



HAL
open science

Approche décentralisée pour un apprentissage constructiviste en environnement continu : application à l'intelligence ambiante.

Sébastien Mazac, Frédéric Armetta, Salima Hassas

► To cite this version:

Sébastien Mazac, Frédéric Armetta, Salima Hassas. Approche décentralisée pour un apprentissage constructiviste en environnement continu : application à l'intelligence ambiante.. Journées Franco-phones sur les Systèmes Multi-Agents (JFSMA), Jun 2015, Rennes, France. hal-01146381

HAL Id: hal-01146381

<https://hal.science/hal-01146381>

Submitted on 8 Oct 2015

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.

Approche décentralisée pour un apprentissage constructiviste en environnement continu : application à l'intelligence ambiante.

S. Mazac^{a,b}
sebastien.mazac@ubiant.com

F. Armetta^a
frederic.armetta@univ-lyon1.fr

S. Hassas^a
salima.hassas@univ-lyon1.fr

^aUniversité Lyon 1, LIRIS, UMR5205, F-69622, France

^bubiant, France

Résumé

Le paradigme constructiviste de l'apprentissage en intelligence artificielle (IA) se développe largement à travers des concepts tels que l'IA incarnée, l'IA éactive ou la Robotique développementale. L'objectif commun à ces approches est de créer des agents autonomes car dotés d'une capacité d'adaptation à leur environnement et même d'apprentissage, à l'image des organismes biologiques. Un vaste champ d'application inclut tous les systèmes en interaction avec un environnement complexe, dont les objectifs sont variés et non prédéfinis. Suivant ce positionnement, nous nous intéressons ici au problème de l'intelligence ambiante. Nous proposons un modèle décentralisé d'apprentissage constructiviste pour un système d'AmI basé sur une architecture multi-agent. Avec ce travail nous abordons notamment le problème de l'amorçage de l'apprentissage sensorimoteur pour des environnements réels continus sans modélisation de l'environnement à priori.

Mots-clés : Apprentissage Développementale, Constructivisme, Intelligence ambiante, Intelligence artificielle, Apprentissage, Systèmes multi-agents

1 Introduction

Commençons par donner un exemple minimal permettant d'illustrer la problématique. Le paradigme de l'intelligence ambiante (AmI) vise à exploiter grâce à l'intelligence artificielle (IA) les possibilités offertes par le développement des environnements connectés, afin d'offrir aux habitants/usagers divers services (voir [Friedewald et al., 2005] et [Augusto, 2007]). Pour cela l'apprentissage est un élément crucial, en particulier la reconnaissance d'activités qui suppose l'apprentissage de motifs d'interactions variés comme présenté

dans [Aztiria et al., 2010]. Dans ce contexte, nous définissons un système d'apprentissage S comme constitué d'un ensemble de capteurs $\{c_1; c_2 \dots c_n\}$ et d'un ensemble d'actionneurs $\{a_1; a_2 \dots a_n\}$. S est considéré comme un *agent*¹ autonome qui doit découvrir et apprendre des relations entre ces différentes variables, par exemple un lien entre a_x , c_y et c_z . Par la suite l'agent peut alors utiliser cette connaissance, par exemple dans le cadre de la réalisation d'un objectif défini par l'utilisateur. Supposons que l'actionneur a_x soit une prise actionnable sur laquelle est branchée un appareil non identifié D , que c_y soit le capteur de la consommation électrique de cette prise et c_z un capteur de luminosité dans la pièce. Si D est une lampe, S doit déterminer qu'il peut agir sur la consommation et la luminosité en activant et désactivant la prise. Il en va de même avec la température si jamais on remplace D par un chauffage électrique. De telles régularités dans l'expérience de l'agent peuvent sembler triviales et l'apprentissage relativement aisé. C'est éventuellement le cas si le concepteur du système le dote d'une représentation à priori, telle que la caractérisation des événements et la façon d'appréhender leurs liens temporels. L'élaboration d'une telle représentation permet en effet de focaliser la tâche d'apprentissage sur un objectif précis et de restreindre l'espace de recherche en ayant préconstruit une partie de la solution, notamment en limitant les façons possibles de « percevoir » les données brutes. Cependant nous nous positionnons ici dans le cas d'un apprentissage plus complet et considérons le cas où l'agent doit construire seul sa propre représentation à partir des données brutes. Dans ce cas, l'agent doit réaliser un double apprentissage puisqu'il doit en parallèle : apprendre à percevoir efficacement, de façon à : apprendre des régularités ou motifs récurrents. Avant d'introduire plus en

1. Un agent intelligent en IA ([Russell and Norvig, 2009])

détail ce *problème d'amorçage*, et de présenter notre contribution, les paragraphes suivants décrivent succinctement les avancées des approches constructivistes et de la robotique développementale sur ces questions.

2 Le problème des environnements continus

Dans le cadre d'environnements réels tels que celui de l'AmI ou de la robotique, le système doit donc faire face à la complexité du flot continu de données brutes provenant du matériel. C'est un problème de *discrétisation* de l'expérience, qui doit être réalisée de manière adaptée à la relation système-environnement. Par exemple l'agent doit pouvoir précisément identifier un évènement dans le temps et l'*espace* (ensemble des domaines de valeur des variables), mais la tâche qui consiste à délimiter dans la trame de l'expérience ce qui peut être considéré comme un évènement pertinent est particulièrement difficile. Le cerveau humain est capable de focaliser son attention sur les détails significatifs et d'ignorer tout le reste. Cette propriété de la cognition n'est pas évidente à reproduire en IA et certains problèmes fondamentaux bien connus en dépendent (*perceptual aliasing, frame problem, etc.*). De façon plus générale ces problèmes sont donc liés à la difficulté de gérer la complexité des informations du monde réel sans passer par une modélisation simplificatrice, connue en IA comme le *paradoxe de Moravec*. Ce paradoxe pointe du doigt le fait plutôt surprenant que les *capacités sensori-motrices* basiques des organismes vivants sont plus difficiles à reproduire artificiellement que les capacités cognitives de plus haut niveau tel que le raisonnement, comme le souligne [Brooks, 1991]. Dans le cadre de l'apprentissage constructiviste appliqué à des systèmes réels, ces problèmes sont bien sûr cruciaux.

3 Apprentissage constructiviste et robotique développementale

En psychologie, la théorie constructiviste de l'apprentissage développée en particulier par [Piaget, 1954] propose une vision de l'apprentissage comme un processus actif de construction et d'adaptation d'une représentation par le sujet, en interaction avec son environnement, plutôt que comme la simple acquisition d'un modèle figé du monde. A la suite de [Drescher, 1991] qui s'inspira directement

de la théorie de Piaget pour proposer un modèle d'apprentissage artificiel constructiviste, certains chercheurs se sont intéressés à l'apprentissage de schémas sensorimoteurs (voir : [Guerin, 2011]). Dans ce modèle la brique élémentaire de représentation est appelée un schéma en référence au concept de *schème* sensorimoteur de la théorie de Piaget. Un schéma est un triplet : contexte C , action A , résultat R qui signifie pour l'agent : « si j'exécute A en observant C , alors j'observe R ». De nouveaux schémas sont appris de façon incrémentale alors que l'agent exécute aléatoirement des actions à la manière d'un enfant qui tâtonne. Le schéma est une structure intéressante car elle permet de relier une action de l'agent aux conséquences qu'il perçoit dans l'environnement, donc basé sur sa propre expérience. Afin de permettre l'abstraction, le mécanisme d'apprentissage de schémas propose également le concept d'*item synthétique*, afin de représenter une notion nouvelle à partir des besoins liés à l'expérience, comme présenté dans [Perroto et al., 2010]. Cependant, ces modèles sont intéressants d'un point de vue théorique mais difficilement applicables tels quels à des problèmes réels en environnement continu à cause de la complexité des algorithmes utilisés. C'est pourquoi la plupart des travaux dans ce domaine concernent des environnements simulés discrets. Le domaine de *robotique développementale* (voir [Lungarella et al., 2003]) nous intéresse donc particulièrement puisqu'il s'inscrit tout à fait dans cette approche, et doit gérer les environnements réels. De plus, habituellement le concepteur ne définit pas une tâche particulière mais le robot doit plutôt explorer librement son environnement. Il faut donc prévoir des mécanismes de *motivation* afin de susciter et guider l'exploration ([Oudeyer et al., 2007]). La capacité de prédire est une composante fondamentale du mécanisme d'apprentissage duquel émerge la cognition, comme énoncé par [Von Glasersfeld, 1984]². Les fonctions cognitives de plus haut niveau, telles que par exemple la planification, peuvent être considérées comme la prédiction basée sur l'expérience, d'une série d'évènements. C'est pour-

2.

"Quite generally, our knowledge is useful, relevant, viable, or however we want to call the positive end of the scale of evaluation, if it stands up to experience and enables us to make predictions and to bring about or avoid, as the case may be, certain phenomena (i.e., appearances, events, experiences). If knowledge does not serve that purpose, it becomes questionable, unreliable, useless, and is eventually devaluated as superstition."

quoi la notion d'évènement est très importante dans le cadre des environnements continus. Sans modélisation préconstruite, l'agent doit en effet être capable de déterminer ce qu'il est pertinent de considérer comme un évènement. Et le seul critère qui permet d'en juger pour l'agent est l'utilisation de cet évènement dans le cadre d'une prédiction.

4 Le problème d'amorçage

Il existe des pistes intéressantes pour rendre plus efficaces les modèles existants et permettre des applications à des problèmes réels aux environnements continus. Par exemple nous pouvons mentionner [Chaput et al., 2003] qui proposent d'utiliser des cartes auto-organisatrices (SOM) pour améliorer l'apprentissage de schémas. Poursuivant dans cette direction, [Provost et al., 2006] proposent un système qui utilise un processus de discrétisation basé sur les SOM couplé à un apprentissage par renforcement dans un problème de navigation de robot. Dans le même esprit, [Linaker and Jacobsson, 2001] conçoivent un système composé d'un mécanisme d'extraction d'évènement (système de classification) qui convertit les données brutes (multidimensionnelles et horodatées) en des séries d'évènements sur une plus grande échelle de temps, ainsi que d'un mécanisme d'apprentissage par renforcement. Le problème majeur d'une telle approche réside dans le fait que le processus de classification à bas niveau est unique et défini à priori. En effet, pour une même variable, les classes d'évènements générées peuvent être différentes en fonction des patterns dans lesquels cette variable est impliquée, et d'autre part une classification réussie peut devenir inadaptée plus tard, suite à un changement de matériel par exemple.

Une solution possible est proposée par [Mugan and Kuipers, 2007] qui présentent un mécanisme d'extraction d'évènements couplé à un système d'apprentissage de règles prédictives, similaires aux schémas. A partir de variables continues, les évènements sont spécifiés comme des transitions entre des états qui ne sont pas définis initialement. Des prédictions sont apprises pour associer ces évènements et une boucle de rétroaction permet de guider le processus en favorisant la création de nouveaux états, pour apprendre de plus précis et de plus nombreux évènements et règles. Ce dernier travail illustre la mise en place d'une dynamique entre le processus de discrétisation de l'expérience et l'apprentissage

de motifs sensori-moteurs, qui est au centre du problème d'amorçage. Un élément de représentation ou un motif peut être considéré comme pertinent s'il contribue à la construction de nouveaux motifs, et donc appartient à une représentation qui le justifie. En d'autres termes, l'agent doit apprendre à percevoir de manière adaptée pour pouvoir apprendre efficacement. Cette définition autoréférentielle, caractéristique de l'étude des systèmes vivants, illustre le *problème d'amorçage* énoncé par [Kuipers et al., 2006] qui réside dans l'apprentissage des motifs sensorimoteurs à partir des données brutes de capteurs inconnus (voir également [Guerin, 2011]). Pour les organismes biologiques, ce double apprentissage est probablement réalisée à la fois durant la phylogénèse et durant l'ontogénèse, mais il s'agit d'un seul et même problème dans le cas des systèmes artificiels. Dans la suite, nous proposons un modèle systémique, se focalisant sur les dynamiques entre les processus de traitement de l'information qui permettent de construire des motifs sensorimoteurs, et autorise de multiples implémentations de ces processus.

5 Présentation du modèle

5.1 Interactions Agent-Environnement

L'apprentissage doit donc reposer sur l'interaction agent-environnement, c'est-à-dire sur les signaux échangés à la frontière entre l'agent et son environnement. Dans notre modèle, cette frontière est modélisée par un ensemble de variables continues qui représentent l'ensemble des capteurs et des actionneurs : $V = \{v_1; v_2...v_n\}$. Tel que représenté sur la Figure 1-(a), ce flot continu de données brutes peut être interprété par l'agent comme une séquence d'évènements. Par la suite nous appelons expérience $E = \{e_1; e_2...e_n\}$ l'ensemble des évènements construits par l'agent, au sein de laquelle nous considérons qu'il existe des régularités. Une régularité est un motif récurrent que l'agent doit être capable d'isoler et d'identifier dans son espace de recherche. Du point de vue du système proposé, il n'y a pas de différence entre les variables représentant les capteurs et les actionneurs. Donc les motifs appris peuvent concerner indifféremment capteurs et actionneurs. Il faut cependant que des actions soient exécutées pour pouvoir apparaître dans des patterns sensori-moteurs. Dans le domaine de l'AmI, il est davantage contraignant qu'en robotique d'utiliser les interactions aléatoires

comme point de départ du développement. En effet un comportement par tâtonnement comme dans [Mugan and Kuipers, 2007] pourrait être trop long et ses conséquences trop gênantes pour les utilisateurs. Heureusement, nous pouvons utiliser à notre avantage une propriété des systèmes ambiants : les utilisateurs peuvent être à l'origine des actions possibles. D'ailleurs pour des raisons de sécurité et d'éthique, la conception d'un système ambiant devrait toujours faire en sorte que l'utilisateur garde le contrôle de son environnement. Donc nous considérons comme contrainte de conception que chaque action qui pourrait être décidée et réalisée par le système, doit aussi pouvoir être provoquée directement par l'utilisateur. D'autre part nous faisons l'hypothèse que l'agent est capable de percevoir une telle action produite par l'utilisateur, comme une forme de *proprioception*³, de sorte que si cette action est impliquée dans un motif, l'agent l'apprend et soit capable de la réaliser par lui-même plus tard. [Najjar and Reignier, 2013] font la même hypothèse et nomment cela « actions observées ». La phase initiale d'exploration aléatoire des systèmes constructivistes peut donc dans le cas des systèmes ambiants être remplacée par des actions provoquées par des utilisateurs, ce qui est intéressant car ainsi l'utilisateur est impliqué dès le début du processus d'apprentissage comme une forme de motivation pour l'agent.

5.2 Structure élémentaire de représentation

Le rôle de la structure élémentaire de représentation est d'exprimer le motif régulier que le système est capable d'apprendre au plus bas niveau. Plus généralement il s'agit d'exprimer une boucle élémentaire d'interaction entre l'agent et son environnement telle que celle exprimée par le *cercle fonctionnel* de [Von Uexküll, 1992]. Notons ici qu'une autre approche possible afin de s'affranchir du problème d'amorçage est de considérer que le schéma (ou toute autre élément représentationnel du même type) constitue une structure élémentaire indivisible intrinsèque au système. Par exemple [Georgeon and Aha, 2013] définissent une interaction sensorimotrice comme une primitive. Cela suppose néanmoins dans le cas de systèmes réels, d'implémenter ces primitives, et donc de connaître parfaitement les capacités

3. i.e. perception de ses propres actions. Dans le corps humain : sensibilité permettant de percevoir ses propres mouvements, la position du corps, etc.

sensori-motrices de l'agent. La même remarque s'applique à l'approche de [Brooks, 1991] avec l'architecture de subsomption pour la robotique où les comportements de bas niveau sont préalablement codés. Donc à l'image d'un schéma, une structure élémentaire de représentation est construite en définissant des éléments de représentation (ex. contexte, action, ...) et des liens logiques pour unir ces éléments (ex. "et", "pendant", "consécutif", etc.). Dans le cas des environnements réels continus il y a une infinité de possibilités pour définir ces structures à partir des données brutes. Le problème d'amorçage se pose donc dès lors que le concepteur laisse tout ou partie de ce travail de discrétisation et d'abstraction à la charge de l'agent. Pour cela, nous choisissons de définir un élément de représentation générique "*évènement*", que l'on peut distinguer en deux sous-catégories génériques. Le premier type d'évènement représente un moment ciblé de l'évolution d'une variable. Rappelons que comme une variable peut représenter aussi bien l'état d'un capteur que d'un actionneur, un évènement peut donc représenter une action. Le deuxième type d'évènement, appelé *association*, permet de décrire une relation entre deux ou plusieurs évènements. Les différentes implémentations possibles de ces éléments et des moyens de les produire constituent donc l'espace de recherche de la discrétisation de l'expérience continue en une série d'évènements. Par exemple, on peut définir un évènement comme étant une variation notable d'une variable, et une association comme étant une durée identifiable entre deux évènements. D'autre part on peut aussi spécifier qu'un évènement est, non pas une variation, mais un état stable et que l'association n'est pas une durée ciblée mais une inclusion, etc. Par combinaison de ces différents types d'évènements il est ainsi possible d'exprimer une régularité de type schéma entre autres. Nous présenterons plus loin les choix d'implémentations que nous utilisons pour réaliser ces structures.

5.3 L'activité du système multi-agent

Notre proposition est basée sur l'utilisation d'un système multi-agent⁴ (SMA) pour réaliser l'apprentissage sensori-moteur décrit précédemment. Le système multi-agent proposé se compose de trois populations d'agents distinctes qui exécutent des fonctions complémentaires et

4. Pour éviter toute confusion, le terme « agent » fera désormais référence à une entité du SMA et non au système global apprenant, que l'on désignera simplement comme « le système »

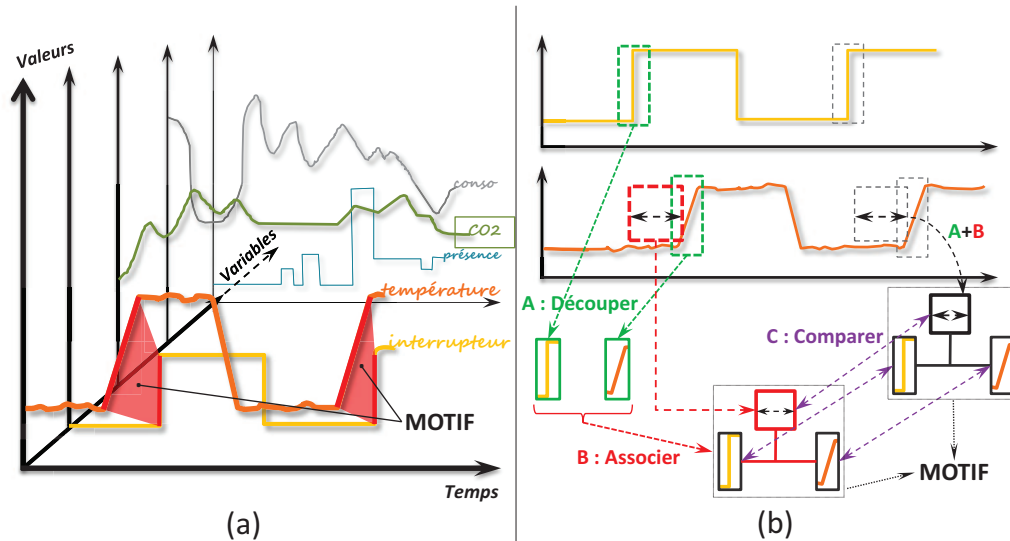


FIGURE 1 – (a) Expérience continue : ensemble de variables. (b) Les 3 types d'opérations. A : Découpage temporel (création d'un évènement). B : Trouver corrélation entre évènement (création d'une association). C : Comparer des éléments de représentation.

interagissent afin de construire la représentation. Ils sont guidés par des boucles de rétroaction provenant de l'évaluation de la construction réalisée. En effet, en lien avec la notion de structure élémentaire de représentation présentée précédemment, nous définissons trois opérations élémentaires qui interviennent pour l'amorçage et l'apprentissage de motifs sensorimoteurs à partir de l'expérience continue, tel qu'illustré sur la Figure 1-(b). Tout d'abord il faut découper l'expérience continue des variables en des moments finis que l'on peut considérer comme des évènements. Ensuite il faut pouvoir associer ces évènements pour créer des évènements plus complexes (évènements *associations*), et enfin il est nécessaire de pouvoir comparer deux éléments de représentation, par exemple pour reconnaître différentes instances d'un même évènement. Un agent implémente donc une fonction correspondant à l'une de ces trois catégories d'opération. Le comportement d'une fonction peut varier en fonction d'un *paramètre* ajustable par l'agent, constituant ainsi un espace de recherche supplémentaire. Une certaine fonction peut convenir pour apprendre une régularité de l'expérience de l'agent à un certain niveau, mais échouer pour un autre aspect de l'expérience. L'exploration conjointe par les agents des différentes implémentations possibles de ces opérations et de leurs espaces de recherche respectifs fournit des propositions de discrétisations et de représentations multiples qui constituent l'espace de recherche global. L'évaluation des solutions

oriente la recherche et renforce les zones intéressantes. La Figure 2 offre une vue synthétique sous forme de diagramme d'activité de ce processus de construction décentralisé.

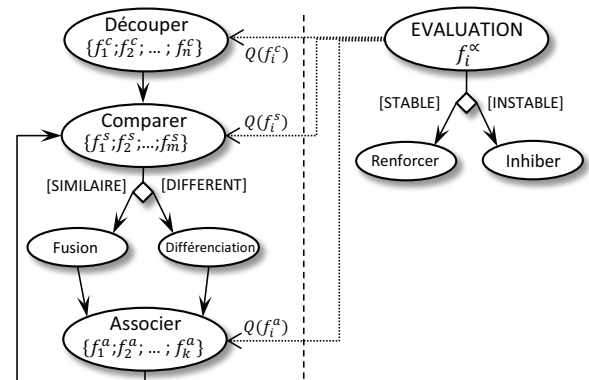


FIGURE 2 – Activité globale du système.

Nous pouvons formaliser ces 3 populations d'agents de la manière suivante :

- Nommons "*découper*" l'opération permettant de générer des évènements à partir des variables. L'ensemble des fonctions possibles de type "*découper*" est défini comme : $F^d = \{f_1^d, f_2^d, \dots, f_n^d\}$. Les différentes fonctions implémentant cette opération peuvent être défini-

nie par :

$$f_i^d : v_i \rightarrow \{e_1^d; e_2^d; \dots; e_n^d\}$$

$$V \rightarrow E$$

Ce type de fonction est donc implémenté par les agents "découper" (D) qui sont connectés à une variable. Par exemple, une implémentation possible est un agent qui utilise des fenêtres glissantes avec une durée paramétrable. L'exploration possible liée à cette fonction est donc la recherche d'une taille de fenêtre pertinente, adaptée à la variable et aux éventuels motifs concernés.

- Deuxièmement l'opération *Association* permet de « relier » les évènements existants. Une fonction association ($F^a = \{f_1^a; f_2^a; \dots; f_n^a\}$) peut être définie par :

$$f_i^a : e_x, e_y \rightarrow e_z$$

$$E \times E \rightarrow E$$

Par exemple une implémentation possible de cette fonction est un agent *Association* (A) qui, à partir d'un élément de référence e_1 , génère une représentation des durées entre les occurrences successives de e_1 et d'autres éléments e_i du système. L'espace de recherche de cet agent réside dans la manière dont l'agent sélectionne les durées des occurrences des éléments.

- Le troisième type d'opération, *Similarité* permet de « comparer », étape nécessaire à toute manipulation des éléments de représentation. Une telle fonction ($F^s = \{f_1^s; f_2^s; \dots; f_n^s\}$) de comparaison peut être définie par :

$$f_i^s : e_x, e_y \rightarrow s$$

$$E \times E \rightarrow [0, 1]$$

où e_x, e_y sont des éléments comparables car provenant d'une même source ($s = 1$ signifie totalement similaire, $s = 0$ totalement différent). L'implémentation de l'agent *Similarité* (S) va dépendre des structures de données utilisées pour les éléments de représentation. Dans cette version nous utilisons des histogrammes, donc une première implémentation d'un agent S est une fonction de comparaison d'histogrammes telle que l'intersection. Le paramétrage conditionne l'échantillonnage de l'histogramme, l'agent pouvant ainsi être plus ou moins précis pour évaluer la similarité de deux éléments.

Le traitement opéré par les agents S est de trier les instances d'évènements produits

par les deux autres types d'agents et d'identifier les classes d'évènements intéressantes. Pour cela, les agents forment des organisations qui sont toujours composées d'au moins un agent S, comme illustré sur la Figure 3 qui montre les formes d'organisations possibles sous la forme d'un diagramme Agent-Groupe-Rôle ([Ferber and Gutknecht, 1998]). Au sein d'une organisation, un agent producteur d'évènements (D ou A) reçoit des données sélectionnées à un niveau inférieur, et les traite pour créer un nouvel évènement. Ces nouveaux évènements sont classés et sélectionnés par l'agent S, et diffusés à leur tour dans le système. Comme illustré sur la Figure 4, les agents peuvent appartenir à différentes organisations, dans lesquelles ils jouent différents rôles, à différents niveaux.

Comme on peut le voir sur la Figure 2, ce mécanisme est un processus circulaire, amorcé par la création d'évènements à partir des variables. A chaque fois qu'un nouvel évènement est créé par un agent D, il est comparé avec les précédents. Il peut alors être fusionné avec un élément similaire (la classe d'évènement se voit ainsi renforcée), ou à défaut initier une nouvelle distinction, c'est-à-dire une nouvelle classe d'évènements possibles. Ces évènements peuvent ensuite être associés, comparés, puis les associations sont utilisées à leur tour comme des évènements et ainsi de suite. Ces opérations génériques et la dynamique de leur interaction avec l'évaluation d'une représentation prédictive permettent de définir un modèle de base pour le mécanisme d'extraction de motifs sensori-moteurs.

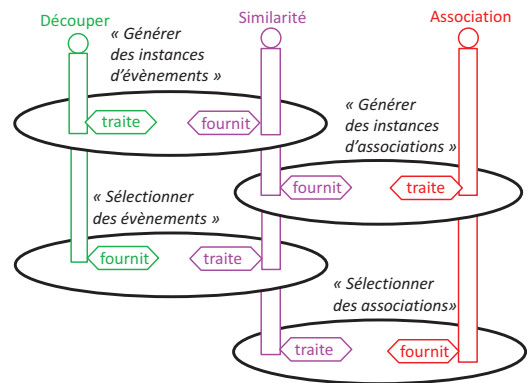


FIGURE 3 – Organisations possibles entre les types d'agents.

6 L'évaluation des organisations

L'espace de recherche du système est donc l'ensemble des fonctions possibles $F = F^D \cup F^A \cup$

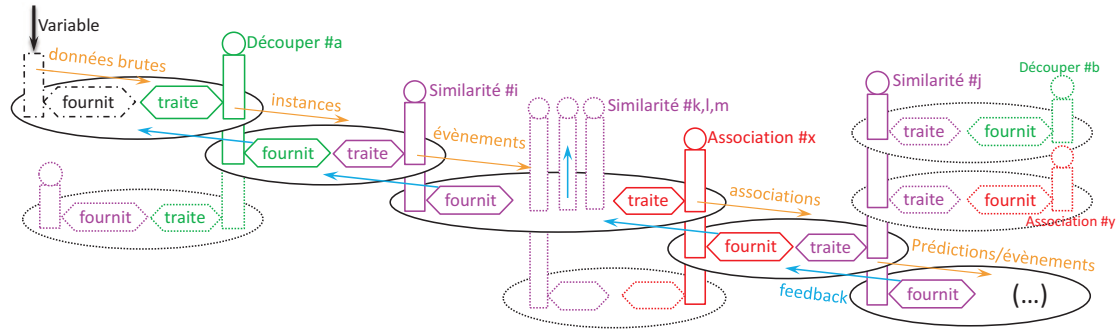


FIGURE 4 – Exemples d’interactions entre instances d’agents.

F^S . La pertinence d’une fonction peut être évaluée d’après sa participation à la construction de motifs, qui se fait nécessairement en collaboration avec d’autres types de fonctions. Nous pouvons représenter la rétroaction par une fonction d’évaluation $Q : f \in F \rightarrow [0, 1]$ qui s’applique aux structures générées, et influence les agents participants. Ainsi le but du système est de trouver les combinaisons de fonctions qui permettent de construire des motifs pertinents, en maximisant l’évaluation. Cet ensemble peut être défini comme $F' \subset F = \{f' \in F' | Q(f') > T_Q\}$ où T_Q est un seuil pour conserver uniquement les structures ayant le meilleur score. Pour guider l’exploration des agents en fonction de cette évaluation, un marquage est effectué dans une grille partagée (espace de marquage) à chaque nouvelle évaluation d’une structure de représentation. La position du marquage dans la grille correspond aux paramètres des différentes fonctions employées pour construire la structure évaluée. L’évaluation des motifs peut correspondre à leur faculté à produire des prédictions fiables, comme présenté dans [Mugan and Kuipers, 2007].

Mais dans le cas du problème d’amorçage, il n’est pas possible d’évaluer la capacité prédictive du système, tant que l’on n’a aucun motif à considérer comme une structure prédictive. Pour pallier cela, nous introduisons une mesure d’intérêt pour orienter la recherche vers les zones intéressantes de l’espace de recherche lors de la phase d’amorçage. L’intérêt des éléments de représentation est défini par leur *spécificité* et leur *poids*. Si les processus de construction sont trop précis, une trop grande quantité d’évènements spécifiques va être générée, mais ayant un poids faible. Si au contraire, le processus est trop général, peu d’éléments seront générés et auront un poids fort mais ne seront pas spécifiques, c’est-à-dire qu’ils ne se démar-

queront pas suffisamment du reste de l’expérience. Cette mesure repose donc sur l’idée de trouver un compromis entre précision et stabilité, et permet de guider le système avant que des motifs puissent être évalués, et ainsi surmonter le problème d’amorçage. La spécificité d’un élément peut être exprimée comme la différence entre cet élément et la classe d’élément la plus générale possible, appelée *référence*. Par exemple, pour une association qui contient les durées entre les occurrences de 2 évènements e_1 et e_2 , la référence est l’association qui contient toutes les durées entre les occurrences de e_1 et des évènements aléatoires e_i . La *spécificité* d’un évènement est : $s(e) = 1 - f^s(e, ref)$ définie entre 0 et 1. Notons $|e|$ le nombre d’occurrences de l’évènement $e \in E^\alpha$, et E^α l’ensemble des classes d’évènements générées par l’agent S à partir de toutes les occurrences produites.

Le *poids* d’un évènement est défini en fonction de la répartition des occurrences de cet évènement par rapport aux autres évènements de spécificité équivalente : $w(e) = \frac{|e|}{\sum_{|s(e_i) - s(e)| < \epsilon; e_i \in E^\alpha} |e_i|}$.

L’intérêt d’un évènement est $i(e) = s(e) * w(e)$.

Une fois qu’un motif est identifié comme intéressant, il peut être considéré comme une structure prédictive entre un évènement e_1 et un évènement e_2 , qui est évaluée de manière classique en fonction de son taux de succès (confiance) et de sa précision. Nous définissons pour cela un score $s = acc * rel$. La confiance (*rel*) est définie comme le rapport du nombre de prédictions réussies sur le nombre d’évènements e_1 observés : $rel = nb(predictions) / nb(e_1)$. Soit *tol* la tolérance de la prédiction, qui est l’écart type de la durée moyenne entre e_1 et e_2 lors des prédictions réussies. La précision (*acc*) est définie comme $acc = 1 - (tol * freq(e_2))$ ($acc = 0$ si

$tol < 0$), où $freq(e_2)$ est la fréquence d'apparition de e_2 .

7 Résultats

Nous proposons d'abord des résultats portant sur des données simulées. Dans cette expérience nous simulons l'interaction simple entre des variables, du type : activations d'interrupteurs provoquant des variations progressives de variables bruitées avec plus ou moins de latence. Ces activations sont répétées après un intervalle de temps aléatoire borné. L'intérêt principal du simulateur est de pouvoir tester le système sur des échelles de temps plus courtes. Dans cette expérience nous ajoutons 7 variables $V1; \dots; V7$, certaines étant liées par des régularités, comme illustré par la Figure 5. L'activation ou la désactivation de $V1$ entraîne des variations de $V2$ et $V3$. Il en va de même pour $V6$ et $V7$ avec des délais et des amplitudes différentes, tandis que $V4$ et $V5$ s'activent régulièrement et aléatoirement sans lien avec aucune autre variable.

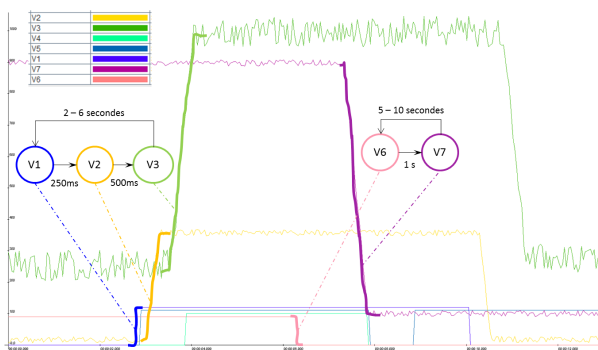


FIGURE 5 – Simulation de régularités entre variables continues.

Ce scénario a pour objectif de montrer que ces régularités émergent de l'activité du système sous forme d'associations qui restent stables. Pour cela nous visualisons les agents et les structures générées au sein du système sous la forme d'un graphe, grâce à la bibliothèque *Graphstream* ([Pigné et al., 2008]). Dans cette expérience le système est initialement composé de trois agents D par variable, des agents S sont générés à la demande, et les agents A apparaissent lorsqu'un évènement créé est suffisamment intéressant. L'état initial est représenté sur la Figure 6-A. Sans rentrer dans les détails, l'implémentation utilisée est la suivante. Agent D : génération d'histogrammes contenant la variance sur des fenêtres glissantes. Agent A : génération des histogrammes contenant la durée entre

les deux dernières occurrences d'évènements. Agent S : comparaison des histogrammes par intersection.

Lorsqu'une organisation est évaluée avec un score suffisamment élevé, les agents exploitent cette position : des évènements ou des associations sont générés (Figure 6-B⁵). Puis éventuellement les prédictions sont activées pour une association suffisamment intéressante (Figure 6-C). Lorsque la prédiction atteint un certain score, elle est confirmée. Lorsque l'évaluation n'est pas probante, les agents explorent d'autres possibilités en modifiant leur paramétrage. Une vidéo que le lecteur pourra consulter permet de mieux visualiser l'activité du système⁶.

Un motif est donc constitué par un ensemble d'agents et un ensemble d'éléments de représentation. Ces éléments contiennent des données qui décrivent précisément la nature de cette régularité. Comme le montre la Figure 6-C, nous constatons donc que dans le cas de la simulation le système est capable de retrouver et de stabiliser toutes les régularités qui existent entre les variables, pour un temps d'apprentissage variable entre une quinzaine et une trentaine d'occurrences des variations simulées, suivant les paramètres. D'autre part rares sont les régularités non existantes (faux positifs) qui apparaissent. Cela correspond généralement à des coïncidences apprises qui ne peuvent être renforcées et disparaissent rapidement. Le nombre d'observations nécessaires à l'apprentissage d'un phénomène peut être réduit d'une expérience sur l'autre si on mémorise les espaces de marquage des agents, car de cette façon le système converge plus rapidement vers des choix de discrétisation adaptés. La figure 7-(a) montre les évènements du motif qui prédit une variation de $V3$ à partir de l'activation de $V1$. De gauche à droite, on visualise l'évènement initial $e1$, l'évènement association, et l'évènement prédit $e2$. Pour les évènements $e1$ et $e2$ les histogrammes montrent la répartition de la variance sur les fenêtres temporelles sélectionnées par rapport à la répartition en général de toutes les fenêtres. Ces évènements ont été détectés comme intéressants car ils sont spécifiques et suffisamment récurrents. Mais en définitive, c'est uniquement la participation de l'évènement à un pattern permettant de réaliser des prédictions effectives (cf. score de la prédiction), qui autorise à les considérer comme des évènements pertinents. L'his-

5. Par souci de lisibilité, l'affichage des différents éléments est épuré à chaque étape sur l'image

6. <http://liris.cnrs.fr/sycosma/wiki/doku.php?id=iadev-intamb>

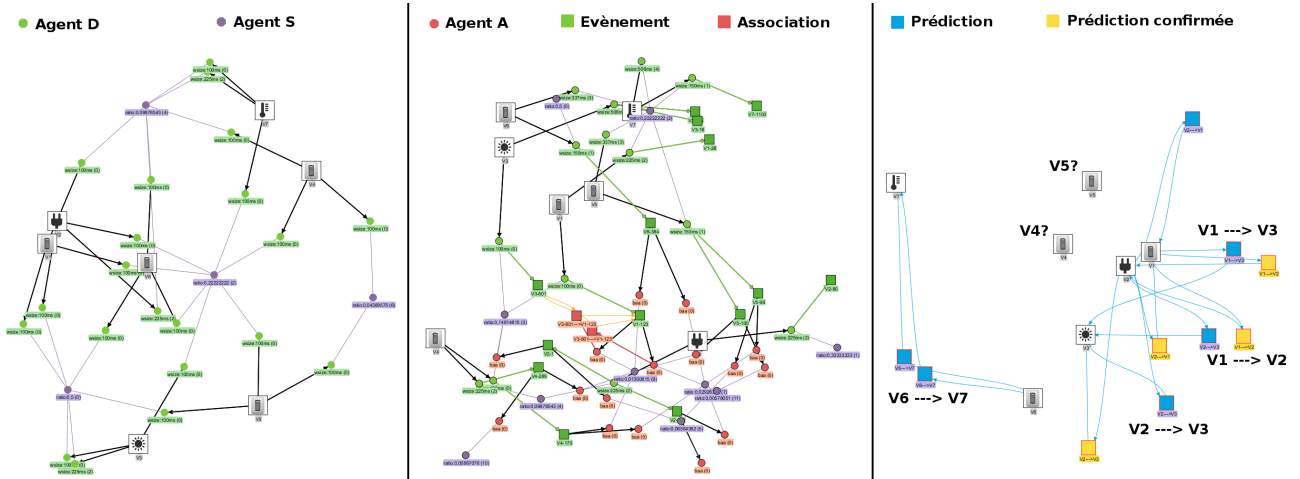


FIGURE 6 – A : Etat initial. B : Création d’associations. C : Motifs intéressants.

togramme de l’évènement association, montre quant à lui la répartition des durées observées entre les instances de $e1$ et $e2$ par rapport à la répartition des durées entre $e2$ et d’autres évènements choisis aléatoirement.

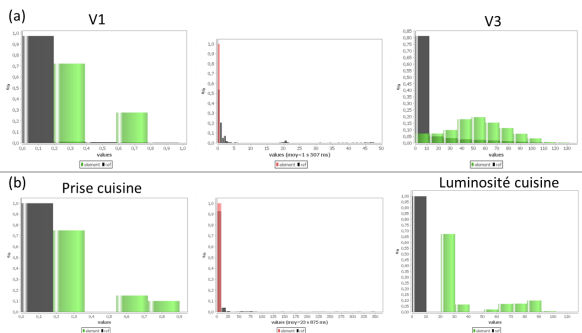


FIGURE 7 – Exemples des données des éléments perception et association d’un motif.

8 Conclusion et perspectives

Dans ce travail nous proposons une architecture qui permet de simuler et mettre en évidence la dynamique de l’amorçage de l’apprentissage de motifs sensorimoteurs à partir de signaux continus dans le cadre d’une approche constructiviste, sans s’attacher à une implémentation spécifique, et de manière décentralisée. Ce modèle autorise la création et le traitement parallèle d’une variété d’agents différents pour implémenter les différents rôles spécifiés, à l’image du concept de « société de l’esprit » développé par [Minsky, 1991]. Outre les expériences en simulation, quelques expériences préliminaires

ont été réalisées à ce jour sur un système réel comprenant une trentaine de variables parmi lesquelles (température, luminosité, CO₂, humidité, ouverture de porte, prise pilotable, etc.), dans trois pièces différentes. Une configuration préalable des plages de paramètres des agents en fonction des variables nous permet de borner un peu les espaces de recherche et d’accélérer le processus d’apprentissage. Au bout d’environ une journée (ce qui est relativement court par rapport aux échelles de temps de l’AmI), il est déjà possible d’observer quelques motifs. Par exemple le lien entre une prise pilotable et la luminosité dans une pièce, comme mentionné dans l’exemple d’introduction. Ce motif est illustré en Figure 7-(b), et les caractéristiques de la prédiction ainsi que son évaluation dans le Tableau 8. Ce travail d’implémentation est toujours en cours, en partenariat avec la société *ubiant*⁷.

EXEMPLE MOTIF SIMULATION	e1	e2	Prédiction
Fréquence	5 s 823 ms	5 s 823 ms	-
Durée moyenne	304 ms	1 s 716 ms	298 ms
Tolérance	-	-	41 ms
Précision	-	-	0,994
Nombre d’observations	17	18	17
Confiance	-	-	1
Score	-	-	0,985
EXEMPLE MOTIF REEL	e1	e2	Prédiction
Fréquence	17 m 35 s 93 ms	16 m 2 s 4 ms	-
Durée moyenne	522 ms	3 s 461 ms	25 s 293 ms
Tolérance	-	-	3 s 207 ms
Précision	-	-	0,996
Nombre d’observations	17	20	17
Confiance	-	-	1
Score	-	-	0,993

FIGURE 8 – Evaluation des motifs.

7. <http://www.ubiant.com>

Pour illustrer la dynamique du système, nous montrons une instanciation minimale des différentes opérations du modèle, reposant sur des agents réactifs simples. L'analyse qualitative des résultats présentés ici démontre la possibilité d'obtenir des motifs satisfaisants de manière efficace par cette méthode, à partir des seules données brutes, en temps réel. Par la suite, nous allons enrichir le système en ajoutant d'autres implémentations d'agents et en faisant varier certains paramètres tels que le nombre d'agents, les choix et la vitesse d'exploration, divers seuils utilisés dans les évaluations, etc. De plus amples tests sont à effectuer pour étudier quantitativement les influences de ces divers changements sur l'efficacité du système. Il est clair que plus les possibilités de discrétiser et d'associer des évènements augmentent, plus le système peut être expressif, mais plus l'espace de recherche croît. Grâce à la nature décentralisée du système, les calculs sont cependant hautement parallélisables. La gestion de la mémoire pourrait être un défi non négligeable, même si très peu de données brutes sont stockées. Enfin nous allons poursuivre le travail commencé avec le système d'AmI réel sur des temps d'expérimentations plus longs, les premiers résultats étant très encourageants.

Références

- [Augusto, 2007] Augusto, J. (2007). Ambient intelligence : the confluence of ubiquitous/pervasive computing and artificial intelligence. *Intelligent Computing Everywhere*, pages 213–234.
- [Aztiria et al., 2010] Aztiria, A., Izaguirre, A., and Augusto, J. (2010). Learning patterns in ambient intelligence environments : a survey. *Artificial Intelligence Review*, 34(1) :35–51.
- [Brooks, 1991] Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, 47(1 - 3) :139 – 159.
- [Chaput et al., 2003] Chaput, H. H., Kuipers, B., and Mikkilainen, R. (2003). Constructivist learning : A neural implementation of the schema mechanism. In *In Proceedings of the Workshop on SelfOrganizing Maps (WSOM03)*.
- [Drescher, 1991] Drescher, G. (1991). *Made-up minds : a constructivist approach to artificial intelligence*. The MIT Press.
- [Ferber and Gutknecht, 1998] Ferber, J. and Gutknecht, O. (1998). A meta-model for the analysis and design of organizations in multi-agent systems. In *Multi Agent Systems, 1998. Proceedings. International Conference on*, pages 128–135.
- [Friedewald et al., 2005] Friedewald, M., Costa, O. D., Punie, Y., Alahuhta, P., and Heinonen, S. (2005). Perspectives of ambient intelligence in the home environment. *Telematics and Informatics*, pages 221–238.
- [Georgeon and Aha, 2013] Georgeon, O. and Aha, D. (2013). The Radical Interactionism Conceptual Commitment. *Journal of Artificial General Intelligence*, 4(2) :31–36.
- [Guerin, 2011] Guerin, F. (2011). Learning like a baby : a survey of artificial intelligence approaches. *The Knowledge Engineering Review*, 26 :209–236.
- [Kuipers et al., 2006] Kuipers, B. J., Beeson, P., Moudayil, J., and Provost, J. (2006). Bootstrap learning of foundational representations. *Connection Science*, 18(2) :145–158.
- [Linaker and Jacobsson, 2001] Linaker, F. and Jacobsson, H. (2001). Mobile robot learning of delayed response tasks through event extraction : A solution to the road sign problem and beyond. In *International Joint conference on artificial intelligence*, volume 17, pages 777–782. Lawrence Erlbaum Associates Ltd.
- [Lungarella et al., 2003] Lungarella, M., Metta, G., Pfeifer, R., and Sandini, G. (2003). Developmental robotics : a survey. *Connection Science*, 15(4) :151–190.
- [Minsky, 1991] Minsky, M. (1991). Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI magazine*, 12(2) :34.
- [Mugan and Kuipers, 2007] Mugan, J. and Kuipers, B. (2007). Learning distinctions and rules in a continuous world through active exploration. In *Proceedings of the Seventh International Conference on Epigenetic Robotics (EpiRob-07)*, pages 101–108.
- [Najjar and Reignier, 2013] Najjar, A. and Reignier, P. (2013). Constructivist Ambient Intelligent Agent for Smart Environments. In *PerCom - IEEE International Conference on Pervasive Computing and Communications*, San Diego, États-Unis.
- [Oudeyer et al., 2007] Oudeyer, P.-Y., Kaplan, F., and Hafner, V. (2007). Intrinsic motivation systems for autonomous mental development. *Evolutionary Computation, IEEE Transactions on*, 11(2) :265–286.
- [Perroto et al., 2010] Perroto, F., Alvarez, I., and Buisson, J.-C. (2010). Constructivist Anticipatory Learning Mechanism (CALM) : Dealing with Partially Deterministic and Partially Observable Environments. In *International Conference on Epigenetic Robotics (EpiRob)*, pages 110–120. Lund University Cognitive Science.
- [Piaget, 1954] Piaget, J. (1954). *The Construction of Reality in the Child*. Basic Books.
- [Pigné et al., 2008] Pigné, Y., Dutot, A., Guinand, F., and Olivier, D. (2008). GraphStream : A Tool for bridging the gap between Complex Systems and Dynamic Graphs.
- [Provost et al., 2006] Provost, J., Kuipers, B. J., and Mikkilainen, R. (2006). Developing navigation behavior through self-organizing distinctive-state abstraction. *Connection Science*, 18(2) :159–172.
- [Russell and Norvig, 2009] Russell, S. and Norvig, P. (2009). *Artificial Intelligence : A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition.
- [Von Glasersfeld, 1984] Von Glasersfeld, E. (1984). An introduction to radical constructivism. *The invented reality*, pages 17–40.
- [Von Uexküll, 1992] Von Uexküll, J. (1992). A stroll through the worlds of animals and men : A picture book of invisible worlds. *Semiotica*, 89(4) :319–391.