, and after pushing a box near a tile, three configurations appear, according to the relative position of the tile.

Figure 7 lists object instances and proto-objects that have the greatest influence on the behavior, at each step, in the environmental configuration described in Figure 6. We observe that, at step 4, the exploitation mechanism proposes *turn left* to interact with the box. When the agent is only driven by the exploitation mechanism, as in the case of previous agents [1][3], the agent effectively turns left and pushes the box (Figure 6a). However, the construction mechanism detects an interesting configuration: after enacting sequence [▷△▷△], the proto-objects that enable construction of *eat* affordance are separated by sequence [△▷▷▽▷] of length 5. After pushing the box (here, sequence [▷]), the two object instances are separated by sequence [△▷▷▽], which is shorter (length=4). Pushing the object instance from position [▷△▷△] is thus interesting as it can gather proto-objects affording *eat*. the construction mechanism thus propose *move forward* with a high utility, related to the high valence of *eat*.

Configurations given in Figure 8 show that the agent considers every elements of its environment: when a wall is added, the agent interacts with the box from another position, as the
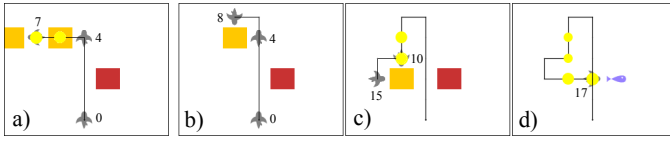
Fig. 6. Behavior of the agent in presence of mobile objects and proto-object affording *eat*. a) the agent is equipped with the exploitation mechanism, such as agents introduced in previous work [2][3]. At step 4, the agent turns left and interact with the object affording *push*, as *push* has a positive valence. b), c) and d) the agent is equipped with both exploitation and construction mechanisms. We observe that the agent goes behind the box and push it toward the tile. At step 10, the agent changes its position to not overpass the tile, and complete pushing the box to construct the affordance of *eat* (step 17). Object instances and proto-object influencing the behavior of the agent are listed in Figure 7.
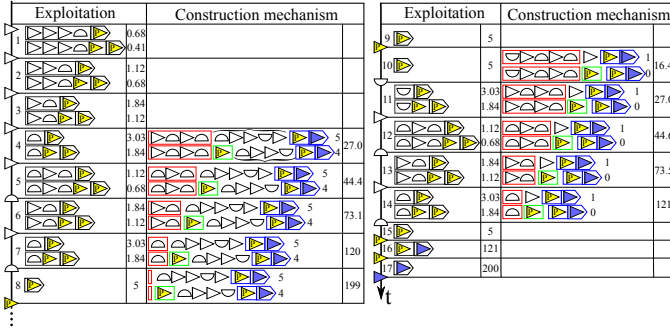

Fig. 7. Utility values given by decisional mechanisms, in configuration showed in Figure 6. From left to right: enacted interaction at each step t, most influent object instances (with their position σ), and their utility values, most influent construction sequence given by the construction mechanism. Positions of the mobile objects are in red, manipulation sequences are in green. The sequence that enables interaction with the constructed object is in blue. The middle sub-sequence characterizes the distance between the two proto-objects required to construct *eat* affordance. We display distance of proto-objects before (top sequence) and after (bottom sequence) manipulating the mobile object, and the construction utility value. At step 4, we observe that mechanisms diverge: construction mechanism propose *turn left*, and directly interact with the box to experience valence of *push*, while construction mechanism proposes *move forward* to construct a *eat* affordance. Influence coefficients can influence either the agent prefers immediate or future satisfaction.

previous position would construct object affording *bump*. The agent can thus construct the object affording *eat* using another sequence of interactions.

VII. Conclusion

This work proposes a model that enable an agent to modify its environment to construct possibilities of interaction that are not present at first. Our implementation in artificial agent shows how this model can extract and integrate object properties from signatures of interactions and its own experience of the environment, and exploit these properties to construct objects affording interactions with high valence and thus satisfy agent's motivational principles.

This model does not rely on plan construction, but increase attractiveness of object instances with which interacting will change the distance between elements that compose the needed object. This model is thus similar to the exploitation mechanism developed in previous works [1], but that is applied between two proto-objects rather than between the agent and an object instance.
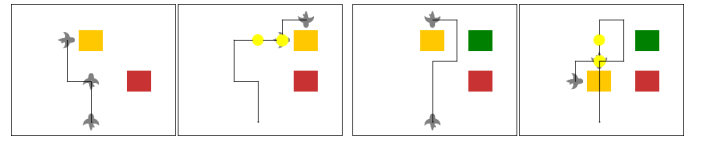

Fig. 8. The agent considers every element of its environment: if a wall block is added (right), the agent will push the box from another side to avoid constructing an affordance of *bump*.

In future works, we will implement this model on more sophisticated RI implementations, such as agent in continuous and dynamic environment, or even on robots, and using variable valences and coefficients that rely on internal states of the agent (such as hunger or tiredness), which can instantly change attractiveness of objects and global behavior of the agent. We also intend to extend the model: in the model presented in this paper, we only consider object instances and proto-objects that are accessible. A possible extension of this model could consider a non-enactable path leading to an object instance as an object that may be constructed. A agent equipped with such a model could construct or modify a path that lead to an object, or construct an object that require several steps to be constructed, which constitutes the bases of an abstract construction plan.

References

[1] S L. Gay., *Mécanismes d'apprentissage développemental et intrinséquement motivés en intelligence artificielle : étude des mécanismes d'intégration de l'espace environnemental*, PhD thesis, Université Claude Bernard Lyon 1, France, 2014.

[2] S. L. Gay, O. L. Georgeon, and C. Wolf, *Autonomous object modeling based on affordances for spatial organization of behavior*, IEEE Fourth Joint International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EPIROB), 2014.

[3] S. L. Gay and S. Hassas, *Autonomous object modeling based on affordances in a dynamic environment*, In proceedings of BICA 2015, Biologically Inspired Cognitive Architecture, pages 150–156, 2015.

[4] O. L. Georgeon and D. Aha, *The Radical Interactionism Conceptual Commitment*, Journal of Artificial General Intelligenc, 4(2):31-36, 2013.

[5] O. L. Georgeon, J. B Marshall, and S. Gay, *Interactional motivation in artificial systems: between extrinsic and intrinsic motivation*. In proceedings of the Second International Conference on Development and Learning, and on Epigenetic Robotics (EPIROB), 2012.

[6] O. L. Georgeon and I. Sakellariou, *Designing Environment-Agnostic Agents*, In proceedings of ALA2012, Adaptive Learning Agents workshop, at AAMAS 2012, pages 25–32, 2012.

[7] J. J. Gibson, *The theory of affordances*, in Perceiving, Acting, and Knowing: Toward an Ecological Psychology, R. Shaw and J. Bransford, Eds. Hillsdale, pages 67–82, 1977.

[8] S. Harnad, *The symbol grounding problem*, Physica D(42), 335–346,1990.

[9] T. E. Horton, A. Chakraborty, and R. Amant, *Affordances for robots: a brief survey*, Published in Avant, 3(2):preceding p71,2012.

[10] J. K. O'Regan, *Red Doesn't Sound Like a Bell: Understanding the feel of consciousness*, Oxford, 2011.

[11] P. Y. Oudeyer, F. Kaplan, and V. V. Hafner, *Intrinsic Motivation Systems for Autonomous Mental Development*, IEEE Transactions on Evolutionary Computation, 11(2):265–286, 2007.

[12] J. Piaget, *The construction of reality in the child*, New York: Basic Books, 1937/1954.

[13] Henri Poincaré, *La Science et l'Hypothèse*, ed. Flammarion, 1902.

[14] E. Sahin et al., *To afford or not to afford: A new formalization of affordances towards affordance-based robot control*, Published in Adaptive Behavior, 15(4):447–472, 2007.

[15] A. Stoytchev, *Learning the Affordances of Tools using a Behavior-Grounded Approach*, In "Affordance-Based Robot Control," Springer Lecture Notes in Artificial Intelligence 4760, pp. 140–158, 2008.