



Intelligence artificielle sans données ontologiques sur une réalité présumée

Olivier Georgeon, Alain Mille, Simon Gay

► To cite this version:

Olivier Georgeon, Alain Mille, Simon Gay. Intelligence artificielle sans données ontologiques sur une réalité présumée. *Intellectica - La revue de l'Association pour la Recherche sur les sciences de la Cognition (ARCo)*, Association pour la Recherche sur la Cognition, 2016, New approaches in cognitive robotics, 65, pp.143-168. <<http://intellectica.org/en/artificial-intelligence-without-using-ontological-data-about-presupposed-reality>>. <hal-01379575>

HAL Id: hal-01379575

<https://hal.archives-ouvertes.fr/hal-01379575>

Submitted on 11 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Artificial intelligence without using ontological data about a presupposed reality

Abstract

This paper introduces an original model to provide software agents and robots with the capacity of learning by interpreting regularities in their stream of sensorimotor experience rather than by exploiting data that would give them ontological information about a predefined domain. Specifically, this model pulls inspiration from: a) the movement of embodied cognition, b) the philosophy of knowledge, c) constructivist epistemology, and d) the theory of enaction. Respectively to these four influences: a) Our agents discover their environment through their body's active capacity of experimentation. b) They do not know their environment "as such" but only "as they can experience it". c) They construct knowledge from regularities of sensorimotor experience. d) They have some level of *constitutive autonomy*. Technically, this model differs from the traditional perception/cognition/action model in that it rests upon atomic sensorimotor experiences rather than separating percepts from actions. We present algorithms that implement this model, and we describe experiments to validate these algorithms. These experiments show that the agents exhibit a certain form of intelligence through their behaviors, as they construct proto-ontological knowledge of the phenomena that appear to them when they observe persistent possibilities of sensorimotor experiences in time and space. These results promote a theory of artificial intelligence without ontological data about a presupposed reality. An application includes a more robust way of creating robots capable of constructing their own knowledge and goals in the real world, which could be initially unknown to them and unmodeled by their designers.

Keywords: artificial intelligence, embodied cognition, constructivist learning, enaction, hierarchical sequence learning, trace-based reasoning, cognitive architecture, self-motivation.

Intelligence artificielle sans données ontologiques sur une réalité présupposée

Olivier Georgeon¹, Alain Mille², et Simon Gay³

¹²³Université de Lyon, CNRS

Université Lyon 1, LIRIS, UMR5205, F-69622, France.

¹olivier.georgeon@liris.cnrs.fr, ²alain.mille@univ-lyon1.fr, ³simon.gay@liris.cnrs.fr

Résumé

Cet article propose un modèle original pour doter des agents informatiques ou des robots de la capacité d'apprendre en interprétant des régularités dans leur flux d'expériences

sensorimotrices plutôt qu'en exploitant des données qui leur apporteraient des informations ontologiques sur un domaine prédéfini. Ce modèle s'inspire en particulier de : a) le courant de la cognition incarnée, b) la philosophie de la connaissance, c) l'épistémologie constructiviste, et d) la théorie de l'énaction. Respectivement à ces quatre influences : a) Nos agents découvrent leur environnement à travers les capacités expérimentales actives de leur corps. b) Ils ne connaissent pas leur environnement « en soi » mais uniquement « en ce qu'ils peuvent en faire l'expérience ». c) Ils construisent leurs connaissances à partir de régularités d'expériences sensorimotrices. d) Ils disposent d'une certaine *autonomie constitutive*. Techniquement, ce modèle se distingue du modèle perception/cognition/action classique par le fait qu'il considère des expériences sensorimotrices atomiques au lieu de séparer les percepts et les actions. Nous présentons des algorithmes qui implémentent ce modèle, et décrivons des expérimentations permettant de les valider. Les expérimentations montrent que les agents exhibent une certaine forme d'intelligence dans leurs comportements en construisant une connaissance proto-ontologique des phénomènes qui apparaissent à eux quand ils constatent des possibilités d'expériences sensorimotrices persistantes dans l'espace et le temps. Ces résultats promeuvent une théorie de l'intelligence artificielle sans données ontologiques sur une réalité présumée, avec, comme perspectives applicatives, des robots capables de construire leurs propres connaissances et objectifs dans le monde réel, initialement inconnu d'eux et non modélisé par leur concepteur.

Mots clés: intelligence artificielle, cognition incarnée, apprentissage constructiviste, éniation, apprentissage séquentiel hiérarchique, raisonnement à partir de trace, architecture cognitive, motivation intrinsèque.

1 Introduction

Nous nous intéressons au problème de concevoir des *agents informatiques* capables, en parallèle, de découvrir des *régularités dans leur flux d'expériences sensorimotrices*, de *construire des connaissances* qui expliquent et synthétisent ces régularités, et d'exploiter ces connaissances pour générer des *comportements intelligents*. Nous désignons ce problème par l'expression *apprentissage par régularités d'expériences*. Dans l'énoncé de ce problème, l'expression *agent informatique* (ou, dans la suite de cet article, simplement *agent*) désigne une entité programmée interagissant avec un environnement, par exemple un robot dans le monde réel ou un agent artificiel dans un monde virtuel. L'expression *régularités d'expériences* désigne des schémas temporels (ou spatio-temporels) d'activité de l'agent rendus possibles de manière persistante par son couplage avec l'environnement. L'expression *construire des connaissances* désigne le fait que l'algorithme qui contrôle l'agent construit des structures informationnelles accompagnées de mécanismes permettant de les exploiter. L'expression *comportements intelligents* désigne des comportements dont des observateurs humains puissent juger qu'ils dénotent une forme d'intelligence de la part de l'agent qui les effectue.

Cet article présente les modèles théoriques et les principes des algorithmes, en renvoyant à des articles plus techniques pour plus de détails. Il présente les expérimentations réalisées, et décrit les comportements exhibés par les agents, en expliquant dans quelle mesure ces comportements peuvent être considérés comme intelligents. Le terme *algorithme* désigne le dispositif

informatique (programme et structures mémorielles) qui contrôle l'agent. Nous rapportons ici uniquement des expérimentations qui n'impliquent qu'un seul agent interagissant avec son environnement.

Notre approche se démarque des approches d'intelligence artificielle symbolique par le fait que les données d'entrée des algorithmes ne constituent pas des symboles qui seraient interprétés selon des règles sémantiques données par le concepteur de l'algorithme. Spécifiquement, nous n'adoptons pas l'hypothèse des symboles physiques (Newell et Simon, 1976), selon laquelle l'intelligence consisterait en un mécanisme de résolution de problème par recherche d'un « état solution » dans un « espace de problème »¹. Nos algorithmes n'ont pas pour but d'atteindre un objectif final spécifié sous la forme d'un état particulier appartenant à un ensemble d'états prédéfini. Cette prise de distance par rapport à l'IA symbolique nous rapproche du mouvement de la *cognition incarnée* (e.g., Varela et al., 1991 ; Lakoff et Johnson, 1999 ; Anderson, 2003). Ces auteurs suggèrent que les systèmes cognitifs seraient mus par des préférences comportementales (« *drives* ») plutôt que par des objectifs finaux prédéfinis. Dans ce cadre, la cognition est vue comme une « façon d'être dans le monde »². En conformité avec cette approche, nous évaluons nos agents par leur capacité à « se débrouiller dans le monde » (« *to cope with the world*», Dreyfus, 2007)³ plutôt que par leur performance à résoudre un problème prédéfini.

Nous nous démarquons également de nombreux travaux d'intelligence artificielle non-symboliques par le fait que les données d'entrée de nos algorithmes ne constituent pas une représentation de l'état du *monde* (pas même une représentation partielle ou bruitée, c.f., Section 2). Le terme *monde* englobe à la fois l'environnement et le « corps » de l'agent (un robot physique ou un corps simulé dans un environnement virtuel). Par exemple, notre position se distingue de la position de Russell et Norvig selon laquelle "*the problem of AI is to build agents that receive percepts from the environment and perform actions*" (Russell et Norvig, 2003, p. iv). A la différence des algorithmes présentés par *Russell et Norvig*, nos algorithmes ne traitent pas leurs données d'entrée comme des percepts mais comme des résultats d'expériences qui résultent d'une interaction active entre l'agent et l'environnement. A la différence des percepts,

¹ A physical symbol system exercises its intelligence in problem solving by search—that is, by generating and progressively modifying symbol structures until it produces a solution structure. (Newell & Simon, 1976, p. 120).

² "This shift in focus from Descartes' "thinking thing", and the picture of human being and subjectivity it suggests, to a more Heideggerian approach to being in the world, in which agency and interactive coping occupy center stage, is an extremely important development, the implications of which are only just beginning to be fathomed." (Anderson, 2003, p. 91)

³ When we solve problems, we do sometimes make use of representational equipment outside our bodies, but Heidegger's crucial insight is that being-in-the-world is more basic than thinking and solving problems; it is not representational at all. That is, when we are coping at our best, we are drawn in by affordances and respond directly to them, so that the distinction between us and our equipment—between inner and outer—vanishes (Dreyfus, 2007, p. 1146).

ces résultats d'expérience n'entretiennent pas une relation de correspondance avec un monde présumé. L'algorithme ignore ce que ces expériences signifient. Il construit des connaissances à partir de régularités constatées dans son flux d'expérience. L'algorithme apprend à connaître l'environnement et l'agent en termes des possibilités d'interaction qui existent entre les deux.

Ce travail vise à la fois des objectifs théoriques et pratiques dans le domaine de l'intelligence artificielle. Sur le plan théorique, il participe à un effort de la communauté de l'intelligence artificielle pour prendre en compte certaines critiques philosophiques auxquelles nous référons ici par les termes de *critique kantienne* et *critique heideggérienne*. Nous proposons ce travail comme une tentative de prise en compte de la critique kantienne selon laquelle la réalité nouménale—la « chose en soi »—n'existe pas ou est inconnaissable. Nous traduisons cette idée philosophique dans les algorithmes par le fait que les données d'entrée de l'algorithme ne représentent pas un monde présumé par le concepteur de l'algorithme.

Nous proposons également ce travail comme tentative de prise en compte de la critique heideggérienne selon laquelle la cognition ne se réduit pas à un mécanisme de résolution de problème, et nécessite une évolution dynamique du couplage agent/environnement. Nous traduisons ces idées philosophiques en implémentant des préférences comportementales qui sous-déterminent les comportements de l'agent. Du fait de l'évolution du couplage algorithme/monde (Section 3), l'agent a la possibilité de développer sa propre « façon d'être dans le monde », en fonction de son histoire individuelle. Nous n'évaluons pas nos agents par leur capacité à résoudre des problèmes prédéfinis ni à maximiser une valeur numérique, mais par leur capacité à générer des comportements que le lecteur de cet article ou un observateur de l'agent pourra juger intelligent.

En ce qui concerne les objectifs pratiques, nous cherchons à construire des robots capables d'interagir avec un environnement non modélisé a priori, par exemple le monde réel (incluant éventuellement d'autres robots et des acteurs humains). Au fur et à mesure des progrès dans ce domaine de recherche, ces robots généreront des comportements qui, du point de vue du grand public et de l'industrie, pourront évoquer des comportements d'animaux. Cela permettra de nouveaux usages en robotique personnelle, pédagogique, ou ludique. Par ce rapprochement avec l'intelligence animale, cette étude s'inscrit dans une approche évolutionniste de l'intelligence artificielle qui considère que les facultés de raisonnement s'appuient sur des capacités d'inférence sensorimotrice déjà présentes chez certains animaux (e.g., Lakoff et Johnson, 1999)⁴.

⁴ "Reason is evolutionary, in that abstract reason builds on and makes use of forms of perceptual and motor inference present in "lower" animals. The result is a Darwinism of reason, a rational Darwinism: Reason, even in its most abstract form, makes use of, rather than transcends, our animal nature. The discovery that reason is evolutionary utterly changes our relation to other animals and changes our conception of human beings as uniquely rational. Reason is thus not an essence that separates us from other animals; rather, it places us on a continuum with them. (Lakoff and Johnson 1999, p. 4)

2 Du modèle classique au modèle expérientiel

La plupart des modèles d'intelligence artificielle sont construits sur la base du cycle perception / cognition / action illustré en Figure 1a. A chaque tour du cycle, l'algorithme reçoit une donnée d'entrée o (souvent appelée *observation*) qui représente l'état du monde, et produit une donnée de sortie a qui représente une *action* effectuée dans le monde. Le concepteur de l'algorithme conçoit l'observation o comme une *représentation* partielle de l'état du monde, au sens étymologique du terme représentation, c'est à dire que l'observation o « rend présente à nouveau » une caractéristique du monde sous la forme d'une donnée accessible à l'algorithme. Si l'observation o est bruitée, les algorithmes typiquement conçus dans le cadre de ce modèle tentent d'extraire l'information représentative du monde en filtrant le bruit par des méthodes statistiques.

Cet article propose des modèles alternatifs qui ne considèrent pas les données d'entrée de l'algorithme comme des représentations du monde. Afin de mettre en évidence leurs différences par rapport au modèle classique, nous affichons dans les figures le point où le cycle d'interaction commence et le point où il se termine. Bien que le cycle d'interaction tourne indéfiniment, nous voulons montrer que ces débuts et fins conceptuelles ont leur importance. La Figure 1b représente le modèle classique en mettant en évidence le fait que le cycle commence par l'observation (rond noir), et se termine par l'action (triangle noir), comme nous l'avons expliqué au paragraphe précédent.

La Figure 1c présente notre premier modèle alternatif appelé modèle expérimentation / résultat (Georgeon et Cordier, 2014). Dans ce modèle, le cycle commence conceptuellement par le fait que l'algorithme sélectionne une expérimentation x dans l'ensemble X des expérimentations à sa disposition. En retour, l'algorithme reçoit un résultat r dans l'ensemble R des résultats possibles. Le développeur de l'algorithme peut penser aux données de sortie comme à des expérimentations *spontanées* au sens où elles ne suivent pas un protocole expérimental réfléchi. Une expérimentation peut produire un effet sur le monde, tout comme une action dans le modèle classique. La différence cruciale par rapport au modèle classique réside dans le fait que les données d'entrée de l'algorithme (les résultats r) *ne constituent PAS nécessairement une représentation de l'état du monde*, puisque, dans un état du monde donné, r peut varier selon l'expérimentation x précédemment effectuée. L'algorithme est un observateur actif du monde car les données qu'il reçoit (les résultats) répondent à des « questions qu'il pose » (les expérimentations). Le modèle expérimentation / résultat offre un moyen de traduire les théories de la cognition incarnée dans les algorithmes car il rend compte du fait que l'algorithme ne peut connaître le monde qu'au travers des expérimentations effectuées par l'agent. Dans ce modèle, le développeur de l'algorithme considère la perception comme une structure de données interne à l'algorithme et non comme ses données d'entrée.

Le modèle interactionnel présenté en Figure 1d prolonge la logique du modèle expérimentation / résultat avec la différence qu'il se base sur des *interactions* définies comme des couples $\langle \text{expérimentation}, \text{résultat} \rangle$. Les données de sortie et d'entrée de l'algorithme appartiennent au même ensemble $X \times R$ des interactions offertes par le couplage agent/environnement. En début de cycle, l'algorithme sélectionne une *interaction intentée* $i = \langle x, r \rangle \in X \times R$. En retour, il reçoit une

interaction énoncée $e = \langle x, r' \rangle \in X \times R$. Nous utilisons l'anglicisme « énoncé » pour exprimer le fait que cette interaction a effectivement été effectuée par l'agent interagissant avec son environnement. Si l'interaction énoncée e est la même que l'interaction intentée i (c'est à dire si $r' = r$), alors nous disons que la tentative d'énoncé de i a *réussi*, sinon, cette tentative a *échoué*. Un exemple d'interaction peut être donné par un robot touchant un objet (le toucher implique de manière indissociable un mouvement relatif et une perception). Le robot peut tenter cette interaction, et réussir à l'énoncer si l'objet est présent, ou échouer si l'objet est absent, auquel cas, le robot a énoncé une autre interaction correspondant à *bouger dans le vide*. Le modèle interactionnel permet de prédéfinir des préférences comportementales de l'agent en associant des valences numériques aux interactions. En section 3, nous présentons un algorithme qui cherche à énoncer des interactions associées à une valence positive, et à éviter d'énoncer les interactions associées à une valence négative. Les agents contrôlés par cet algorithme exhibent une forme de motivation propre que nous appelons *motivation interactionnelle* (Georgeon et al., 2012). Ces comportements sont décrits en Section 4.

Notons que cette approche permet également d'implémenter des agents « curieux » si l'algorithme sélectionne des interactions qui ont le moins été testées dans un contexte donné. Il permet également d'implémenter des agents qui semblent aimer être « au contrôle » de leur propre activité—une forme de motivation appelée principe autotélique par Steels (2004). Pour cela l'algorithme doit sélectionner les interactions dont il anticipe que l'énoncé réussira.

Le modèle expérientiel de la Figure 1e simplifie le modèle interactionnel en supprimant les notions d'*expérimentation* et de *résultat*, en remplaçant le terme *interaction* par le terme *expérience*, et en renommant E l'ensemble $X \times R$. Le développeur de l'algorithme peut penser aux données de sortie et d'entrée comme à des expériences respectivement voulues et vécues par l'agent. Le terme *expérience* est à comprendre dans le sens anglais de « *to experience something* ». Comme le modèle interactionnel, le modèle expérientiel commence par le fait que l'algorithme sélectionne une *expérience intentée* i dans l'ensemble E des expériences offertes par le couplage agent/environnement. En retour, l'algorithme reçoit une expérience énoncée $e \in E$. L'expérience intentée peut changer l'état du monde. Si $e = i$ alors la tentative d'énoncé de i a réussi, sinon elle a échoué. L'algorithme est incarné et actif : ses données d'entrée (les expériences énoncées) ne constituent pas une représentation de l'état du monde. Les expériences encapsulent à la fois un mouvement et un signal sensoriel ; elles peuvent donc représenter des schèmes sensorimoteurs qui sont les briques de base de la théorie de l'apprentissage constructiviste de Piaget (1951). Notons que l'expression *schème sensorimoteur* pourrait suggérer que le *senseur* précéderait le *moteur*. Ici, nous l'utilisons pour désigner une expérience d'interaction dont les aspects *moteur* et *senseur* sont simultanés.

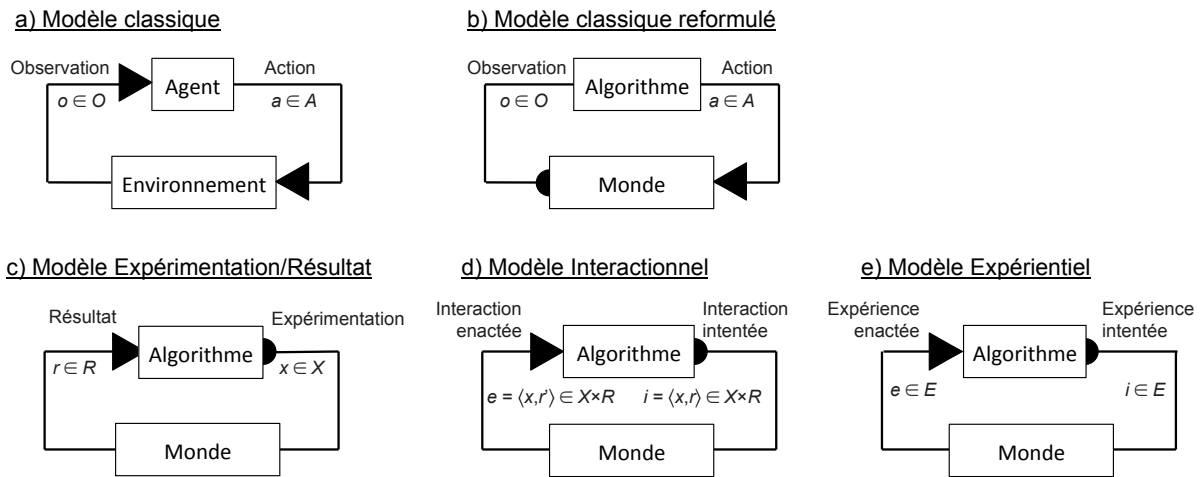


Figure 1 : Du modèle classique au modèle expérientiel. a) Modèle classique. b) Modèle classique reformulé : le début conceptuel du cycle est mis en évidence par le point noir. c) Modèle expérimentation/résultat : le cycle commence par le fait que l’algorithme sélectionne une expérimentation et finit par le fait qu’il reçoit un résultat. d) Modèle interactionnel : l’algorithme sélectionne une *interaction intentée* i composée d’une expérimentation x et d’un résultat anticipé r . En retour, il reçoit une *interaction éncatée* e composée de l’expérimentation x et d’un résultat r . e) Modèle expérientiel basé sur un seul type de primitives, les *expériences*, que le développeur de l’algorithme peut considérer comme des schèmes sensorimoteurs ou des expériences phénoménologiques faites par l’agent.

Le modèle expérientiel présenté en figure 1c traduit une inversion radicale du point de vue porté sur un agent cognitif par rapport au modèle classique des figures 1a et 1b (Georgeon et Aha, 2013). Le modèle expérientiel invite le concepteur de l’algorithme à raisonner initialement sur le flux d’expérience phénoménologique de l’agent au lieu de présupposer le couplage agent / environnement. Nous attendons que nos algorithmes interprètent les régularités constatées dans leur flux d’expérience comme résultantes de l’interaction entre l’agent qu’ils contrôlent et un environnement persistant qu’ils apprennent progressivement à observer et sur lequel ils apprennent progressivement à agir. Cette approche rejoint la théorie des *actions intentionnelles* d’Engel et al. (2013). Comme les actions intentionnelles, les expériences intentées incorporent la prédiction ou l’anticipation d’un résultat⁵. Par le fait qu’ils manipulent des schèmes sensorimoteurs, nos algorithmes se rapprochent également d’autres algorithmes qui suivent une approche sensorimotrice de l’intelligence artificielle (e.g., Drescher, 1991 ; Brooks, 1991 ; Pierce et Kuipers, 1997). Cependant, nos algorithmes se différencient de ceux-ci par le fait qu’ils effectuent, en un sens, un processus inverse. En effet, les algorithmes de ces auteurs utilisent les observations et les actions comme primitives pour construire les schèmes sensorimoteurs sous la forme de couples \langle observation, action \rangle , alors que nos algorithmes utilisent des schèmes sensorimoteurs primitifs pour construire des actions intentionnelles et des catégories d’observations qui sont des structures secondaires (Garnier et al., 2013).

Si nous augmentons la complexité des données d’entrée et de sortie, les algorithmes implémentés selon le modèle expérientiel souffrent tout autant d’une baisse de performance que

⁵ “Intentional actions [...] involve prediction or anticipation of an intended outcome” (Engel et al., 2013, p. 203).

ceux implémentés avec le modèle classique. Cependant, notre objectif n'est pas de traiter des données d'entrée complexes mais de générer des comportements de plus en plus intelligents en gardant la complexité des données d'entrée limitée. Nos algorithmes ne sont pas soumis à la complexité intrinsèque d'un problème modélisé a priori puisqu'ils ne cherchent pas à atteindre un état solution. Sur ces points, notre approche est en phase avec, par exemple, le point de vue constructiviste radical de Riegler (2007)⁶. Au contraire, le modèle classique a été critiqué au motif qu'il nécessitait des observations dont la complexité soit proportionnelle à la complexité du monde afin de limiter un effet de *perceptual aliasing* (Whitehead et Ballard, 1991). Le *perceptual aliasing* est lié au fait que les observations sont traitées comme des percepts qui représentent le monde, ce que le modèle expérientiel évite justement de faire. Le fait de ne pas augmenter inutilement la complexité des expériences ne nous empêche cependant pas de rester attentifs à la performance de nos algorithmes, comme nous le montrons en fin de section 3.

3 Algorithme d'abstraction d'expérience

Nous avons développé un algorithme qui apprend des régularités séquentielles hiérarchiques à partir du flux d'expériences énoncées par l'agent au cours de son existence. Nous l'avons initialement développé sur la base du modèle interactionnel (Georgeon et Ritter, 2012), puis simplifié sur la base du modèle expérientiel (Georgeon et al., 2013). Cet algorithme vise des objectifs similaires aux algorithmes de *hierarchical sequence learning* réalisés, par exemple, par Sutton et al. (1999). Cependant, il s'en différencie par le fait qu'il apprend à partir d'une séquence d'expériences qu'il contribue activement à générer, à la différence des algorithmes de ces auteurs qui apprennent à partir d'une séquence reçue passivement ou donnée a priori. Notre algorithme utilise sa position active pour tester des régularités hypothétiques, ce qui l'aide à faire face à la complexité du monde. Une fois qu'il a appris une séquence d'expériences qui reflète une régularité confirmée, il considère cette séquence comme une expérience abstraite qu'il pourra tenter d'énoncer à son tour. L'apprentissage est récursif et produit des expériences de plus en plus abstraites. La Figure 2 illustre ce processus.

⁶ « As we can no longer speak of information input and the vicissitude of stimuli, organisms are no longer exposed to information overload as a result of processing the entirely available information. They no longer need to devote their cognitive resources to filter out irrelevant information in order to retain useful knowledge. It becomes clear that even insect brains can accomplish navigational tasks and sophisticated cognitive deeds in nontrivial environments without falling prey to the frame problem. Therefore, cognitive research on perception should not focus on filtering mechanisms and data reduction. Information anxiety (Wurman 1990) and cognitive overload (Kirsh 2000) should not be considered a problem of the environment, as it is the case when talking, e.g., about the overload that comes with the information flood on the internet. Perception has to be explored in terms of the organism that performs the perceptive act » (Riegler, 2007, p109).

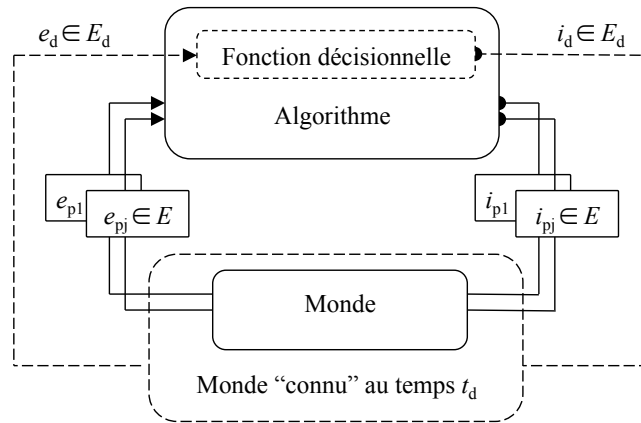


Figure 2 : Énaction récursive d'expériences. E : ensemble des expériences primitives prédéfinies. E_d : ensemble des expériences primitives ou abstraites au temps de décision t_d . Boucles primitives (traits pleins) : cycles d'énaction des expériences primitives. Boucle décisionnelle (trait pointillé) : cycle d'énaction d'une expérience abstraite. Un tour de la boucle décisionnelle génère plusieurs tours de la boucle primitive.

Les expériences permises par le couplage agent/environnement (appartenant à l'ensemble E en figures 1e, et 2) sont maintenant appelés *expériences primitives*, notées i_p ou e_p selon qu'elles sont intentées ou énaquées. Leur processus d'énaction (impliquant la commande d'actionneurs et la lecture de capteurs) est programmé par le développeur de l'agent. Au cours du temps, l'algorithme construit des expériences abstraites qui correspondent à des séquences $\langle e_{p1}, \dots, e_{pn} \rangle$ d'expériences primitives. Nous qualifions ces expériences *d'abstraites* car elle ne peuvent pas être énaquées directement par l'agent mais doivent être décomposées en expériences primitives pour être énaquées. L'ensemble de toutes les expériences connues par l'algorithme au temps décisionnel t_d (primitives ou abstraites) est noté E_d .

Tenter d'énacter une expérience abstraite i_d consiste à tenter d'énacter successivement les n expériences primitives i_{p1}, \dots, i_{pn} de i_d (boucles en trait plein de la Figure 2). Si l'énaction de la $j^{\text{ème}}$ expérience primitive i_{pj} échoue, alors l'énaction de i_d est interrompue. A la fin de la tentative d'énaction de i_d (complète ou interrompue), la fonction décisionnelle de l'algorithme reçoit l'expérience abstraite énaquée e_d construite à partir des j expériences primitives énaquées $\langle e_{p1}, \dots, e_{pj} \rangle, j \leq n$.

La boucle primitive est gérée par une sous-fonction de l'algorithme qui est indépendante de la fonction décisionnelle, comme si les expériences abstraites étaient énaquées machinalement sans nécessiter l'attention de l'algorithme tant que leur énaquée réussit. De ce fait, l'apprentissage peut s'appliquer récursivement, indépendamment de la longueur des expériences énaquées. Pour la fonction décisionnelle de l'algorithme, e_d semble avoir été énaquée dans un monde abstrait « connu au temps t_d » (boucle en trait pointillé). Du fait que chaque agent exécutant cet algorithme apprend des expériences abstraites en fonction de sa propre histoire, il connaît progressivement le monde d'une façon qui lui est propre. Cette possibilité d'évolution de la boucle cognitive rejoint les principes exprimés par Froese et Ziemke (2009). Ces auteurs s'appuient sur la philosophie de Heidegger et sur la théorie de l'énaction pour argumenter que la possibilité d'évolution du *couplage structurel* entre un être cognitif et son environnement est une propriété importante des systèmes cognitifs. Nous proposons le modèle expérientiel comme une réponse informatique pour traduire cette attente théorique dans les algorithmes.

Ce modèle participe également à un effort de la recherche en intelligence artificielle pour concevoir des agents capables de se « programmer eux-mêmes » (e.g., Thórisson et al., 2013). Parce que les expériences abstraites peuvent être énoncées machinalement comme des séquences d'expériences primitives, elles constituent une sorte de code ré-exécutable appris. L'apprentissage par régularité d'expérience génère donc un effet d'auto-programmation de l'agent par *sédimentation d'habitudes* de bas en haut—une expression que nous empruntons à David Hume (1739) et Husserl (e.g., cité par De Loor et al., 2010). Nous considérons les propriétés *d'évolution du couplage cognitif* et *d'auto-programmation* comme deux propriétés liées entre elles qui permettent une forme *d'autonomie constitutive* de l'agent (Froese et Ziemke, 2009).

Techniquement, l'algorithme encode les expériences abstraites $e_d \in E_d$ de manière hiérarchique sous forme de couples composés d'une pré-expérience et d'une post-expérience : $e_d = \langle e_{pre}, e_{post} \rangle$ | $e_{pre}, e_{post} \in E_d$. La Figure 3 illustre l'apprentissage d'un niveau d'abstraction supérieur à partir d'un niveau d'abstraction donné.

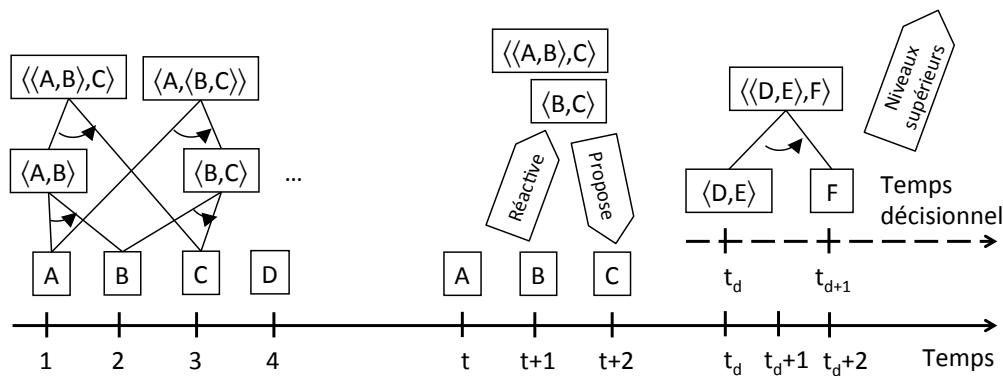


Figure 3 : Apprentissage hiérarchique d'expériences. Chaque pas de temps décisionnel (t_d, t_{d+1} sur la flèche de temps pointillée) représente une exécution de la fonction décisionnelle de l'algorithme (un tour de la boucle en pointillé de la **Figure 2**).

Dans l'exemple de la Figure 3, une expérience $A \in E_1$ est énoncée au temps 1, puis une expérience $B \in E_2$ au temps 2. L'algorithme enregistre une expérience abstraite de niveau supérieur correspondant à la séquence $\langle A,B \rangle$. Si l'expérience A est énoncée à nouveau au temps t , l'algorithme réactivera la séquence $\langle A,B \rangle$ car sa pré-expérience (A) est identique à la dernière expérience énoncée au temps t . Une fois activée, la séquence $\langle AB \rangle$ propose de tenter d'énoncer sa post-expérience (B) au temps $t+1$. Si cette tentative réussit souvent, alors l'algorithme peut considérer la séquence $\langle A,B \rangle$ comme une régularité d'interaction offerte par le couplage agent/environnement. Le fait que les tentatives réussissent souvent ou non est évalué par un mécanisme de pondération des séquences apprises détaillé par Georgeon et Ritter (2012).

Quand une expérience C est énoncée au temps 3, l'algorithme enregistre les séquences $\langle B,C \rangle$ ainsi que les séquences hiérarchiques $\langle \langle A,B \rangle, C \rangle$ et $\langle A, \langle B,C \rangle \rangle$. Quand une expérience D se produit au temps 4, l'algorithme pourrait enregistrer toutes les combinaisons hiérarchiques des expériences précédentes. Ceci montre que le nombre de séquences croît exponentiellement avec le temps, et qu'un mécanisme de limitation du nombre des séquences apprises est nécessaire pour limiter cette croissance.

Si l'expérience B est effectivement énoncée au temps $t+1$, l'algorithme réactive la séquence $\langle B,C \rangle$, ainsi que la séquence $\langle \langle A,B \rangle, C \rangle$ car sa pré-expérience s'identifie à la séquence énoncée de deuxième niveau $\langle A,B \rangle$ aux temps t et $t+1$. Ce mécanisme permet à l'algorithme de représenter son contexte courant par un ensemble de séquences d'expériences énoncées. En pratique, l'algorithme peut réactiver au même moment deux séquences qui proposent des expériences intentionnelles différentes. Un mécanisme de décision est donc nécessaire pour sélectionner la prochaine expérience intentée. Ce mécanisme de décision traduit une forme de préférence de l'agent. La motivation interactionnelle (introduite en Section 2) est implémentée par le fait que l'algorithme choisit préférentiellement les expériences qui ont la meilleure probabilité de réussir (estimées en fonction des expériences antérieures) et qui ont les valences les plus élevées, comme si l'agent trouvait agréable d'énoncer les expériences positives, et désagréable d'énoncer les expériences négatives. Les valences d'expérience offrent donc un moyen de définir des préférences innées sans contraindre l'agent à des objectifs prédéfinis. L'agent doit construire des connaissances sur le monde pour mieux réussir à énoncer les expériences qui ont une valence positive élevée et à éviter les expériences qui ont une valence négative.

Pour limiter l'explosion combinatoire du nombre d'expériences abstraites apprises, l'algorithme limite l'apprentissage en ne construisant des séquences de plus haut niveau qu'à partir de séquences intentionnellement énoncées. Par exemple, au temps 3, l'algorithme n'enregistre pas la séquence $\langle \langle A,B \rangle, C \rangle$ car la sous-séquence $\langle A,B \rangle$ n'a pas été décidée intentionnellement en tant que séquence entière. En revanche, si, au temps de décision t_d , l'agent décide d'énoncer la séquence $\langle DE \rangle$ (au cours des temps t_d et t_d+1), et si, au temps décisionnel t_{d+1} , l'agent énonce l'expérience F, alors l'algorithme enregistre la séquence $\langle \langle D,E \rangle, F \rangle$ car elle est basée sur deux sous-séquences énoncées intentionnellement. Ce mécanisme, détaillé plus précisément par Georgeon et Ritter (2012), assure qu'un niveau de régularités soit confirmé et utile avant d'apprendre des régularités de plus haut niveau.

4 Expérimentations

Cette section rapporte deux expérimentations qui illustrent les comportements générés par des agents contrôlés par l'algorithme présenté en Section 3. Noter que le même algorithme peut être utilisé pour contrôler des agents dont les expériences sont de natures totalement différentes puisque l'algorithme n'exploite pas de présupposés sur la signification des expériences.

4.1 Expérimentation 1 : robot e-puck

La **Figure 4** présente le dispositif expérimental. Pour plus de détails, nous renvoyons le lecteur à l'article de Georgeon et al., (2013), ainsi qu'à une vidéo de démonstration et à une expérimentation interactive similaire disponibles en ligne⁷. Une analyse plus complète des comportements générés par une expérimentation plus sophistiquée en environnement simulé a été effectuée par Georgeon et Marshall (2013).

⁷ <http://youtu.be/t1R05S4mBEY> <https://youtu.be/LVZ0cPpmSu8>



Figure 4 : Gauche : dispositif expérimental, le robot e-puck (Mondada et al., 2009) dans son environnement. Droite : robot e-puck face à un mur (vert) et cônes de détection de murs à gauche, devant, et à droite (zones grises dégradées).

Dans cette expérimentation, le couplage robot/environnement offre les dix expériences listées dans le tableau 1.

Tableau 1 : Expériences à la disposition de l'algorithme. Le seuil de détection de mur est réglé à 5cm environ. Le seuil de détection de collision est réglé à 0,5cm environ.

Expérience	Implémentation	Valence
Sentir un mur devant	Activer la LED infrarouge frontale et mesurer une lumière réfléchie supérieure au seuil de détection de mur.	-1
Sentir un espace vide devant	Activer la LED infrarouge frontale et mesurer une lumière réfléchie inférieure au seuil de détection de mur.	-1
Sentir un mur à gauche	Activer la LED infrarouge gauche et mesurer une lumière réfléchie supérieure au seuil de détection de mur.	-1
Sentir un espace vide à gauche	Activer la LED infrarouge gauche et mesurer une lumière réfléchie inférieure au seuil de détection de mur.	-1
Sentir un mur à droite	Activer la LED infrarouge droite et mesurer une lumière réfléchie supérieure au seuil de détection de mur.	-1
Sentir un espace vide à droite	Activer la LED infrarouge droite et mesurer une lumière réfléchie inférieure au seuil de détection de mur.	-1
Avancer	Activer les deux roues et activer la LED frontale pendant le temps de parcourir environ 5 cm, et la lumière réfléchie ne dépasse jamais le seuil de collision de mur.	5
Cogner un mur	Activer les deux roues et activer la LED frontale. La lumière réfléchie dépasse le seuil de collision avant d'avoir parcouru 5cm, provoquant la fin anticipée du déplacement.	-10
Tourner à gauche	Activer la roue gauche en arrière et la roue droite en avant pendant le temps d'avoir tourné sur place d'environ 90° à gauche.	-4
Tourner à droite	Activer la roue gauche en avant et la roue droite en arrière pendant le temps d'avoir tourné sur place d'environ 90° à droite.	-4

Lorsque l'algorithme tente une expérience, non seulement il ignore quelle diode infrarouge ou moteur il actionne, mais aussi le fait même qu'il s'agisse d'une diode ou d'un moteur. Il ignore également l'existence de murs, ainsi que le fait qu'il contrôle un robot qui se déplace sur une surface plane. L'algorithme n'a pas d'autre moyen de connaître le robot et son environnement que par les régularités d'expériences énoncées. Des valences numériques sont associées aux

expériences : *avancer* (+5), *collision* (-10), *tourner* (-4), *sentir* (-1). Les expériences *sentir* et *tourner* ont des valences légèrement négatives pour représenter l'effort de les énoncer.

Au début, l'algorithme doit « babiller » pour apprendre à donner du sens aux possibilités d'expérience offertes par le couplage robot/environnement. Une analyse du comportement observable du robot est rapportée en Figure 5.



Figure 5 : 100 premières expériences énoncées par le robot. Bande 1 : expériences primitives énoncées au cours du temps : *avancer* (triangle blanc), *collision* (triangle rouge), *tourner* (demi-cercles), *sentir vide* (carré blanc), *sentir mur* (carré vert). Les positions des carrés haut/milieu/bas représentent les directions gauche/devant/droite. Bande 2 : valence des expériences énoncées représentée en bar-graphe (vert positive, rouge négative). Bande 3 : Niveau hiérarchique des expériences intentionnellement énoncées (gris : énonciation primitive réussie ; noir : énonciation primitive échouée, interrompant l'énonciation de l'expérience abstraite). L'expérience abstraite *sentir_vide_devant* – *avancer* est énoncée pour la première fois aux pas 27-28. Aux pas 64, 74 et 92, cette expérience abstraite a été interrompue (segment noir de second niveau en bande 3) en raison du fait que l'expérience primitive tentée *sentir_vide_devant* a résulté en l'expérience *sentir_mur_devant*, ce qui dissuada le robot d'aller vers le mur et le fera plutôt tourner à gauche.

L'analyse de comportements présentée en **Figure 5** montre que le robot apprend à utiliser les expériences *sentir* comme une perception active pour éviter d'entrer en collision avec les murs (bien que les expériences *sentir* aient une valence légèrement négative). Cet apprentissage ne consiste pas seulement à ne pas avancer quand un mur est senti devant, mais, plus important, à activement utiliser les expériences *sentir* comme une perception pour évaluer la situation avant de prendre une décision. Dans cette approche, la perception émerge de l'activité du robot, par l'utilisation appropriée de schèmes sensorimoteurs. C'est une différence radicale avec les approches classiques dans lesquelles la perception est prédéfinie. Le fait que le robot prenne des dispositions pour pouvoir avancer sans se cogner fait penser à l'observateur qu'il aime avancer et qu'il n'aime pas se cogner. Dans une expérience plus complète en environnement simulé (e.g., Georgeon et Marshall, 2013), l'observateur peut voir que le robot apprend à associer les sensations latérales et les rotations latérales, c'est à dire qu'il découvre implicitement la structure à deux dimensions de son environnement. Cette découverte de la structure spatiale du monde à partir de régularités d'expériences sensorimotrices rejoint, par exemple, les travaux de Philipona et al. (2004).

Il est à noter que les comportements rapportés en **Figure 5** reposent sur le fait que les seuils de détection des capteurs utilisés par les expériences *sentir* sont réglés pour correspondre approximativement à la distance parcourue par les expériences *avancer* (5 cm). Si ces distances sont trop différentes, le robot apprend que les expériences *sentir* sont inutiles ; dans ce cas, il apprend à avancer sans utiliser les expériences *sentir* et n'évite pas les collisions. Si nous réduisons le « coût » de tourner (e.g., en donnant aux expériences *tourner* la valence -1), on

observe que le robot finit par apprendre à éviter les collisions la plupart du temps en tournant au centre de la boîte en alternant *avancer* et *tourner*.

4.2 Expérimentation 2 : système sensoriel distal rudimentaire

La **Figure 6** présente le dispositif expérimental. Nous renvoyons le lecteur à l'article de Georgeon et al. (2011) pour plus de détails. Une vidéo de démonstration est en ligne, ainsi que d'autres vidéos dans lesquelles l'agent se déplace dans un espace continu (au lieu d'une grille de cellules) et poursuit des cibles mobiles⁸.

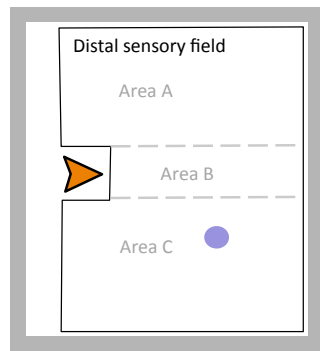


Figure 6 : Dispositif expérimental. La pointe orange représente l'agent placé dans une grille de 6 x 7 cellules. La zone délimitée par un trait plein représente son champ sensoriel couvrant 180°, capable de distinguer les déplacements relatifs de l'objet cible (point bleu) dans les zones A, B ou C.

L'agent dispose d'un système sensoriel rudimentaire (simulant une forme de vision ou d'olfaction) qui détecte les déplacements relatifs d'un objet cible immobile quand l'agent bouge. Ce système sensoriel renvoie une information de direction sommaire : à droite, en face, ou à gauche. Lorsque l'agent arrive sur une cellule contenant la cible, elle est retirée de la grille, comme si l'agent la mangeait. L'expérimentateur peut insérer de nouvelles cibles en cliquant sur la grille. Les expériences offertes par le couplage agent/environnement sont listées dans le Tableau 2.

Tableau 2 : Expériences à la disposition de l'algorithme. Les expériences impliquant la détection d'une cible existent en trois variantes selon la zone du déplacement de la cible (A : à gauche, B : en face, ou C : à droite). L'algorithme ignore qu'il s'agit de variantes d'une même expérience. Il traite les 18 expériences de manière initialement indifférenciée (une expérience par ligne de la colonne "Variantes").

Événements	Variantes	Implémentation	Valence
Avancer sans détecter de cible	-	Avancer d'une case. Pas de cible dans le champ sensoriel	0
Collision de mur	-	Tenter d'avancer d'une case mais échouer à cause d'un mur.	-1
Avancer vers une cible	A B C	Avancer d'une case. Une cible apparaît ou grossit dans la zone A, B ou C.	1
Avancer avec sortie de la cible du champ	A C	Avancer d'une case. La cible disparaît du champ sensoriel depuis la zone A ou C. (La variante B n'existe	-1

⁸ <https://youtu.be/91kKzybt8XY> <https://youtu.be/vSUEoh-sjwU>

visuel		pas dans ce couplage agent/environnement.)	
Manger la cible	-	Avancer d'une case sur la cible. La cible est retirée de la grille.	1
Tourner sans voir de cible	Gauche Droite	L'agent tourne de 90° à gauche ou à droite.	0
Tourner vers une cible	Gauche A Gauche B Gauche C Droite A Droite B Droite C	L'agent tourne de 90° à droite ou à gauche et la cible apparaît dans la zone A, B ou C.	1
Tourner avec sortie de la cible du champ visuel	Gauche A Droite C	L'agent tourne de 90° et la cible disparaît de son champ visuel depuis la zone A (tourne à gauche) ou C (tourne à droite).	-1

Encore une fois, l'algorithme ignore la signification des expériences, ainsi que le fait qu'il contrôle un agent qui se déplace dans une grille à deux dimensions contenant des cibles qu'il peut détecter et des murs qu'il peut cogner. Les valences associées aux expériences sont définies par l'expérimentateur de telle sorte que l'agent semble aimer se rapprocher des cibles.

Comme pour l'expérimentation précédente, au début, l'algorithme sélectionne des expériences arbitrairement, puis il apprend progressivement des régularités qui conduisent à rapprocher l'agent d'une cible. Le temps nécessaire pour atteindre la première cible dépend de la position initiale de l'agent et de la cible mais est de l'ordre d'une cinquantaine d'expériences. Une fois que l'agent a appris à atteindre la première cible, l'expérimentateur introduit de nouvelles cibles en cliquant sur la grille, en attendant à chaque fois que l'agent ait « mangé » la cible précédente. On observe qu'à partir de la deuxième cible, l'agent reproduit le même schéma de comportement appris, qui peut être différent d'une expérimentation à l'autre, en fonction des conditions initiales. La **Figure 7** montre deux comportements typiques observés. Une fois qu'un agent a appris un de ces comportements, il le répète sur toutes les nouvelles cibles introduites par l'expérimentateur.



Figure 7 : Exemples de comportements appris par l'agent au cours de différentes exécutions de l'expérimentation. Gauche : comportement consistant à avancer en escalier jusqu'à s'aligner avec la cible. Droite : comportement consistant à avancer en ligne droite jusqu'à ce que la cible sorte du champ sensoriel, puis revenir en arrière d'une cellule pour s'aligner sur la cible. L'agent ne perçoit pas la distance de la cible, il apprend à la localiser par le fait qu'elle sorte de son champ sensoriel.

L'émergence des deux schémas de comportements décrits à la **Figure 7** montre que chaque instance d'agent apprend une stratégie à partir de son histoire individuelle, au lieu d'appliquer une stratégie pré-codée, comme c'est souvent le cas avec les techniques classiques de résolution de problème. Notons que les différences de stratégie ne résultent pas de choix aléatoires (l'algorithme ne fait pas appel à la fonction *random*), mais uniquement de petites différences

dans la position initiale de l'agent, produisant un effet d'imprédictibilité déterministe (e.g., Zwirn et Delahaye, 2013).

Cette façon d'encoder les motivations propres sous forme de valences associées aux expériences sensorimotrices offre une réponse aux desiderata exprimés par Sun (2004) de considérer les comportements comme premiers⁹. L'agent est attiré par les cibles avant même d'avoir mangé sa première cible.

Du point de vue de l'observateur, les comportements reflètent une individualité acquise à partir de l'expérience personnelle de l'agent. Les agents peuvent apprendre des comportements différents pour satisfaire des désirs identiques. Nous pensons que cette approche ouvre la voie à la conception d'agents qui pourront développer une personnalité individuelle plus complexe par un processus d'individuation (Simondon, 2007). Cette possibilité d'individuation ne provient pas du fait que l'agent échappe à ses désirs prédéfinis, mais du fait que ses désirs prédéfinis sous-déterminent ses comportements (*needful freedom*, Jonas, 1966). A l'avenir, nous souhaitons développer des algorithmes plus sophistiqués dans lesquels des données représentant la situation homéostatique de l'agent interviendraient dans les critères de sélection des expériences intentées. Par exemple, en situation d'équilibre homéostatique, l'algorithme sélectionnerait les expériences qui ont le moins été testées dans le contexte courant plutôt que celles qui ont la plus haute valence attendue. L'observateur d'un tel agent pourrait interpréter ces comportements par le fait que l'agent est plus joueur ou curieux lorsqu'il n'a pas faim.

5 Vers la construction de connaissances ontologiques

Les êtres cognitifs naturels (animaux) existent dans un environnement (le monde réel) qui peut leur offrir des possibilités d'expériences persistantes pendant un certain temps et en certains lieux. Par exemple, une proie peut offrir des possibilités d'expériences visuelles, auditives, tactiles, olfactives, ou gustatives. La possibilité persistante de faire ces expériences en une certaine région de l'espace peut être interprétée par le fait qu'il existe une proie dans cette région. Cette section examine comment concevoir des agents capables de réaliser cette interprétation. L'algorithme ne connaît pas la proie « en soi » mais la proie « en ce qu'elle offre à l'agent la possibilité d'énacter certaines expériences en un certain lieu ». Nous désignons les entités présentes dans l'environnement « en ce qu'elles sont expérimentées par l'algorithme » par le terme de *phénomènes*. Cet usage du mot phénomène correspond à son sens courant qui

⁹ « Comportement, according to Heidegger, [...] “precedes every possible mode of activity in general,” prior to explicit beliefs, prior to explicit knowledge, prior to explicit conceptual thinking, and even prior to explicit desire. Comportement is thus primary, in exactly this sense. The traditional mistake of representationalism lies in the fact that they treat explicit knowledge and its correlates as the most basic instead, and thus they turn the priority upside-down; and in so doing, “every act of directing oneself toward something receives [wrongly] the characteristics of knowing” (Heidegger, 1927) » (Sun, 2004, p. 361);

désigne la façon dont la chose se présente à la cognition d'un agent, par opposition à la chose « en soi » désignée par le terme *noumène*.

Dans les sections 3 et 4 nous avons présenté des algorithmes capables d'interagir avec un seul type de phénomène. L'agent était incapable de catégoriser les phénomènes présents dans son environnement et d'adapter ses comportements en fonction des catégories des phénomènes avec lesquelles il interagissait. Nous présentons maintenant des algorithmes dotés de la capacité d'apprendre et de reconnaître des catégories de phénomènes différentes. Ces algorithmes encodent chaque catégorie de phénomène apprise sous la forme d'un ensemble P d'expériences ($P \subset E_d$) qui regroupe les *expériences offertes* par les phénomènes de cette catégorie. Nous utilisons l'expression *expériences offertes* pour traduire le terme « *afforded* » venant de la théorie des affordances de Gibson (1977). L'algorithme utilise les ensembles d'expériences P comme des *représentations opératives* de phénomènes, c'est à dire des représentations capables de provoquer des comportements (e.g., Weill-Fassinà et al., 1993).

Pour que l'algorithme puissent encoder, mémoriser et exploiter les catégories de phénomènes (les ensembles P), nous le dotons de deux mémoires supplémentaires : une mémoire *proto-ontologique* et une *mémoire de travail*, qui viennent s'ajouter à sa mémoire séquentielle hiérarchique présentée en Section 3. La mémoire proto-ontologique mémorise les catégories de phénomènes connus par l'algorithme à un instant donné. Techniquement, c'est l'ensemble des ensembles P construits à l'instant t . Nous qualifions cette mémoire de *proto-ontologique* car elle contient les prémices d'une ontologie des phénomènes construite par l'algorithme, sans toutefois posséder les mécanismes d'inférence habituellement présents dans les ontologies informatiques.

La mémoire de travail mémorise les instances de phénomènes persistantes dans l'environnement à un instant donné. Techniquement, une instance de phénomène est représentée par un pointeur qui renvoie à sa catégorie (un ensemble P particulier) stockée en mémoire proto-ontologique. Dans l'expérimentation de la Section 5.1, la mémoire de travail n'a la capacité de mémoriser qu'une seule instance de phénomène présent à un instant donné. Dans l'expérimentation de la Section 5.2, la mémoire de travail consiste en une mémoire spatiale égocentrée qui peut mémoriser plusieurs instances de phénomènes qui coexistent en différents lieux de l'espace autour de l'agent.

5.1 Inférence de phénomènes à partir de régularités d'expérience

Dans cette étude (Georgeon et al., 2015), la mémoire de travail est encodée sous la forme d'un réseau de Pétri. Initialement, ce réseau ne contient aucun arc et un seul nœud nommé *état ignorant*. Lorsque le jeton est sur ce nœud, l'algorithme sélectionne les expériences sans faire d'hypothèse sur le phénomène avec lequel l'agent est en train d'interagir. Lorsqu'une nouvelle catégorie de phénomène est construite en mémoire proto-ontologique, un nouveau nœud correspondant à cette catégorie est ajouté au réseau. Lorsque le jeton est sur ce nœud, l'algorithme sélectionne les expériences intentées en faisant l'hypothèse que l'agent est en train d'interagir avec une instance de phénomène de cette catégorie. L'algorithme construit un arc vers un nœud représentant une catégorie lorsqu'il infère que l'agent commence à interagir avec un phénomène de cette catégorie. Au fur et à mesure que l'algorithme construit le réseau de

Pétri, il l'utilise pour évaluer les conséquences possibles des expériences qu'il pourrait tenter : l'expérience éactée qui résulterait d'une expérience intentée particulière, ainsi que le phénomène avec lequel l'agent interagirait à la suite de cette éaction. Lorsqu'une expérience est éactée, l'algorithme déplace le jeton le long de l'arc associé à cette expérience. Les catégories de phénomènes (mémoire proto-ontologique) et le réseau de Pétri (mémoire de travail) cessent d'évoluer quand ils permettent à l'algorithme de faire des anticipations correctes. La difficulté provient du fait que l'algorithme doit construire le contenu de ces deux mémoires en parallèle.

Dans cette expérimentation, le monde est composée d'une chaîne de 11 chiffres arbitraires initialisée à la valeur $C_0 = [1, 7, 3, 2, 9, 3, 5, 6, 7, 8, 9]$, et d'un entier p dans l'intervalle $[0, 9]$, initialisé à $p_0 = 0$, qui représente la position de l'agent sur les dix premiers chiffres de cette chaîne. $C_t(p)$ désigne le chiffre à la position p de l'agent au temps t . L'agent peut se déplacer dans la chaîne vers la droite. Quand l'agent arrive sur le dixième chiffre $C_t(9)$, un nouveau déplacement vers la droite le ramène sur le premier chiffre $C_{t+1}(0)$. Les expériences sont listées dans le Tableau 3.

Tableau 3 : Expériences dont dispose l'algorithme. $C_t(p)$ désigne le chiffre à la position p au temps t .

Expériences	Implémentation	Valence
Sentir supérieur	$C_t(p+1) \geq C_t(p)$	0
Sentir inférieur	$C_t(p+1) < C_t(p)$	0
Avancer vers supérieur	$C_t(p+1) \geq C_t(p)$; If $(p < 9)$ then $p \leftarrow p+1$ else $p \leftarrow 0$;	1
Avancer vers inférieur	$C_t(p+1) < C_t(p)$; If $(p < 9)$ then $p \leftarrow p+1$ else $p \leftarrow 0$;	-1
Permuter vers supérieur	$C_{t+1}(p+1) \leftarrow C_t(p) \mid C_{t+1}(p) \leftarrow C_t(p+1)$; $C_{t+1}(p+1) \geq C_{t+1}(p)$	0
Permuter vers inférieur	$C_{t+1}(p+1) \leftarrow C_t(p) \mid C_{t+1}(p) \leftarrow C_t(p+1)$; $C_{t+1}(p+1) < C_{t+1}(p)$	0

Rappelons que l'algorithme ignore les effets produits par les expériences, ainsi que le fait même qu'il contrôle un agent localisé à l'emplacement p dans une chaîne de chiffres. Les valences des expériences sont initialisées pour simuler un agent qui semblerait aimer avancer vers un chiffre supérieur ou égal à celui sur lequel il est à l'instant t , et ne pas aimer avancer vers un chiffre inférieur.

Ce couplage agent/environnement offre deux catégories de phénomènes qui sont intéressantes à découvrir pour l'algorithme—catégories que nous désignons arbitrairement par *montée* et *descente*. L'algorithme doit découvrir qu'il peut utiliser les expériences *sentir* pour observer l'instance de phénomène avec lequel l'agent interagit à l'instant t , et connaître sa catégorie. Il doit également découvrir qu'il peut utiliser les expériences *permuter* pour transformer une instance de phénomène d'une certaine catégorie en une instance de phénomène d'une autre catégorie. Ces phénomènes sont donc bien des « façons dont le monde se présente à l'algorithme ». La chaîne de chiffres, quant à elle, constitue « le monde en soi, inconnu de l'algorithme », qui pourrait donc être appelé, en termes philosophiques, le *monde nouménal* de cette simulation.

Nous avons implémenté un algorithme qui s'inspire de techniques de *process mining* (e.g., Van der Aalst, et al. 2003). Le *process mining* est une discipline qui vise à construire de manière non supervisée un modèle causal d'un système, à partir d'une trace d'activité générée par ce

système. La Figure 8 montre le réseau de Pétri construit par l’algorithme après environ 70 expériences. L’algorithme a appris les deux catégories de phénomène P1 et P2 que nous (observateurs) connaissons pour être les catégories *montée* et *descente* ainsi que les différentes expériences qui permettent à l’agent de changer sa situation (arcs du réseau de Pétri).

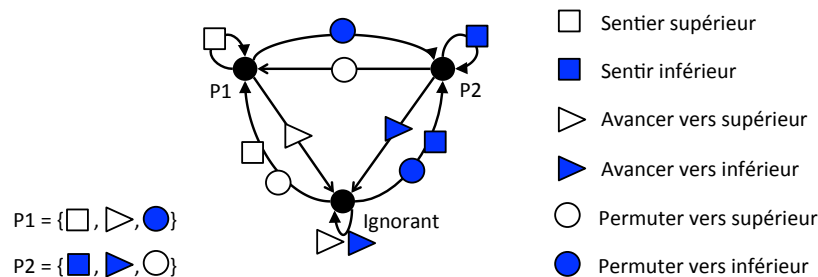


Figure 8 : La proto-ontologie (gauche) et le réseau de Pétri (centre) représentant le système de connaissance construit par l’algorithme (adapté de Georgeon et al., 2015). Les catégories de phénomènes *montée* et *descente* sont représentées par les ensembles P1 et P2 des expériences offertes par les phénomènes de chaque catégorie. Les nœuds du réseau de Pétri représentent des états de connaissances que l’algorithme peut avoir sur l’état du monde : agent interagissant avec un phénomène *montée* (gauche), avec un phénomène *descente* (droite), ou *ignorant* du phénomène courant (bas). Les arcs du graphe montrent les expériences qui font passer l’algorithme d’un état de connaissance à un autre, pour refléter les changements du monde provoqués par l’énaction des expériences attachées à ces arcs.

Une fois que le réseau de Pétri est construit, l’algorithme l’exploite dans le but d’énacter des expériences positives et d’éviter les expériences négatives. Du point de vue de l’observateur, l’agent semble comprendre progressivement comment il peut observer sa situation et comment il peut la transformer à son avantage. Quand il est dans l’état *ignorant*, il tente une expérience *sentir* pour connaître la catégorie du phénomène avec lequel il est en train d’interagir. Quand il semble se rendre compte qu’il est confronté à un phénomène de la catégorie *montée*, il énonce l’expérience *avancer vers supérieur*. Quand il est confronté à un phénomène de la catégorie *descente*, il énonce successivement les expériences *permuter vers supérieur* puis *avancer vers supérieur*.

Bien entendu, le développeur de l’algorithme pourrait facilement programmer directement ces comportements en s’appuyant sur sa connaissance des effets des expériences. Cependant, pour l’observateur, l’agent semblerait connaître la structure du monde dès le départ. Au contraire, dans notre expérience, l’observateur voit l’agent apprendre progressivement comme s’il parvenait à comprendre de lui-même la structure de son environnement. Nous constatons que l’observateur est davantage enclin à attribuer une intelligence à l’agent quand il est témoin de ce processus d’apprentissage (sinon, l’observateur pense que « l’agent est seulement un automate qui exécute un comportement préprogrammé »). L’article de Georgeon et al. (2015) présente une analyse approfondie du comportement d’apprentissage en analysant une trace d’activité.

Notons que cette expérimentation illustre aussi le fait que l’algorithme ne parvient pas à construire une connaissance phénoménale qui rende exactement compte de la structure du monde nouménal. En effet, dans la situation rare où le chiffre courant est égal au chiffre suivant ($C_t(p) = C_t(p+1)$), si l’agent énonce l’expérience *permuter* il reste confronté à un phénomène de la

catégorie *montée*, contrairement à ce que le réseau de Pétri laisse anticiper. Pour comprendre ce qui se passe dans cette situation, il faudrait que l'agent puisse construire la troisième catégorie de phénomènes « *horizontal* », ce qu'il est incapable de faire dans l'état actuel de l'algorithme. Une des difficultés provient du fait que les catégories de phénomènes ne seraient alors plus représentées par des ensembles d'expériences disjointes. Nous continuons à travailler sur des algorithmes capables de gérer cette difficulté.

5.2 Architecture cognitive

Cette section examine une extension du modèle expérientiel (Figure 1d) qui permet à l'algorithme d'apprendre des régularités d'expériences spatio-séquentielles effectuées par un agent évoluant dans un espace euclidien. La **Figure 9** présente cette extension appelée *modèle expérientiel spatial*. A chaque tour du cycle d'interaction, l'algorithme peut traiter plusieurs expériences qui sont énoncées simultanément, et peut suivre les déplacements spatiaux du corps de l'agent.

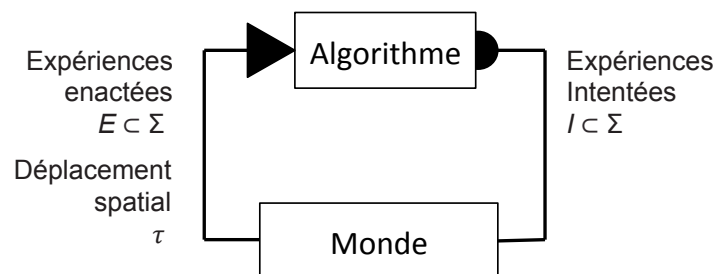


Figure 9 : Modèle expérientiel spatial. L'ensemble des *expériences localisées* que peut traiter l'algorithme est noté Σ . A chaque cycle d'interaction, l'algorithme tente un ensemble d'expériences localisées intentées $I \subset \Sigma$. En retour, il reçoit un ensemble d'expériences localisées énoncées $E \subset \Sigma$ et une information de déplacement spatial τ .

Le modèle expérientiel spatial présenté en **Figure 9** permet de modéliser des agents capables de localiser approximativement leurs expériences dans l'espace, ainsi que de percevoir leurs propres déplacements. Ce modèle fait l'hypothèse que l'agent dispose d'un dispositif sensoriel lui apportant ces informations. Cette hypothèse s'appuie sur des études qui montrent, par exemple, que les humains utilisent des capteurs kinesthésiques pour localiser leurs expériences tactiles, la convergence oculaire pour localiser leurs expériences visuelles, et le déphasage interaural pour localiser leurs expériences auditives. Ils utilisent leur système vestibulaire et le flux optique pour connaître leurs déplacements dans l'espace (e.g., Gross et Graziano, 1995). Ce modèle fait également l'hypothèse que l'agent dispose d'une mémoire spatiale prédéfinie. Cette mémoire est inspirée du colliculus supérieur du cerveau des mammifères. Dans d'autres travaux, nous étudions comment cette mémoire spatiale pourrait être construite à partir de régularités d'expériences sensorimotrices (Gay et al., à paraître).

Dans le modèle expérientiel spatial, l'ensemble Σ des expériences contient des *expériences localisées*. Une expérience localisée est une expérience similaire aux expériences du modèle expérientiel précédent, à laquelle sont rattachées des propriétés de localisation spatiale par rapport à un référentiel centré sur le corps de l'agent. Par exemple, appliqué à l'expérimentation de la Section 4.2, le modèle expérientiel spatial permet de traiter les trois types d'expérience

avancer vers une cible comme un seul type d'expérience auquel est rattaché l'information de localisation spatiale A, B, ou C.

L'algorithme reçoit également une information τ représentant le déplacement spatial effectué par l'agent pendant le cycle d'interaction. Si l'environnement possède une structure d'espace euclidien à n dimensions (e.g., un espace simulé à deux dimensions ou le monde réel à trois dimensions), et si le corps de l'agent est un bloc indéformable, alors le développeur de l'algorithme peut encoder τ comme une fonction de translation et de rotation euclidienne de dimension n représentant le mouvement de l'agent dans l'espace (Georgeon et al. 2013). Ce modèle ne suppose pas que les informations de localisation et de déplacement soient très précises. Nous n'avons pas encore étudié comment l'agent pourrait, de lui-même, calibrer et raffiner ce système sensoriel.

L'algorithme est maintenant organisé sous la forme de l'architecture cognitive représentée en **Figure 10**.

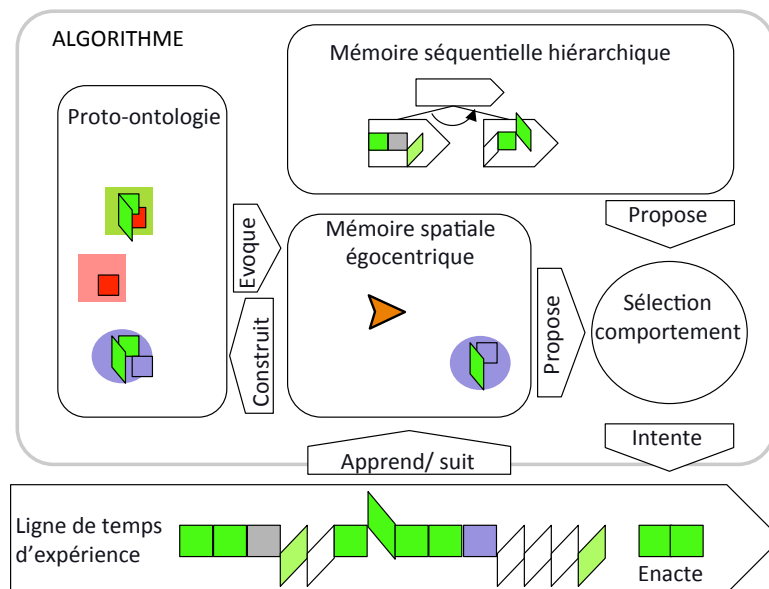


Figure 10 : Architecture cognitive pour l'apprentissage de régularités spatio-séquentielles. Flux d'expérience (bas) : expériences primitives (petits carrés et parallélogrammes) énoncées au cours du temps, similaire à la Figure 2. Mémoire séquentielle hiérarchique (haut) : le système d'abstraction d'expériences présenté en Section 3. Mémoire spatiale égocentrique (MSE, centre) : mémoire spatiale à court terme des expériences énoncées localisées dans l'espace relativement à l'agent et mise à jour en fonction des déplacements de l'agent (flèche « suit »). Proto-ontologie (gauche) : mémoire des catégories de phénomènes apprises par expérience. Les catégories de phénomènes (représentées par le rond bleu, le triangle rouge, et le triangle vert) sont définies par l'ensemble des expériences offertes par les phénomènes de cette catégorie. Les expériences en MSE peuvent activer des catégories de phénomènes mémorisées dans la proto-ontologie qui proposent les expériences que ces phénomènes offrent (flèche « évoque »). Les expériences sont proposées en fonction du contexte spatio-séquentiel (flèches « propose »), avant d'être choisies comme futures expériences à tenter d'énoncer (bas à droite).

L'architecture cognitive présentée en **Figure 10** utilise les informations spatiales attachées aux expériences énoncées pour les placer en mémoire spatiale égocentrique (MSE). Elle utilise les déplacements τ pour mettre à jour la MSE à chaque cycle d'interaction. Par exemple, si l'agent effectue une rotation vers la gauche, l'algorithme fait pivoter les expériences précédemment énoncées de l'angle opposé (vers la droite) en MSE pour refléter ce déplacement. La MSE est une

mémoire à court terme ; elle ne vise pas à construire une carte de l'environnement ; nous ne considérons pas que les informations spatiales disponibles soient suffisamment précises pour permettre cela. En revanche, elle permet à l'algorithme de détecter des recouvrements spatiaux des expériences énoncées sur le court terme. Lorsque des expériences sont énoncées dans des endroits proches, l'algorithme infère l'existence d'un phénomène qui offre ces expériences à cet endroit. Par exemple, en **Figure 10**, le parallélogramme vert et le carré bleu représentent des expériences énoncées à différents instants dans la même zone de l'espace, révélant la présence d'un phénomène qui offre ces expériences (rond bleu).

Une fois que l'agent a construit des catégories permettant de classer les phénomènes qu'il rencontre, et des séquences d'expériences adaptées aux catégories de phénomènes, il peut effectuer des simulations spatio-temporelles internes de séquences d'expérience adaptées à sa situation courante. La MSE permet des simulations de comportements dans un référentiel égocentré. A l'avenir, nous souhaitons implémenter d'autres structures mémorielles permettant des simulations spatio-temporelles dans un référentiel allocentré, par exemple inspirées de l'hippocampe. Cette approche s'inscrit dans le cadre de l'hypothèse formulée par certains chercheurs (e.g., Hesslow, 2002 ; Buzsáki, 2013) selon laquelle l'intelligence « de haut niveau » (raisonnement logique, langage) pourrait reposer sur des mécanismes de simulation spatio-temporelle de flux d'expérience déjà présents dans la cognition animale.

6 Conclusion

Nous avons présenté quatre modèles pour concevoir des algorithmes qui apprennent à partir de régularités d'expériences sensorimotrices : le modèle expérimentation/résultat, le modèle interactionnel, le modèle expérientiel, et le modèle expérientiel spatial. Ces modèles s'inspirent de théories cognitives qui remettent en question le cycle perception / cognition / action classique (e.g., Pfeifer et Scheier, 1994 ; Hurley, 1998), ou qui proposent une conception active de la perception (e.g., O'Regan et Noë, 2001 ; Findlay et Gilchrist, 2003). Dans ces modèles, l'absence de données ontologiques sur une réalité présupposée se traduit en évitant d'assimiler le processus de génération des données d'entrée à une fonction mathématique définie sur l'ensemble des états possibles du monde : le même état du monde peut produire différentes données d'entrée selon l'expérience intentée. Les algorithmes n'effectuent pas une recherche d'un état solution dans un ensemble d'états prédéfinis. Les agents ne sont pas évalués par leur capacité à atteindre un but prédéfini ni à maximiser une valeur numérique.

Nos objectifs sont de développer des agents qui génèrent des comportements susceptibles d'être jugés intelligents par des observateurs. Ces objectifs rejoignent ceux d'autres auteurs qui travaillent sur la *motivation intrinsèque* d'agents informatiques (e.g., Steels, 2004 ; Oudeyer et al., 2007) et l'apprentissage développemental (e.g., Lungarella et al., 2003). Pour progresser dans ce domaine de recherche, il convient de s'accorder sur des critères partagés par la communauté scientifique pour évaluer l'intelligence d'un comportement. En nous inspirant de méthodes utilisées en éthologie pour évaluer l'intelligence animale (e.g., Martin et Bateson, 1993), nous suggérons de produire des démonstrations de comportements intelligents basés sur les traces d'activité de l'agent. Une trace d'activité est un flux de données représentant l'activité de l'agent

(e.g., **Figure 5** et **Figure 7**) qui permet à un observateur humain de juger du niveau d'intelligence des comportements. Nous gardons l'idée proposée par Turing (1950) d'une évaluation basée sur le jugement d'observateurs humains, pour proposer ce que nous pourrions appeler un *test de Turing des comportements*. Un agent artificiel réussirait un *test de Turing des comportements* si un groupe d'observateurs (pouvant éventuellement interagir avec l'agent) s'avérait incapable de dire si les traces ont été générées par l'agent ou par un animal. En imaginant ce genre de tests intermédiaires, nous espérons favoriser une recherche incrémentale qui passerait par l'implémentation de différents niveaux d'intelligence sensorimotrice pour progresser vers l'implémentation d'une intelligence artificielle capable de manipuler un langage qui serait ancré dans l'expérience de l'agent.

Ces travaux soulèvent de nombreuses questions qui restent à résoudre pour progresser dans la conception d'agents qui exhibent des comportements de plus en plus intelligents. Ces questions incluent : comment catégoriser des phénomènes qui offrent certaines expériences en commun mais diffèrent par d'autres, comment l'agent peut-il réviser ses catégories de phénomènes, comment permettre à l'agent de calibrer et affiner la localisation spatiale de ses expériences, comment apprendre, encoder, et simuler des régularités spatio-temporelles complexes, comment étendre l'effet d'apprentissage récursif (présenté en **Figure 2**) au modèle expérientiel spatial (présenté en **Figure 9**) afin de permettre l'évolution du couplage cognitif dans le cadre du modèle expérientiel spatial.

7 Références

- Anderson M. (2003). Embodied cognition: A field guide. *Artificial Intelligence*, V, 149, 91–130.
- Brooks R. A. (1991). New Approaches to Robotics. *Science*, V, 253, 1227–1232.
- Buzsáki G. (2013). Time, space and memory. *Nature*, V, 497, 568-569.
- De Loor P., Manac'h K., & Tisseau J. (2010). Enaction-Based Artificial Intelligence: Toward Co-evolution with Humans in the Loop. *Minds and Machine*, V, 19, 319–343.
- Dreyfus H. (2007). Why Heideggerian AI failed and how fixing it would require making it more Heideggerian. *Artificial intelligence*, V, 171, 1137–1160.
- Drescher G. L. (1991). *Made-up minds, a constructivist approach to artificial intelligence*. Cambridge, MA: MIT Press.
- Engel A. K., Maye M., Kurthen M., & König P. (2013). Where's the action? the pragmatic turn in cognitive science. *Trends in Cognitive Sciences*, V, 17, 202–209.
- Froese T. & Ziemke T. (2009). Enactive artificial intelligence: Investigating the systemic organization of life and mind. *Artificial Intelligence*, V, 173(3-4), 466–500.
- Findlay J., & Gilchrist I. (2003). *Active Vision: The Psychology of Looking and Seeing*. USA: Oxford University Press.
- Gay S., Mille A., Georgeon O., Dutech A. (à paraître). Autonomous construction and exploitation of a spatial memory by a self-motivated agent. *Cognitive Systems Research*.
- Garnier J., Georgeon O., & Cordier A. (2013). Inferring actions and observations from interactions. *Proceedings of the Goal Driven Autonomy Workshop (GDA) at Advanced Cognitive Systems (ACS2013)*, Baltimore (pp. 26–35).
- Georgeon O., Marshall J., & Gay S. (2012). Interactional motivation in artificial systems: between extrinsic and intrinsic motivation. *Proceedings of the Second International Conference on*

- Development and Learning and on Epigenetic Robotics (ICDL-EPIROB 2012), San Diego (pp. 1-2).
- Georgeon O. & Aha D. (2013). The Radical Interactionism Conceptual Commitment. *Journal of Artificial General Intelligence*, V, 4(2), 31–36.
- Georgeon O. & Ritter F. (2012). An intrinsically-motivated schema mechanism to model and simulate emergent cognition. *Cognitive Systems Research*, V, 15-16, 73–92.
- Georgeon O. & Marshall J. (2013). Demonstrating sensemaking emergence in artificial agents: A method and an example. *International Journal of Machine Consciousness*, V, 5(2), 131–144.
- Georgeon O., Marshall J., & Manzotti R. (2013). ECA: An enactivist cognitive architecture based on sensorimotor modeling. *Biologically Inspired Cognitive Architectures*, V, 6, 46–57.
- Georgeon O., Wolf C., & Gay S. (2013). An Enactive Approach to Autonomous Agent and Robot Learning. *Proceedings of the Third Joint International Conference on Development and Learning and on Epigenetic Robotics*, Osaka (pp. 1–6).
- Georgeon O. & Cordier A. (2014). Inverting the interaction cycle to model embodied agents. *Procedia Computer Science*, V, 41, 243–248. *Proceedings of the the Fifth international conference on Biologically Inspired Cognitive Architecture*, Boston, MA.
- Georgeon O., Bernard F., & Cordier A. (2015). Constructing Phenomenal Knowledge in an Unknown Noumenal Reality. *Procedia Computer Science*, V, 71, 11–16. *Proceedings of the Sixth international conference on Biologically Inspired Cognitive Architectures*, Lyon, France.
- Gibson J. (1977). The theory of affordances. In R. E. Shaw & J. Bransford (éds.), *Perceiving, acting, and knowing*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Gross C. & Graziano M. (1995). Multiple representations of space in the brain. *The Neuroscientist*, V, 1(1), 43–50.
- Hesslow G. (2002). Conscious thought as simulation of behavior and perception. *Trends in Cognitive Sciences*, V, 6(6), 242–247.
- Hume D. (1739). *A treatise of human nature*. Oxford University Press.
- Hurley S. (1998). *Consciousness in action*. Cambridge, MA: Harvard University Press.
- Jonas H. (1966). *The Phenomenon of Life: Toward a Philosophical Biology*. Evanston, Illinois: Northwestern University Press.
- Kant (1781). *Critique of Pure Reason (Kritik der reinen Vernunft)*.
- Lakoff G., Johnson M. (1999). *Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Thought*. New York: Basic Books.
- Lungarella M., Metta G., Pfeifer R., & Sandini G. (2003). Developmental robotics: A survey. *Connection Science*, V, 15(4), 151–190.
- Martin P. & Bateson P. (1993). *Measuring behavior, An introductory guide*. Cambridge University Press.
- Mondada F., Bonani M., Raemy X., Pugh J., Cianci C., Klaptocz A., Magnenat S., Zufferey J.-C., Floreano D., Martinoli A. (2009). The e-puck, a robot designed for education in engineering. *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions* (pp. 59–65).
- Newell A. & Simon H. (1976). Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, V, 19(3), 113–126.
- O'Regan J. K., Noë A. (2001). A sensorimotor account of vision and visual consciousness. *Behavioral and Brain Sciences*, V, 24, 939–1031.
- Oudeyer P.-Y., Kaplan F., & Hafner V. (2007). Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, V, 11(2), 265–286.
- Pfeifer R. & Scheier, C. (1994). From perception to action: The right direction? In P. Gaussier and J.-D. Nicoud (éds.), *From Perception to Action* (pp. 1–11). IEEE Computer Society Press.

- Philipona D., O'Regan J.K., & Nadal J.-P. (2004). Perception of the structure of the physical world using unknown sensors and effectors. *Advances in Neural Information Processing Systems*, V, 16, 945–952.
- Piaget J. (1951). *The psychology of intelligence*. London: Routledge and Kegan Paul.
- Pierce D. & Kuipers B. (1997). Map learning with uninterpreted sensors and effectors. *Artificial Intelligence*, V, 92, 169–227.
- Riegler A. (2007). The radical constructivist dynamics of cognition. In: B. Wallace (éd.) *The Mind, the Body and the World: Psychology After Cognitivism?* (pp. 91–115). Imprint: London.
- Russell S., & Norvig P. (2003). *Artificial Intelligence, A Modern Approach*. Prentice Hall.
- Simondon G. (2007). *L'individuation psychique et collective : à la lumière des notions de forme, information, potentiel et métastabilité*. Paris: Aubier.
- Steels L. (2004). The Autotelic Principle. In I. Fumiya, R. Pfeifer, L. Steels, & K. Kuniyoshi (éds), *Embodied Artificial Intelligence* (pp. 231- 242), Springer Verlag.
- Sun R. (2004). Desiderata for cognitive architectures. *Philosophical Psychology*, V, 17(3), 341–373.
- Sutton R., Precup D., & Singh S. (1999). Between MDPs and semi- MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, V, 112, 181–211.
- Thórisson K., Nivel E., Sanz R., Wang P. (2013). Approaches and Assumptions of Self-Programming in Achieving Artificial General Intelligence. *Journal of Artificial General Intelligence*, V, 3(3), 1–10.
- Turing A. (1950). Computing machinery and intelligence. *Mind*, V, 59(236), 433–460. Oxford University Press.
- Van der Aalst W., Van Dongen B., Herbst J., Maruster L., Schimm G., & Wei-jters A. (2003). Workflow mining: A survey of issues and approaches. *Data & Knowledge Engineering*, V, 47(2), 237–267.
- Varela F., Thompson E., & Rosch E. (1991). *The embodied mind: Cognitive science and human experience*. Cambridge: MIT Press.
- Weill-Fassina A., Rabardel P., & Dubois D. (1993) *Représentations pour l'action*. Toulouse : Octares.
- Whitehead S. D. & Ballard D. H. (1991). Learning to perceive and act by trial and error. *Machine Learning*, V, 7(1), 45–83.
- Zwirn H. & Delahaye J.-P. (2013) Unpredictability and computational irreducibility. In H. Zenil (éd.), *Irreducibility and Computational Equivalence: 10 Years After Wolfram's A New Kind of Science (Emergence, Complexity and Computation)* (pp. 273–295). Springer.