International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:14, No:10, 2020

A Constructivist Approach and Tool for Autonomous Agent Bottom-up Sequential Learning

Jianyong Xue, Olivier L. Georgeon, Salima Hassas

Abstract—During the initial phase of cognitive development, infants exhibit amazing abilities to generate novel behaviors in unfamiliar situations, and explore actively to learn the best while lacking extrinsic rewards from the environment. These abilities set them apart from even the most advanced autonomous robots. This work seeks to contribute to understand and replicate some of these abilities. We propose the Bottom-up hiErarchical sequential Learning algorithm with Constructivist pAradigm (BEL-CA) to design agents capable of learning autonomously and continuously through interactions. The algorithm implements no assumption about the semantics of input and output data. It does not rely upon a model of the world given a priori in the form of a set of states and transitions as well. Besides, we propose a toolkit to analyze the learning process at run time called GAIT (Generating and Analyzing Interaction Traces). We use GAIT to report and explain the detailed learning process and the structured behaviors that the agent has learned on each decision making. We report an experiment in which the agent learned to successfully interact with its environment and to avoid unfavorable interactions using regularities discovered through interaction.

Keywords—Cognitive development, constructivist learning, hierarchical sequential learning, self-adaptation.

I. Introduction

LEARNING from interactions plays an important role in an agent's cognitive development. It not only makes up the deficiencies in artificial designing, but also gives us a way to improve the agent's performance with requiring little or no manual intervention. Meanwhile, in the domain of artificial intelligence (AI), it is becoming increasing accepted as a viable alternative paradigm in designing a self-motivated and self-adaptive agent that can behave in an intelligent and flexible manner under dynamic conditions [1]–[3]. Seeking and designing this paradigm for the agent interacting by AI systems has become a hot topic in research [4], [5].

Imagine the following scenario: an agent is placed in an unfamiliar environment without any prior knowledge, only with innate actions that enable it to perform elementary functions such as moving forward, turning its direction and touching the environment. Different with other scenarios, the agent's interaction is under conditions that without any structured behaviors and prior knowledge about the environment, nor the final goals for the agent to achieve. It leaves us a big challenge, that of how can we design an alternative learning paradigm that the agent could successfully interact with its environment and learn to avoid unfavorable interactions using the structured behavior it has learned from interactions.

In the study of the beginnings of mental development and the origins of intelligence in children, Piaget [6] proposes a knowledge acquiring theory, constructivism, which describes the cognitive development of infants. It is acknowledged that infants are good at playing. Within the initial phase of cognitive development, they exhibit amazing abilities to generate novel behaviors with unfamiliar situations and explore actively to learn the best with lacking extrinsic rewards from the environment. With the skills and abilities they were born with (such as looking, sucking, grasping and listening), babies experience the world and gain knowledge through movements between senses and motor interactions. As the interact with the world around them, they continually absorb new knowledge build upon existing knowledge and adapt previously held ideas to accommodate new information.

Inspired from the theory of constructivism, we propose a bottom-up hierarchical sequential learning model with comstructivist paradigm (BEL-CA), an algorithm for autonomous and continuous learning of environment representations and agent's self-adaptation. Our approach neither initially endows the agent with the prior knowledge of its environment, nor supplies it with knowledge during its learning process. Instead, we propose a way for the agent to autonomously encode the interactional experiences and reuse behavioral patterns based on the agent's intrinsic motivation (we prefer to call it the interactional motivation [7]). Therefore, the agent gets the perception of its world and generates proper behaviors in different and also complicated situations. Thus, the agent could be moving around freely and learn regularities of the environment. Meanwhile, our agent can discover a long sequence of "correct" actions to find a configuration of the environment that yields the non-stationary valence.

The paper is structured as follows. In section II, we introduce current developments in solving the problems we are faced with. Section III provides the theoretic foundation of constructivist learning. In section IV, we explain the learning process with constructivist paradigm and give the formal model proposed of bottom-up hierarchical sequential learning with constructivist paradigm, as well as the BEL-CA. In section V, we introduce the methodology and the experimental settings and also introduce the implementation of toolkit GAIT for analyzing all interaction results. Finally, we conclude and

J. Xue and S. Hassas are with the Laboratory of Computer Science on Image and System Information (LIRIS CNRS, UMR5205), Department of Computer Science, University Claude Bernard Lyon 1, Lyon, 69100 France (e-mail: {jianyong.xue, salima.hassas}@liris.cnrs.fr).

O. Georgeon is with the unit of CONFLUENCE Sciences & Humanities research and the Laboratory of Computer Science on Image and System Information (LIRIS CNRS, UMR5205), Catholic University of Lyon, Lyon, 69002 France (e-mail: ogeorgeon@univ-catholyon.fr).

provide open issues for possible improvements of our work in the future.

II. Related Work

Traditional artificial intelligence approaches strongly rely on an abstraction of the environment proposed by the system designer. In other words, how well the agent connects with the environment (for example, through actuators/sensors patterns) determines the performance of the agent. Thus its capability of adaptation to different problems/environments is limited.

Reinforcement Learning (RL) [8] and with its advances [9]–[11] as a most popular and successful way, tries to solve this problem by learning to get the maximum cumulative reward of actions, which is used to help in making decisions of actions in response to interactions with the environment. Behavior in such systems is not predefined, but learned from interactions with the environment, enabling the partial self-adaptation to situations that were unexpected or unknown within designing time [5], [12]. However, conditions in such environment change and RL processes designed and deployed at the start of the system operation will become unsuitable and will need further self-adapting during the system execution. Besides, RL needs to explore the environment repeatedly in conditions that the agent is placed in an environments without any final goals or the "terminal state" to achieve. [5].

Another way is inspirations from intrinsic motivations (like curiosity [13], [14]), which drive the development of the world-model making, as a way to replicate some abilities of infants' playing [15]. Playing capacity in this period likely interacts with infants' powerful abilities to understand and model their environment, which amazingly generates flexible actions with familiar environments and novel behaviors with unfamiliar environments.

A related but alternative idea is from the constructivist learning paradigm [1], [5], [6]. In the constructivist paradigm, the agent is not a passive observer of reality, but rather constructs a perception of reality through active interaction [3]. The constructivist theory proposes that humans build internal frameworks of knowledge, and acquire new knowledge either through *assimilation* (incorporating new knowledge into their existing framework) or *accommodation* (re-framing internal representations to the newly acquired external knowledge) [3]. Previously [16], we proposed a causality reconstruction model with constructivist which could let an autonomous agent organize its behavior to fulfill a form of intentionality defined independently of a specific task. With the PetriNet that the agent has learned to predict the consequences of the agent's actions, which explains regularities of interaction through the presence of objects in the agent's surrounding space. The work of [17] introduces a model for self-motivated hierarchical sequence learning with inspirations from Piaget's theories of early-stage developmental learning. The behavior organization is driven by pre-defined values associated with primitive behavioral patterns. The agent learns increasingly elaborated behaviors through its interactions with its environment. These learned behaviors are gradually organized in a hierarchy that reflects how the agent exploits the hierarchical regularities afforded by the environment.

III. The Constructivist Theory

The constructivism as a knowledge acquisition theory proposing that learning happens as a result of an internal mental representations and external perceptions from interactions [18], the process of cognitive development is focused on how to construct a mental model of the world and as a process that occurs due to biological maturation and interaction with the environment [19].

From the theory of constructivism, behavior acts as the adaptation to the environment is controlled through mental organizations called *schema*, which the individual uses to represent the world and generate corresponding actions. Each schema describes both the mental and physical actions involved in understanding and knowing, as a category of knowledge and the process of obtaining that knowledge. For example, as experience happens, the newly obtained information is used to modify, append, or change previous existing schemas. The adaptation is driven by a biological drive to obtain a balance between schemas and the environment (equilibration). Schemas are the basic building blocks of such cognitive models, which is a way of organizing knowledge that enables us to form a mental representation of the world. As schemas become increasingly more complex, which means they are responsible for more complex behaviors, structures are termed. While structures become complex, they could be organized in a hierarchical manner which means from general to specific. In our proposed model, the mechanism behind our bottom-up hierarchical sequential model comes from the way to effectively organize different schemas into different structured behaviors (or hierarchical manners).

There are three processes used in learning and self-adaption: assimilation, accommodation, and equilibration. *Assimilation* refers to a part of the adaptation process which initially proposed by Jean Piaget, as one of the processes by using or transforming the environment so that it can be placed in preexisting cognitive structures. Through assimilation, the agent incorporates new information or experiences into the existing knowledge base, sometimes that reinterprets these new experiences so that they will fit in with previously existing information.

Another part of adaptation involves altering existing schemas to accept new information from the environment, a process known as *accommodation*. The process of accommodation involves changing current cognitive structures, as a result of newly acquired external knowledge or new experiences [5]. New schemas may also be developed during this process. Schemas become more refined, detailed, and nuanced as new information is gathered and accommodated into agent's mind and beliefs about how the environment works.

For keeping the balance between assimilation and accommodation, *equilibration* is proposed as the force that drives the learning process keep going. Reaching a state of equilibrium between the assimilation and accommodation processes is what helps create a sense of stability between the agent and its environment.

As shown in Fig. 1, all processes are used simultaneously

and alternately throughout life as the agent increasingly adapts to the environment in a more complex manner.

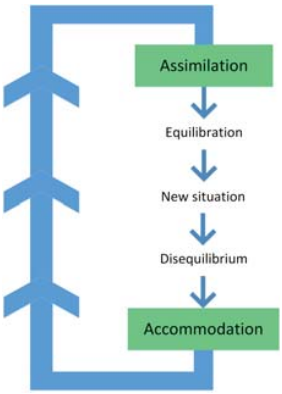


Fig. 1 Three processes of assimilation, accommodation and equilibration. When the agent is faced with a new situation, it takes this new information into the existing schemas is known as assimilation. Accommodation involves modifying existing schemas to accept new information. Equilibration is the force that moves development along

IV. BEL-CA: Bottom-up Hierarchical Sequential Learning with Constructivist Paradigm

A. The Definition of Interaction

As shown in Fig. 2, learning starts with interactions. In our work, the primitive interaction is defined as a tuple of an experiment e_t with its corresponding feedback f_t at time *t*, $i_t = \langle e_t, f_t \rangle$. Additionally, we associate each primitive interaction with a scalar *valence* v_t to qualify the agent's "feeling" from each environmental feedback f_t. For example, if the agent decides to "move forward" from the current position, there are two possible situations afforded for the agent: bumping with the wall or not bumping. Suppose we associate positive valence for the experiment of "move forward" with the feedback of "not bumping", the agent will be satisfied with this interaction and continue enacting interactions that have experiment "move forward" and expecting the feedback of "not bumping". Otherwise, the agent enjoys enacting interactions of the experiment "move forward" and receiving the feedback of "bumping". In order to be consistent with our commonsense, we associate positive valence for "not bumping" and negative valence for "bumping". With primitive interactions, the agent will find ways to organize and experience more interactions for "moving forward" with the feedback of "not bumping" and reduce the interactions with less "bumping".

As for interactions with negative valences, we prefer the valence of experiment "touch front is empty" is better than "touch front is a wall ", then the agent will prefer to "feel front is empty" interactions to lead the agent to "move forward" in a correct direction. The valence assignment for different experiments is an important issue in constructivist learning. The optimal allocation strategy could apparently accelerate and improve the learning process, otherwise it slows down the agent's interaction and even interferes with the learning process. Currently, there is no clear research result to prove which allocation strategy is optimal. In this paper, experiments' valence are allocated mainly based on experimental experience, we will discuss this issue in detail in the final section.

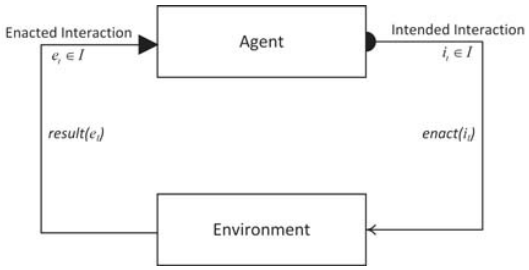


Fig. 2 The interaction cycle between the agent and the environment

Enacting an interaction i_t means that the agent intends an experiment e_t and receives feedback f_t that composites a given interaction at step *t*. The experiment e_t could be a primitive interaction or a series of interactions that the agent is going to enact recursively. Furthermore, the agent intends an interaction i_t which expresses that it performs an experiment e_t while expecting its corresponding feedback f_t at step *t*. With different interactive situations, this "intention" could be that the agent actually enacts interaction $\langle e_t, f_t^{'} \rangle$ if it receives feedback $f_t^{'}$ instead of f_t.

From the perspective of constructivism, intended interaction as it represents the sensorimotor schema that the agent intends to enact, and constitutes the agent's output that is sent to the environment. While the enacted interaction represents the sensorimotor schema that the agent records as actually enacted, which constitutes the agent's input received from the environment. If the enacted interaction equals the intended interaction, then the attempted enaction of intended interaction is considered a *success*, otherwise *failure*.

B. Learning Process with Constructivist Paradigm

At the beginning of each interaction cycle *t* (as shown in Fig. 3), the agent decides an intended interaction $i_t^i = \langle e_t, f_t \rangle$ and tries to enact with reference to the reactive part of the environment. As a result, the agent receives the enacted interaction i_t^e and memorizes the two-step enacted interaction sequence $c_t = \langle i_{t-1}^e, i_t^e \rangle$ as a tuple of $\langle contextInteraction, enactedInteraction \rangle$ made by the previously enacted interaction i_{t-1}^e of i_t^e. The sequence of interaction $\langle i_{t-1}^e, i_t^e \rangle$ called a *composite interaction*, as the pattern of structured behaviors corresponds to the *assimilation* process in constructivism. The interaction i_{t-1}^e is called c_t's pre-interaction, noted as $pre(\langle i_{t-1}^e, i_t^e \rangle)$, and i_t^e is called c_t's post-interaction and is noted as $post(\langle i_{t-1}^e, i_t^e \rangle)$. The tuple of composite interaction expresses that in the context of i_{t-1}^e, the agent learns to recognize its interactive situation in terms of affordances related to its own prior experience, then enacts the proposed enacted interaction in the future to verify its assumptions.

As interaction goes on, more complex composite interactions will be emerged with combinations of different kinds of primitive interactions. To better reflect the

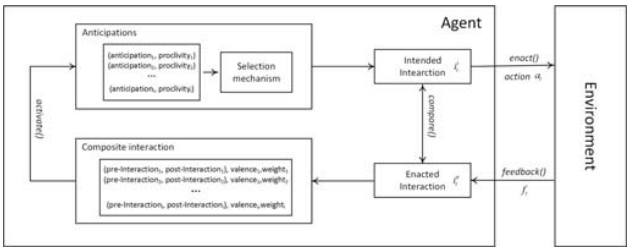


Fig. 3 The learning process with constructivist paradigm

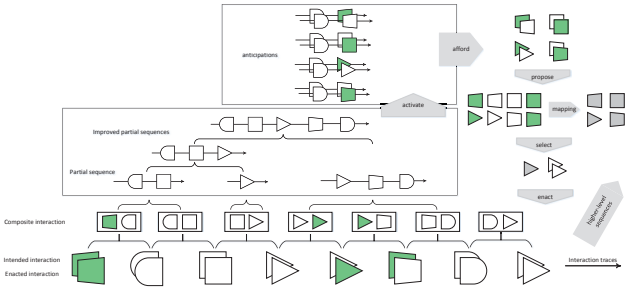


Fig. 4 The model of bottom-up hierarchical sequential learning with constructivist paradigm

closeness of pre-interaction and post-interaction in composite interactions, we associate each composite interaction with a *weight* (initialized as "1") and the *weight* will be incremented when the same composite interaction has learned again (coincide with the *accommodation* process in constructivism). Moreover, composite interaction as one type of interaction, its *valence* is the sum of its pre-interaction's and post-interaction's *valence*.

The set of composite interactions known by the agent at time t is defined as C_t and the set $J_t = I \cup C_t$ is the all interactions known to the agent at time *t*. For the next step interaction, the current enacted interaction i_t^e activates previously learned composite interactions as it matches their pre-interaction, then the agent gets the *activated composited interaction* set A_t. For example, if $i_t^e = a$ and the composite interaction $\langle a, b\rangle$ has been learned before time *t*, then the composite interaction $\langle a, b\rangle$ is activated, as if it is recalled from the memory. Activated composite interactions propose their post-interaction as anticipations for the next round, in this case: the interaction of *b*. The agent's decision making comes from these anticipations.

For each post-interactions, *anticipation* is created with a scalar value *proclivity* $p_i \in \Re$ which is computed from the weight w_{a_i} of the activated composite interaction a_i multiplied by the valence of the proposed post-interaction $v(post(a_i))$. The proclivity value as a way to reflect the regularity of the interaction based on its probability of occurrence and the motivations of the agent.

$$p_i = w_{a_i} \times v(post(a_i)) \quad (1)$$

As a result, the anticipation which is the most likely to result in the primitive interaction that has the highest valence receives the highest proclivity, and that have the biggest proclivity are the most likely to be enacted in the next round.

C. BEL-CA

As shown in Fig. 4, the proposed BEL-CA contains seven parts: initialization, activation, proposition, selection, enaction, learning, and construction. Initialization, agent's interaction starts with innate experiments and possible primitive interactions, with receiving subsequent enacted interactions, the agent constructs composite interactions with previously enacted interactions. Activation, with current enacted interaction, the agent retrieves previously learned composite interactions whose pre-interaction matches with current enacted interactions, then those composite interactions are activated. Proposition, the activated composite interactions propose their post-interactions as anticipations for enacting in the next round. Selection, the anticipations' selection is based on their proclivity values. The anticipation that has the biggest proclivity value will be selected to enact. Learning, after enacting the proposed intended interaction, the agent receives the newly enacted interaction. The composition of current enacted interaction and its previous enacted interaction will be examined in the current learned composite interactions. If it exists, its weight will be incremented. Otherwise, a new more higher-level composite interaction is constructed, which means more complicated structured behavior is learned in this interaction, and hierarchical sequential learning then starts. The mechanism underlying the algorithm could be implemented as a sequential learning process that relies on the interactions between the agent and its environment. We present a high-level overview of BEL-CA in Algorithm 1.

At the beginning of the interaction, with innate experiments and primitive interactions, the agent starts interacting with the environment. Each primitive interaction is associated with a scalar *valence* pre-defined with positive or negative valences according to the nature of different experiments. Results are defined with possible feedback from interacting with the environment. For example, if the agent currently moves forward, moving successfully and bumping with the wall are two different feedbacks from the environment. To distinguish these two kinds of feedback, we used "1" to perform moving successfully and "0" for bumping with the wall. Since there are no enacted interactions and previous experiences, the composite interaction set C_0 is empty as performed with ϕ.

With initialization experiments and primitive interactions, the agent retrieves the previously learned composite interactions whose pre-interaction belongs to the current context B_t, then activates them and forms the set A_t of activated interactions defined as $A_t = \{a_i \in C_t | pre(a_i) \subset B_t\}$ (as shown in Algorithm 2). The activated interactions in A_t propose their post-interaction for anticipations, with proclivity values that are computed from activated interaction a_i's weight and its post interaction's valence $v(post(a_i))$ forming the set AN_t of proposed anticipations: $AN_t = \{anti_i \in AN_t | a_i \in A_t, proclivity_{anti_i} = w_{a_i} \times v(post(a_i))\}$.

With anticipations that start with the same experiment, or called the *partial similar anticipations (PSAs)*, we regroup

Algorithm 1 BEL-CA

Initial:
  experiments = {move forward, turn left, turn right, touch front, touch left, touch right};
  Valences V = {v_moveSucess, v_moveFailture, v_turn, v_feelEmpty, v_feelWall };
  Experiments E = addOrGetExperience("experiment labels",e4:xperiments); anticipatio
  Results Re = [0,1]; 6:
E.resetAbstract();
  primitive interactions I = addOrGetPrimitiveInteraction(E, Re, V);
  default Interactions De = defaultInteractions(I);
  E.setIntendedInteraction(De);
  composite Interaction $C_0 = \phi$ and all interactions $J_0 = I$.
while interactions continues **do**
  anticipations = anticipate();
  selectedAnticipation = selectInteraction(anticipations);
  intendedInteraction = selectedAnticipation.intendedInteraction;16:
  enactedInteraction = enact(intendedInteraction);
  if $intendedInteraction = enactedInteraction$ **then**
    this interaction is *success*;
  else
    this interaction is *failure*;
  end if
  c_t = learnCompositeInteraction(intendedInteraction, enactedInteraction);
  if $c_t \notin C_t$ **then**
    Initial c_t's weight as 1;
    Add c_t in C_t interaction;
  else
    Increase c_t's weight by one;
    Reinforce c_t in C_t;
  end if
end while

Algorithm 2 Activation function and anticipations construction

context interaction B_t = [enacted interaction, composite interaction's post-interaction, super-interaction];
activated composite interaction A_t = [];
default anticipations $Default_t$ = [];
  ns AN_t = [];
for each $i_t \in I$ **do**
  if i_t is primitive **then**
    $anti^i_{default} = createAnticipation(i_t.experiment, 0)$;
    $anti^i_{default}.anticipationsList = []$;
    Add $anti^i_{default}$ in $Default_t$;
  end if
end for
for each $c_i \in C_t$ **do**
  if $pre(c_i) \in B_t$ **then**
    Add c_i in A_t;
  end if
end for
for each $a_i \in A_t$ **do**
  $anti_i$ = $createAnticipation(post(a_i).experiment, 0)$;
  $anti_i.intendedInteraction = post(a_i)$;
  $proclivity_{anti_i} = w_{a_i} \times v(post(a_i))$;
  if $anti_i \notin AN_t$ **then**
    Add $anti_i$ in AN_t;
  end if
end for
for each $anti^i_{default} \in Default_t$ **do**
  for each $anti_i \in AN_t$ **do**
    firstPrimitiveIntearction = getFirstPrimitiveInteraction($anti_i$);
    if $firstPrimitiveIntearction.experiment = anti^i_{default}.experiment$ **then**
      Add $anti_i$ in $anti^i_{default}.anticipationsList$;
      $proclivity_{anti^i_{default}} += proclivity_{anti_i}$;
    end if
  end for
end for

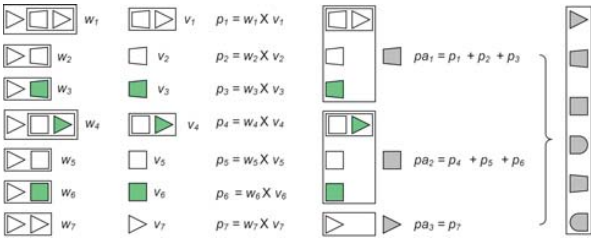


Fig. 5 Mapping partial similar anticipations to experiments.

all anticipations according to their first experiment of the intended interactions and map them all as different lists with corresponding experiments respectively (at Fig. 5). Each experiment's proclivity value is calculated as follows:

$$proclivity_{anti^i_{default}} = \sum_{i=1}^{n} w_{a_i} \times v(post(a_i)) \qquad (2)$$

The $proclivity_{anti^i_{default}}$ from the equation above is the proclivity of experiment *i*, *n* refers to the number of anticipations that share the same first-experiment of primitive interaction in their intended interaction, w_{a_i} is the weight of activated composite interactions a_i and the $v(post(a_i))$ is the valence of a_i's post-interaction.

The intended interaction i^i_t is selected with function *selectInteraction()* from anticipations whose experiment has the biggest proclivity and enacts the intended interaction of its anticipation which the one has the biggest proclivity. If the intended interaction is a composite interaction, the enaction of this intended composite interaction is subject to its anticipation's weight w_{a_i} and the threshold $d \in \Re$. The parameter d is the threshold which encodes the limited of enacting the intended interaction as a whole. If the weight of the anticipation is greater than d and its proclivity value is positive, then the agent will effectively enact all primitive interactions within this intended interaction according to the hierarchical sequential structure recursively. Otherwise, the agent just needs to enact the first primitive interaction of this

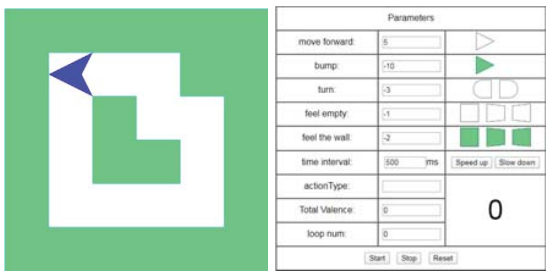
intended interaction. In essence, this mechanism ensures that higher-level schemas are sufficiently rehearsed before being enacted as a whole. If the sequence of intended interaction corresponds to the regularity of interaction, then it is possible that the sequence of this intended interaction can be enacted again. Therefore, the agent can anticipate that performing post interaction's experience will likely produce its result. The agent can thus base its choice of the next interaction on this anticipation.

While enacting intended interactions (as shown in Algorithm 3), the agent checks each enacted primitive interaction with intended primitive interaction and compares the result between them. For enacting composite interactions, the flat sequence of enacted primitive interactions constructs a hierarchical structure according to the enaction sequence and the intended composite interaction's structure.

With enacted interactions, new composite interactions are constructed or reinforced with their pre-interaction belonging to the context and their post-interaction i_t^e, forming the set of learned or reinforced interaction c_t to be included in C_{t+1}, for supporting affordance of more complicated interaction situations in the future. The set c_t is defined as $c_t = \{< i_{t-1}^e, i_t^e >, < i_{t-2}^e, < i_{t-1}^e, i_t^e >>, << i_{t-2}^e, i_{t-1}^e >, i_t^e >\}$, $C_{t+1} = C_t \cup c_t$. A new context B_{t+1} is constructed to include the stabilized interactions in i_t^e and $post(i_t^e)$.

V. Methodology and Experimental Scenario

In order to evaluate the model we proposed in this paper, an experimental scenario was set up in which the agent could move around and touch the environment in three directions: front, left and right. The environment is designed as a Small Loop [20], [21] environment, which composed of white squares represent paths surrounded by green "walls" (as shown in the left figure of Fig. 6). The Small Loop Problem (SLP) as a benchmark to evaluate agents that implement four principles of emergent cognition: environment agnosticism, self-motivation, sequential regularity learning, and spatial regularity learning. Different from most existing benchmarks, the small loop environment does not involve a final goal for the agent to reach, instead, the agent's self-motivation comes from the fact that primitive interactions have different valences.



(a) The classic SLP environment. (b) Parameter settings

Fig. 6 The designed environment and experimental settings

The agent is presented as a blue head arrow and initialized with a random direction. Different with classic Small Loop

Algorithm 3 Enaction intended interaction

```
given proposedIntendedInteraction;
Nodes = [];
nodeStack = [];
if proposedIntendedInteraction.weight ≥ threshold and proclivity is positive then
    Add proposedIntendedInteraction in nodeStack;
else
    Add proposedIntendedInteraction.firstPrimitiveInteraction in nodeStack;
end if
while nodeStack not null do
    topInteraction = nodeStack.pop();
    if topNode is not primitive then
        nodeStack.push(topInteraction.postInteraction);
        nodeStack.push(topInteraction.preInteraction);
    else
        newNode = createNode(topInteraction);
        Add newNode in Nodes;
    end if
end while
node = getLeftNode(Nodes);
while the root node is not visited do
    intendedInteraction = node.getInteraction;
    enactedInteraction = enact(intendedInteraction);
    if intendedInteraction = enactedInteraction then
        previousRecordInteraction = recordWithStructure(enactedInteraction);
        node = getNearestRightNode(node);
    else
        previousRecordInteraction = recordWithStructure(enactedInteraction);
        break;
    end if
end while
learnCompositeInteraction(proposedIntendedInteraction, previousRecordInteraction);
```

environment, our environment is designed as changeable to verify the adaptability of agent in different environments. We use different shapes like a triangle, left and right half-circle square, left and right trapezoid and square to represent moving forward, turn left and turn right, touch left, touch right and touch front, respectively. Colors (like green and white) indicate interactions with the same experiment but receive different possible feedbacks from the environment. The experimenter can preset the parameters to control the interaction process (as shown in the right figure of Fig. 6), such as valences allocation for primitive interactions, using interaction "interval" speed up or slow down the interaction, "actionType" indicates the current experiment the agent enacts, "Total valence" represents the cumulated valence the agent has received from each interactions, and "loopNum" presents the decision-making times. In order to better observe the interaction between agents and the environment, as well as the agent's gradually learning process, we proposed and developed a toolkit named

"Generating and Analyzing Interaction Traces toolkit" (GAIT) as a way to investigate the detailed learning process for agent interacting with the environment and each structured behavior it has learned within each decision-making step.

A. Generating and Analyzing Interaction Traces (GAIT)

The framework of GAIT records each agent's enacted interaction and forms a streamline of continuous interaction traces. In the area of interaction traces, each time select a primitive intended/enacted interaction, the constructed composite interaction tip windows will pop out all composite interactions that the agent has learned. The surrounded green rectangles indicate that the composite interactions are newly learned, while the blue ones are already learned before but enforced in this interaction. Left-clicking on any enacted interactions, the sorted experiment list will pop out with its proclivities. For the case that experiments have proposed anticipations, we surround them with pink rectangles. If continue selecting the experiment, more detailed anticipations are displayed with detailed intended interactions and sorted with their proclivities. As for enacting composite interactions, the light green rectangle fields on the "loop number" could pop out a tip window that includes the intended composite interactions with its enacted composite interaction to show the detailed interaction process. In order to identify the range of interactions in this composite interaction, we surround all primitive interactions the agent has enacted with a yellow rectangle.

The scroll button at the bottom of the interaction traces could be used to show all previous interactions. In our experiment, the proposed toolkit could support tens of thousands interactions which makes it easy to look back at all previous interactions.

B. Interaction Traces Analysis

At the beginning of the interaction, the agent randomly selects an experiment (touch right) and intends the default intended interaction (the green right trapezoid), with feedback from the environment, the agent receives the same result. At step 2, the agent memorizes the previously enacted interaction with the current enacted interaction and forms a composite interaction. Especially in step 3, the agent not only memorizes the previously enacted interaction, and also combines the previously learned composite interaction to construct more higher-level composite interaction (as shown in Fig. 7).

The proposed intended interaction appears at step 9 (as shown in Fig. 8), the experiment "move forward" has the highest proclivity and its intended interaction (the white triangle) is activated with the highest proclivity (the proclivity value is 15). Hence the agent selected this white triangle (move forward) and got the same white triangle (the agent successfully moves forward a step), with previous statement, this enaction is considered as a *success*. While the opposite situation happens in step 14, the agent intends the same white triangle but bumps with the wall (a green triangle), thus this interaction as *failure*.

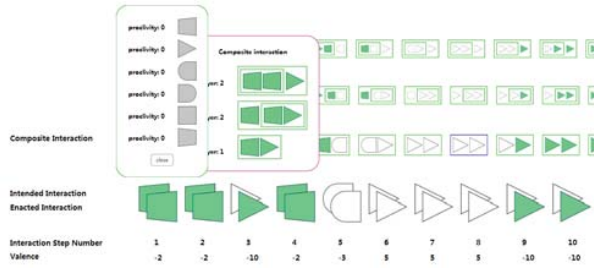


Fig. 7 The first several interactions and composite interactions construct process

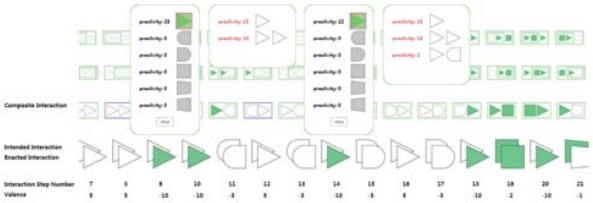


Fig. 8 Enact the same intended interaction with different feedback

At step 23, the agent is going to enact a composite interaction, with the reason that this anticipation's weight is less than the threshold, then the agent intends the first primitive interaction (left half-circle) and receives the same enacted interaction (as shown in Figure 9). In our implementation, we use different colors of proclivities to identify their anticipations' weight beyond the threshold or not, the red color presents the weight is less than the threshold while the green color presents the weight is higher than the threshold.

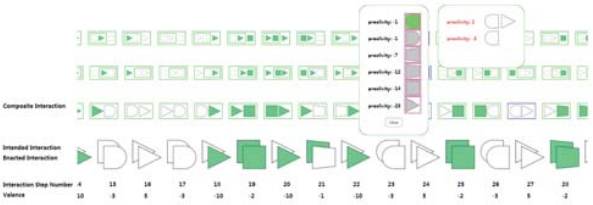


Fig. 9 The enacted composite interaction's weight less than the threshold

At step 119, the agent gets an intended composite interaction with the weight beyond the threshold, then it sequentially intends this composite interaction successfully receives the same enacted composite interaction (as shown in Fig. 10). The agent combines this enacted composited interaction with previously enacted interaction and learned composite interaction to construct higher-level and more complex composite interaction.

With interactions continue, the agent could successfully interact with the environment and learn to avoid unfavorable interactions (bumping with the wall) by using regularities it has learned. More complicated behavioral patterns have constructed and it could generate properly with different situations (as shown in Fig. 11).

From the agent's 600 interactions, we can find that the agent bumps with the wall at the beginning several interactions,

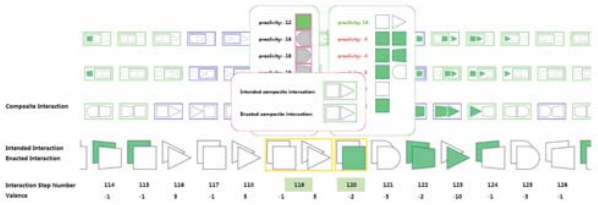


Fig. 10 The agent enacts composite interaction and constructs higher-level composite interaction

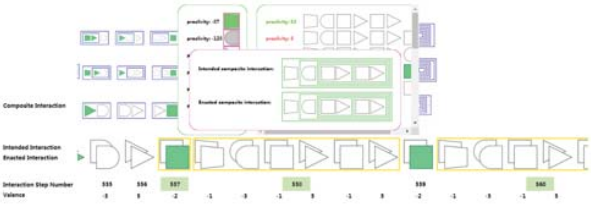


Fig. 11 Enacting complicated composite interaction

however, from 357th interaction (as shown in the top Fig. 12), the bumping phenomenon disappears and the agent could successfully interact with its environment with starting to prevent unfavorable interactions using regularities that it has learned. This could be confirmed from the perspective of cumulated valences the agent has received from interactions (as shown in the bottom Fig. 12). In the beginning interactions, the agent could move forward successfully, but in most cases, the agent is taking various explorations and attempts, hence the total valence reduces in this period. Since 357th interaction, we find that the total valence begins to rise, and the increase gradually increases until stable, which as a way to prove the emergence of sense-making and cognitive development in the agent's interactions with the environment.

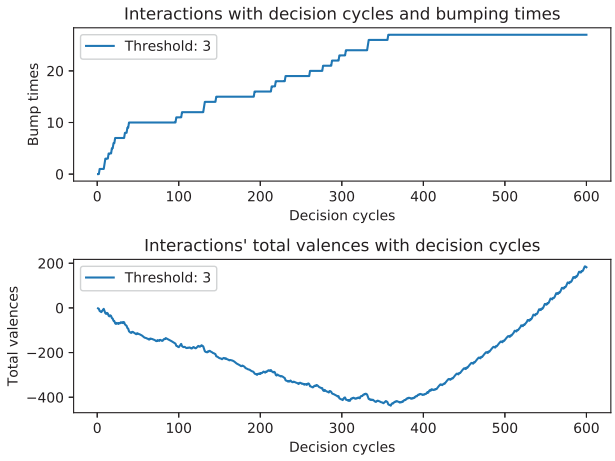


Fig. 12 Agent's bumping times and cumulated valence with interactions

VI. Conclusion and Open Issues

This paper presents BEL-CA, a bottom-up hierarchical sequential learning model for autonomous and continuous learning of environment representation and agents' self-adaptation. The agent autonomously organizes schemas it has learned from interactions into hierarchically structured behaviors, which allows the agent gradually understand the meaning of divers experiments and infers the structure of the environment simultaneously based on the patterns in the stream of interaction feedback traces. Meanwhile, an implementation of GAIT for autonomously generating and analyzing interactions at run-time, which could allow us observe the detailed learning process for agent interacting with the environment and each structured behavior it has learned within each decision-making step.

We evaluated the agent's cognition emergence with an improved Small Loop Problem (SLP) environment, in which the changeable environment is designed for simulating agent's performance in different levels of complex scenarios. With interaction traces from GAIT and the hierarchically structured behaviors the agent has learned, the agent gradually exploits the hierarchical regularities afforded by the environment and learns to avoid unfavorable interactions using regularities that it has learned. Nevertheless, the agent has to retrospect all composite interactions to retrieve the one whose pre-interaction matches the current enacted interaction. With interactions continuing, interaction traces grow progressively longer, hence the agent spends a long time activating all eligible composite interactions for anticipations. In addition, the utility rate of composite interactions needs to be improved. As shown in Fig. 13, the agent activated almost all composite interactions but few are proposed for intending. Although the agent can be easily qualified for the work in the environment designed in this paper, when the environment becomes more complex, this shortcoming will be easily shown.

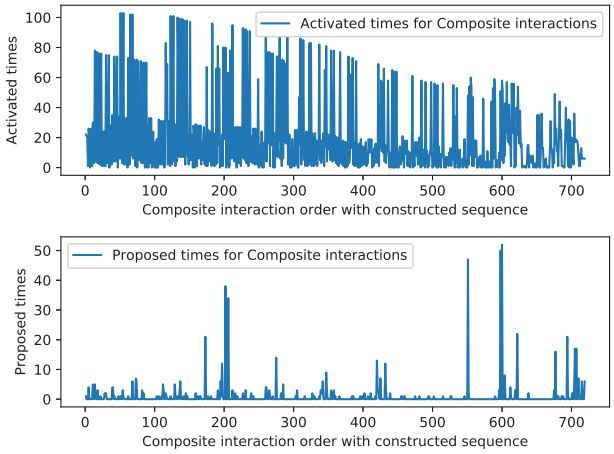


Fig. 13 Utility rate of composite interaction

Valence initialization is another issue that we need to face. According to the common sense of human beings, assume the agent gets positive hints if it can successfully take a step forward and the negative feedback for collisions with the wall. As for the agent, it starts interactions without any prior knowledge, which means it should comprehend the feedback of different behaviors from their own interaction with the

environment. For the initialization of valence, it is inevitable to have a certain influence on the cognitive process of the agent to some extent. In the following research, we need to study to reduce human intervention as much as possible, in order to allow the agent explore for itself and discover how to find the optimal valence allocation strategy from the interaction.

Further work will be mainly focused on optimizing our model and upgrading the proposed toolkit. For example, we could use a predictive model for better-proposing anticipations for the agent of interacting with the environment in the next round. This could be achieved with memorizing patterns that could improve the learning efficiency and eliminating composite interactions that probably will not be used to simplify the activation and proposition processes in BEL-CA in the future. With this hierarchical sequential learning model, the aim would be like to evaluate the performance of the agent in a multi-agent scenario, which provides more challenges and opportunities to improve its learning ability.

REFERENCES

[1] M. Guériau, F. Armetta, S. Hassas, R. Billot, and N.-E. El Faouzi, "A constructivist approach for a self-adaptive decision-making system: application to road traffic control," in *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2016, pp. 670–677.

[2] Y. Bu, J. Lu, and V. V. Veeravalli, "Active and adaptive sequential learning," *arXiv preprint arXiv:1805.11710*, 2018.

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

[4] G. Anthes, "Lifelong learning in artificial neural networks," *Communications of the ACM*, vol. 62, no. 6, pp. 13–15, 2019.

[5] M. Guériau, N. Cardozo, and I. Dusparic, "Constructivist approach to state space adaptation in reinforcement learning," *Learning*, vol. 4, no. S3, p. S2, 2019.

[6] J. Piaget, *The construction of reality in the child*. Routledge, 2013, vol. 82.

[7] O. L. Georgeon, J. B. Marshall, and S. Gay, "Interactional motivation in artificial systems: Between extrinsic and intrinsic motivation," in *2012 IEEE international conference on development and learning and epigenetic robotics (ICDL)*. IEEE, 2012, pp. 1–2.

[8] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[9] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," in *Advances in neural information processing systems*, 2016, pp. 3675–3683.

[10] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[11] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 1.

[12] M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degrave, T. Van de Wiele, V. Mnih, N. Heess, and J. T. Springenberg, "Learning by playing-solving sparse reward tasks from scratch," *arXiv preprint arXiv:1802.10567*, 2018.

[13] N. Haber, D. Mrowca, L. Fei-Fei, and D. L. Yamins, "Emergence of structured behaviors from curiosity-based intrinsic motivation," *arXiv preprint arXiv:1802.07461*, 2018.

[14] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE transactions on evolutionary computation*, vol. 11, no. 2, pp. 265–286, 2007.

[15] A. E. Stahl and L. Feigenson, "Observing the unexpected enhances infants' learning and exploration," *Science*, vol. 348, no. 6230, pp. 91–94, 2015.

[16] J. Xue, O. L. Georgeon, and M. Gillermin, "Causality reconstruction by an autonomous agent," in *Biologically Inspired Cognitive Architectures Meeting*. Springer, 2018, pp. 347–354.

[17] O. L. Georgeon, J. H. Morgan, and F. E. Ritter, "An algorithm for self-motivated hierarchical sequence learning," in *Proceedings of the International Conference on Cognitive Modeling. Philadelphia, PA. ICCM-164*. Citeseer, 2010, pp. 73–78.

[18] F. Guerin, "Constructivism in ai: Prospects, progress and challenges." in *AISB Convention*, 2008, pp. 20–27.

[19] W. Huitt and J. Hummel, "Piaget's theory of cognitive development," *Educational psychology interactive*, vol. 3, no. 2, pp. 1–5, 2003.

[20] O. L. Georgeon, C. Wolf, and S. Gay, "An enactive approach to autonomous agent and robot learning," in *2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*. IEEE, 2013, pp. 1–6.

[21] O. L. Georgeon and J. B. Marshall, "Demonstrating sensemaking emergence in artificial agents: A method and an example," *International Journal of Machine Consciousness*, vol. 5, no. 02, pp. 131–144, 2013.

Jianyong Xue is a PhD student in the Laboratory of Computer Science on Image and Information System (LIRIS CNRS UMR5205), Department of computer science, University Claude Bernard Lyon 1, France. His work mainly focuses on developmental learning and constructivist paradigm. He obtained Master's Degree from North China Electric Power University with a major of Computer Applied Technology and Bachelor's Degree in computer science and technology at University of Datong.

Olivier L. Georgeon Olivier L. Georgeon is an associate professor at the Catholic University of Lyon and a researcher in computer science at the LIRIS Laboratory (CNRS-UMR 5205). He has an interdisciplinary background with a master of engineering in computer science from the Ecole Centrale de Marseille and a PhD in cognitive psychology from the Université Lumière Lyon 2. His research focuses on the implementation of developmental learning mechanisms and intrinsic motivation in artificial agents.

Salima Hassas Salima Hassas is Full Professor at the Department of Computer Science of Polytech Lyon at Claude Bernard-Lyon1 University. She is affiliated to the LIRIS-CNRS Laboratory, where she leads a research activity on Multi-Agents Systems , Autonomous Agents and Self-* Systems.