



HAL
open science

**Approche décentralisée de l'apprentissage constructiviste
et modélisation multi-agent du problème d'amorçage de
l'apprentissage sensorimoteur en environnement continu.
Application à l'intelligence ambiante.**

Sébastien Mazac

► **To cite this version:**

Sébastien Mazac. Approche décentralisée de l'apprentissage constructiviste et modélisation multi-agent du problème d'amorçage de l'apprentissage sensorimoteur en environnement continu. Application à l'intelligence ambiante.. Intelligence artificielle [cs.AI]. Université de Lyon, 2015. Français. NNT : 2015LYO10147 . tel-01213353v2

HAL Id: tel-01213353

<https://hal.science/tel-01213353v2>

Submitted on 26 Oct 2015 (v2), last revised 2 May 2016 (v3)

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.

THÈSE

APPROCHE DÉCENTRALISÉE DE L'APPRENTISSAGE CONSTRUCTIVISTE ET MODÉLISATION MULTI-AGENT DU PROBLÈME D'AMORÇAGE DE L'APPRENTISSAGE SENSORIMOTEUR EN ENVIRONNEMENT CONTINU. APPLICATION À L'INTELLIGENCE AMBIANTE.

Présentée publiquement le :
06/10/2015

Pour obtenir :
Doctorat de l'Université de Lyon

Spécialité :
Intelligence artificielle

Formation doctorale :
Informatique

École doctorale :
Ecole Doctorale Informatique et Mathématiques - Université Lyon 1

Par :
Sébastien MAZAC

SOUTENUE PUBLIQUEMENT LE 06 OCTOBRE 2015 DEVANT LE JURY COMPOSÉ DE :

Patrick REIGNIER, Professeur, LIG, Grenoble INP	Rapporteur
Pierre DE LOOR, Professeur, Lab-STICC, Ecole Nationale d'Ingénieurs de Brest	Rapporteur
Yves DUTHEN, Professeur, IRIT, Université de Toulouse	Examineur
Stéphane DONCIEUX, Professeur, ISIR, Université Pierre et Marie Curie	Examineur
Jacques FERBER, Professeur, LIRMM, Université de Montpellier	Examineur
Olivier LEFÈVRE, Docteur, Ubiant	Invité
Frédéric ARMETTA, Maître de conférence, LIRIS, Université de Lyon	Co-encadrant
Salima HASSAS, Professeur, LIRIS, Université de Lyon	Co-encadrante et Directrice

THÈSE

APPROCHE DÉCENTRALISÉE DE
L'APPRENTISSAGE CONSTRUCTIVISTE
ET MODÉLISATION MULTI-AGENT DU
PROBLÈME D'AMORÇAGE DE
L'APPRENTISSAGE SENSORIMOTEUR
EN ENVIRONNEMENT CONTINU.
APPLICATION À L'INTELLIGENCE
AMBIANTE.

Sébastien MAZAC

Avant-propos

Résumé

L'intelligence ambiante

L'*intelligence ambiante* (AmI) est un paradigme qui émane naturellement de la conjonction de deux facteurs, à savoir la concrétisation progressive de l'*informatique ubiquitaire* annoncée par [Weiser91] (multiplication et décentralisation des unités de calcul et des appareils communicants) et bien sûr le développement de l'*intelligence artificielle* (IA). Du point de vue applicatif, ce domaine est motivé par deux principaux objectifs.

D'une part, une attention croissante est consacrée aux préoccupations écologiques au sein de nos sociétés, notamment à la consommation énergétique du bâtiment. L'AmI devrait ainsi permettre d'assurer le confort de l'utilisateur tout en réduisant considérablement le coût énergétique par une gestion habile des appareils connectés. D'autant plus que le développement des énergies renouvelables amène de nouvelles problématiques, relatives à leur nature transitoire et décentralisée. Le partage et le stockage de l'énergie, ainsi que la gestion des pics de consommation et de surproduction sont des contraintes qui viendront s'ajouter à cette gestion active du bâtiment dans le cadre des *smart grids*.

D'autre part, la deuxième idée principale qui sous-tend l'AmI est l'hypothèse que la mise en relation intelligente de tous les constituants d'un système ambiant puisse offrir en termes de services, davantage que la somme de leurs fonctions respectives. Cela suppose que le système soit conscient de ses propres possibilités d'interaction ainsi que des attentes de l'utilisateur. Il est clair qu'un système d'AmI doit alors présenter des capacités d'adaptation et d'apprentissage évoluées, pour pouvoir par exemple distinguer différentes situations, activités, évènements, etc. Du point de vue de l'IA, le domaine applicatif de l'AmI est donc extrêmement complexe, si on le considère dans son entièreté (voir [Friedewald05]). Les caractéristiques suivantes illustrent ces difficultés :

- Hétérogénéité : de l'environnement, des objectifs, du système lui-même.
- Evolutivité du système : ses capacités dépendent du matériel choisi par l'utilisateur et des retraits et ajouts possibles de matériel.
- Prise en compte de l'humain dans la boucle.

- Environnement continu / temps réel.

Problématique et contributions de la thèse

Il existe de nombreuses approches se focalisant sur un ou plusieurs aspects de l'AmI (par exemple l'optimisation de la gestion du chauffage, la reconnaissance d'activités spécifiques, etc.), et appliquant avec plus ou moins de succès des techniques existantes d'IA, par exemple de *machine learning*. Bien souvent, elles reposent sur une modélisation avancée du système d'AmI et de son environnement. L'apprentissage semble être l'aspect incontournable dans la plupart des cas, mais il n'y a pas vraiment d'approche permettant de couvrir toutes les facettes de l'AmI, comme le remarque [Aztiria10] qui met en avant la nécessité de créer des techniques spécifiques pour l'AmI. Nous considérons cette remarque comme l'expression plus générale de la difficulté en IA de gérer la complexité des agents réels incarnés (tels que les robots) dans des environnements réels (complexes et continus). Notamment, le *paradoxe de Moravec* pointe du doigt le fait plutôt surprenant que les *capacités sensorimotrices* basiques des organismes vivants sont plus difficiles à reproduire artificiellement que les capacités cognitives de plus haut niveau tel que le raisonnement. Nous proposons un état de l'art ouvert, en examinant diverses pistes offertes par la recherche en IA en réponse à ces problèmes fondamentaux, tout en confrontant ces idées aux caractéristiques spécifiques de l'AmI.

Ancrée dans l'épistémologie constructiviste et supportée par des arguments forts de la psychologie cognitive et des neurosciences (voir [Ziemke01]), l'approche constructiviste de l'apprentissage constitue une piste prometteuse en intelligence artificielle pour dépasser les limites des techniques classiques issues du cognitivisme ou du connexionnisme (voir [Guerin11]). De ce fait, des courants voisins tels que : IA incarnée, IA éactive ou Robotique développementale, sont de plus en plus étudiés. L'objectif commun à ces approches est de créer des agents intelligents autonomes car dotés d'une forte capacité d'adaptation à leur environnement à l'image des organismes biologiques. L'apprentissage se fait alors par l'expérience (*sensorimotrice*), permettant ainsi de construire une représentation adaptée au couple agent-environnement. Le champ d'application est très vaste et concerne tous les systèmes en interaction avec un environnement complexe, dont les objectifs sont variés et non prédéfinis, tels que la robotique ou évidemment l'AmI.

Nous proposons donc un modèle original d'apprentissage constructiviste adapté pour un système d'AmI. Ce modèle repose sur une approche décentralisée, permettant de multiples implémentations convenant à un environnement hétérogène. Dans les environnements réels continus sans modélisation a priori, se pose la question de la modélisation des structures élémentaires de représentation et particulièrement le *problème d'amorçage* de l'apprentissage sensorimoteur (comme décrit par [Kuipers06]). Dans le cadre du modèle général proposé, nous explicitons ce problème particulier et proposons de le traiter comme une forme d'auto-organisation modélisée par un *système*

multi-agent. Cette approche permet de construire des motifs d'interaction élémentaires à partir des seules données brutes, sur lesquels peut reposer la construction d'une représentation plus élaborée (voir [Mazac14]).

Nous présentons enfin une série d'expérimentations illustrant la résolution de ce problème d'amorçage : tout d'abord grâce à un environnement simulé, qui permet de maîtriser les régularités de l'environnement et autorise des expérimentations rapides ; ensuite en implémentant ce système d'apprentissage au sein d'un environnement d'AmI réel. Pour cela le modèle est intégré dans le système d'AmI développé par l'entreprise partenaire¹ de cette thèse CIFRE. Puis nous présentons une possible application industrielle des résultats de cette première étape implémentée d'amorçage de l'apprentissage sensorimoteur. Nous concluons par l'analyse des résultats et des perspectives de ce type d'approche pour l'AmI et l'application en général de l'IA aux systèmes réels en environnements continus.

1. ubiant - www.ubiant.com

Abstract

The theory of cognitive development from Jean Piaget (1923) is a constructivist perspective of learning that has substantially influenced cognitive science domain. Within AI, lots of works have tried to take inspiration from this paradigm since the beginning of the discipline. Indeed it seems that constructivism is a possible trail in order to overcome the limitations of classical techniques stemming from cognitivism or connectionism and create autonomous agents, fitted with strong adaptation ability within their environment, modelled on biological organisms. Potential applications concern intelligent agents in interaction with a complex environment, with objectives that cannot be predefined. Like robotics, Ambient Intelligence (AmI) is a rich and ambitious paradigm that represents a high complexity challenge for AI. In particular, as a part of constructivist theory, the agent has to build a representation of the world that relies on the learning of sensori-motor patterns starting from its own experience only. This step is difficult to set up for systems in continuous environments, using raw data from sensors without a priori modelling. With the use of multi-agent systems, we investigate the development of new techniques in order to adapt constructivist approach of learning on actual cases. Therefore, we use ambient intelligence as a reference domain for the application of our approach.

Remerciements

Tout d'abord merci à Salima Hassas, Frédéric Armetta, Olivier Lefèvre et Saber Mansour, pour leur encadrement agréable et bénéfique. Merci à *ubiant* de m'avoir permis de réaliser cette thèse dans des conditions idéales.

Je remercie beaucoup les membres du jury : examinateurs (Yves Duthen, Stéphane Doncieux et Jacques Ferber) pour avoir accepté d'évaluer mon travail, et rapporteurs (Patrick Reignier, Pierre De Loor) pour leurs critiques et commentaires précis et profitables.

Pour leur aide à la rédaction du présent mémoire, je remercie particulièrement Salima et Frédéric. Également, merci à Romuald Thion pour avoir partagé le modèle de thèse Latex utilisé ici, ainsi qu'à Isabelle Mazac pour son travail de relecture efficace.

Je salue chaleureusement tous les amis et collègues qui m'ont accompagné et aidé aux cours de ces quelques années. Parmi eux : Pierre Lehonneur, Michael Guyon-Gellin, Robert Drowilal, Laetitia Sévérac, Guillaume Laurent, Natalia Borsato, Jordi Gendreau, Morgane Denzler, Adrien Vermont, André Fabbri, Matthis Gaciarz, Maxime Gasse, Maxime Guériau, Nicolas Gastineau, Pierre Geth, Brahim Negazzi, Kamel Madi, Pascal François Mbissane Faye, Laetitia Matignon, Véronique Deslandres, Rana Chamsi Abu Quba, Eric Duchêne, Brice Effantin, Olivier Georgeon, Emmanuel Olivier, Romain Bazile, Nicolas Wiest, François Desmares, Matthieu Olivier, Franck Bercegay, Alexandre Gilibert, Cédric Morin, Nicolas Mauti, Kevin Dos Santos, Victor Lequay, Samar Benamar.

Je saisis l'occasion pour faire une dédicace à Isabelle, Serge, Élise, Germain et au reste de ma famille.
Enfin, merci à Célia.

Table des matières

Avant-propos	i
Résumé	i
Abstract	iv
Remerciements	v
Table des matières	vii
1 Contexte et positionnement	1
<i>Présentation de l'intelligence ambiante et identification des problématiques. Analyse des problèmes fondamentaux du point de vue de l'IA et choix d'un positionnement.</i>	
1.1 Sur L'Intelligence Ambiante (AmI)	3
1.1.1 Qu'est-ce que l'Intelligence Ambiante?	3
1.1.1.1 Contexte et Définition	3
1.1.1.2 Enjeux	4
1.1.2 Le rôle de l'intelligence artificielle pour l'AmI	5
1.1.3 Positionnement au sein du domaine de l'AmI	7
1.1.4 Présentation des caractéristiques de l'AmI et identification des problématiques	8
1.1.4.1 Les caractéristiques liées à l'agent et l'environnement	8
1.1.4.2 Les caractéristiques liées aux objectifs et à la représentation	10
1.2 Analyse théorique du point de vue de l'intelligence artificielle	12
1.2.1 Introduction des problèmes fondamentaux	12
1.2.2 La représentation et le problème de l'ancrage des symboles	13
1.2.3 L'IA incarnée (<i>Embodied AI</i>)	16
1.2.4 L'apprentissage sensorimoteur	18
1.2.5 Approches postcognitivistes et positionnement épistémologique	19
1.2.6 Les approches constructivistes en IA	21
1.3 Positionnement et choix théoriques	22
1.3.1 L'approche constructiviste dans le domaine de l'AmI	22
1.3.2 Principes généraux du modèle	23
1.3.2.1 Modélisation systémique	23
1.3.2.2 Approche décentralisée	25

2	Etat de l'art	27
	<i>La problématique de cette thèse se situe à la croisée de différents domaines de l'IA, à savoir : l'apprentissage artificiel, les systèmes multi-agents, ainsi que les approches neuro-inspirées et le paradigme constructiviste du point de vue de notre positionnement fondamental. Cet état de l'art présente ces domaines en se focalisant sur les problématiques identifiées relatives à l'AmI.</i>	
2.1	L'apprentissage artificiel	29
2.1.1	Introduction	29
2.1.2	L'apprentissage supervisé	31
2.1.3	L'apprentissage par récompense	32
2.1.4	L'apprentissage non-supervisé	36
2.1.5	Commentaires sur l'IA incarnée et l'apprentissage	37
2.1.6	L'apprentissage dans le cadre de l'AmI	39
2.1.6.1	Application de diverses techniques d'apprentissage artificiel	39
2.1.6.2	Exemple d'une approche globale combinant diverses techniques	41
2.2	Les systèmes multi-agents	43
2.2.1	Présentation des SMA	43
2.2.2	L'apprentissage et les systèmes multi-agents	45
2.2.2.1	Le problème de l'attribution de la récompense	46
2.2.2.2	Caractéristiques de l'apprentissage décentralisé	47
2.2.3	Les systèmes multi-agents dans le cadre de l'AmI	48
2.2.3.1	Modélisation directe d'une maison intelligente par un SMA	48
2.2.3.2	Utilisation de SMA pour résoudre le problème de la gestion de l'énergie pour l'habitat	49
2.3	Les approches neuro-inspirées	52
2.3.1	Introduction	52
2.3.2	Memory Prediction Framework	53
2.3.2.1	La plasticité du cerveau	53
2.3.2.2	Le fonctionnement du cerveau	54
2.3.2.3	La représentation invariante	55
2.3.2.4	L'importance de la prédiction	55
2.3.3	Approches reposant sur l'utilisation de cartes corticales	56
2.3.4	Conclusion	57
2.3.5	Les approches connexionnistes dans le cadre de l'AmI	58
2.4	Le paradigme constructiviste	60
2.4.1	L'intelligence comme adaptation	60
2.4.2	Application en IA	62
2.4.3	L'apprentissage sensorimoteur	64
2.4.4	Mécanismes d'apprentissage de schémas	65

2.4.4.1	Principe général	65
2.4.4.2	Amorçage de l'apprentissage sensorimoteur	65
2.4.4.3	Capacité d'abstraction et de généralisation	67
2.4.5	La robotique développementale	67
2.4.5.1	Présentation des principaux concepts	67
2.4.5.2	Motivation intrinsèque	68
2.4.6	Applications des approches constructiviste ou développementales aux environnements continus	70
2.4.7	Le problème d'amorçage	72
2.4.7.1	Définition	72
2.4.7.2	Exemples de travaux répondant au problème d'amorçage	75
2.4.8	Les approches constructivistes dans le cadre de l'AmI	77
3	Proposition d'un modèle	79
	<i>Présentation du contexte Industriel et du modèle proposé</i>	
3.1	Contexte : un modèle d'apprentissage intégré à un système d'intelligence ambiante	81
3.1.1	Présentation du système HEMIS	81
3.1.1.1	Fonctions principales	81
3.1.1.2	Architecture	82
3.1.1.3	Notions principales du modèle HEMIS	83
3.1.1.4	Le SMA cognitif	85
3.1.1.5	Le SMA réactif	86
3.1.2	Le rôle de l'apprentissage	86
3.1.2.1	Bas niveau : Apprentissage des capacités sensorimotrices	87
3.1.2.2	Niveau intermédiaire : Reconnaissance d'activités, de situations, d'évènements complexes	87
3.1.2.3	Haut niveau : Apprentissage des objectifs et des préférences utilisateurs	88
3.1.3	Intégration du système d'apprentissage	88
3.2	Présentation générale du Modèle	89
3.2.1	Aperçu global	90
3.2.2	Perception à partir d'évènements	90
3.2.2.1	Définition d'un évènement	91
3.2.2.2	Remarques concernant les actions	92
3.2.2.3	Évaluation des structures internes	93
3.2.3	L'organisation du système multi-agent	94
3.2.3.1	Les rôles	96
3.2.3.2	Couple d'agents : Producteur + Similarité	98
3.2.4	L'évaluation des organisations	99
3.2.4.1	Feedback transitionnel (F_t)	100
3.2.4.2	Feedback prédictif (F_p)	102

3.2.5	Utilisation d'espaces de marquage pour l'interaction inter-agents	104
3.2.5.1	Présentation des espaces de marquage	105
3.2.5.2	Principe d'exploration et remarques générales	106
3.2.6	Modèle étendu et propositions	107
3.2.6.1	Généralisation	107
3.2.6.2	Spécialisation	109
3.3	Présentation d'une implémentation du modèle	110
3.3.1	Élément de représentation	110
3.3.2	Comportement générique des agents	111
3.3.2.1	Exploration	113
3.3.2.2	Exploitation	113
3.3.3	Les agents D	114
3.3.3.1	Comportement général	114
3.3.3.2	Implémentations proposées	114
3.3.4	Les agents A	117
3.3.4.1	Comportement général	117
3.3.4.2	Implémentations proposées	118
3.3.5	Les agents S	120
3.3.5.1	Comportement général	120
3.3.5.2	Implémentations proposées	121
4	Application et résultats	123
	<i>Description des expérimentations réalisées et analyse des résultats</i>	
4.1	Présentation des Expérimentations	125
4.2	Amorçage et apprentissage de motifs élémentaires sur simulateur	126
4.2.1	Expériences concernant la mesure d'intérêt	126
4.2.1.1	Adaptation de la mesure à différentes échelles de temps	127
4.2.1.2	Influence du bruit sur la qualité de la mesure d'évaluation de l'intérêt.	129
4.2.1.3	Discussion sur les expériences liées à la mesure d'intérêt	131
4.2.2	Création de motifs sensorimoteurs primaires	132
4.2.2.1	Analyse globale de l'activité du système	133
4.2.2.2	Analyse des motifs appris	137
4.2.3	Temps d'apprentissage : influence du nombre d'agents et du partage des espaces de marquage	139
4.2.3.1	Influence du nombre d'agents sur le temps d'apprentissage	140
4.2.3.2	Remarques sur le temps d'apprentissage	141
4.2.3.3	Influence du partage d'un espace de marquage sur le temps d'apprentissage	142
4.3	Expérimentations en environnement réel	142

4.3.1	Commentaires sur l'exploitation des données réelles et sur les performances	143
4.3.2	Création de motifs sensorimoteurs primaires	144
4.3.3	Bilan des expériences en environnement réel	150
5	Conclusion et perspectives	151
	<i>Synthèse, présentation des travaux futurs et des perspectives.</i>	
5.1	Conclusion	153
5.1.1	Synthèse	153
5.1.1.1	Positionnement	153
5.1.1.2	Modèle	154
5.1.2	Bilan des expérimentations	155
5.1.3	Application des résultats actuels : Autocalibrage d'actions dans HEMIS	156
5.2	Perspectives	157
5.2.1	Développement du modèle et améliorations	157
5.2.1.1	Enrichissement du comportement des agents au sein du système	157
5.2.1.2	Enrichissement des interactions afin de favoriser l'auto-organisation	157
5.2.1.3	Implémentation d'une plus large variété d'agents	158
5.2.1.4	Développement des mécanismes de spécialisation et de généralisation	158
5.2.2	Problématiques et pistes de recherche	159
5.2.2.1	Etudier les possibilités d'apprentissage commun inter-systèmes	159
5.2.2.2	Feedback utilisateur	159
5.2.2.3	Apprentissage actif	160
A	Annexes	161
A.1	Arbre des concepts et paradigmes	163
A.2	Publications	164
	Bibliographie	174

Table des figures

1.1	Un système d'intelligence ambiante est composé d'un ensemble d'appareils communicants disséminés dans l'environnement.	3
1.2	Les domaines de l'intelligence ambiante. (Figure par [Augusto07]) . . .	4
1.3	Illustration du modèle cognitiviste	15
1.4	Illustration du concept de <i>Umwelt</i> (monde interne)	19
1.5	Illustration du modèle constructiviste	20
2.1	Apprentissage par renforcement	33
2.2	Principe de l'algorithme génétique	35
2.3	Les trois types de capteurs du modèle de [Aztiria09]	42
2.4	Architecture du système MAHAS [Abrás09a]	50
2.5	Composantes principales du système MAHAS [Abrás09a]	51
2.6	(a) Découpage du cortex en cartes corticales composées d'un ensemble de colonnes. (b) Théorie neuronale de la cognition. Réseau hiérarchique de cartes corticales.	57
2.7	Architecture du modèle ACHE (The neural network house)	59
2.8	Schéma	65
2.9	L'architecture proposée par [Linaker01] pour le robot mobile confronté au <i>delayed response task problem</i>	72
2.10	L'environnement simulé du robot nourrisson ([Mugan07a])	75
2.11	Processus de discrétisation de l'expérience ([Mugan07b])	76
3.1	Principe : le système HEMIS en tant que système d'exploitation de l'habitat.	82
3.2	Architecture globale du système HEMIS.	83
3.3	Du point de vue théorique : frontière agent-environnement comme un ensemble de variables continues	89
3.4	Modèle général d'apprentissage constructiviste.	91

3.5	(a) L'expérience continue modélisée comme un ensemble de variables. Un motif est une régularité récurrente dans l'évolution des variables.	
	(b) Les trois types d'opérations élémentaires composantes de la construction d'un motif sensorimoteur.	
	<i>A</i> : Découper : proposer une discrétisation de l'expérience (création d'un évènement à partir de l'évolution continue).	
	<i>B</i> : Associer : proposer des corrélations entre évènements (création d'une association).	
	<i>C</i> : Comparer : évaluer la similarité entre des éléments de représentation comparables.	95
3.6	Activité globale du système.	96
3.7	Organisations possibles entre les types d'agents.	98
3.8	Exemples d'interactions entre instances d'agents.	99
3.9	Le feedback transitionnel permet de guider à priori la discrétisation de l'expérience vers des structures stables et potentiellement intéressantes.	100
3.10	Illustration du rôle des notions de "spécificité" et de "poids" dans la mesure d'intérêt proposée.	102
3.11	Le feedback prédictif permet d'évaluer l'intérêt utile des structures produites.	103
3.12	Illustration de la construction d'un motif par la collaboration des trois rôles d'agents. Les agents sont représentés par des ronds. Le motif est composé d'une association reliant deux évènements issus des variables : e_1 et e_2 . e_1 est l'évènement de référence de l'agent A_1 , comme modélisé par la variable interne v'_1 à laquelle il est attaché.	104
3.13	Représentation d'un espace de marquage	106
3.14	L'étape de généralisation	108
3.15	L'étape de spécialisation	109
3.16	Méthode <i>execute()</i> d'un agent.	111
3.17	Méthode <i>evaluate()</i> d'un agent.	112
3.18	Agent D : variations sur fenêtre glissante	115
3.19	Test de différentes fenêtres temporelles pour l'interprétation d'évènements en tant que variations	116
3.20	Interprétation d'évènements en tant que variations pour le CO2 et la luminosité sur 24 heures de données réelles	117
3.21	Agent association : binaire durée relative (l'agent est connecté à l'évènement e_2 , et génère une représentation de l'association avec l'évènement relatif e_1 , à partir des occurrences successives)	118
3.22	Les évènements de plusieurs facteurs environnementaux générés par le mécanisme de variation sur 24 heures	119
3.23	Les associations générées entre 8 heures et 10 heures	120
3.24	Méthode <i>performTask()</i> d'un agent S.	120
3.25	Agent similarité : intersection	121

4.1	Les différents paramètres du simulateur pour caractériser l'activité des variables (légende présentée dans le Tableau 4.2).	126
4.2	Les différents paramètres du simulateur pour caractériser l'activité des variables (voir Figure 4.1).	127
4.3	Paramétrage du simulateur. Dans cette expérience seule la durée de variation (DV) change et les autres paramètres de durée sont ajustés proportionnellement.	128
4.4	Visualisation de l'espace de marquage de la mesure d'intérêt en fonction de la durée de variation (x : nombre de classes de l'agent AS_{HSI} ; y : durée de la fenêtre de l'agent AD_{FGV} en ms ; z : mesure d'intérêt)	129
4.5	Dans cette expérience seule l'amplitude du bruit varie.	130
4.6	Visualisation de l'espace de marquage de la mesure d'intérêt en fonction de l'amplitude du bruit par rapport à l'amplitude de variation (en %) (x : nombre de classes de l'agent AS_{HSI} ; y : durée de la fenêtre de l'agent AD_{FGV} en ms ; z : intérêt)	131
4.7	Visualisation de l'espace de marquage de la mesure d'intérêt pour la luminosité (expérience en système réel).	132
4.8	Simulation de régularités entre variables continues (voir Tableau 4.9). . .	133
4.9	Simulation de régularités entre variables continues (voir Figure 4.8). . .	133
4.10	A : Etat initial. B : Création d'évènements et d'associations. Un agent explorateur est représenté par un rond et un agent exploiteur par un rond cerclé de noir.	135
4.11	Un exemple d'état stable des motifs produits par le système. Visualisation simplifiée des liens temporels entre les variables (évènements cachés)	136
4.12	Visualisation de 2 motifs primaires entre V1 et V2. Les histogrammes représentent (de la gauche vers la droite) : le modèle du premier évènement, le modèle de l'association (durée entre les 2 évènements) et le modèle du deuxième évènement (voir Tableau 4.13).	137
4.13	Caractéristiques des 2 motifs primaires entre V1 et V2 (voir Figure 4.12). . .	138
4.14	Visualisation de 2 motifs primaires entre V6 et V7. Les histogrammes représentent (de la gauche vers la droite) : le modèle du premier évènement, le modèle de l'association (durée entre les 2 évènements) et le modèle du deuxième évènement (voir Tableau 4.15).	138
4.15	Caractéristiques des 2 motifs primaires entre V6 et V7 (voir Figure 4.14). . .	139
4.16	Distribution du temps d'apprentissage (en nombre d'observations) par rapport au nombre d'agents par variable (explorateur/exploiteur) sur 10 exécutions.	141
4.17	Utilisation du CPU (en %, en haut) et de la mémoire (en MB, en bas) au cours de l'activité en mode "temps accéléré".	144
4.18	Visualisation des motifs obtenus au bout d'une trentaine d'heures. . . .	145

4.19	Visualisation de 2 motifs primaires. En haut : une régularité entre l'ouverture des portes de deux pièces. En bas, le lien entre l'activation d'une <i>smartplug</i> et la luminosité (voir Tableau 4.20).	147
4.20	Caractéristiques des 2 motifs primaires entre : d'une part "porte R&D" et "porte prod" dans les zones "R&D" et "prod", d'autre part " <i>smartplug</i> " et "luminosité" dans la zone "cuisine" (voir Figure 4.19).	147
4.21	Visualisation de 2 motifs primaires. L'ouverture d'une porte donnant sur l'extérieur provoque une légère variation de la température dans deux zones adjacentes.	148
4.22	Visualisation de 2 motifs primaires. En haut : une régularité entre une variation d'humidité et une variation de COV dans une même zone. En bas, une prédiction d'une ouverture de porte à partir d'une variation de COV (voir Tableau 4.23).	149
4.23	Caractéristiques des 2 motifs primaires entre : d'une part "humidité" et "COV" dans la zone "prod", d'autre part "COV" dans la zone "prod" et porte R&D (voir Figure 4.22).	149

1

Contexte et positionnement

▷ Dans ce chapitre nous commençons par présenter le domaine de l'intelligence ambiante, ses enjeux et ses problèmes. Nous présentons notre positionnement à l'intérieur de ce domaine, avec une approche orientée avant tout vers l'IA et l'apprentissage. Ensuite nous identifions les problématiques fondamentales que nous rencontrons du point de vue de l'IA et nous justifions un positionnement théorique plus général, épistémologique et scientifique. ◁

Plan du chapitre

1.1	Sur L'Intelligence Ambiante (AmI)	3
1.1.1	Qu'est-ce que l'Intelligence Ambiante?	3
1.1.2	Le rôle de l'intelligence artificielle pour l'AmI	5
1.1.3	Positionnement au sein du domaine de l'AmI	7
1.1.4	Présentation des caractéristiques de l'AmI et identification des problématiques	8
1.2	Analyse théorique du point de vue de l'intelligence artificielle	12
1.2.1	Introduction des problèmes fondamentaux	12
1.2.2	La représentation et le problème de l'ancrage des symboles . .	13
1.2.3	L'IA incarnée (<i>Embodied AI</i>)	16
1.2.4	L'apprentissage sensorimoteur	18
1.2.5	Approches postcognitivistes et positionnement épistémolo- gique	19
1.2.6	Les approches constructivistes en IA	21
1.3	Positionnement et choix théoriques	22
1.3.1	L'approche constructiviste dans le domaine de l'AmI	22
1.3.2	Principes généraux du modèle	23

confondu avec des concepts proches tels que : *domotique, informatique ubiquitaire, environnements pervasifs, Internet des objets*, etc. Or, ce qui distingue l'AmI de ces précédentes notions c'est bien l'apport d'une intelligence artificielle pour coordonner les composantes de l'environnement afin que celui-ci devienne adaptatif et proactif ; le but étant ainsi d'améliorer l'expérience de l'habitant. C'est pourquoi les termes suivants que l'on peut souvent rencontrer, sont davantage en accord avec la notion d'intelligence ambiante : « *environnements intelligents* » (*smart environments*). [Augusto07] donne une analyse détaillée de ce concept et de ses possibles définitions, et propose une architecture générale pour modéliser un système d'AmI. Pour résumer, on peut définir l'intelligence ambiante comme étant : *l'application de l'intelligence artificielle aux environnements qui réalisent le paradigme de l'informatique ubiquitaire, dans le but d'améliorer l'expérience de l'habitant*. La Figure 1.2 illustre les relations entre l'AmI et les domaines qui lui sont liés.

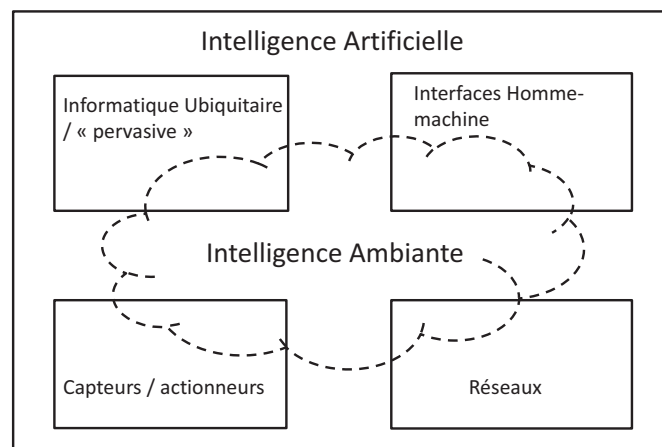


FIGURE 1.2 – Les domaines de l'intelligence ambiante. (Figure par [Augusto07])

1.1.1.2 Enjeux

Du point de vue applicatif, ce domaine est motivé par deux principaux objectifs.

L'enjeu écologique

Premièrement, à notre époque une attention croissante est consacrée aux préoccupations écologiques, notamment à la consommation énergétique du bâtiment. L'évolution technologique et industrielle a permis d'atteindre un niveau de confort important dans les lieux de vie des pays développés, mais globalement au prix d'une dépense énergétique élevée. De manière générale, la réponse à ce problème passe naturellement par l'optimisation *passive* des moyens techniques (ex. : appareils plus efficaces, meilleure isolation, etc.) mais aussi par l'optimisation *active* des comportements (ex. : éteindre les appareils inutiles, maintenir la température optimale, la qualité de l'air optimale, etc.).

En effet, ce dernier aspect est loin d'être négligeable, car une bonne gestion des appareils consommateurs dans le temps peut réduire considérablement la dépense énergétique. Cependant, cette tâche de gestion et d'optimisation des ressources d'une maison, peut rapidement devenir complexe et fastidieuse pour un utilisateur, et ce d'autant plus que les appareils sont de plus en plus nombreux et sophistiqués. L'AmI peut intervenir pour suppléer l'humain dans cette tâche de gestion active de la consommation énergétique du bâtiment. L'une des idées principales qui motivent l'intelligence ambiante serait donc de permettre de maintenir le confort des utilisateurs dans leurs lieux de vie et même de l'améliorer tout en réduisant considérablement le coût énergétique. L'AmI devrait ainsi permettre d'atteindre cet objectif par une gestion habile des appareils connectés. Cette tâche est d'autant plus pertinente que le développement des énergies renouvelables amène de nouvelles problématiques propres à leur nature transitoire et décentralisée. Le partage et le stockage de l'énergie, ainsi que la gestion des pics de consommation et de surproduction sont des contraintes qui viendront s'ajouter à cette gestion active du bâtiment dans le cadre de ce que l'on nomme les *smart grids*.

De nouveaux usages et services

D'autre part, la deuxième idée principale qui sous-tend l'AmI est l'hypothèse suivant laquelle la mise en relation intelligente de tous les constituants d'un système ambiant peut offrir davantage de services que la somme de leurs fonctions respectives. L'éventail des possibilités offertes dépend du nombre et de la variété des fonctions offertes par les objets connectés ainsi que de l'imagination des utilisateurs et concepteurs de ces systèmes. Les maisons intelligentes (*smart homes*) ne sont qu'une application parmi d'autres de l'AmI. Toutes sortes d'environnements : bureaux, magasins, transports, usines, salles de sports, musées, etc. pourraient être concernés par l'AmI et posséder leurs propres innovations dans ce domaine. Nous choisissons dans le cadre de ce travail de considérer le bâtiment intelligent classique (maison, bureau) comme cas d'étude. Dans ce type d'environnement [Friedewald05] présente une analyse de ces services possiblement émergents grâce à l'AmI. Nous ne détaillons donc pas davantage cet aspect prospectif de la discipline dans cet état de l'art. De même l'intelligence ambiante est un domaine multidisciplinaire et à ce titre comporte par exemple des enjeux sociaux et philosophiques, tels que le respect de la vie privée ou la dépendance à la technologie. Nous n'abordons pas ici ces aspects pour nous focaliser plutôt sur le problème technique et scientifique lié à l'intelligence artificielle et à l'apprentissage dans ce domaine ciblé qu'est l'AmI.

1.1.2 Le rôle de l'intelligence artificielle pour l'AmI

Nous avons vu que pour réaliser certains objectifs (ex. : consommer moins d'énergie), un système d'AmI doit pouvoir contrôler un éventail de matériel (principalement des capteurs et des actionneurs). Pour cela, il ne suffit pas de fournir les moyens à un utilisateur de paramétrer et d'automatiser le plus possible d'options de l'environne-

ment. Il est crucial que la technologie soit intuitive et discrète pour l'utilisateur et ne lui impose pas une charge supplémentaire (ex. installation, configuration, programmation, maintenance). En effet, on observe que les systèmes domotiques « classiques » (*home automation*) ne rencontrent pas un large développement, car ils ne sont pas adaptatifs et sont donc fastidieux à utiliser pour un usager non initié. Par conséquent, la plupart des utilisateurs potentiels ne trouvent pas un bénéfice réel à l'achat et à l'utilisation de tels systèmes, et ce malgré l'intérêt écologique qui pourrait en découler. Outre les questions d'optimisation, l'utilisation d'une *intelligence artificielle* pourrait permettre de décharger l'utilisateur de la contrainte de la gestion des équipements, qui tend à devenir de plus en plus complexe dès lors que l'on veut réaliser des objectifs potentiellement antagonistes, tels que confort et consommation énergétique, tout en ayant à prendre en compte une grande quantité d'informations et d'équipements.

Comme nous l'avons dit, le système doit être au service de l'habitant, et non pas lui imposer un comportement ciblé pour pouvoir assurer un bon fonctionnement. Dans certains systèmes domotiques (ou de *Gestion technique de bâtiment* (GTB)), l'utilisateur ne peut pas piloter à volonté certains appareils tels que les volets ou le chauffage. Ce genre de contraintes est généralement très mal vécu par les utilisateurs qui se plaignent souvent de l'inefficacité de tels systèmes. Ce type d'approche repose souvent sur des modélisations physiques à priori, et fonctionne en automate sans avoir les facultés d'adaptation ni d'apprentissage nécessaires à la prise en compte de l'utilisateur humain. Or dans le paradigme de l'AmI, c'est bien l'utilisateur qui doit être au centre du dispositif, et non pas le concepteur qui lui impose son système. La diversité des environnements (propriétés thermiques du bâtiment, climat), des habitants (préférences, habitudes) et des équipements, rend inenvisageable une résolution à ces problèmes par une simple automatisation à priori.

Si l'on tient compte des contraintes précitées, l'IA peut apporter des solutions pour mettre en place des systèmes adaptatifs, capables d'apprendre, d'anticiper, et de prendre des décisions pertinentes pour agir et/ou informer l'utilisateur, sans le gêner ou le perturber dans ses activités. Le rôle de l'IA dans le contexte de l'AmI est central mais aussi très prospectif. Dans l'idéal, un système d'AmI devrait présenter des capacités d'adaptation et d'apprentissage évoluées pour pouvoir, par exemple, distinguer différentes situations, activités, événements, etc. se produisant dans l'environnement et pouvoir utiliser cette connaissance efficacement. Les questions principales auxquelles nous essayons de répondre au cours de ce premier chapitre sont donc : quelles sont les solutions offertes par le domaine de l'IA pour affronter ces problèmes ? Lesquelles nous proposons de privilégier et pourquoi ?

1.1.3 Positionnement au sein du domaine de l'AmI

Cette thèse² s'inscrit dans le développement d'un système applicatif d'intelligence ambiante : *HEMIS* (Home Energy Management Intelligent System), développé par l'entreprise *ubiant*³. Le système HEMIS est un système d'intelligence artificielle reposant sur l'utilisation de *système multi-agents* (SMA) permettant de réduire la consommation d'énergie des bâtiments tout en maximisant le bien-être de leurs occupants. Nous présentons plus en détails ce système dans la partie 3.1.1. Notre objectif est de créer un module d'apprentissage intégré au sein de ce système afin de pousser plus loin ses capacités d'adaptation notamment grâce à l'identification de motifs dans le flux de données générés.

Ainsi nous travaillons à partir d'un système d'AmI autonome (HEMIS), possédant ses propres mécaniques internes de fonctionnement. L'objectif dans cette thèse n'est donc pas de proposer un système d'AmI complet en soi, mais d'étudier les possibilités d'apprentissage dans ce contexte d'interaction entre un système d'AmI et l'environnement complexe dans lequel il évolue. Nous choisissons de considérer le problème de l'apprentissage d'un point de vue fondamental en IA. Tout d'abord on peut formaliser le problème en ces termes : *l'environnement intelligent* (ou système d'AmI) est un système composé d'éléments interconnectés, qui interagit avec un domaine extérieur. Ce domaine est constitué de l'environnement physique (ex. température), le domaine biologique (les occupants) et le domaine informationnel (le réseau). Ces différents domaines sont perceptibles par le système grâce à différents types de capteurs. Le système peut également "impacter" ces domaines grâce à des actionneurs. D'un point de vue théorique, rien ne différencie radicalement un environnement intelligent d'un robot : il s'agit d'un système constitué d'un ensemble de capteurs et d'actionneurs en interaction avec un environnement. Du point de vue du système, tout ce qui lui est extérieur constitue son environnement, c'est-à-dire le monde physique, les individus et tout le reste. Dans la suite, le terme "*environnement*" fera référence à cet "extérieur"; *l'environnement intelligent* (ou système ambiant) étant quant à lui désigné comme "le système" ou "l'agent". Donc on peut considérer le système comme un agent situé et incarné, dans un environnement dynamique complexe (« monde réel »). On souhaite que l'agent puisse prendre des décisions « intelligentes » en utilisant les connaissances acquises grâce à l'apprentissage, c'est-à-dire bénéficier de son expérience pour améliorer ses performances. On peut simplement modéliser l'interface entre l'agent et l'environnement par un ensemble de variables. De manière générale, le but d'un système d'apprentissage consiste à déceler entre ces variables des relations (ou régularités) qui représentent la structure sous-jacente de l'environnement. Il s'agit donc de construire une représentation du monde, ou en d'autres termes un ensemble de prédictions sur les interactions agent-environnement. Nous optons donc pour une conception du système ambiant comme un robot doté de capacités d'interaction avec un environnement

2. Convention CIFRE

3. www.ubiant.com

inconnu, ne disposant pas de représentation fournie à priori mais devant apprendre par son expérience. Le module d'apprentissage devra ainsi tirer profit d'un ensemble de variables au sein du système d'AmI afin d'en extraire des motifs d'interaction utiles à l'amélioration des capacités d'adaptation du système global.

1.1.4 Présentation des caractéristiques de l'AmI et identification des problématiques

On peut énumérer six caractéristiques essentielles des difficultés liées à l'AmI pour la conception d'un système IA : *Continuité de l'environnement*, *Temporalité complexe*, *Indéterminisme et évolutivité*, *Hétérogénéité*, *Multimodalité et abstraction*, *Intelligibilité*, que nous proposons de distinguer en deux catégories : d'une part les caractéristiques relatives à l'agent et son environnement, d'autre part les caractéristiques liées aux objectifs et à la représentation. Nous introduirons dans cette partie les problèmes fondamentaux associés à ces différentes caractéristiques.

1.1.4.1 Les caractéristiques liées à l'agent et l'environnement

Caractéristique 1 : *Continuité de l'environnement*

Le système fait partie d'un environnement continu complexe (le monde réel et les utilisateurs) et doit gérer son activité en temps réel. Il doit ainsi gérer un important flot continu de données brutes. Pour supporter ce "flot" de l'expérience –ou de manière imagée : boire à une lance d'incendie ("the firehose of experience" ([Kuipers05])– le système doit pouvoir isoler les aspects intéressants et "laisser passer" le reste. Le cerveau humain est ainsi capable de gérer un flux d'information bien supérieur à celui de la plupart des systèmes artificiels que l'on conçoit. Dans le cas des systèmes artificiels on doit par ailleurs gérer les problèmes techniques liés à la collecte et à l'interprétation des données produites par les capteurs. Les données bruitées, incomplètes ou imprécises sont fréquentes. C'est pourquoi, il est nécessaire d'avoir une redondance des sources d'information pour repérer et gérer des erreurs.

Caractéristique 2 : *Temporalité complexe*

- Besoin de réactivité (court-terme). Par exemple, si le système doit baisser les volets roulants lorsque l'activité "regarder un film" débute, il doit pouvoir reconnaître l'activité immédiatement, sinon cette connaissance est inutile pour la réalisation de cet objectif.
- Besoin de planification / anticipation (long-terme). Ainsi, dans le cas de l'optimisation de la consommation énergétique, le système doit anticiper les besoins de chauffage en fonction de la présence des habitants et des contraintes du réseau.

- Différentes notions temporelles : date, durées, cycles, séquences, périodes, etc. En effet, que ce soit dans l'apprentissage de régularités, ou dans la communication avec l'humain, ces différentes notions temporelles sont essentielles.
- Différents ordres de grandeurs : secondes, minutes, heures, jours, semaines, mois, saisons, années. Les méthodes d'apprentissage notamment devraient être indépendantes de l'échelle de temps. Par exemple, pour apprendre indifféremment à reconnaître un évènement rapide tel qu'une variation de luminosité ou un évènement étendu tel qu'une variation de température. Ou encore reconnaître un lien temporel immédiat tel qu'entre l'activation d'un interrupteur et une variation de luminosité, mais également un lien temporel plus long mais régulier (ex. entre le début de l'activité "prendre le petit déjeuner" et l'évènement "quitter le logement").

Les caractéristiques 1 (continuité) et 2 (temporalité) sont liées au problème de la *discrétisation de l'expérience*, c'est à dire le passage du domaine spatio-temporel continu (des données brutes) à une représentation discrète interprétée et compressée.

Caractéristique 3 : Indéterminisme et évolutivité

L'environnement physique dans lequel opère le système est à priori indéterminé, et il est de plus susceptible d'évoluer. Il en va exactement de même pour les objectifs du système. Même si l'on peut supposer que les systèmes d'AmI possèdent tous plus ou moins des fonctions essentielles communes, leurs objectifs sont avant tout liés à leurs utilisateurs, à leur environnement et surtout dépendent des capacités du matériel dont est constitué le système. Il va de soi que les objectifs sont susceptibles d'évoluer, par exemple en fonction d'une période (semaine/weekend ou hiver/été), ou en fonction d'utilisateurs différents. Le système est à priori indéterminé dans sa constitution même. De plus, cette constitution évolue :

- Les capacités de perception et d'action du système dépendent du matériel choisi par l'utilisateur.
- Des mises à jour du matériel peuvent changer son comportement.
- Des changements dans l'environnement modifient le comportement (ex. déplacement d'un capteur d'un endroit à un autre).

Ce point est très important car il rend très difficile toute approche experte reposant sur une forte modélisation à priori, sauf si on limite le système ambiant à des éléments connus prédéterminés. Mais dans ce cas le système n'est plus entièrement conforme au paradigme de l'AmI.

La caractéristique 3 (indéterminisme et évolutivité) est relative au problème de l'*incarnation de l'intelligence*, c'est à dire la dépendance entre le processus cognitif et les capacités de perception et d'action (ie. sensorimotrices) du système. Lorsqu'on ne peut

maîtriser totalement ces capacités comme dans le cas de l'AmI, la conception d'un système intelligent par une approche experte devient moins aisée. Cela conduit alors l'IA à aborder la question de l'apprentissage développemental, c'est à dire l'apprentissage réalisé par un organisme pour prendre possession de ses moyens d'interagir avec son environnement, à l'image d'un enfant au cours de ses premières années.

D'un point de vue combinatoire, assurer les multiples objectifs de l'AmI est extrêmement complexe en tant que tâche d'optimisation et de planification abstraite ; mais comme pour beaucoup de systèmes d'IA évoluant en environnement réel, la principale difficulté réside avant tout dans l'élaboration du lien entre l'environnement physique et la *représentation* ou le *contrôle* du système. Le problème de la *discrétisation de l'expérience* et celui de l'*incarnation de l'intelligence* constituent un problème fondamental en IA que nous nommons problème du *couplage agent-environnement* ou problème de l'*apprentissage sensorimoteur*.

1.1.4.2 Les caractéristiques liées aux objectifs et à la représentation

Caractéristique 4 : *Hétérogénéité*

L'hétérogénéité liée au domaine de l'AmI est multiple :

- Hétérogénéité de l'environnement : phénomènes continus / discontinus ; événements ponctuels / récurrents ; des données quantifiables / qualitatives ; données réelles / ressenties ; certaines / incertaines ; etc.
- Hétérogénéité des objectifs (potentiellement antagonistes)
 - Confort : Régulation de variables (température, luminosité, humidité, qualité de l'air), reconnaissance de préférences utilisateurs, contrôle d'équipements, etc.
 - Réduction de la consommation énergétique : contrôle du matériel, planification, etc.
 - Sécurité : détection de présence, détection d'anomalie, reconnaissance d'activités, etc.
- Hétérogénéité des composantes du système : capteurs et actionneurs de différents types, appareils connectés variés, interfaces de communications, etc.

Cette diversité des données et des concepts susceptibles d'intervenir est une contrainte qui rend difficile la conception d'un modèle basé sur un principe unique, ou sur une représentation trop figée de la connaissance. Un problème fondamental soulevé ici est le problème d'*expressivité de la représentation*. Dans ce type de contexte, un système d'apprentissage doit en effet pouvoir exprimer une large panoplie de concepts.

Caractéristique 5 : *Multimodalité et abstraction*

Il est souvent nécessaire de prendre en compte des informations de natures différentes pour reconnaître un concept abstrait. Des capteurs de mouvements ont été utilisés dans de nombreuses applications pour détecter une présence ou suivre les dé-

placements des utilisateurs. Mais il faudrait également pouvoir identifier un utilisateur particulier. Pour cela, de nouvelles technologies telles que les tags RFID ont été proposées pour permettre de tracer et d'identifier précisément les utilisateurs et les objets. Pour chaque besoin on peut de cette façon développer un matériel spécifique et un programme dédié pour gérer ce matériel mais ce type d'approche est limitée et peu souple.

On peut plutôt généraliser la problématique en considérant ces différents problèmes de reconnaissance comme l'identification de motifs complexes dans le flot de données global du système. En effet des données d'une unique source peuvent ne pas être suffisamment expressives ou trop inconstantes pour la tâche d'identification qu'on souhaite effectuer. Une combinaison de plusieurs sources de données brutes différentes peut permettre de reconnaître plus efficacement des concepts non triviaux. Ce point est analysé par [Perotto07] qui évoque le lien entre les deux caractéristiques suivantes d'un environnement : *accessibilité* et *déterminisme*. Plus un environnement est accessible à l'agent (plus il en a de perceptions différentes), plus l'environnement peut lui sembler déterministe. Ceci fait écho à ce que Brooks, nomme *sensory integration* : l'utilisation de la complémentarité de plusieurs modalités facilite l'apprentissage et diminue la complexité ([Brooks98]). Cela concorde avec le fait qu'un être humain est capable de distinguer davantage d'objets, lorsqu'il peut les caractériser sur plusieurs modalités plutôt que sur une seule dimension [Miller56]. [Lefort12] propose un état de l'art intéressant sur cette question de la multimodalité.

En l'absence d'équipements dédiés, on peut faire l'hypothèse que le système pourrait par exemple identifier la présence de manière robuste grâce à une combinaison d'autres capteurs non forcément prévus à cet effet. De manière générale, l'idée serait de ne pas limiter un matériel à un usage unique et tout miser sur son bon fonctionnement pour assurer un objectif, mais plutôt d'exploiter l'ensemble des capacités du système, y compris celles que l'on ne peut prévoir à l'avance. Lorsqu'on considère la reconnaissance de motifs plus complexes tels que des activités (ex. prendre le petit-déjeuner, jouer à la console, faire le ménage, ...) il est clair que la prise en compte de multiples informations sera nécessaire, mais qu'on ne peut simplement pas prévoir à l'avance quels capteurs seront impliqués et de quelle façon, pour chacun des motifs possibles.

Il est ainsi pertinent de considérer un apprentissage *multimodal* qui se fait en fonction des propres capacités du système à collecter différents types de données, quelles qu'elles soient. En conséquence, deux systèmes radicalement différents dans leurs constitutions (ex. types de capteurs) pourraient être capables d'apprendre à reconnaître des choses différentes et inaccessibles pour l'autre, mais aussi d'apprendre un même concept par des moyens différents. Par exemple, pour identifier l'activité "quitter le logement" un système particulier pourrait se baser sur la diminution du taux de CO2 dans le logement ainsi que sur l'absence de mouvement, un autre ne possédant ni capteur de mouvement ni capteur de CO2, pourrait se baser sur une série d'évènements : activation puis désactivation de la lumière dans le hall ; activation du mode économie d'énergie du thermostat ; et ouverture/fermeture de la porte d'entrée.

D'autre part, la reconnaissance d'un concept suppose une distinction entre le *concept* et une *instance du concept*. Par exemple, à chaque fois que l'activité "*faire la cuisine*" se produit, elle est sensiblement différente, mais le système doit être capable de généraliser. De plus il pourrait créer une représentation hiérarchique de concepts liés les uns aux autres par des relations de généralisation. Par exemple : "cette forme < cet objet perçu < une boule < une balle de pingpong < une balle < un objet < un jeu < etc.". A tout moment lorsqu'un concept est perçu, cet ensemble de représentations associées plus ou moins générales serait alors "activé". Cette caractéristique illustre le problème de la *capacité d'abstraction*.

Caractéristique 6 : *Intelligibilité*

Enfin, pour que les concepts appris soient convenablement exploités, ils doivent pouvoir être intégrés à un système capable de prendre des décisions de haut niveau. C'est pourquoi il est nécessaire que le système d'apprentissage soit aisément interconnecté avec une gamme plus large de systèmes. Si les résultats de l'apprentissage sont exploités par un autre système ou combinés avec d'autres types de connaissances pour obtenir des informations plus pertinentes, de tels processus vont normalement demander des connaissances "compréhensibles". De plus, il sera peut-être nécessaire que le mécanisme d'apprentissage puisse interagir avec l'utilisateur, notamment pour expliquer les décisions prises. Ainsi il est essentiel que le processus et ses "sorties" soient facilement transformables en langage compréhensible par l'utilisateur. Cette caractéristique introduit le besoin de communication du système avec un autre agent (humain ou artificiel), que l'on nommera le problème de la *communication*.

Les problèmes présentés ici : *expressivité*, *capacité d'abstraction* et *communication* constituent des aspects du problème fondamental de la *représentation*. Le problème de la représentation désigne la question de savoir comment le processus cognitif se réalise en interaction avec les structures qui constituent le système et contiennent l'information dont il dispose. Comment est stockée l'information, comment est-elle utilisée dans le processus cognitif pour produire du sens, et comment peut-elle être échangée avec un système extérieur ?

1.2 Analyse théorique du point de vue de l'intelligence artificielle

1.2.1 Introduction des problèmes fondamentaux

Les problèmes identifiées dans le cadre de notre positionnement par rapport à l'AmI, que nous avons présentés comme le problème de l'apprentissage sensorimoteur et le problème de la représentation, sont des problèmes fondamentaux de l'IA, comme l'énonce [Leahu08] :

"In many ways, the central problem of ubiquitous computing - how computational systems can make sense of and respond sensibly to a complex, dynamic environment laden with human meaning - is identical to that of Artificial Intelligence (AI). Indeed, some of the central challenges that ubi-comp currently faces in moving from prototypes that work in restricted environments to the complexity of real-world environments - e.g. difficulties in scalability, integration, and fully formalizing context - echo some of the major issues that have challenged AI researchers over the history of their field. ([Leahu08])"

En fonction des approches de l'IA, ces problèmes apparaissent sous différentes formes et sont plus ou moins abordés, mais ils reflètent les questions ouvertes de la discipline concernant la compréhension des mécanismes de l'intelligence réelle. Une autre façon de le présenter, est de dire que les problématiques de l'AmI induisent un problème *IA-complet*, comme le pose [Leahu08] :

"The problems of AI-completeness became particularly obvious and challenging in the area of autonomous agents, i.e. software and robotic systems that are intended to correspond to a complete (though perhaps simplistic) animal, person, or character. By necessity, agents require functional integration. In light of the growing complexities faced by tackling AI-complete problems head on, a number of alternative approaches arose in this area to work around issues pertaining to AI-completeness, such as behaviorbased AI, situated action, and believable agents. [...] these researchers saw themselves as providing concrete, technically feasible approaches for supporting real-time, intelligent interaction with a changing environment, we will refer to these approaches as interactionist AI. Interactionist AI questions the assumptions of generality, rationality, and complete and correct world models underlying classical AI, focusing instead on embedding autonomous systems within specific, real-world contexts."([Leahu08])

C'est pourquoi nous nous intéressons dans la suite de ce chapitre à différents paradigmes de l'IA dans la lignée de ce que [Leahu08] nomme *Interactionist AI* et aux techniques d'apprentissage potentiellement intéressantes pour se confronter à ces problèmes.

1.2.2 La représentation et le problème de l'ancrage des symboles

Une formulation possible des problèmes mentionnés précédemment, est le problème de *l'ancrage des symboles* (*Symbol grounding problem*) explicité notamment par Searle avec l'expérience de la chambre chinoise [Searle80]. Ce problème expose *l'absence de signification* de la représentation pour un système artificiel et donc la raison de son incapacité à comprendre ce qu'il fait et à agir intelligemment.

Considérons le système suivant : un homme est isolé à l'intérieur d'une chambre close. Il dispose seulement d'un livre d'instructions, d'un morceau de papier et d'un stylo. De l'extérieur on lui fait parvenir un texte écrit en chinois associé à une question qui porte sur le texte, auxquels il ne comprend pas le moindre mot car il ne parle pas le chinois. Cependant, appliquant les instructions décrites par le livre de façon mécanique, l'homme est capable de produire une réponse en chinois sur un papier qu'il retourne à l'extérieur de la chambre. Si un Chinois à l'extérieur lit la réponse, il découvre qu'elle est juste et peut éventuellement en conclure que le système de la chambre chinoise est intelligent. L'analogie avec l'IA est intéressante : l'homme représente le processeur qui exécute les calculs, le livre d'instructions représente le programme, le texte et la question sont les données en entrée, et la réponse représente les données en sortie. Ayant répondu à la question, peut-on dire que le système dans sa globalité comprend le chinois ? Est-ce qu'il comprend le texte ? Il est clair que l'intelligence permettant de résoudre le problème ne se situe dans aucune des parties du système indépendamment. Si intelligence il y a, il est plus cohérent de penser qu'elle émerge de la synergie entre les composantes du système. Cependant on peut également considérer que l'intelligence ne réside qu'en amont du système au sein des personnes qui l'ont créé, le système n'étant en fait qu'un intermédiaire entre l'intelligence d'un humain concepteur et un problème particulier. Si un nouveau texte est fourni au système contenant un vocabulaire plus récent qui n'existait pas au moment auquel le livre d'instruction a été écrit, alors le système sera incapable de donner une réponse correcte, par exemple en déduisant le sens des nouveaux mots à partir du contexte de la phrase. Cette faculté d'adaptation est au cœur de la notion d'intelligence et ceci incite à penser qu'une des propriétés fondamentale de l'intelligence est l'apprentissage.

Les symboles manipulés par un programme d'IA étant dénués de signification pour le programme qui les manipule, on peut difficilement dire que le programme comprend ce qu'il fait. Ainsi on aurait beau pousser toujours plus loin la complexité des algorithmes à représentation symbolique reposant sur des approches computationnelles de la cognition, le résultat ne serait toujours qu'un automate sophistiqué incapable d'une "réelle" intelligence, au sens où il exécute seulement une procédure dénuée de sens. La signification, l'intention et la conscience sont ici les concepts au cœur de cette remarque, et suscitent de nombreux débats philosophiques. On pourrait définir intuitivement la signification d'un symbole en disant que ce symbole est indissociable d'une quantité de relations qu'il entretient avec d'autres symboles de la représentation, et que tous ces symboles sont directement connectés à l'expérience de l'agent. En d'autres termes la représentation existe par rapport à des objectifs ou des intentions préexistantes qui la génèrent.

D'autres problèmes que l'on peut trouver mentionnés dans la littérature sont des expressions différentes de ce même problème. Par exemple le problème de la "*connaissance de fond*" (*commonsense knowledge problem*), qui désigne la lacune propre aux systèmes d'IA d'une quantité de connaissances évidentes qui devraient en théorie aller de pair avec leurs connaissances ou facultés cognitives plus avancées. Un exemple amusant

est présenté par [Dreyfus92], concernant un programme d'IA à base de *scripts* pouvant dialoguer à propos d'une situation (sorte de test de Turing). Le programme en question était capable de répondre avec succès à des questions concernant une scène se déroulant dans un restaurant, en utilisant des raisonnements logiques. Cependant, comme le remarque Dreyfus, le programme aurait été incapable de répondre à des questions triviales telles que : Est-ce que la serveuse portait des habits ? Est-ce qu'elle marchait en avant ou en arrière ? Est-ce que les clients mangeaient avec leur bouche ou avec leurs oreilles ? L'agent ne possède pas une représentation générée par lui et ancrée dans son expérience sensorimotrice, seulement des symboles qui ne peuvent avoir de signification. Tant que l'agent est confronté à des situations prévues, le comportement qu'il exhibe peut paraître sensé et les symboles appropriés, mais à la moindre déviation tout l'édifice s'effondre.

Ces critiques sont principalement adressées au paradigme cognitiviste qui conçoit la cognition comme simple processus de traitement de l'information. Cette vision s'inscrit dans l'épistémologie réaliste c'est à dire la croyance en un réel ontologique indépendant de l'expérience humaine, comme l'illustre la Figure 1.3.

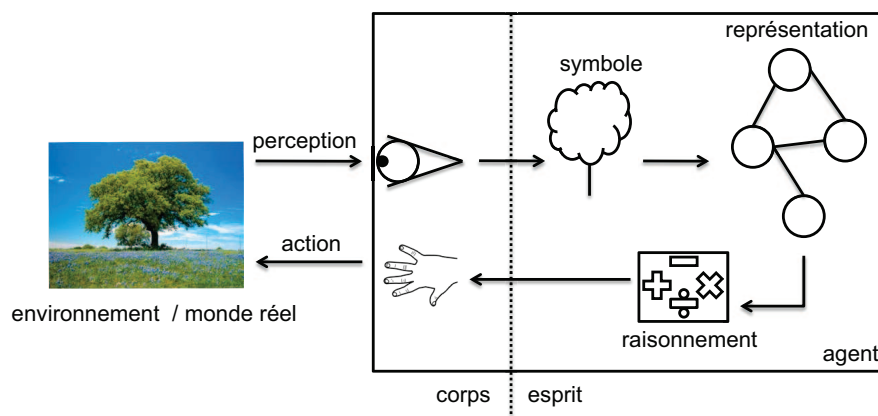


FIGURE 1.3 – Illustration du modèle cognitiviste

Les modèles cognitivistes reposent généralement sur une représentation prédéfinie, c'est à dire une spécification par le concepteur des connaissances du système. C'est pourquoi on parle de *connaissances expertes*, ou de *système expert* pour désigner le fait que la compétence du système est une modélisation des compétences d'un expert. Il est évident que plus le problème est complexe, plus cette modélisation devient difficile. *L'ingénierie des connaissances (knowledge engineering)* est la discipline qui étudie les méthodes permettant de créer de tels systèmes.

La notion de *représentation* est largement employée pour désigner des concepts différents suivant le paradigme dans lequel on se place en sciences cognitives et en IA. [Dorffner97] en distingue trois principaux types d'usage :

- Usage "Classique" : Un encodage spécifique des structures du monde réel, la connaissance étant vue comme un "miroir" de la réalité (usage classique en IA cognitive).
- Définition interactive : L'ensemble des structures internes qui conditionnent le comportement de l'agent (compatible avec le point de vue constructiviste d'une représentation construite, subjective).
- Signification la plus générale : Englobe les relations causales et les correspondances. Par exemple le fait qu'un stimulus lumineux entraîne une réponse neuronale dans la rétine est alors considéré comme une forme de représentation.

1.2.3 L'IA incarnée (*Embodied AI*)

La question de l'ancrage de la représentation a été largement commentée par Brooks dont la position est que la capacité d'abstraction du concepteur (en tant qu'humain) incite à croire que l'intelligence passe par la représentation abstraite de l'environnement en des symboles manipulables indépendamment pour raisonner [Brooks91]. Il remarque qu'il est simple de trouver des algorithmes de résolution efficaces lorsque l'on fournit par un travail intellectuel humain préalable la représentation formelle abstraite et utile à la machine qui ainsi n'a plus qu'à explorer un espace de recherche. En se concentrant sur ce genre de problème, il affirme que l'IA « classique » aurait esquivé la partie "difficile" du problème, parce que c'est le processus même d'abstraction qui est la clef de l'intelligence, et la représentation symbolique semble en ce sens être un obstacle à la construction d'un système intelligent :

"I claim that AI researchers are guilty of the same (self) deception. They partition the problems they work on into two components. The AI component, which they solve, and the non-AI component which, they don't solve. Typically, AI "succeeds" by defining the parts of the problem that are unsolved as not AI. The principal mechanism for this partitioning is abstraction. Its application is usually considered part of good science, not, as it is in fact used in AI, as a mechanism for self-delusion. In AI, abstraction is usually used to factor out all aspects of perception and motor skills. I argue below that these are the hard problems solved by intelligent systems, and further that the shape of solutions to these problems constrains greatly the correct solutions of the small pieces of intelligence which remain. ([Brooks91])"

Brooks réfute l'utilisation d'une représentation (au sens classique) et essaie de créer des robots purement réactifs grâce à des comportements simples qui se combinent. L'idée de Brooks est de décomposer le problème en sous-systèmes producteurs d'activité ou comportements (layers) qui connectent directement perception et action sans passer par une représentation intermédiaire. Les niveaux sont organisés en une *architecture de subsomption* par niveaux d'abstraction, activables en parallèle suivant différents critères.

Par exemple : un robot simple peut avoir comme niveaux par ordre croissant d'abstraction :

- 1) Eviter obstacles
- 2) Se balader
- 3) Explorer zones inconnues.

C'est une technique intéressante du point de vue de la conception de contrôleur de robot, car cela permet par l'arbitrage entre des comportements élémentaires, de créer un comportement général robuste qui paraît naturel. Ce type d'approche nécessite cependant un codage préalable des comportements de base, ce qui implique d'avoir une connaissance parfaite des capacités du robot. D'autre part, cette approche est limitée du point de vue de l'apprentissage, notamment par rapport au problème de l'abstraction.

De manière plus générale l'IA incarnée est soutenue par le paradigme de la cognition incarnée en sciences cognitive ([Varela93]) qui replace le corps et l'action au cœur de la cognition. De nombreuses expériences en psychologie cognitive illustrent cela. Par exemple les expériences de [Held63] ont montré l'importance de l'action pour la vision en exhibant la différence de développement entre deux groupes de chatons confrontés aux mêmes perceptions visuelles, l'un d'eux étant cependant actif et l'autre passif. Les chatons sont tous placés peu après leur naissance dans le noir complet. Ils sortent dans un environnement contrôlé pendant des sessions d'entraînement d'une heure. Durant ces sorties, certains chatons (pilotes) peuvent se déplacer dans l'environnement. Ils tirent derrière eux grâce à une ficelle une remorque miniature dans laquelle est placé un autre chaton immobile (passager). Au bout de quelques jours les chatons sont libérés dans le monde extérieur. Les résultats montrent que les chatons pilotes ont pu s'adapter normalement, mais pas les chatons passagers qui ne réagissaient pas normalement aux stimulus visuels, comme s'ils étaient aveugles. Même s'ils ont pu voir tout ce que les pilotes ont vu durant les phases de développement communes, le fait qu'ils soient inactifs ne leur a pas permis de construire un comportement sensé ou utile guidé par la vision. Un autre exemple est l'expérience proposée par [Strack88] qui cette fois concerne les humains. Cette expérience analyse l'impact de certaines positions corporelles sur le ressenti des sujets face à une même situation. En l'occurrence la situation est la lecture de bandes dessinées que les sujets doivent évaluer ensuite comme plus ou moins amusantes. Trois positions sont possibles en fonction de la façon dont le sujet tient un stylo : "dans la main" (position neutre), "entre les dents et sans toucher les lèvres" (position souriante) et enfin "entre les lèvres et sans toucher les dents" (position triste). Les résultats des évaluations montrent clairement que l'évaluation est la meilleure en moyenne dans la position souriante, et la moins bonne en position triste.

1.2.4 L'apprentissage sensorimoteur

Le problème difficile de l'intelligence artificielle dépendrait donc des aspects liés à la perception et aux fonctions motrices, c'est-à-dire le traitement de l'expérience de l'agent au plus bas niveau. Cette idée est également formulée par le fameux *paradoxe de Moravec*, qui indique que les tâches de bas niveau de la cognition sont les plus difficiles à reproduire artificiellement, alors que les tâches de haut niveau sont aisément reproduites. Les tâches sensorimotrices consistent en des facultés de la cognition réalisées de manière quasi inconsciente : coordination des gestes, reconnaissance de forme (distinction entre fond et objet), gestion de l'attention, repérage spatial, etc. ; par opposition les tâches de haut niveau sont : raisonnement, planification, etc. Un argument possible consiste à expliquer ce paradoxe par la théorie de l'évolution : les facultés de bas niveau sont en partie issues d'un long processus d'évolution réalisé sur des millions d'années et sont fortement couplées au corps biologique. Ce processus d'évolution peut ainsi être assimilé à une forme d'apprentissage à l'échelle phylogénétique. D'autres problèmes bien connus sont directement liés à cette difficulté de reproduire les capacités sensorimotrices dans les domaines complexes et continus : le problème du cadre (*Frame Problem*) ([Murray Shanahan09]), l'aliasing perceptuel (*Perceptual aliasing*) ([Shani05]), perception catégorielle (*Categorical perception*) ([Harnad03]), dérive conceptuelle (*Concept Drift*) ([Tsymbal04]), etc. Tous ces problèmes sont susceptibles d'être rencontrés dans un effort de création d'un système d'AmI si l'on crée un système doté d'une représentation élaborée déconnectée de l'expérience de l'agent.

L'intérêt pour un système artificiel d'avoir une représentation générée à partir de ses propres capacités sensorimotrices, est que cette représentation est adaptée au système. En tant que concepteur humain, on possède une vision du monde qui nous est propre, pouvant être incompatible avec les moyens de percevoir et d'agir du système, risquant de compromettre son efficacité. Cette idée est présente en biologie si l'on considère les différences entre les espèces. Selon [Von Uexküll09] chaque organisme vit dans un *Umwelt* ("environnement sensoriel" ou "monde propre") qui lui est spécifique, dépendant de ses capacités d'interaction avec son environnement, comme illustré par la Figure 1.4. L'*Umwelt* d'une autre espèce nous est ainsi par définition inaccessible, comme l'explique [Nagel74]. Par exemple le monde dans lequel vit une chauve-souris utilisant l'écholocation est certainement radicalement différent de celui de l'homme. Il est clair que le comportement de l'organisme va dépendre de ce *Umwelt*, c'est pourquoi il est parfois difficile de comprendre le comportement de certains animaux d'un point de vue humain. Dans le cas des systèmes artificiels une problématique analogue intervient. On ne peut que supposer ce qu'un robot est capable d'apprendre ou non avec les capacités qu'il possède et les comportements qu'il est capable d'effectuer. Par exemple, dans le cas d'un système ambiant un capteur de CO2 est généralement installé pour mesurer la qualité de l'air. C'est en tout cas la raison pour laquelle ce matériel a été créé, mais pour le système il s'agit d'une source d'information sur son environnement qui n'aura de sens que si son exploitation participe à une forme de représentation utile

au système. Peut-être que certaines variations de CO2 bien identifiées associées à des variations d'autres modalités (ex. température, infrarouges, etc.) peuvent permettre au système de distinguer la présence d'un humain, ou celle d'un animal, ou de détecter le nombre approximatif de personnes présentes, ou l'activité en cours dans la pièce.

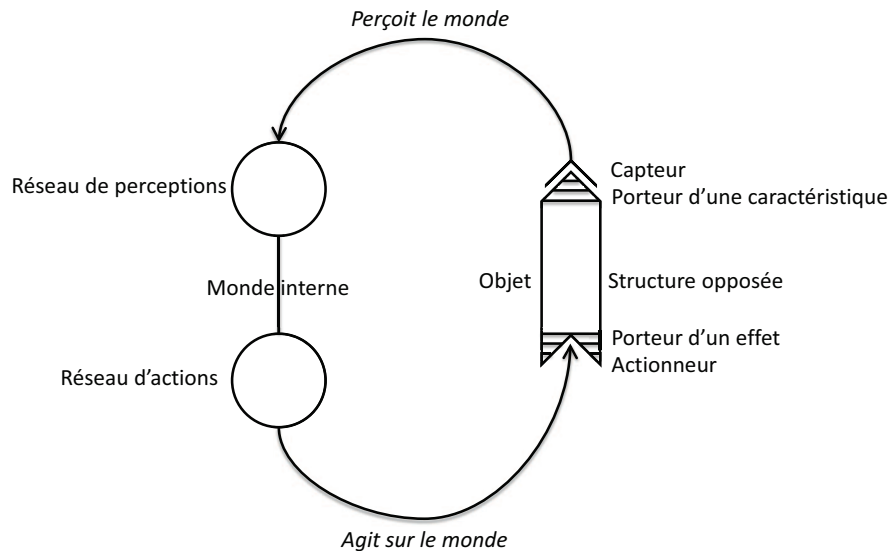


FIGURE 1.4 – Illustration du concept de *Umwelt* (monde interne)

On utilise souvent le terme "*motifs*" (*patterns*) pour désigner la structure qui résulte de l'activité d'apprentissage de régularités. Un motif peut désigner toute structure représentationnelle mettant en relation différentes variables dans le temps. Le choix du mot "motif" est particulièrement intéressant. Etymologiquement, ce mot vient du latin *motivus* : "Mobile, relatif au mouvement". Cela illustre l'idée qu'un motif n'est pas un état figé, mais l'expression d'un changement. D'autre part une définition du mot est : "Ce qui pousse à agir ; Raison d'une action", ce qui suggère qu'un motif est défini en relation avec son rôle par rapport à une action ou un objectif. Enfin, un motif est également un "élément graphique répété sur un tissu ou un papier". C'est à dire un phénomène récurrent dans le temps, lorsque l'on parcourt du regard le tissu. En effet le "motif" en tant qu'élément de représentation de la connaissance apprise dans le cadre d'un système d'IA est "morceau" d'expérience récurrent, qui est identifié au cours du temps parce qu'il a une utilité particulière relativement à un objectif du système ou de l'utilisateur.

1.2.5 Approches postcognitivistes et positionnement épistémologique

Il existe des approches que l'on peut qualifier de postcognitivistes à l'image de l'IA incarnée ([Brooks98]) présentée précédemment : l'IA énative ([McGee05, McGee06]) et

[De Loor09]), l'apprentissage constructiviste ([Drescher91]), ou la robotique développementale ([Meeden06]), qui tentent de différentes façons de dépasser la conception de la cognition comme simple processus de traitement de l'information issue du cognitivisme. Les approches post-cognitivistes s'en distinguent en considérant plutôt la cognition comme une organisation émergeant du couplage de l'agent et de son environnement. On remarque que dans ces approches, l'apprentissage n'est pas une composante de l'IA envisagée à part, mais qu'IA et apprentissage se confondent. Ce qui relie également ces approches c'est leur compatibilité avec la position épistémologique du *constructivisme radical* [Von Glasersfeld84].

"Une telle radicalisation témoigne d'une rupture avec la notion traditionnelle selon laquelle toute connaissance humaine devrait ou pourrait s'approcher d'une représentation plus ou moins «vraie» d'une réalité indépendante ou «ontologique». À la place de cette notion de représentation, le constructivisme radical introduit une nouvelle relation, plus tangible, entre connaissance et réalité, relation que j'ai appelée «viabilité». Pour expliquer la notion de viabilité, disons simplement qu'on jugera «viable» une action, une opération, une structure conceptuelle ou même une théorie tant et aussi longtemps qu'elles servent à l'accomplissement d'une tâche ou encore à l'atteinte du but que l'on a choisi. Ainsi, au lieu de prétendre que la connaissance puisse représenter un monde au-delà de notre expérience, toute connaissance sera considérée comme un outil dans le domaine de l'expérience." ([Von Glasersfeld84])

La Figure 1.5 illustre le modèle constructiviste de la cognition en opposition à la Figure 1.3.

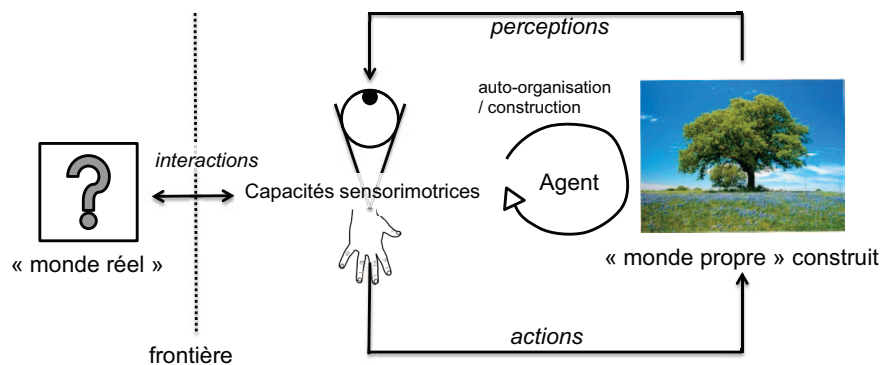


FIGURE 1.5 – Illustration du modèle constructiviste

[Riegler01] et [Stewart01] expliquent respectivement pourquoi une position épistémologique constructiviste peut avoir une importance significative pour la science,

et pour les sciences cognitives en particulier. D'autres sources d'influence communes aux approches post-cognitivistes sont directement liées au constructivisme épistémologique dans les différents travaux cités précédemment : la cybernétique, la notion d'*Umwelt* de [Von Uexküll09], la théorie de l'*autopoïèse* et de l'*énaction* ([Varela92]), ainsi que la *théorie constructiviste de l'apprentissage* ([Piaget54]). D'autre part, ces différents paradigmes possèdent des points communs avec un autre type d'approche que nous nommons "neuro-inspirée" consistant à s'inspirer directement du fonctionnement du cerveau humain en vue de construire de nouveaux systèmes d'IA, dont l'apprentissage est bien entendu une composante essentielle⁴. Ces approches connexionnistes peuvent être une réalisation possible d'un système constructiviste comme proposé par [Dorffner97].

1.2.6 Les approches constructivistes en IA

En psychologie, la théorie constructiviste de l'apprentissage développée en particulier par [Piaget54] 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 [Drescher91] 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 : [Guerin11]). Dans ces modèles, la brique élémentaire de représentation (le motif) 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 [Perotto07]. 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. 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 [Chaput03] qui proposent d'utiliser des cartes auto-organisatrices (SOM) pour améliorer l'apprentissage de schémas.

4. Nous proposons en annexe A.1, de donner un aperçu sous forme d'arbre, des principaux concepts et paradigmes introduit dans cette thèse.

Le problème principal auquel on est confronté en appliquant des méthodes d'apprentissage constructivistes en environnements continus est le *problème d'amorçage*. Ce problème est associé à la dynamique entre le processus de discrétisation de l'expérience et l'apprentissage de motifs sensorimoteurs. 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 [Kuipers06] qui réside dans l'apprentissage des motifs sensorimoteurs à partir des données brutes de capteurs inconnus. Nous concentrons particulièrement notre attention sur ce problème qui est présenté plus précisément dans la partie 2.4.7. Pour les organismes biologiques, ce double apprentissage est probablement réalisé 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.

L'approche constructiviste de l'apprentissage constitue une piste intéressante pour la conception d'agents autonomes, fortement adaptés à leur environnement. Le domaine de *robotique développementale* (voir [Lungarella03]) nous intéresse donc particulièrement puisqu'il s'inscrit tout à fait dans cette approche, et doit gérer les environnements réels. Dans cette approche le concepteur ne définit pas une tâche particulière mais le robot doit plutôt explorer librement son environnement. Nous présentons plus en détails les approches constructivistes et la robotique développementale ainsi que des travaux représentatifs dans le chapitre suivant (chapitre 2).

1.3 Positionnement et choix théoriques

1.3.1 L'approche constructiviste dans le domaine de l'AmI

Nous avons donc montré que l'apprentissage sera une composante essentielle du domaine de l'AmI. Tout d'abord parce qu'un système d'AmI est évolutif et situé dans un environnement imprévisible. Mais aussi parce que ce genre de système doit pouvoir apprendre à reconnaître des situations ou des activités qui sont d'un niveau d'abstraction élevé. Une des particularités des systèmes d'AmI est que leur composition et leurs objectifs sont à priori indéterminés car ils dépendent des choix des utilisateurs, et demeurent susceptibles d'évoluer. Il s'agit donc de développer un mécanisme d'apprentissage total d'un système artificiel physique indéterminé (ex. pour un robot en général sans rien supposer de son corps) dans un environnement complexe (sans rien supposer des propriétés de l'environnement). Nous avons introduit le concept d'IA incarnée et l'idée d'apprentissage constructiviste d'une représentation ancrée dans l'expérience sensorimotrice pour les systèmes cognitifs. Autrement dit, un système artificiel doit apprendre uniquement par l'expérience à interagir dans son environnement, à l'image d'un organisme biologique depuis sa naissance à l'âge adulte (Apprentissage au sens

large, comme faculté, ou propriété de l'intelligence). Nous avons donc passé en revue différentes approches de l'IA allant dans cette direction. Dans le domaine applicatif, la *robotique développementale* offre des avancées importantes dans ce domaine, comme nous le montrons dans la partie qui y est consacrée (section 2.4.5) du chapitre suivant. Nous pensons qu'à l'instar de la robotique, une approche de ce type est pertinente et prometteuse dans le cadre de l'AmI, comme argumenté par [Leahu08] et [Najjar13].

L'objectif général du modèle proposé est donc d'appliquer les idées des approches constructivistes et développementales de l'apprentissage artificiel au problème de l'intelligence ambiante. L'intérêt à long terme est d'obtenir un environnement intelligent, qui a appris les propriétés du monde et ses capacités d'interactions avec lui par sa propre expérience. Le but étant que le système soit ainsi capable d'atteindre un niveau d'adaptation suffisant pour satisfaire les ambitions du paradigme de l'AmI dans son sens plein. A court terme, l'apprentissage des motifs sensorimoteurs primaires et secondaires pourra permettre d'apporter une solution à des problèmes concrets rencontrés dans le cadre applicatif industriel, par l'identification automatique d'évènements pertinents, et la prédiction d'évènements par rapport à d'autres. La principale difficulté va être d'adapter à un environnement complexe des méthodes jusqu'à maintenant assez théoriques et pouvant devenir très coûteuses en temps de calcul et en occupation d'espace mémoire.

1.3.2 Principes généraux du modèle

Les modèles d'apprentissage constructiviste proposés dans la littérature sont variés comme nous le verrons dans la section 2.4. Ils utilisent des techniques algorithmiques et des structures représentationnelles différentes mais des principes fondamentaux communs existent, que l'on retrouve dans d'autres domaines telles que celui des approches neuro-inspirées (voir 2.3). Dans ce travail nous n'essayons pas d'appliquer ou d'adapter l'une de ces techniques en particulier, mais plutôt de créer une approche nouvelle adaptée à nos besoins, partageant ces mêmes principes communs fondamentaux.

L'approche développementale et constructiviste démarre par l'étape de l'apprentissage sensorimoteur qui permet de construire les structures représentationnelles de base sur lesquelles s'appuie la construction d'une représentation plus évoluée. Cet apprentissage suppose deux processus complémentaires et interdépendants qui interviennent dans le *problème d'amorçage*.

- Un processus de discrétisation de l'expérience
- Un processus d'apprentissage prédictif

Dans ce modèle nous abordons cette problématique sous un angle particulier : nous proposons une *modélisation systémique* du problème et une *méthode de résolution décentralisée*.

1.3.2.1 Modélisation systémique

On peut définir un programme ou un *algorithme* comme :

"un processus systématique de résolution, par le calcul, d'un problème permettant de présenter les étapes vers le résultat à une autre personne physique (un autre humain) ou virtuelle (un calculateur). En d'autres termes, un algorithme est un énoncé d'une suite d'opérations permettant de donner la réponse à un problème. ([Marc Lavielle09])"

Le phénomène cognitif tel que nous l'avons présenté jusqu'ici est un système composé de différents processus, pouvant être individuellement modélisé par des algorithmes. Un système peut difficilement être réduit à un algorithme car on ne peut l'énoncer comme une suite d'opérations déterministes. Une modélisation systémique vise à décrire un système par sa structure et son organisation, sa frontière, c'est à dire énoncer les composants essentiels du système, leurs rôles et leurs échanges d'informations. Cela permet notamment de laisser ouverts les moyens d'implémenter chacune des composantes et opérations du système. Plutôt que de choisir un algorithme ou une technique d'apprentissage et d'essayer de l'adapter au problème ou de circonscrire le problème pour l'adapter à l'algorithme, il s'agit de décrire une solution à haut niveau. Cela permet ensuite de choisir les techniques et algorithmes les plus adaptés pour chaque composante. Cela permet de plus d'envisager l'exécution de plusieurs techniques pour une même opération et aussi de les faire travailler en parallèle pour participer à différents objectifs.

Minsky regrette que l'intelligence soit envisagée comme un phénomène régi par quelques lois précises qu'il s'agirait de découvrir, à l'instar de la physique. Il prône plutôt la création de systèmes complexes, utilisant diverses techniques à différents niveaux :

"The secret of what something means lies in how it connects to other things we know. That's why it's almost always wrong to seek the real meaning of anything. A thing with just one meaning has scarcely any meaning at all. To get around these limitations, we must develop systems that combine the expressiveness and procedural versatility of symbolic systems with the fuzziness and adaptiveness of connectionist representations. Why has there been so little work on synthesizing these techniques? I suspect that it is because both of these AI communities suffer from a common cultural-philosophical disposition : They would like to explain intelligence in the image of what was successful in physics by minimizing the amount and variety of its assumptions. But this seems to be a wrong ideal. We should take our cue from biology rather than physics because what we call thinking does not directly emerge from a few fundamental principles of wave-function symmetry and exclusion rules. Mental activities are not the sort of unitary or elementary phenomenon that can be described by a few mathe-

mathematical operations on logical axioms. Instead, the functions performed by the brain are the products of the work of thousands of different, specialized subsystems, the intricate product of hundreds of millions of years of biological evolution. We cannot hope to understand such an organization by emulating the techniques of those particle physicists who search for the simplest possible unifying conceptions. Constructing a mind is simply a different kind of problem : how to synthesize organizational systems that can support a large enough diversity of different schemes yet enable them to work together to exploit one another's abilities ([Minsky91])."

Nous proposons tout d'abord de distinguer les opérations élémentaires génériques qui sont nécessaires à la discrétisation de l'expérience continue pour construction de motifs prédictifs. Ces opérations sont systématiquement présentes dans les modèles d'apprentissage souvent implicitement et avec une unique façon de les réaliser qui est choisie à priori. Ensuite nous analysons les moyens de coordonner ces opérations de manière à obtenir un système dont l'organisation équivaut à la construction de la représentation souhaitée. Cette organisation passe par la création de liens entre les composantes du système, des mécanismes de renforcement et de rétroaction. Enfin nous proposons une architecture permettant de concrétiser ce modèle.

1.3.2.2 Approche décentralisée

Comme nous le décrirons dans l'état de l'art le cerveau semble fonctionner de manière décentralisée. Le choix d'une modélisation systémique est intrinsèquement lié à une conception décentralisée. C'est l'interaction de différents processus qui permet au système de réaliser la tâche désirée, en l'occurrence ici l'apprentissage constructiviste et la résolution du *problème d'amorçage*. Le choix de la décentralisation est une contrainte de conception forte mais permet aussi beaucoup de souplesse et ouvre des possibilités différentes, comme nous le verrons en présentant l'*intelligence artificielle distribuée* et les *systèmes multi-agents* dans la partie 2.2 du chapitre suivant. Notamment, la parallélisation des traitements peut être intéressante à l'instar du fonctionnement du cerveau. Ainsi, concernant la gestion du flux de données provenant du matériel, [Kuipers05] indique que le traitement doit être simple, local et décentralisé :

"Robot senses are nowhere near as rich as human senses, but they still provide information at an overwhelming rate. (A stereo pair of color cameras alone generates data at over 440 megabits per second.) *With such a high data rate, any processing applied to the entire sensor stream must be simple, local, and parallel.* In the human brain, arriving sensory information is stored in some form of short-term memory, remains available for a short time, and then is replaced by newly arriving information." ([Kuipers05])

Même si des techniques connexionnistes sont souvent employées pour l'étape de discrétisation dans les modèles constructivistes (ex. SOM), les modèles globaux d'appren-

tissage constructiviste décentralisés sont peu fréquents. Il nous semble aussi intéressant d'explorer cette piste pour cette raison.

Un système multi-agent peut permettre de modéliser les différentes opérations du modèle grâce à des rôles exécutés par des agents. La structure du système est l'ensemble des agents et des éléments de représentation. L'organisation est l'ensemble des interactions entre ces composantes. Ce modèle autorise la création et le traitement parallèle d'une variété d'agents différents pour implémenter les différentes opérations spécifiées, à l'image du concept de « *société de l'esprit* » développé par [Minsky91].

"The system could consist of localized clumps of expertise. At the lowest levels, these clumps would have to be densely connected to support the sort of associativity required to learn low-level pattern-detecting agents. However, as we ascend to higher levels, the individual signals must become increasingly abstract and significant, and accordingly, the density of connection paths between agencies can become increasingly (but only relatively) smaller.([Minsky91])"

Dans un domaine extrêmement hétérogène tel que celui de l'AmI, cette vision possède l'avantage considérable que l'on peut définir un agent particulier adapté à la nature d'une variable ou d'un concept particulier à identifier. Un agent d'un certain type peut être générique et paramétrable. On peut envisager qu'un tel agent soit créé de manière experte, ou bien généré automatiquement par le système pour exploration ou sur la base d'un apprentissage dédié. Par exemple, plusieurs agents paramétrés différemment peuvent être associés à une même variable en parallèle, afin de prendre en compte plusieurs échelles de temps pour détecter des événements. Enfin, un modèle modulaire tel que celui-ci possède l'avantage de permettre un ajustement facile du système pour assurer des contraintes industrielles ou encore pour tester différents aspects indépendamment. Par exemple, on peut implémenter des agents un peu plus "experts" pour faciliter l'apprentissage de certains motifs ciblés bien connus.

2

Etat de l'art

▷ Le chapitre précédent a introduit le sujet de cette thèse, son contexte industriel et les problématiques scientifiques associées. Dans ce chapitre, nous proposons un état de l'art qui s'articule en quatre parties. Dans la première partie nous présentons le domaine de l'apprentissage artificiel et ses principales techniques. La deuxième partie introduit l'intelligence artificielle décentralisée, fait la présentation des systèmes multi-agents, et aborde la question de l'apprentissage dans les systèmes multi-agents. Dans la troisième partie nous abordons les approches neuro-inspirées de la cognition en IA, qui proposent une vision décentralisée et basée sur l'apprentissage directement inspiré du fonctionnement du cerveau. Enfin, la quatrième partie présente les travaux liés à l'approche constructiviste de l'apprentissage en IA, et notamment la robotique développementale qui est son application directe aux systèmes artificiels en interaction avec des environnements continus complexes. Dans chacune des quatre parties, nous présentons des travaux existants en lien avec l'intelligence ambiante. ◁

Plan du chapitre

2.1	L'apprentissage artificiel	29
2.1.1	Introduction	29
2.1.2	L'apprentissage supervisé	31
2.1.3	L'apprentissage par récompense	32
2.1.4	L'apprentissage non-supervisé	36
2.1.5	Commentaires sur l'IA incarnée et l'apprentissage	37
2.1.6	L'apprentissage dans le cadre de l'AmI	39
2.2	Les systèmes multi-agents	43
2.2.1	Présentation des SMA	43
2.2.2	L'apprentissage et les systèmes multi-agents	45
2.2.3	Les systèmes multi-agents dans le cadre de l'AmI	48
2.3	Les approches neuro-inspirées	52
2.3.1	Introduction	52
2.3.2	Memory Prediction Framework	53
2.3.3	Approches reposant sur l'utilisation de cartes corticales	56
2.3.4	Conclusion	57
2.3.5	Les approches connexionnistes dans le cadre de l'AmI	58
2.4	Le paradigme constructiviste	60
2.4.1	L'intelligence comme adaptation	60
2.4.2	Application en IA	62
2.4.3	L'apprentissage sensorimoteur	64
2.4.4	Mécanismes d'apprentissage de schémas	65
2.4.5	La robotique développementale	67
2.4.6	Applications des approches constructiviste ou développe- mentales aux environnements continus	70
2.4.7	Le problème d'amorçage	72
2.4.8	Les approches constructivistes dans le cadre de l'AmI	77

2.1 L'apprentissage artificiel

Dans cette partie nous proposons d'analyser la notion d'apprentissage artificiel, de passer en revue les principales techniques existantes puis d'envisager leur lien avec un positionnement constructiviste. Enfin, nous présentons quelques exemples d'application de l'apprentissage artificiel au domaine de l'AmI.

2.1.1 Introduction

Comme nous l'avons vu, la capacité d'apprentissage devient une composante essentielle des systèmes d'IA évolués. La notion d'apprentissage est intimement liée à la notion d'intelligence. Peut-on qualifier d'intelligent un système qui ne sait qu'exécuter parfaitement une tâche pour laquelle il a été programmé sans pouvoir s'adapter à un changement dans son environnement ? Ou bien encore sans pouvoir exécuter une nouvelle tâche similaire en profitant de l'expérience acquise au cours de sa précédente activité ? Dans le cas de l'AmI, l'apprentissage est essentiel de par la nature hétérogène du problème : chaque système est doté de capacités propres (un certain nombre de capteurs et d'actionneurs de différentes natures) et d'objectifs divers (préférences des utilisateurs) ainsi que d'un environnement particulier (environnement physique (propriétés du bâtiment, zone géographique, habitudes des utilisateurs)). Le système intelligent doit donc être capable d'apprendre les caractéristiques de son environnement ainsi que ses capacités d'interaction avec celui-ci, car elles ne peuvent être programmées à l'avance efficacement.

Définition

L'apprentissage en général (pour un organisme biologique, en particulier l'humain) consiste à « acquérir ou à modifier sa *représentation d'un environnement* de façon à permettre avec celui-ci des interactions efficaces ou de plus en plus efficaces ». Dans le cadre de l'IA, la notion d'apprentissage est bien plus vaste, puisque l'apprentissage regroupe « toutes les techniques par lesquelles un programme est capable d'évoluer au cours du temps pour résoudre des problèmes complexes ». La relation entre apprentissage et intelligence est extrêmement forte, comme l'exprime [Sen99] :

"Learning and intelligence are intimately related to each other. It is usually agreed that a system capable of learning deserves to be called intelligent; and conversely, a system being considered as intelligent is, among other things, usually expected to be able to learn. Learning always has to do with the self-improvement of future behavior based on past experience. More precisely, according to the standard artificial intelligence (AI) point of view learning can be informally defined as follows : The acquisition of new knowledge and motor and cognitive skills and the incorporation of the acquired knowledge and skills in future system activities, provided that this acqui-

sition and incorporation is conducted by the system itself and leads to an improvement in its performance." ([Sen99])

Si un programme évolue pour maintenir son niveau de performance dans la réalisation de ses objectifs, pendant que l'environnement change, on parle plutôt d'*adaptation*. Si par contre le programme augmente sa performance, ou devient capable d'atteindre de nouveaux objectifs, on parle alors d'*apprentissage*. On suppose que pour être parvenu à ce résultat, la représentation de l'agent a qualitativement changée. Dans les deux cas, l'évolution que connaît ce programme est interprétée comme un gain de performance par un observateur extérieur, relativement à un ou des objectifs. C'est cette évolution que l'on nomme "processus d'apprentissage" par analogie avec la notion d'apprentissage des êtres vivants. Cependant les méthodes employées peuvent être radicalement différentes des mécanismes en œuvre chez les êtres vivants.

Différents types d'apprentissage artificiel

Les caractéristiques suivantes ([Sen99]) sont valables aussi bien pour l'apprentissage centralisé ou distribué dans les systèmes multi-agents (SMA) :

La méthode d'apprentissage : (Par effort d'apprentissage croissant)

- Mémorisation par cœur (*rote learning*) : implémentation directe du savoir et des compétences, pas d'évolution.
- Apprendre par instruction : transformation du savoir en représentation interne pour intégration au savoir préexistant.
- Apprendre par l'exemple et par entraînement : à partir d'exemples positifs ou négatifs et expérience pratique.
- Par analogie : extension d'un savoir d'un problème résolu vers un problème non résolu.
- Par découverte : faire des observations, des expériences. Construire et tester des hypothèses et théories.

Une caractéristique importante de l'apprentissage pour les systèmes artificiels est la suivante :

- *Apprentissage hors ligne* (offline) : la période d'apprentissage est préalable et indépendante à l'exécution du programme.
- *Apprentissage en ligne* (online) : le programme apprend au cours de son activité.
- *Apprentissage par lots* (batch learning) : technique intermédiaire (des phases d'apprentissages offline à intervalle de temps régulier durant l'activité)

L'apprentissage hors ligne suppose que l'on dispose d'un ensemble de données qui reflètent précisément le problème auquel sera confronté le programme durant son exécution. L'apprentissage en ligne suppose que le système fasse constamment le lien entre son contrôle et l'apprentissage en parallèle. Cela suppose de fréquentes opérations de vérifications et de mise à jour. L'apprentissage par lots est une alternative à l'apprentissage en ligne afin de réduire ce nombre d'opérations.

Comme nous l'avons mentionné précédemment, l'apprentissage artificiel repose généralement sur un signal de rétroaction (récompense, feedback, fonction de fitness, etc.) pour orienter le système par l'évaluation de son activité. La façon dont on met en place cette rétroaction peut permettre de classer les principaux types d'apprentissage que l'on rencontre en IA : (comme proposé par [Panait05]).

- *Apprentissage supervisé* : le feedback spécifie exactement l'activité désirée de l'apprenant. L'objectif de l'apprentissage est de s'y conformer le plus possible.
- *Apprentissage basé sur la récompense* : le feedback spécifie seulement l'utilité du comportement par une valeur, l'objectif étant de la maximiser. Cette catégorie peut se diviser en 2 sous-catégories :
 - *Apprentissage par renforcement* (estimation de fonctions évaluées) (ex. : Q-learning ; TD-learning)
 - *Apprentissage par recherche stochastique* (apprentissage direct de comportements) (ex. : algorithmes génétiques)
- *Apprentissage non supervisé* : Pas de feedback explicite. L'objectif est de trouver les activités utiles et désirées sur la base du tâtonnement (trial-and-error).

Dans le cas de l'apprentissage non supervisé le feedback est effectivement implicite. Il est dissimulé dans la notion d'*utilité* des activités ou des représentations apprises, ou dans la notion de conformité des structures apprises avec le modèle de régularités que l'on recherche. L'apprentissage artificiel (ou automatique) étant un domaine de recherche extrêmement vaste, il est possible de catégoriser différemment les méthodes existantes et nous n'en donnons ici qu'un modeste aperçu (voir [Cornuéjols11]). Nous nous contentons dans cet état de l'art de présenter les caractéristiques essentielles des principaux courants.

2.1.2 L'apprentissage supervisé

L'apprentissage supervisé désigne l'ensemble des techniques pour lesquelles on cherche à faire apprendre un concept au système, à partir d'une quantité suffisante d'exemples que l'on peut évaluer comme appartenant ou non à ce concept. Chaque exemple est ainsi *labellisé* de manière binaire. Lorsque le système intègre les exemples les uns à la suite des autres, il modifie sa représentation de manière à ce qu'elle soit cohérente avec le maximum d'exemples.

Une technique qui illustre bien cette manière de procéder est *l'espace des versions* avec l'algorithme d'*élimination des candidats* proposé par [Mitchell82]. Il s'agit tout d'abord de définir un espace des hypothèses, qui est un ensemble de solutions possibles pour modéliser le concept cible. On définit ensuite deux ensembles G et S . G est l'ensemble des solutions les plus générales qui sont cohérentes avec les exemples connus du concept (cohérent : tous les exemples vrais déjà observés sont modélisés par G et tous les exemples faux ne sont pas modélisés par G). S est inversement l'ensemble des solutions les plus spécifiques cohérentes avec les exemples. Alors que le système

parcourt de nouveaux exemples, des modifications sont appliquées pour ajuster G et S . Ces modifications sont faites par des règles de *généralisation* et de *spécification*. Par exemple lorsque S ne permet pas de modéliser le nouvel exemple, il faut généraliser S .

Si la solution peut être définie par l'espace des hypothèses spécifié initialement, alors l'algorithme doit converger vers cette solution au fur et à mesure que l'on élimine des solutions candidates dans les ensembles G et S . Les deux ensembles permettent d'exprimer à tout moment l'ensemble des hypothèses compatibles avec les exemples connus qui est potentiellement très grand.

Un autre exemple d'apprentissage supervisé est celui du *perceptron*. Il s'agit d'un réseau de neurone simple (*feedforward neural network*) entraîné à partir d'une base d'exemples à reconnaître un concept. Le réseau de neurone doit déterminer si chaque exemple appartient ou non au concept, et en cas d'erreur les poids des nœuds du réseau sont ajustés.

De manière générale l'apprentissage supervisé est donc utile lorsque l'on veut apprendre un ou des concepts ciblés. Il faut aussi disposer d'une base d'exemples pouvant être évalués et présentés en fonction des concepts. Si l'on fait une analogie avec le domaine biologique, cette forme d'apprentissage correspond à l'apprentissage d'un concept par un élève sous la supervision d'un professeur.

2.1.3 L'apprentissage par récompense

L'apprentissage par renforcement

L'apprentissage par renforcement est le problème auquel est confronté un agent qui doit apprendre son comportement par tâtonnement (interactions par essais et erreurs (*trial-and-error*)) dans un environnement dynamique ([Kaelbling96]).

L'agent est connecté à l'environnement par la perception et l'action. A chaque pas de l'interaction, l'agent reçoit en entrée une perception p de l'état courant, et génère une action a comme sortie qui impacte l'environnement. La valeur de cette transition est communiquée à l'agent par un signal de renforcement r (ou récompense), comme illustré par la Figure 2.1. Le comportement de l'agent (ou *politique*) doit permettre de choisir les actions qui vont maximiser la somme des valeurs des récompenses sur le long terme. La politique Π fait correspondre les états aux actions.

Formellement le modèle est représenté par le triplet : (S, A, R) avec S un ensemble fini d'états, A un ensemble fini d'actions et R une fonction de renforcement $R : S, A \rightarrow 0, 1$ par exemple.

Hypothèses :

- L'environnement est non déterministe : pour une action exécutée deux fois dans un même état, l'état résultant ou la récompense peuvent être différents.

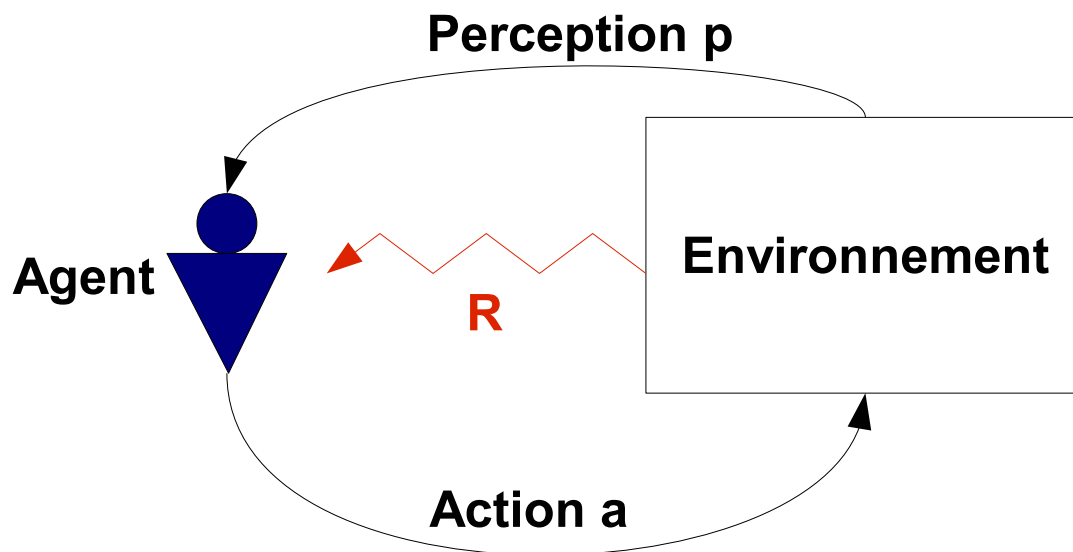


FIGURE 2.1 – Apprentissage par renforcement

- L'environnement est stationnaire : la probabilité des transitions et des valeurs de renforcement ne changent pas au cours du temps.

Après avoir choisi une action, l'agent ne connaît que l'état résultant et la récompense reçue. Il ne sait pas quelle action aurait été la meilleure sur le long terme. L'agent doit donc nécessairement collecter de l'expérience sur les possibles états, actions, transitions et récompenses pour pouvoir agir de façon optimale.

Les modèles de comportements optimaux répondent à la question de comment exploiter l'expérience acquise par l'agent pour construire un comportement optimal. Plus précisément, de quelle façon faut-il prendre le futur en considération pour le choix d'une action ?

- Horizon fini (h) : à un instant donné un agent optimise sa récompense pour les h étapes suivantes.
- Horizon infini : à chaque instant, on envisage la récompense sur le long terme mais plus la récompense est éloignée dans le futur (et donc incertaine) plus son influence est réduite (négligée) par un facteur de réduction γ .
- Récompense moyenne : l'agent choisit des actions qui maximisent sa récompense moyenne sur le long terme. Cette politique correspond à l'horizon infini avec γ à 1.

Il faut choisir la politique la plus appropriée en fonction du problème à traiter. Le modèle à horizon fini est approprié lorsque la durée de vie de l'agent est connue. L'apprentissage par renforcement permet de laisser l'agent construire son comportement

en fonction d'un signal de récompense qui représente sa réussite dans l'exécution d'un objectif. Néanmoins il faut pour cela modéliser préalablement l'ensemble des actions et des états, ainsi que la fonction de renforcement.

Si l'on fait une analogie avec le domaine biologique, cette forme d'apprentissage correspond au conditionnement d'un animal (ex. un chien) pour obtenir de lui un comportement ciblé en l'influençant par récompenses et châtements.

L'apprentissage par recherche stochastique

L'algorithme génétique est certainement la technique la plus représentative de cette forme d'apprentissage. Ces algorithmes permettent notamment de trouver une solution à des problèmes d'optimisation lorsqu'il n'existe pas de méthode exacte ou que la forme de la solution n'est pas spécifiée précisément. En effet, l'algorithme génétique permet de définir un langage plus ou moins expressif de façon à construire une solution par exploration des combinaisons qu'offre ce langage. L'inspiration de cette approche provient de l'"apprentissage" réalisé par les organismes biologiques à l'échelle de l'évolution. Ainsi le langage défini pour coder la solution est une sorte de code génétique. La recherche de solution se fait par une stratégie de mutations et de combinaisons aléatoires d'une population initiale de solutions, engendrant une nouvelle population que l'on évalue grâce à la fonction de récompense. Dans le cas des algorithmes génétiques cette fonction s'appelle *fonction de fitness* par analogie avec le terme "*fit*" (adapté) désignant l'organisme adapté à son environnement (les individus les mieux adaptés survivent). A chaque génération, seuls les individus les mieux notés sont sélectionnés pour engendrer des combinaisons ou des mutations (Figure 2.2).

Ce type d'approche a de multiples intérêts.

- L'exploration est garantie par le caractère aléatoire des opérations de mutations et de combinaisons. Suivant le langage défini la mutation d'un seul gène peut changer radicalement le comportement de la solution générée.
- C'est un algorithme "*anytime*". C'est à dire que l'on peut avoir à tout moment accès à la meilleure solution trouvée.
- L'apprentissage peut disposer d'une grande expressivité et échapper aux biais de représentation du concepteur, car seul un langage de construction bas niveau de la solution est spécifié. De même, la fonction d'évaluation exprime un objectif de haut niveau, et n'influence pas directement la forme de la solution. C'est pourquoi les solutions proposées par les algorithmes génétiques sont parfois surprenantes pour le concepteur.
- La plus grande partie des calculs correspond à l'évaluation des solutions par la fonction de fitness. De ce fait, cet algorithme est fortement parallélisable car toutes les solutions d'une population vont être évaluées en même temps.

Ce dernier point peut aussi être un inconvénient si la fonction de fitness est trop complexe. Généralement, cet algorithme est efficace lorsque l'on peut spécifier une fonction de fitness simple. La fonction de fitness doit être capable d'évaluer n'importe quelle solution de manière pertinente, ce qui est une contrainte forte. Malheureusement pour

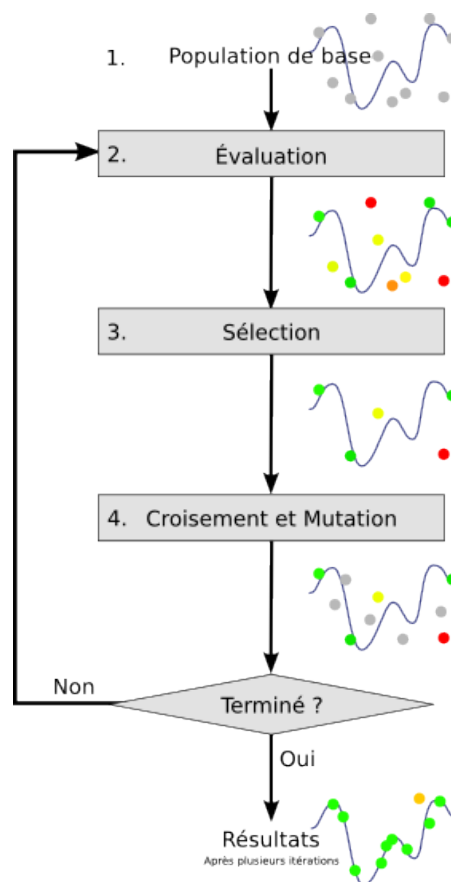


FIGURE 2.2 – Principe de l'algorithme génétique

certaines problèmes, une telle fonction n'est pas toujours possible. Il en va de même pour la spécification du langage génétique définissant l'espace de recherche, qui n'est pas toujours évidente. Enfin, l'évaluation d'une grande quantité de solutions possibles en un temps très court est adaptée aux environnements virtuels. L'application aux environnements réels (ex. robots) oblige à passer par une simulation du robot et de l'environnement posant de nombreux problèmes (ex. *reality gap* : le contrôleur évolué en simulation ne fonctionne pas une fois appliqué au robot réel (voir [Koons13])). Ces aspects conditionnent fortement les domaines d'application possibles de cette technique.

L'utilisation de ces algorithmes pour l'apprentissage du comportement d'agents virtuels incarnés dans un environnement simulé a suscité l'émergence d'une variété de nouvelles approches de l'IA notamment dans le domaine de la *vie artificielle*. Ce sont des approches *bottom-up* de l'IA dans la lignée des idées développées par Brooks sur l'IA incarnée. C'est le cas des *animat*, idée initiée par [Wilson86] qui propose un animal simulé minimal évoluant dans une grille mais dont le comportement est entièrement

appris en fonction des possibilités qu'offre son environnement. Il utilise pour cela un algorithme évolutionnaire et une représentation du comportement reposant sur un système de classeurs. Plus tard, [Sims94] utilise des algorithmes génétiques pour faire évoluer en parallèle la morphologie et le contrôleur de créatures virtuelles, afin de trouver les comportements les plus adaptés au couple agent-environnement. Le résultat est saisissant car on observe alors des comportements qui semblent "naturels" et inattendus, qu'il aurait été très difficile d'obtenir par une modélisation experte. Cette constatation renforce l'idée que l'apprentissage à partir de l'expérience est l'approche permettant de créer des systèmes artificiels qui peuvent réellement s'adapter à des environnements complexes changeants.

2.1.4 L'apprentissage non-supervisé

L'apprentissage non supervisé est une méthode d'apprentissage automatique, consistant à détecter des structures cachées (des régularités) dans des données non labellisées. C'est-à-dire sans feedback explicite sur les données (signal d'erreur ou de récompense). Ces approches sont souvent basées sur l'utilisation d'outils statistiques. Il existe différentes façons d'extraire ces régularités des données :

- La classification : il s'agit pour un programme de diviser un groupe hétérogène de données en sous-groupes, de manière à ce que les données considérées comme les plus similaires soient associées au sein d'un groupe homogène et qu'au contraire les données considérées comme différentes se retrouvent dans des groupes distincts. L'objectif étant de permettre une extraction de connaissance organisée à partir de ces données.
- La recherche d'associations consiste à découvrir des relations ayant un intérêt statistique entre deux ou plusieurs variables stockées dans de très importantes bases de données (ex. algorithme Apriori).
- *cartes auto-organisatrices* (SOM : *Self Organizing Map* [Kohonen90]) sont un type de réseau de neurones permettant de créer une représentation (carte) de dimensionnalité réduite (ex. 2 dimensions) d'un ensemble de variables. Un *voisinage* (fonction de similarité) doit être défini pour comparer et intégrer les exemples à la topologie de la carte. Les SOM peuvent être utilisées pour faire de la classification.

Les méthodes de classification et de discrétisation telles que les SOM peuvent être très intéressantes pour traiter les données brutes continues. Il existe actuellement de nombreuses méthodes pour classer et identifier l'information, qui peuvent être efficaces pour certaines catégories de problèmes. Mais toutes ces méthodes sont limitées car elles ne reposent que sur le caractère ou la structure des éléments eux-mêmes par le choix d'une fonction de similarité unique. Or, deux éléments peuvent être similaires du point de vue d'un objectif, et complètement différents dans un autre cas. C'est pourquoi, la représentation des relations entre les objets doit être contextuelle et non absolue. Ce type de méthode est très adapté pour la résolution de certaines opérations nécessaires

au sein d'un système d'IA, mais l'on ne peut réduire une vraie IA à ces seules fonctions comme l'exprime [Minsky91] :

"Each method has its advocates, but I contend that it is now time to move to another stage of research : Although each such concept or method might have merit in certain domains, none of them seem powerful enough alone to make our machines more intelligent. It is time to stop arguing over which type of pattern-classification technique is best because that depends on our context and goal. Instead, we should work at a higher level of organization and discover how to build managerial systems to exploit the different virtues and evade the different limitations of each of these ways of comparing things. Different types of problems and representations may require different concepts of similarity. Within each realm of discourse, some representation will make certain problems and concepts appear more closely related than others. To make matters worse, even within the same problem domain, we might need different notions of similarity for descriptions of problems and goals, descriptions of knowledge about the subject domain, and descriptions of procedures to be used.([Minsky91])"

En d'autres termes, l'évaluation de la similarité est tout à fait liée au problème du sens, c'est une notion subjective issue de l'expérience de l'agent situé. Par exemple, si l'on veut s'asseoir on peut considérer comme similaires un tabouret et un banc, et l'on choisira l'un ou l'autre indifféremment. Si l'on veut s'allonger au contraire, ces deux objets paraissent beaucoup moins similaires, et on privilégiera sans doute le banc. Pour un robot qui n'a ni la faculté de s'asseoir, ni celle de s'allonger ces objets pourraient paraître totalement similaires, ou bien être complètement différents, pour une raison qui nous dépasse et qui est propre à l'expérience du robot.

2.1.5 Commentaires sur l'IA incarnée et l'apprentissage

Dans la plupart des cas, les notions d'*agent*, d'*environnement*, et d'*expérience* n'apparaissent pas explicitement dans le domaine de l'apprentissage artificiel, en particulier pour son application à des problèmes abstraits. Mais on peut toujours considérer que l'agent correspond au système d'apprentissage, que l'environnement correspond aux données, et que l'expérience consiste en les *inputs/outputs* successifs du système. Il est clair que si le système évolue, c'est qu'il y a donc un changement d'état (une mémoire) et donc une représentation qui évolue, même si elle peut être extrêmement simple ou non interprétable (ex. réseau de neurones). Prenons deux exemples afin d'illustrer la tâche d'apprentissage et l'évolution de la représentation.

Premièrement, un réseau de neurone (*feedforward*) doit apprendre à reconnaître les lettres de l'alphabet. Il dispose en entrée d'une matrice de pixels, et en sortie de 26 valeurs possibles. C'est un algorithme supervisé, donc on lui fournit des exemples d'en-

traînement qui modifient sa représentation interne (poids des nœuds). L'agent progresse dans sa tâche et, à la fin, il peut reconnaître une lettre inconnue avec 95% de chances de succès. Supposons que le programme doive désormais apprendre à reconnaître des chiffres, il n'a pas acquis une expérience assez générique lors de l'apprentissage des lettres ou une sorte de savoir-faire qui lui permettrait d'apprendre plus rapidement cette nouvelle tâche pourtant très similaire.

Deuxièmement, un algorithme de classification non supervisé (ex. *k-moyennes*) doit identifier des maladies à partir d'une base de données contenant des informations sur des patients (infos personnelles, symptômes, etc.). L'algorithme parcourt les données qui lui permettent de modifier sa représentation interne (classes définies par des variables et des valeurs). Une interprétation des données est ainsi proposée par le programme. Cette interprétation est destinée à être analysée et validée par un utilisateur humain comme correspondant à sa propre réalité et lui étant utile ou non. La représentation n'est pas utile au système et ne peut être confirmée par lui par l'usage.

Dans ces exemples, on peut éventuellement dire que ces systèmes ont évolué grâce à leur interaction avec leur environnement, et que cela leur a permis d'améliorer cette interaction d'un point de vue externe. En tout cas, il y a amélioration du comportement, c'est à dire des résultats attendus par le concepteur. Néanmoins la représentation acquise par l'agent est exclusivement dédiée à cet objectif. Cette représentation ne permet pas à l'agent d'améliorer son propre rapport dynamique avec son environnement et donc de se développer et de s'adapter à de nouvelles tâches. Dans ces cas applicatifs, les systèmes possèdent les caractéristiques suivantes, résultant l'une de l'autre :

- L'agent n'est pas couplé à l'environnement : l'agent n'exerce aucune influence sur son environnement
- Sa motivation n'est pas intrinsèque : la représentation que l'agent construit est orientée vers un but qui ne provient pas de l'agent lui-même. Elle n'est pas *utile* à l'agent.
- Il n'y a pas d'évolution qualitative ou de capacité d'abstraction : ce que l'agent a appris ne lui permet pas d'apprendre de nouvelles choses radicalement différentes.

Ainsi sont absentes les capacités de généralisation et d'incrémentation qualitative des facultés de l'agent, qui distinguent l'apprentissage (au sens fort) de la simple évolution (ou adaptation dans des limites fixées) d'un processus en fonction d'un objectif externe.

Dans le cas de l'apprentissage par récompense, on peut dire que la représentation est modifiée par l'expérience active de l'agent confronté à son environnement. Au fur et à mesure que l'agent agit dans l'environnement, il renforce en effet une partie de sa représentation en fonction d'un objectif final (certaines transitions dans le cas de l'apprentissage par renforcement, certains "génomés" dans le cas des algorithmes évolutionnistes). Etant donné que l'objectif final intervient dans l'apprentissage par l'intermédiaire d'une fonction d'évaluation numérique relativement abstraite, cela laisse au système davantage de possibilités de faire évoluer sa représentation, de développer des

sous-comportements, etc. Dans le cas des algorithmes génétiques notamment, on peut retrouver une idée de construction d'une représentation à partir des éléments de base spécifiés (éléments du langage génétique). Dans cette optique, la principale différence avec l'apprentissage constructiviste réside dans le processus de construction et de validation de cette représentation. Pour l'apprentissage constructiviste il est essentiel que la représentation soit construite incrémentalement. Des concepts déjà acquis ayant leur utilité propre permettent d'en acquérir de nouveaux auxquels ils resteront liés. Ainsi la représentation doit être modulaire et polyvalente. C'est pourquoi certains travaux en *robotique évolutionniste* s'attachent à trouver des moyens de rendre la représentation générée par les algorithmes génétiques plus modulaire afin de permettre par exemple le transfert de compétences apprises d'une tâche à une autre, comme par exemple dans [Doncieux13].

2.1.6 L'apprentissage dans le cadre de l'AmI

2.1.6.1 Application de diverses techniques d'apprentissage artificiel

[Aztiria10] insiste sur l'importance de l'apprentissage pour l'AmI, en particulier sur l'apprentissage des actions des utilisateurs et la reconnaissance de situations et d'activités qui constituent des informations cruciales pour un environnement intelligent. En effet, pour que les décisions du système ne soient pas mal vécues par l'habitant, elles ne doivent pas le perturber dans ses habitudes. Pour cela, le système doit être plus ou moins "conscient" des événements ou des contextes liés à l'activité humaine en cours dans l'environnement, et d'une certaine manière être conscient des attentes de l'utilisateur. Mais il est évident que ce genre d'information est de nature variée et imprévisible, et ne peut donc pas être représenté ou calculé à l'avance.

[Aztiria10] passe en revue différentes techniques issues du domaine de l'apprentissage automatique (machine learning) appliquées à l'AmI pour l'apprentissage de motifs, et discute les avantages et inconvénients de chacune. Nous présentons dans cette partie les principales caractéristiques de ces différentes méthodes évoquées dans l'état de l'art effectué par [Aztiria10] et les principales conclusions.

Les techniques de classification

Dans ce type d'approche, un algorithme basé sur des outils statistiques permet de repérer des corrélations entre des variables au sein d'une base de données. Par exemple, un algorithme d'apprentissage supervisé basé sur des arbres de décision peut permettre d'apprendre à identifier le contexte adapté pour l'activation d'un objectif, comme présenté dans [Brdiczka05] (ex. apprendre à activer ou désactiver une musique en fonction de la présence et des actions de personnes dans une pièce). Ce type d'algorithme est assez efficace pour apprendre un concept ciblé dans un environnement maîtrisé. Les principaux avantages sont l'interprétation aisée des motifs appris, et la facilité de vérification et de mise à jour de la représentation. Cependant ce genre de programme

nécessite une préalable présentation et évaluation des données. D'autre part, l'expressivité de la représentation et la capacité d'abstraction de l'apprentissage sont absentes, puisqu'il s'agit seulement de repérer des corrélations ciblées. Ce type d'approche ne peut constituer en soi un modèle pour la création d'un agent autonome qui s'adapte à son environnement. Par contre, ces techniques sont des outils très utiles pour des tâches d'apprentissage ciblées. Comme par exemple pour identifier les conditions d'apparition d'un motif préalablement acquis. Ainsi, ces méthodes peuvent être employées judicieusement au sein de systèmes d'IA plus larges.

L'utilisation de règles pour représenter la connaissance est certes un peu rigide, mais l'utilisation de règles de la logique floue peut permettre d'atténuer cet inconvénient. Ces techniques conservent les avantages des modélisations à base de règles présentées ci-dessus mais permettent plus de robustesse pour gérer des domaines continus. Malgré cette différence, ce type d'approche ne diffère pas fondamentalement de la précédente, et les mêmes conclusions s'appliquent.

La découverte de séquence est une forme d'apprentissage non supervisé dont l'objectif est de détecter des régularités temporelles entre des événements dans une base de données. Par exemple, [Jakkula07] s'attaque au problème de la gestion du temps continu (modèle ne reposant pas seulement sur des pas de temps instantané). Pour cela il modélise les événements et leurs relations grâce à l'utilisation d'une logique temporelle (Utilisation de la logique temporelle d'Allen (logique modale)) et ramène la recherche de régularités au problème classique de la recherche de plus longue sous-séquence commune dans un ensemble de séquences. L'emploi de la logique temporelle permet d'exprimer des relations temporelles qualitatives (ex. pendant, après, etc.) mais ne permet pas de décrire une relation quantitative (ex. 5 minute plus tard). D'autre part, le langage proposé ne permet pas d'exprimer des conditions d'activation contextuelles d'un motif, mais seulement de relier deux événements dans le temps. De manière plus générale, cette approche nécessite également qu'on fournisse des exemples d'apprentissage à priori et repose sur une représentation préétablie relativement figée.

L'apprentissage par renforcement

L'apprentissage par renforcement peut être utilisé pour apprendre au système quelle est l'action la plus adaptée à effectuer en fonction du contexte dans lequel il se trouve. Par exemple [Zaidenberg09] étudie les moyens d'implémenter ce type d'algorithme dans le contexte de l'AmI en utilisant un feedback provenant de l'utilisateur. Il n'est pas aisé d'utiliser l'apprentissage par renforcement dans le cadre des systèmes ambiants pour plusieurs raisons. Tout d'abord, la représentation sous la forme d'un graphe contenant l'espace des états et l'espace des actions doit être définie à priori. Deuxièmement il est difficile d'interpréter le feedback provenant d'un utilisateur pour l'utiliser efficacement dans l'algorithme d'apprentissage. Enfin, la discrétisation du temps est faite de manière arbitraire et même si cela est peu contraignant en simulation, le problème se pose pour les applications réelles.

Bilan

Le constat général qui ressort de cette étude est que l'on trouve principalement des travaux qui se focalisent sur un aspect de l'AmI (ex. : optimisation de la consommation énergétique) en essayant d'utiliser les techniques connues les plus adaptées au(x) sous-problème(s) traité(s). Il n'y a pas vraiment pour l'instant d'approche globale ou holistique qui permette au système d'apprendre des motifs faciles à comprendre, qui représentent des activités perçues par le système en fonction de ses capacités quelles qu'elles soient. Il n'y a pas d'approche permettant de représenter le comportement humain par des séquences d'évènements, de relier des actions à d'autres évènements, de découvrir des conditions, et d'agir intelligemment tout en enregistrant des données et en adaptant sa représentation aux évolutions de l'environnement. Selon l'auteur la solution est sans doute de combiner différentes techniques présentées et il faut chercher une nouvelle approche plus globale, propre au domaine de l'AmI, afin d'envisager le problème directement dans son ensemble. Jusqu'à présent les recherches dévolues à l'AmI se sont focalisées surtout sur les technologies (les composants physiques et les middlewares). Cet article met en évidence l'importance des avancées en apprentissage automatique pour réaliser pleinement le paradigme de l'AmI. [Aztiria10] considère que nous avons besoin de développer de nouvelles techniques d'apprentissage globales pour l'AmI plutôt que de se contenter d'appliquer des techniques d'apprentissage automatique bien connues à des sous-problèmes indépendants.

2.1.6.2 Exemple d'une approche globale combinant diverses techniques

Pour avancer dans cette direction, [Aztiria09] propose une approche (PUBS : *Patterns of User Behaviour System*) qui combine des techniques de fouille de données (algorithme de recherche de règles d'association (algorithme *Apriori*)) et de classification (JRip (algorithme d'apprentissage de règles)), pour extraire des motifs (*patterns*) de comportements des utilisateurs par l'observation des capteurs (voir également [Aztiria12]). Les données sont préalablement divisées en 3 types, suivant qu'elles proviennent des objets (O) ou bien de deux différents types de capteurs : les capteurs contextuels (C) et les capteurs de mouvement (M) (voir Figure 2.3). On retrouve ici la "présentation" des données, faite à priori de manière experte, qui est souvent nécessaire à l'emploi de méthodes d'apprentissage non supervisé.

Dans son modèle, la représentation des motifs est définie par un triplet E, C, A (Evènement, Condition, Action). Ce sont les capteurs de type O qui sont au centre du modèle car ils correspondent à des actions de l'utilisateur. Le processus d'apprentissage cherche tout d'abord à trouver des corrélations entre une action (O) et un évènement (O ou M), grâce à une technique d'apprentissage non supervisé similaire à l'algorithme *Apriori*. Puis dans une deuxième étape le système vérifie si des régularités temporelles existent entre ces deux évènements. Pour cela les durées entre les instances successives de ces évènements sont classées en groupes, et si un groupe est dominant il peut consti-

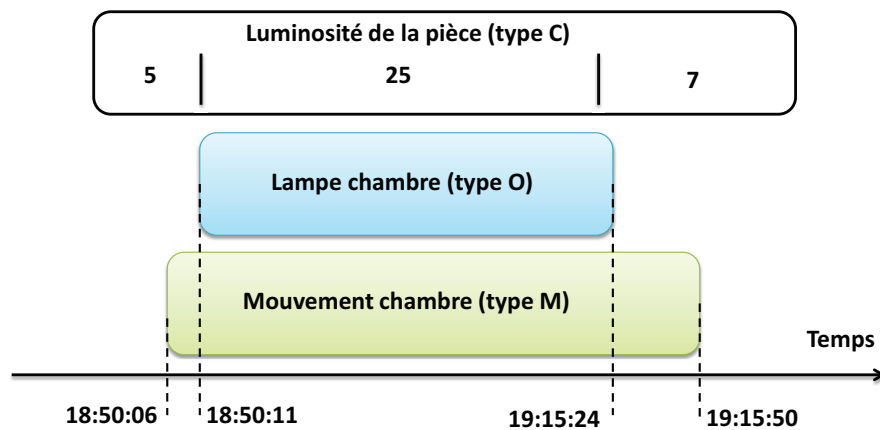


FIGURE 2.3 – Les trois types de capteurs du modèle de [Aztiria09]

tuer un motif. Par exemple un groupe qui représente une durée d'environ 5 secondes entre les deux évènements et qui contient 80% des instances peut être choisi. Enfin dans la troisième étape, le système découvre quelles conditions (par rapport au contexte C ou bien par rapport à une information temporelle (date, heure)) permettent de rendre plus fiable l'occurrence de cette régularité. Cette étape est réalisée par un algorithme de classification (JRip). Les auteurs montrent que leur système est capable de trouver des motifs d'interaction simples à partir des données.

Ce processus constitue une manière intéressante d'envisager l'apprentissage d'un concept pour un apprentissage non supervisé, car il permet de construire progressivement la régularité, en trouvant d'abord une relation générale, puis en précisant peu à peu les détails (temporalité et conditions contextuelles). Dans ce modèle tout comme dans celui de [Jakkula07], la structure de la représentation est assez générique (des séquences d'évènements), mais il y a un choix conceptuel sur la nature et le rôle que doivent jouer les différentes perceptions et actions au sein du modèle. C'est-à-dire qu'il y a une connaissance à priori sur la sémantique de certaines composantes de l'environnement. Cela peut permettre d'apprendre plus efficacement des choses que l'on s'attend à apprendre, mais cette orientation peut aussi empêcher le système d'apprendre des régularités non prévues par le concepteur. D'autre part, l'apprentissage n'est pas *multimodal*, les évènements sont limités à une seule variable. En l'occurrence l'*expressivité* d'un tel système est assez limitée (bornée à la forme E, C, A). Cela provient aussi du fait qu'il n'y a pas de capacité d'*abstraction* possible des concepts appris. Par exemple en permettant de considérer l'occurrence d'un motif comme un évènement constitutif potentiel d'un nouveau motif.

Ainsi nous considérons qu'un modèle constructiviste permettrait d'approfondir et d'étendre ce type d'approche dans un cadre théorique pertinent. Une question est de savoir jusqu'où peut-on aller dans la généralité du modèle d'apprentissage et de la

représentation qu'il autorise, pour ne pas s'interdire d'apprendre certains concepts tout en restant efficace (gestion d'un espace de recherche raisonnable).

2.2 Les systèmes multi-agents

Comme nous l'avons mentionné dans le chapitre d'introduction, cette thèse est effectuée en partenariat avec l'entreprise *ubiant* qui a développé une application d'AmI basée sur l'utilisation de *système multi-agents* (SMA). Dans cette partie nous présentons les SMA et l'intelligence artificielle distribuée de manière générale et considérons les problématiques d'apprentissage dans ce contexte. Nous nous intéressons particulièrement aux travaux existants dans le domaine de l'AmI appliqués aux bâtiments intelligents.

2.2.1 Présentation des SMA

Les systèmes multi-agents sont issus d'une branche de l'IA, l'*Intelligence Artificielle Décentralisée/Distribuée* (IAD) ayant deux principaux sujets d'étude. D'une part les programmes d'IA reposant sur une architecture distribuée permettant la parallélisation des calculs (décentralisation algorithmique). D'autre part, les approches concernant l'*intelligence distribuée*, c'est-à-dire une vision de l'intelligence sans contrôle central (décentralisation conceptuelle). Les réseaux de neurones sont un exemple d'une telle approche décentralisée de l'intelligence. La notion d'intelligence distribuée provient des propriétés d'organisation que semblent posséder des systèmes composés d'une population d'individus qui interagissent selon des règles simples. Pour un observateur au niveau global, le système dans son ensemble paraît posséder un comportement qui lui est propre. On parle alors de phénomène *émergent*, car ce phénomène n'est pas explicitement codé dans les composantes locales du système. Un phénomène collectif peut être prévisible en théorie si l'on peut connaître précisément tous les comportements des composants du système et l'état complet de l'environnement à un instant donné. Mais si l'on considère que le comportement global du système peut avoir un effet sur l'environnement et sur les composantes du système, des changements radicaux peuvent avoir lieu rendant le système difficilement prédictible ; c'est pourquoi on parle généralement de *système complexe*. De tels effets sont provoqués par des *boucles de rétroaction* (*feedback*), une notion clef de la *systemique* (l'étude des systèmes). La systemique est issue de la *cybernétique* introduite par [Wiener65] qui propose une théorie des communications et de leurs régulations dans les systèmes vivants (organismes biologiques) et artificiels (machines). Plutôt que d'étudier de tels systèmes de façon réductionniste par description des composants et des chaînes de causalité, cette discipline propose d'étudier les flux d'informations au sein de ces systèmes. Cette théorie est de fait liée à la *théorie de l'information* ([Shannon01]). L'autorégulation, l'auto-adaptation et l'auto-organisation sont des propriétés que l'on peut observer dans les systèmes vivants et qui motivent l'orien-

tation de cette étude systémique, car comprendre ces phénomènes pourraient permettre de créer des systèmes artificiels possédant de telles propriétés. Cette approche a eu une influence considérable, en particulier sur les sciences cognitives. En IA, de nouveaux domaines de recherche se sont développés autour de l'étude des systèmes complexes en prenant comme source d'inspiration directe des phénomènes biologiques, tels que l'IAD ou la vie artificielle (voir [Rennard02]).

[Ferber99] donne une vision complète du domaine des SMA. Nous restituons seulement ici quelques aspects principaux. Un système multi-agent est un système constitué d'agents situés dans un *environnement* au sein duquel ils interagissent. Un *agent* est une entité virtuelle ou physique pouvant percevoir, agir et communiquer avec d'autres agents, qui est autonome et possède des compétences pour réaliser des objectifs. Un agent peut être plus ou moins capable de raisonnement (agent cognitif) ou purement soumis à des réflexes (agent réactif). L'interaction de ces agents fait émerger un comportement global du système qui peut constituer la réponse à un problème complexe.

L'exemple le plus classique de phénomène biologique pour illustrer les SMA, est celui des colonies de fourmis. Par exemple considérons le cas de la découverte et du fourragement d'une source de nourriture. Lorsque des individus trouvent de la nourriture, ils déposent sur le chemin du retour une substance (*phéromone*) perceptible par d'autres fourmis. La perception de ces phéromones incite les fourmis à emprunter le même chemin les amenant ainsi à trouver la nourriture. A leur retour, elles laisseront aussi des phéromones, ce qui amplifie le phénomène : il y a ainsi une boucle de rétroaction positive. On appelle *stigmergie* ce phénomène de coordination indirecte des agents. Lorsque la source de nourriture se tarit, le dépôt de phéromones cesse et ils finissent par s'effacer au cours du temps. L'aspect intéressant de ce moyen de fonctionnement est la robustesse et la capacité d'adaptation de la solution. Dans cet exemple, une source de nourriture importante est susceptible d'attirer un nombre considérable de fourmis du fait que le chemin va être beaucoup renforcé. Si au contraire la source est restreinte, peu de fourmis seront attirées avant que le chemin ne disparaisse. Il y a donc adaptation des moyens alloués, en fonction de l'ampleur de la tâche. De plus, si le chemin est soudainement coupé par un obstacle, un nouveau chemin d'approvisionnement peut être rapidement rétabli. Certaines fourmis parmi celles qui arrivent à l'obstacle en suivant le chemin, trouveront un chemin alternatif. Le marquage du nouveau chemin sera plus rapidement renforcé étant donné que le chemin précédent attirait déjà beaucoup de fourmis. L'exemple des fourmis n'est pas seulement illustratif mais sert aussi d'inspiration à certains *algorithmes de colonie de fourmis* (ACO : *Ant Colony Optimisation*) qui adaptent directement ces principes pour la résolution de problèmes combinatoires ([Colorni91]), par exemple la recherche du plus court chemin dans un graphe.

Les systèmes multi-agents ont ainsi de multiples intérêts applicatifs, parmi lesquels :

- Résolution de problèmes : alternative à la résolution centralisée de problèmes en IA.

- La Simulation : expérimentations au service d'autres disciplines (ex. biologie, sociologie...).
- Comme paradigme de programmation et de conception logicielle.

Le choix d'utiliser les SMA se prête naturellement au problème de l'AmI du fait de sa nature décentralisée et son aspect complexe et dynamique incitant plutôt aux méthodes robustes et adaptatives qu'aux méthodes de résolution exactes. De plus dans le cadre de l'AmI, chacun des trois points d'intérêt cités ci-dessus est susceptible d'intervenir. C'est pourquoi il existe de nombreux travaux dans ce domaine, dont [Abras09a] donne un état de l'art assez complet. Nous en donnons ci-après quelques exemples, afin de donner une idée des différents types d'application possibles des SMA dans le cadre de l'AmI. Avant cela, nous proposons de considérer les liens possibles entre apprentissage et SMA.

2.2.2 L'apprentissage et les systèmes multi-agents

Comme l'explique [Sen99], les premiers travaux en IAD étaient essentiellement focalisés sur l'organisation structurelle des SMA, le comportement du système étant assez précisément déterminé. Or, les SMA sont pourtant particulièrement utilisés dans des environnements complexes, vastes, ouverts, dynamiques et imprévisibles. C'est pourquoi il est difficile voire impossible dans certains cas, de spécifier à la conception le comportement optimal d'un agent au sein d'un système multi-agent, avant exécution. Les agents doivent donc pouvoir s'adapter et apprendre durant l'exécution, en fonction de leur objectif et de l'environnement. Pourtant l'auteur indique que les domaines des SMA et celui de l'apprentissage automatique sont longtemps restés cloisonnés. L'intersection de ces deux domaines est pour lui un sujet crucial, qui ne se résume pas à l'application de l'apprentissage automatique aux agents individuels ; mais étend le concept d'apprentissage vers une vision décentralisée qui diffère qualitativement de la vision classique de l'apprentissage. Cela offre des perspectives de nouvelles techniques et algorithmes pour l'apprentissage. Il y a plusieurs façons d'envisager l'apprentissage décentralisé dans le cadre des SMA, comme explicité par [Alonso01].

Premièrement on peut considérer que pour améliorer l'efficacité du système, on dote les agents le constituant de capacités d'apprentissage qui leurs sont propres. On peut alors implémenter diverses techniques d'apprentissage connues, bien adaptées aux rôles spécifiques de chaque agent pour que chacun d'eux améliore son comportement. De cette façon on peut considérer que les SMA sont un domaine d'application de l'apprentissage classique (machine learning) avec ses propres caractéristiques, en essayant d'étendre les méthodes centralisées existantes directement aux multiples agents d'un SMA.

Deuxièmement on peut considérer le SMA global comme un système d'apprentissage distribué, sans avoir d'apprentissage centralisé pour les agents qui le constitue. Plusieurs agents ou groupes d'agents peuvent apprendre de manière distribuée et in-

teractive comme un ensemble unique et cohérent. C'est alors le système qui réalise l'apprentissage par l'évolution de ses agents, de ses structures ou de son organisation.

On peut donc distinguer deux catégories d'apprentissage dans les SMA comme suit :

- Apprentissage centralisé (ou individuel) : un agent apprend seul en toute indépendance.
- Apprentissage décentralisé (ou interactif) : plusieurs agents sont impliqués dans une même activité d'apprentissage.

Dans un système multi-agent réalisant un apprentissage décentralisé ces deux formes d'apprentissage peuvent cohabiter de multiples façons. Un même agent peut très bien participer à plusieurs d'entre elles simultanément.

2.2.2.1 Le problème de l'attribution de la récompense

Une des principales difficultés d'une approche décentralisée de l'apprentissage provient du fait que l'apprentissage repose généralement sur un signal de rétroaction. C'est-à-dire une information qui indique au système s'il est ou non sur la bonne voie. Lorsque le système est composé de différentes unités, une question difficile se pose : comment attribuer correctement le feedback parmi les agents ? Pour un changement observable au niveau global, comment savoir quelles sont les entités qui ont effectivement participé à ce changement et de quelle façon ? Il s'agit là du *problème de l'attribution de la récompense* (*Credit Assignment Problem* (CAP)).

Dans le cas des SMA, l'auteur signale que l'on peut décomposer le problème en deux sous-problèmes.

- *CAP inter-agent* : Rétribution des actions des agents en fonction d'un changement de performance au niveau global. Ce problème est particulièrement difficile dans le cas des SMA : Quelle(s) action(s) de quel(s) agent(s) a/ont contribué au changement de performance et à quel point ?
- *CAP intra-agent* : Rétribution des inférences internes ou décisions d'un agent en fonction d'une action externe effectuée. Quelles connaissances, quelles inférences et quelles décisions ont réellement mené à l'action souhaitée ?

L'auteur précise que la distinction entre ces deux sous-problèmes n'est pas toujours évidente, et que souvent l'un est traité, alors que l'autre est fortement simplifié.

Une solution simple est celle de la récompense globale. Il s'agit de diviser équitablement la récompense entre tous les apprenants concernés. Cela ne suffit pas aux problèmes complexes avec des agents hétérogènes car le feedback n'est pas assez fin et corrélé aux actions des agents.

Prenons un exemple pour illustrer ce problème. Supposons que le système d'AmI doit atteindre l'objectif "ambiance tamisée". Réagissant à cet impératif, un agent *lampe* modifie l'éclairage et un agent *volet roulant* provoque la fermeture du volet pour obscurcir la pièce, de sorte que l'objectif soit atteint avec succès. Mais dans une situation

particulière, peut-être que l'agent *volet roulant* n'a eu aucun impact sur la réalisation de l'objectif car, par exemple, il fait nuit. Cet agent ayant participé à la réalisation de l'objectif, il est pourtant susceptible de recevoir un feedback positif.

Une autre solution, à l'extrême inverse, est d'évaluer les agents seulement sur leurs propres actions. L'effet pervers est que cela nuit à la coopération et tend à faire agir les agents indépendamment.

Une solution alternative peut être que le feedback est global mais compris par les agents comme la somme des feedbacks pour leur action collective, et que chaque agent puisse "filtrer" sa part de feedback. Un moyen d'évaluer la part d'un agent par rapport au feedback global est par exemple de comparer la différence de résultat si on retire cet agent.

2.2.2.2 Caractéristiques de l'apprentissage décentralisé

La dimension de l'espace de recherche croît rapidement avec la complexité des comportements des agents, le nombre d'agents, et la taille du réseau d'interactions entre les agents. C'est pourquoi il est très difficile d'apprendre un comportement collectif total ciblé d'un système multi-agent large, hétérogène et fortement communicant. Il faut donc, pour obtenir un apprentissage désiré qui soit effectif, pouvoir faire certains sacrifices comme expliqué dans [Panait05], par exemple :

- En isolant les comportements appris sur certains agents
- En réduisant l'hétérogénéité des comportements
- En réduisant la complexité des capacités des agents

[Panait05] mentionne des exemples de techniques possibles permettant cela qui sont entre autres : l'apprentissage par équipes ou la décomposition du comportement. La décomposition du comportement consiste à réduire la complexité en séparant heuristiquement le problème en sous-problèmes, et donc le comportement collectif en sous-comportements séparés plus simples à apprendre pour les agents. Une telle décomposition peut se faire à plusieurs niveaux et de différentes façons. Par exemple on peut envisager une approche ascendante en apprenant d'abord les comportements élémentaires et en les assemblant afin de créer un comportement plus complexe. Cette méthode est appelée apprentissage par niveaux (*Layered learning*). La problématique ici serait de pouvoir trouver comment décomposer automatiquement le problème et apprendre à générer la coordination entre ces sous-comportements. On peut envisager une approche hiérarchique où les agents coordonnent seulement des comportements de haut niveau, et pas chaque action élémentaire.

[Panait05] présente les trois principaux domaines d'application de l'apprentissage distribué : Les agents embarqués, l'environnement de la théorie des jeux, et les problèmes du monde réel. Clairement l'AmI se situe dans la troisième catégorie, dont les problèmes courants sont les suivants : logistique, planification, problème de satisfaction de contrainte, etc. Parce que ce sont des problèmes très complexes et hautement distri-

bués, il y a relativement peu de travaux. On peut citer cependant certaines applications : contrôle de trafic (routier, aérien), gestion de réseaux, soins médicaux distribués, chaîne de fournisseurs, modèle d'interactions sociales, planification de rencontres.

2.2.3 Les systèmes multi-agents dans le cadre de l'AmI

2.2.3.1 Modélisation directe d'une maison intelligente par un SMA

La manière la plus intuitive d'utiliser les SMA dans le contexte de l'AmI consiste en une modélisation directe du système à un haut niveau. C'est à dire que chaque matériel et son contrôleur (application) est modélisé par un agent dans le système. [Conte03] proposent un modèle général qui illustre cette façon de procéder. Dans le travail qu'ils présentent, l'objectif du système est triple :

- Réguler la consommation énergétique (éviter les pics, planifier pour réduire les coûts)
- Gérer les comportements des applications (optimisation, détection d'erreurs)
- Faciliter l'interaction de l'utilisateur (commande à distance, planification, visualisation des données)

Le système est construit autour d'un agent central, l'agent "capteur/limiteur de consommation" (CLC), qui est capable de percevoir la consommation d'énergie des autres agents, et de communiquer avec les autres agents des informations concernant l'énergie (disponibilité, coût). L'énergie constitue une ressource commune que les agents peuvent "consommer". Pour gérer les accès concurrents à cette ressource, un agent peut réguler sa consommation (activation ou attente), éventuellement au détriment de sa performance (mesurée par un index de satisfaction de l'utilisateur). Cette régulation est gérée par la stratégie du "premier arrivé : premier servi", modérée par un système de priorité. Pour cela il y a trois niveaux d'agents plus ou moins cognitifs. Au premier niveau les agents tiennent compte des informations de l'agent CLC pour réguler leur comportement, au deuxième niveau un agent peut communiquer ses besoins (ex. priorité aux autres agents), et au troisième un agent peut en outre prendre en compte les informations des autres agents. Globalement, la performance du système est évaluée comme le succès à maintenir la consommation énergétique au-dessous d'un seuil fixé, tout en respectant des indices de satisfactions pour les tâches qui lui sont attribuées. Par exemple, réaliser une tâche en un temps voulu.

Dans une démarche similaire, on peut citer les travaux de [Davidsson00]. Ici la liberté d'action est limitée à l'activation des lumières (on/off), du chauffage et de quelques appareils. Les préférences des utilisateurs sont programmées. Un système de badge actif est utilisé pour gérer la présence et l'identité des utilisateurs dans différentes zones. Le comportement des agents est prédéfini par des règles. Cette fois les agents ne correspondent pas forcément à un matériel en particulier. Quatre catégories d'agents sont définies :

- BSA (Badge System Agent) : repère les personnes dans la maison.

- PC (Personal Comfort) : connaît les préférences des utilisateurs.
- ROOM : décide de la valeur des paramètres environnementaux en fonction des préférences des utilisateurs.
- EP (Environmental Parameter) : contrôle des capteurs et actionneurs, afin d'influer sur les paramètres.

Le fonctionnement repose sur les communications entre agents. Exemple : lorsqu'une personne change de pièce, l'agent BSA informe l'agent PC du déplacement. L'agent PC informe les deux agents ROOM concernés, en leur donnant les préférences. L'agent ROOM fournit des conditions sur l'environnement aux agents EP qui s'occupent de les réaliser et les maintenir. Le système doit gérer deux buts conflictuels :

- Agent ROOM : Minimiser l'énergie consommée
- Agent PC : Maximiser le confort

Ce système est évalué relativement à l'objectif de maintien de la température, en le comparant à un thermostat simple et à un thermostat programmable.

- Un thermostat
- Un thermostat programmable (heures d'activation/désactivation du chauffage)
- SMA réactif (chauffage activé seulement lorsqu'il y a une présence)
- SMA proactif (chauffage activé en fonction des données des agendas des utilisateurs)

Les auteurs indiquent que les deux dernières méthodes permettent d'économiser plus d'énergie que les deux premières, ce qui n'est guère surprenant.

Dans ce type de travaux, l'utilisation d'un système multi-agents repose sur une forte modélisation des agents et se focalise sur la résolution d'un problème très ciblé avec des méthodes clairement définies a priori. La modélisation de l'environnement intelligent par un SMA est ici surtout motivée par l'affinité d'ordre conceptuel du domaine avec les SMA. L'intérêt du choix du paradigme multi-agents pour traiter les problèmes présentés en termes de capacité d'adaptation et d'apprentissage n'est pas réellement mis en évidence dans ce type d'approche.

2.2.3.2 Utilisation de SMA pour résoudre le problème de la gestion de l'énergie pour l'habitat

Les travaux suivants ([Abrás08] [Abrás09b] [Abrás09a]) se concentrent sur un des aspects de la domotique : la gestion de l'énergie pour l'habitat. Le système MAHAS (Multi-Agents Home Automation System) proposé doit essentiellement permettre de trouver dynamiquement une politique de consommation d'énergie en fonction des contraintes provenant de l'utilisateur ou des conditions matérielles. Cette approche en vue d'optimiser la consommation énergétique répond notamment au besoin de gérer les pics de consommation.

Dans ce modèle, l'auteur caractérise les équipements selon deux perspectives : *pilotables / non-pilotables* et *permanents / temporaires*. Un agent n'est pas directement associé

à un équipement mais à la notion plus abstraite de *service*, qui correspond à une catégorie d'équipement. Ainsi un service peut contrôler un ou plusieurs équipements. Un service sera classé suivant les critères précédemment cités. Suivant que le service est permanent ou temporaire, son comportement suit respectivement un modèle dynamique (système d'équation) ou un automate à état fini. Chaque "agent service" possède des connaissances internes qui correspondent à des informations propres au fonctionnement de l'équipement associé, ainsi que des connaissances partagées qui sont des messages standardisés échangés avec les autres agents.

L'objectif du système est d'assurer le confort de l'utilisateur. Or le confort est une notion abstraite, aussi l'auteur choisit d'utiliser la notion de *satisfaction de l'utilisateur*. Les agents sont dotés d'une satisfaction [de 0%(inadmissible) à 100%(parfait)] qui reflète les sentiments de l'utilisateur vis-à-vis d'un service.

La liberté d'action des agents repose essentiellement sur deux possibilités : décalage temporel (avance, retard, interruption) pour les services temporaires, et modification de la puissance pour les services permanents. Ceci est suffisant pour permettre une planification des activités au cours du temps afin d'optimiser la consommation. Cependant pour pouvoir planifier, il faut avoir des informations sur le futur.

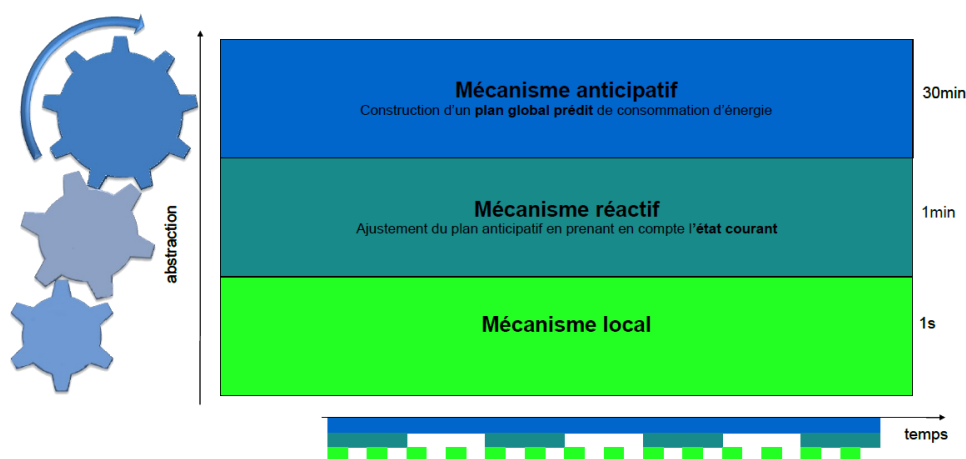


FIGURE 2.4 – Architecture du système MAHAS [Abrás09a]

C'est pourquoi le système est multi-niveaux : un mécanisme réactif et un mécanisme anticipatif sont associés (voir Figure 2.4). Le mécanisme réactif vise à répondre à des phénomènes exigeant des temps de réponse très courts (ex. : contrainte de puissance maximale), tandis que le mécanisme anticipatif doit répondre aux phénomènes lents (ex. : inertie thermique, variations du prix de l'énergie, variation des capacités de production). Ce dernier permet de planifier des événements prévus à l'avance pour réaliser un plan d'affectation. C'est le mécanisme réactif qui réalise ce plan d'affectation et gère

les évènements imprévus. Le mécanisme anticipatif repose sur les prévisions météorologiques et sur les commandes utilisateurs. Il est associé à un agent solveur qui communique avec les agents services afin d'obtenir des informations sur toutes les prochaines activités. Cet agent utilise ensuite une méthode de résolution exacte (GLPK) pour planifier au mieux ces activités. Le mécanisme réactif repose sur un comportement très simple des agents : dès que leur satisfaction atteint un seuil critique, l'agent demande de l'aide aux autres via un protocole de communication de type *contract net*. Ce seuil critique évolue dynamiquement en fonction des requêtes, ainsi les agents gardent un seuil de satisfaction homogène, par conséquent ils sont exigeants lorsque les contraintes sont faibles et font des concessions lorsqu'elles sont fortes. La Figure 2.5 illustre l'interaction de ces différents mécanismes au sein du système MAHAS.

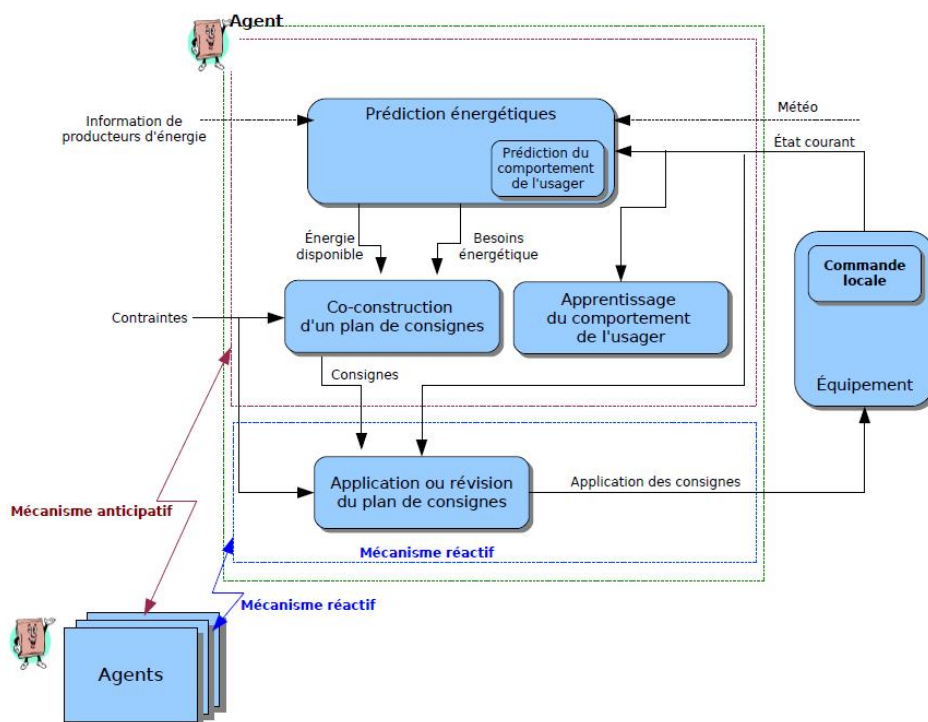


FIGURE 2.5 – Composantes principales du système MAHAS [Abrás09a]

Globalement les auteurs indiquent que les performances du système sont moins bonnes qu'avec une approche centralisée, en termes de temps d'exécution et aussi en termes d'efficacité (non optimalité de la solution). Néanmoins l'auteur montre que la force de cette approche réside dans sa robustesse et son adaptabilité au changement, qui correspond bien à la nature dynamique du problème.

Il s'agit donc ici d'une approche hybride utilisant différents outils de l'IA pour résoudre différents aspects du problème. Elle repose sur deux couches : un SMA réactif pour gérer le bas niveau et une couche anticipative pour les tâches de plus haut-niveau. La couche réactive permet d'assurer une adaptation en temps réel aux imprévus sans forcément remettre en question les tâches en cours. Ce type d'approche autorise la gestion dynamique du matériel et des objectifs, et illustre davantage l'intérêt des SMA en tant qu'approche décentralisée pour la résolution des problèmes complexes de l'AmI. Par contre, ce modèle est focalisé sur la problématique de l'optimisation énergétique avec une modélisation clairement orientée dans ce sens, et n'aborde pas les problématiques d'apprentissage de l'AmI présentées dans la partie 1.1.

2.3 Les approches neuro-inspirées

Pour dépasser les difficultés liées à une vision computo-représentationnelle de la cognition (approche descendante), des approches *connexionnistes* ont été envisagées (approche ascendante). Les approches connexionnistes modélisent les phénomènes mentaux ou comportementaux comme des processus émergents de réseaux d'unités simples interconnectées. Cependant les modèles connexionnistes, même si techniquement et conceptuellement opposés aux modèles computationnels, car ne possédant pas de représentation initiale et étant par nature décentralisés, ne répondent pas à toutes les problématiques liées au problème de la représentation. Notamment, pour la plupart ils consistent essentiellement à relier directement perception et action en un processus de traitement de l'information. En effet, comme l'explique [Ziemke01], le couplage agent-environnement et les rétroactions nécessaires à une intelligence incarnée ne sont pas forcément pris en compte dans ce type d'approche. Néanmoins les *réseaux de neurones récurrents* (RNR) ainsi que de nouvelles approches similaires permettent cela en ajoutant un feedback permettant de prendre en compte l'état interne de l'agent ([Elman90]). Nous présentons dans cette partie certains travaux neuro-inspirés illustrant des principes fondamentaux du fonctionnement du cerveau. Pour un état de l'art complet des techniques connexionnistes, voir par exemple [Panzoli08].

2.3.1 Introduction

Les réseaux de neurones et plus particulièrement les réseaux de neurones récurrents et les mémoires auto-associatives sont des techniques qui sont en accord avec de nombreuses observations biologiques concernant le fonctionnement du cerveau. Par exemple, [McIntosh00] présente une vision du fonctionnement du cerveau comme un réseau auto-organisé. S'éloignant d'une vision plus ancienne du cerveau où les régions sont vues comme des modules correspondant à des fonctions cognitives précises, il insiste sur la nature du réseau neuronal, sa connectivité, sa plasticité et son uniformité, pour signifier que c'est l'organisation et les échanges au cours du temps entre

les ensembles de neurones qui fait émerger la cognition. On retrouve ici une vision systémique de la cognition. Les approches présentées dans cette partie se basent sur l'étude du fonctionnement du cerveau, et plus précisément de la partie appelée "néo-cortex". Cette zone du cerveau, la plus récente dans l'évolution, est considérée comme le siège de l'intelligence. L'approche consistant à démarrer par l'étude du cerveau pour la construction d'une IA est moins courante qu'on pourrait le croire de prime abord, car il y a bon nombre d'objections en sa défaveur. Certains pensent qu'il est simplement impossible de comprendre comment fonctionne le cerveau, d'autres que l'intelligence est inhérente aux caractéristiques biologiques du cerveau et donc impossible à reproduire artificiellement. D'autres considèrent que l'esprit est indépendant du cerveau, de la même manière que le programme est indépendant de la machine qui l'exécute, et donc que comprendre l'un ne permettra pas de comprendre l'autre.

Néanmoins on peut aussi considérer qu'il est possible de concevoir des systèmes réellement intelligents (c'est à dire possédant les capacités essentielles telles que : apprentissage, adaptation, compréhension, créativité) à condition de comprendre préalablement ce qu'est l'intelligence. Le moyen d'y parvenir est notamment d'étudier le cerveau, unique exemple que l'on connaisse de l'intelligence au sens fort. A l'origine, le paradigme de l'intelligence artificielle n'est pas forcément destiné à atteindre cet objectif, et d'ailleurs quelques différences l'indiquent :

- IA classique : préprogrammée - fonctionnement invariable - processeur central
- Intelligence : apprenante - tolérance aux erreurs - pas de contrôle centralisé

En effet les programmes d'Intelligence artificielle tels que les systèmes experts, les solveurs logiques ou encore les planificateurs se contentent d'exécuter de façon extrêmement rapide des algorithmes spécifiques conçus spécialement pour un type de problème à traiter ; alors que l'intelligence réelle réside avant tout dans la compréhension.

2.3.2 Memory Prediction Framework

2.3.2.1 La plasticité du cerveau

[Hawkins05] suggère qu'il est erroné d'associer intelligence et comportement. On peut en effet raisonner et comprendre intérieurement, mais sans extérioriser aucun comportement. Le comportement n'est qu'une manifestation de l'intelligence, aussi il ne faut pas en faire une référence pour l'intelligence artificielle et c'est là le danger du célèbre test de Turing qui tend à conforter cette confusion entre comportement adéquat et intelligence. Dans son modèle *Memory Prediction Framework*, [Hawkins05] s'inspire des connaissances actuelles sur le néocortex et définit l'esprit comme une mémoire associative qui enregistre des motifs spatiaux-temporels (*patterns*) de manière hiérarchique. L'idée générale est tirée de [Mountcastle78] qui, observant que le cortex est extrêmement uniforme, pose l'hypothèse que son fonctionnement est le même dans toutes les régions, et met en avant l'organisation fonctionnelle du cortex en colonnes.

Plutôt que d'analyser les différences entre les zones du cerveau, Hawkins considère qu'il faut mieux se concentrer sur les similitudes afin d'en dégager un fonctionnement commun, sorte d'*algorithme cortical* générique (*common cortical algorithm*). Ceci implique que le cerveau applique toujours la même fonction, pour des traitements qui intuitivement nous paraissent totalement différents (ex. vision, raisonnement, mouvement, etc.). Du fait de cette homogénéité, le cerveau est fortement adaptable car capable de se câbler en fonction des variations auxquelles il est soumis. Cette plasticité du cerveau permet à des personnes nées aveugles d'attribuer à l'audition une zone normalement allouée à la vision. Comme présenté dans [Bach-y Rita03], par un mécanisme de *substitution sensorielle* des personnes aveugles peuvent "voir" grâce au sens du toucher par l'intermédiaire d'une caméra et d'un dispositif installé sur la peau qui retranscrit les informations visuelles de la caméra. L'algorithme cortical générique qui opère identiquement à tous les niveaux du néocortex doit permettre de rendre compte de la *capacité d'abstraction* par la plasticité de la structure neuronale et des liens entre les différentes couches. L'hypothétique algorithme cortical serait donc ce que l'on doit réussir à reproduire dans des systèmes artificiels qui auront leurs propres perceptions et leurs propres matériaux constitutifs, totalement différents de ceux de l'être humain. On retrouve ici la notion de *Umwelt* et l'expression de la nécessité que l'agent construise lui-même sa représentation, en fonction de ses capacités propres.

2.3.2.2 Le fonctionnement du cerveau

Hawkins explique également que ce qui doit être compris et reproduit est l'organisation, la structure et le processus de fonctionnement indépendamment des constituants matériels avec lesquels elle est implémentée. La cognition est assimilée à une « organisation » de même que pour la vision autopoïétique de la cognition. L'auteur affirme que la principale fonction du cerveau consiste à traiter des motifs, c'est-à-dire des motifs spatio-temporels. L'aspect spatial vient du fait qu'un récepteur possède une dimension sur laquelle il prend des valeurs, et plusieurs récepteurs sont stimulés simultanément. L'aspect temporel vient du fait qu'un motif varie constamment au cours du temps. Le temps occupe en effet une place centrale dans la reconnaissance et la mémorisation des motifs. Par exemple pour la vision, l'œil effectue sans cesse des microsaccades qui fournissent des informations variantes lorsqu'on fixe un unique objet. Ce serait cet enchaînement de motifs qui nous permettrait d'identifier l'objet plutôt que l'image fixe de l'objet envoyée par la rétine à un instant unique.

Le fonctionnement du cerveau est fondamentalement différent de celui d'un ordinateur : le cortex tout entier est une mémoire, constituée par les neurones et le cerveau n'effectue pas des opérations pour résoudre un problème, il extrait la solution de sa mémoire. Par exemple, pour faire attraper une balle en plein vol à un bras articulé, on peut faire calculer la trajectoire de la balle à partir des positions initiales, puis inférer une suite d'instructions pour chaque actionneur du bras. Cette méthode est celle du calcul. Ou bien on peut utiliser la mémoire en extrayant la séquence d'instructions

nécessaires issue d'un apprentissage sur des années d'expériences. Car la mémoire du cortex ne range pas des informations exactes et complètes comme pourrait le faire celle d'un ordinateur. Au contraire, la mémoire semble enregistrer seulement les relations importantes du monde, indépendamment des détails. Même si il y a une infinité de lancers possibles, des régularités sont invariantes à la tâche de rattraper une balle, et ont été construites par l'ensemble des cas particuliers de l'expérience. Jeff Hawkins appelle cela le mécanisme de « représentation invariante ».

2.3.2.3 La représentation invariante

C'est cette représentation invariante qui nous permet par exemple de reconnaître un chien particulier alors qu'il se déplace dans l'espace et qu'à chaque instant l'image que l'on en perçoit est différente ; et également c'est ce qui permet de former le concept plus général de chien au-dessus de chaque instance perceptible d'un chien. De même lorsqu'on mémorise un morceau de musique, ce sont les relations entre les notes que nous mémorisons, ce qui permet de reconnaître le morceau joué dans différentes gammes et par différents instruments.

Certains réseaux de neurones artificiels (mémoire auto-associative) peuvent reconnaître une image si on en fournit une version incomplète. Mais si on fournit l'image complète dans des conditions différentes (rotation, décalage, changement de taille) alors la reconnaissance échoue. Pourtant reconnaître une image complète, même dans différents contextes, est pour notre cerveau plus trivial que de reconnaître une image incomplète. Cette différence fondamentale viendrait de la capacité de notre cerveau à former un motif invariant associé à l'entité particulière qu'est l'image en ayant enregistré non pas une sorte de cliché figé de l'image pris à un instant donné, mais un motif spatio-temporel qui représente les relations, les intervalles, les rapports, bref la structure de l'image.

Pour Hawkins, les quatre attributs fondamentaux de la mémoire sont les suivants :

- Range des séquences de motifs
- Rappelle les motifs de façon auto-associative
- Range les motifs sous forme invariante
- Range les motifs dans une hiérarchie

La notion de structure est très importante. Notre cerveau est capable d'appréhender le monde car à l'image du cerveau le monde est fait de structures à tous les niveaux. Ainsi le cerveau trouve naturellement les structures existantes et les capture. Lorsque la structure est absente, nous sommes confrontés à la confusion et au chaos.

2.3.2.4 L'importance de la prédiction

La fonction primaire du cerveau, qui est la cause de l'intelligence et du comportement, c'est de faire des prédictions constamment, à chaque instant et de façon incons-

ciente. La prédiction est possible par les correspondances entre les motifs mémorisés qui sont reliés dans une structure complexe et dynamique. Pour être capable de réaliser une prédiction, il faut associer une représentation courante (une instance), à une représentation invariante mémorisée. Ce processus est permanent, décentralisé et inconscient. Le cerveau permettrait plus généralement de mémoriser des séquences de motifs (des séquences de séquences, etc.), mais la notion de séquence est davantage vue comme un *ensemble* de motifs dont l'ordre peut varier. Une séquence à un niveau de la hiérarchie est interprétée comme un élément à un niveau supérieur.

Le processus de pensée se déroule ainsi : il y a d'abord prédiction puis action. Par exemple si je décide de me lever et d'aller dans la pièce voisine, je "visualise" (charge en mémoire) la pièce ainsi que la suite d'actions à effectuer, ensuite les actions sont effectuées éventuellement. On peut très bien seulement penser à se déplacer, sans pour autant exécuter l'action. Ce processus est permanent donc modifié en temps réel selon les perceptions qui surviennent. Cet apprentissage reposant sur la prédiction permet un comportement orienté en fonction d'un but, par opposition par exemple à un comportement issu d'un apprentissage par renforcement standard, qui est orienté en fonction du contexte (dans tel contexte, l'agent privilégie telle action car elle permet de maximiser la récompense sur le long terme).

Tous les événements non prédits (information inconnue, nouveauté, alerte) remonteront plus haut dans la hiérarchie alors que les informations courantes seront prédites et traitées quasi inconsciemment aux niveaux bas. Par exemple il est bien plus difficile de raisonner en marchant dans un endroit étranger où tout est nouveau que dans un endroit familier. Lorsqu'une information inconnue arrive en haut de la hiérarchie dans la zone correspondant en théorie à l'hippocampe, elle est mémorisée temporairement. Cette mémoire courante est accessible dans l'instant, et peut se renforcer vers les couches plus basses du cortex si l'information est beaucoup sollicitée. Cette vue de l'hippocampe comme sommet de la hiérarchie du cortex expliquerait le rôle décisif que joue l'hippocampe pour la mémoire. La perte de l'hippocampe implique en effet l'incapacité de mémoriser quoi que ce soit de nouveau, les souvenirs anciens étant néanmoins conservés.

2.3.3 Approches reposant sur l'utilisation de cartes corticales

La *théorie neuronale de la cognition* proposée par [Touzet13] est une autre proposition pour utiliser les observations de la neurophysiologie pour bâtir une théorie de la cognition. L'auteur insiste lui aussi sur le rôle des colonnes corticales et leur appartenance à des *cartes corticales*. Ces cartes sont selon lui le niveau naturel d'abstraction des processus cognitifs (Figure 2.6 (a)). Ainsi ce sont les colonnes corticales qui constituent les unités représentationnelles pertinentes pour implémenter une cognition neuro-inspirée, et non pas le neurone. Il propose un modèle basé sur un réseau de *cartes auto-organisatrices* (SOM) (Figure 2.6 (b)).

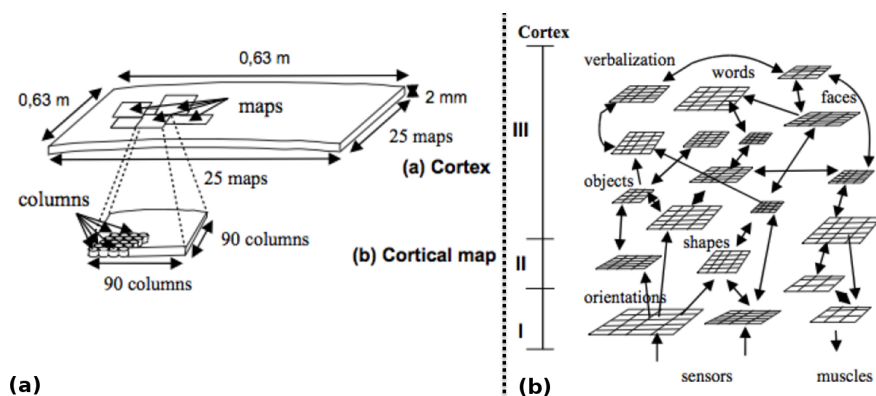


FIGURE 2.6 – (a) Découpage du cortex en cartes corticales composées d'un ensemble de colonnes. (b) Théorie neuronale de la cognition. Réseau hiérarchique de cartes corticales.

[Lefort10, Lefort13] s'intéressent au problème de l'apprentissage sensorimoteur en environnement continu, c'est à dire l'apprentissage de motifs spatio-temporels dans un flux de données multimodales. Ils proposent un modèle inspiré des principes introduits précédemment : *SOMMA* (Self-Organizing Maps for Multimodal Association) qui repose sur une architecture décentralisée connexionniste. Les unités computationnelles sont des *colonnes corticales* composées de trois couches, qui forment des cartes. Chaque modalité de l'environnement est associée à une carte modale auto-organisée (similaire aux SOM). Dans cette approche la multimodalité est prise en compte au niveau le plus bas de la hiérarchie, par une connectivité latérale entre les cartes de différentes modalités. Le *problème de l'amorçage* est donc abordé ici par l'auto-organisation parallèle de plusieurs cartes qui s'influencent mutuellement. Les travaux présentés par la suite dans [Lefort14] introduisent un processus d'évaluation basé sur la prédiction pour guider l'apprentissage.

Pour finir, on peut mentionner également un intérêt croissant dans les techniques de deep learning qui consistent généralement en des architectures multi-niveaux pour permettre la montée en abstraction par des opérations non linéaires successives sur les données. Il s'agit d'approches connexionnistes mais qui mettent donc l'accent sur l'aspect itératif de l'apprentissage, la discrétisation et la construction d'une représentation par des éléments reposant les uns sur les autres à différents niveaux d'abstraction. Voir [Bengio09] pour un état de l'art sur ce type d'approches.

2.3.4 Conclusion

Ces différentes idées convergent vers une vision active de la cognition en tant que réseau constitué d'unités représentationnelles qui s'organisent en fonction de l'expé-

rience suivant un processus constructif. La cognition serait davantage basée sur la mémoire (le résultat d'associations) que sur le calcul. Enfin il s'agit de processus décentralisés, fortement basé sur les mécanismes de rétroaction. Les aspects essentiels que nous retiendrons de ces approches sont les suivants :

- Discrétisation de l'expérience en motifs
- Capacité de représentation abstraite
- Mémorisation des structures de représentation en un réseau hiérarchisé
- Cognition comme un processus d'auto-organisation
- Ce processus utilise les motifs pour faire des prédictions

Pour résumer les principales idées évoquées jusqu'ici, on peut citer cette définition de l'intelligence proposée par [Delahaye14] :

"Quand on tente de formuler une définition générale de l'intelligence, vient assez naturellement à l'esprit l'idée qu'être intelligent, c'est repérer des régularités, des structures dans les données dont on dispose, quelle qu'en soit leur nature, ce qui permet de s'y adapter et de tirer le maximum d'avantages de la situation évolutive dans laquelle on se trouve. L'identification des régularités, on le sait par ailleurs, permet de compresser des données, et de prédire avec succès les données suivantes qu'on recevra. Intelligence, compression et prédiction sont liées ([Delahaye14])."

2.3.5 Les approches connexionnistes dans le cadre de l'AmI

Le système *ACHE* (*Adaptive Control of Home Environments*) proposé par [Mozer98] utilise les réseaux de neurones artificiels (RNA) afin de créer une "maison qui se programme toute seule". Le système mis en place doit être capable :

- D'observer le style de vie et les habitudes de ses habitants
- D'anticiper et satisfaire leurs besoins

Pour cela l'auteur indique qu'un apprentissage est nécessaire pour réaliser notamment les fonctions suivantes : reconnaître les activités des habitants, les motifs d'occupation, identifier les tolérances des utilisateurs, etc. Le système proposé dispose de plusieurs moyens de contrôle pour agir sur l'environnement :

- Chauffage de l'air
- Chauffage de l'eau
- Lumière
- Ventilation

Le principe général de ce système consiste à inférer des motifs afin de prédire des actions dans l'environnement. Lorsque les habitants interagissent avec le système pour modifier un paramètre de l'environnement, l'auteur considère que cela signifie que leurs besoins ne sont pas satisfaits. C'est cette information qui est directement utilisée pour entraîner le système. Pour cela, plusieurs réseaux de neurones traitent indépendamment différentes composantes de l'environnement (ex : lumière, ventilation, chauffage). Chacune de ces composantes est gérée indépendamment par un module

spécialisé. Ces modules reposent sur une même architecture, représentée par la Figure 2.7.

- *State Transformation* : transforme les informations instantanées des perceptions en valeurs temporelles ou statistiques (min, max, moy, var...) en se basant sur une fenêtre temporelle. Cela résulte en une représentation d'états qui donne plus d'informations sur l'environnement que les données brutes instantanées.
- *Occupancy model* : détermine si chaque zone est occupée ou non, par un ensemble de règles.
- *Prediction* : détermine l'état futur à partir de l'état courant. Ex. : *patterns* d'occupation, usage d'eau chaude. Il s'agit ici d'un réseau de neurone « *feed forward* » entraîné par rétro-propagation.
- *Set point Generator* : Spécifie la valeur de préférence (ex. température) sur une fenêtre temporelle.
- *Device Regulator* : Contrôle le matériel pour obtenir la valeur cible (*setpoint*).

Le *Set point Generator* et le *Device Regulator* sont implémentés par diverses techniques suivant les cas (ex. contrôle direct basé sur un modèle, apprentissage par renforcement, RNA).

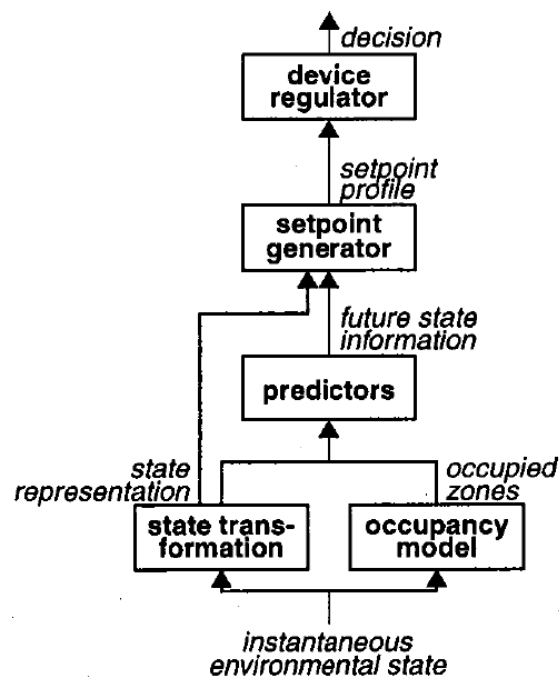


FIGURE 2.7 – Architecture du modèle ACHE (The neural network house)

Notons d'une part l'utilisation de modèles thermiques (connaissances expertes) à partir desquels les réseaux de neurones apprennent des déviations. D'autre part, les réseaux de neurones ne sont pas utilisés pour apprendre automatiquement des mo-

tifs à partir des données brutes. Ils sont utilisés en intermédiaire dans le traitement de l'information entre des modules qui interprètent les données brutes selon un modèle prédéfini et une couche d'apprentissage par renforcement qui permet d'apprendre les préférences utilisateurs. L'apprentissage qui est supervisé se fait ici en rapport à des objectifs prédéfinis, par exemple la minimisation de la consommation d'énergie et les objectifs sont traités indépendamment. Une des limites dans ce cas est que l'on ne peut utiliser la connaissance acquise que dans le cadre de l'objectif défini. Par exemple, on aimerait éventuellement pouvoir maximiser la consommation dans certaines circonstances (ex. : période de pic de production) au lieu de la restreindre. Enfin la nature de "boîte noire" des réseaux de neurones, du fait que leur structure interne est non compréhensible peut être préjudiciable dans le cadre de l'AmI. En effet, si l'on considère comme on l'a vu au chapitre précédent qu'il est important que la connaissance soit interprétable pour la communication avec l'utilisateur ou d'autres systèmes, l'utilisation de réseaux de neurones n'offre pas la possibilité d'extraire facilement les motifs qui représentent l'expérience du système. Les méthodes reposant sur l'utilisation de modèles connexionnistes telles que les réseaux de neurones sont très efficace pour réaliser un apprentissage ciblé d'un concept difficilement caractérisable à priori, grâce à la l'expressivité et la généralité de cette forme de représentation. Comme on le voit dans l'exemple présenté, l'utilisation de ce genre de méthode se fait ponctuellement dans une architecture plus large dans le contexte de l'AmI.

2.4 Le paradigme constructiviste

Dans cette partie, nous présentons tout d'abord le paradigme de l'apprentissage constructiviste en sciences cognitives. Ensuite nous poursuivons par une présentation des principaux travaux en IA, qui s'inscrivent dans cette approche. Puis, nous examinons les problématiques qui se posent pour l'application de ces techniques à des problèmes réels en environnement continu, notamment le *problème d'amorçage*. Pour cela nous nous intéressons plus spécifiquement à la robotique développementale. Pour finir nous présentons des travaux appliquant une approche constructiviste de l'apprentissage dans le domaine de l'AmI.

2.4.1 L'intelligence comme adaptation

En psychologie, la théorie constructiviste de l'apprentissage développée en particulier par [Piaget54] 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. Autrement dit, le sujet apprend en s'adaptant à un milieu ; c'est en agissant sur le monde qu'il le construit.

Piaget divise le développement psychologique de l'enfant en plusieurs périodes, chacune elle-même divisée en stades, conditionnant le suivant. Les différents moments du développement sont ([Wikipédia15a]) :

- La période de l'intelligence sensorimotrice (de la naissance à 2 ans).
- La période de l'intelligence préopératoire (de 2 à 6 ans).
- La période des opérations concrètes ou de l'intelligence opératoire (de 6 à 10 ans).
- La période des opérations formelles (de 10 à 16 ans).

Nous nous intéressons plus particulièrement à la première période. On peut définir la période de l'intelligence sensorimotrice ainsi :

"L'intelligence sensorimotrice est l'activité ordonnatrice et auto-organisée grâce à laquelle le sujet structure ses actions et les milieux avec lesquels il interagit, de manière à atteindre les buts qu'il se fixe. L'autoconstruction de cette intelligence commence dans les mois qui suivent la naissance et s'achève vers dix-huit mois, avec la construction des instruments de base de l'intelligence représentative, notamment la fonction symbolique" ([Fondation Jean Piaget15]).

La construction effectuée repose sur la notion de *schème*, élément de représentation de base d'une interaction. Un schème peut être défini comme :

"ce qui, dans une action, est transposable, généralisable ou différenciable d'une situation à la suivante, autrement dit, ce qu'il y a de commun aux diverses répétitions ou applications de la même action" ([Piaget73])

Cette notion est similaire à l'idée de représentation invariante décrite par Hawkins concernant la mémorisation de *patterns* (motifs).

Un schème est relatif à l'action et à ses conséquences perceptibles par le sujet et il peut être découvert par hasard et renforcé par le mécanisme de *réaction circulaire*. Ceci exprime le fait que lorsqu'une action aboutit à apporter une sensation agréable au sujet, ce mouvement tend spontanément à se répéter. Ce mécanisme est étudié par Piaget à partir de l'observation de ses enfants, et il en distingue plusieurs sortes constituant les différents stades de cette période ([Wikipédia15b]).

- La réaction circulaire innée (ou héréditaire) : Désigne les tout premiers réflexes moteurs possibles pour l'enfant. Par exemple le *réflexe de préhension* est présent dès la naissance chez le nourrisson. Lorsqu'on met un objet au contact de la paume sa main, l'enfant serre fermement ses doigts pour saisir l'objet.
- La réaction circulaire primaire (1-4 mois) : L'enfant répète des actions pour explorer un résultat intéressant trouvé auparavant de façon fortuite sur son corps, en répétant plusieurs fois la même action. Par exemple, le bébé va "par hasard" porter le pouce à sa bouche, et trouver le résultat intéressant. Il va donc répéter l'action plusieurs fois afin de reproduire la sensation plaisante que lui procure cette action.
- La réaction circulaire secondaire (4-8 mois) : L'enfant explore un résultat intéressant, trouvé lui aussi fortuitement, mais sur des objets. Par exemple, le bébé qui

tient une cuillère dans la main va, pour la première fois, la taper contre la table "sans faire exprès", et trouver le résultat intéressant (le son, le mouvement, la sensation que cela produit). Il va donc recommencer plusieurs fois d'affilée à taper la cuillère contre la table.

- La réaction circulaire tertiaire (12-18 mois) : L'enfant introduit des variations au fur et à mesure des répétitions. Il expérimente ainsi les différentes propriétés d'un objet. Il va, par exemple, comparer comment la cuillère en bois sonne par rapport à la cuillère en acier en les tapant toutes les deux sur la table l'une après l'autre, avec une intensité différente, de façon plus ou moins rapide.

Du point de vue du constructivisme, le processus cognitif peut être assimilé au processus d'adaptation de la théorie de l'évolution. C'est-à-dire que les structures mentales (la représentation) construites sont sans cesse mises à l'épreuve au regard de leur efficacité ou de leur utilité dans l'expérience, comme l'exprime [Von Glasersfeld84] :

"[...] pour un organisme cognitif, l'adaptation fonctionne à deux niveaux. Au niveau biologique, elle vise à la survie ; au niveau conceptuel, elle vise à l'élaboration de structures cohérentes et non contradictoires."

Cette adaptation est concrètement réalisée par l'évolution de l'organisation du système cognitif, comme explicité par les concepts d'*assimilation* et d'*accommodation*. L'assimilation consiste à intégrer de nouvelles données issues de l'expérience au sein des schèmes existants, et ainsi à les renforcer. En cela, l'expérience s'adapte à la représentation (*fit practice to theory*). L'accommodation consiste au contraire à modifier les schèmes existants en fonction des données. La représentation ne suffit plus à interpréter efficacement l'expérience, elle n'est plus *viable* à un certain point. La représentation s'adapte à l'expérience (*fit theory to practice*). Il n'est pas étonnant de constater que cette vision de l'acquisition de la connaissance correspond également à la démarche scientifique en général. Piaget considère en effet la science comme une forme évoluée de l'adaptation au même titre que l'intelligence, comme analysé par [Glasersfeld97].

Remarquons ainsi que du point de vue de l'épistémologie constructiviste, l'analogie entre la théorie de l'évolution et l'intelligence qui est utilisée en IA pour la résolution de problèmes complexes dans le cadre des algorithmes évolutionnistes, semble ainsi moins surprenante. L'efficacité des algorithmes génétiques pour trouver des solutions originales à partir d'une représentation réduite de bas niveau prouve en tout cas que cette approche est pertinente.

2.4.2 Application en IA

Pour utiliser ces concepts en IA, l'idée est donc d'assimiler l'agent artificiel (ex. le robot, le système ambiant) à l'organisme biologique (ex. l'humain) durant sa phase de développement. C'est à dire la période pendant laquelle l'agent explore les possibilités

de ses interactions sensorimotrices avec son environnement et construit une représentation du monde par incréments successifs qui gagnent en complexité, les nouveaux incréments reposant sur les précédents. Par conséquent cette vision de l'apprentissage est indissociable de la notion d'agent et d'interaction avec l'environnement, alors que cela importe peu dans d'autre domaine d'application de l'IA. Ce positionnement a été très clairement explicité par [Perotto10] dans sa thèse de doctorat.

"Selon la théorie psychologique constructiviste, le grand potentiel de l'intelligence humaine est la capacité de transformer ses propres structures intellectuelles, en les améliorant progressivement, dans un processus de complexification, gouverné par un besoin constant d'assimiler le monde avec lequel il interagit. Le constructivisme propose la notion de développement cognitif, en élargissant le concept d'apprentissage. Le développement cognitif est le processus qui conduit le sujet à la construction de nouveaux modèles de compréhension, à la création de nouveaux outils intellectuels, à l'intensification de l'élaboration des structures de la connaissance, à l'enrichissement des formes de représentation, pour rendre possible un traitement efficace des expériences complexes. Selon le modèle constructiviste, le potentiel de l'intelligence humaine découle du passage d'une intelligence sensorimotrice à une intelligence symbolique. C'est-à-dire que le sujet surpasse la frontière de la compréhension basée uniquement sur des sensations et des mouvements, pour organiser désormais ses expériences dans un univers de relations, de concepts, et d'objets abstraits. ([Perotto10])"

[Weng01] énonce la différence de principe entre la programmation traditionnelle d'une IA (*knowledge-based*) et l'approche développementale (*Autonomous Mental Development*) :

L'approche traditionnelle peut être résumée comme cela :

- On part d'une tâche (connue par le concepteur)
- On conçoit une représentation spécifique pour cette tâche
- On crée un programme de résolution de la tâche qui utilise cette représentation
- On lance le programme sur la machine

L'approche développementale quant à elle se résume par ces étapes :

- On conçoit un corps adapté à l'environnement du robot
- On conçoit un programme de développement
- Initialement (à sa "naissance") le robot démarre le programme de développement à partir d'une représentation vierge.
- Le robot apprend à partir de son expérience. L'humain peut "éduquer" le robot à effectuer des tâches en interagissant avec lui.

Dans ce paradigme, le robot doit traverser une période plus ou moins longue de développement, de "l'enfance à l'âge adulte". [Weng01] rappelle les caractéristiques clés des programmes développementaux :

- Ils ne sont pas spécifiques à une tâche (non *task-specific*).

- Les tâches sont globalement inconnues à priori.
- Le robot génère donc une représentation pour des tâches inconnues (la représentation n'est pas extrinsèquement orientée)
- Il s'agit d'un apprentissage en temps réel.
- C'est un apprentissage ouvert (open-ended) : l'apprentissage d'un concept offre la possibilité d'apprendre un nouveau concept.

2.4.3 L'apprentissage sensorimoteur

La représentation construite par l'agent est un modèle anticipatoire des interactions possibles entre l'agent et son environnement, comme l'explique Perotto.

"Lorsque l'agent ne connaît pas les règles qui déterminent la dynamique de fonctionnement de son environnement, il faut qu'il les découvre peu à peu en se basant sur ses propres observations. Apprendre un modèle du monde signifie, pour un agent, construire de façon autonome une représentation interne de la dynamique d'interaction avec l'environnement à partir de son expérience. Particulièrement, dans une approche constructiviste, cet apprentissage doit être fait de façon progressive, où un modèle précédent plus grossier et moins adapté est graduellement raffiné. En général, l'entrée pour ce type de problème est un flux ininterrompu de perceptions successives faites par l'agent à travers ses senseurs, décrites dans l'espace défini par les propriétés de l'environnement qu'il est capable d'observer, parallèlement à un flux d'actions exécutées par l'agent à travers ses effecteurs. La tâche de l'algorithme d'apprentissage de modèles du monde est de conduire à l'induction d'une structure telle qu'il soit possible de prédire les perceptions futures, en se basant sur les perceptions et les actions actuelles. Un modèle du monde est donc un modèle anticipatoire, qui décrit, du point de vue de l'agent, la régularité des transformations des propriétés de l'environnement au fil du temps, en fonction de l'observation qu'il fait et des actions qu'il exécute." ([Perotto10])

Dans la théorie de l'apprentissage constructiviste, la première étape consiste en un apprentissage sensorimoteur. L'apprentissage sensorimoteur doit permettre à l'agent d'identifier des régularités dans ses interactions avec l'environnement, c'est-à-dire mettre en relation des actions et des perceptions. Même si l'intérêt de l'approche réside dans l'idée de complexification de la cognition de l'agent à partir de cette première phase, il n'en reste pas moins que cette phase d'apprentissage sensorimoteur pose à elle seule certains problèmes complexes, en particulier dans son application à des environnements complexes. Il y a notamment deux catégories de travaux en IA développementale. Certaines méthodes sont implémentées dans des environnements discrets, alors que d'autres sont appliquées à des systèmes physiques à environnement

continu comme c'est le cas en robotique. [Guerin11] offre une présentation complète de ces différentes approches.

2.4.4 Mécanismes d'apprentissage de schémas

2.4.4.1 Principe général

Le *schema learning* est une technique d'apprentissage constructiviste proposée initialement par Drescher [Drescher87] ([Drescher91]) qui s'inspire de la théorie de Piaget sur l'apprentissage dans les premiers stades du développement. Dans ce modèle, les régularités primitives de l'interaction sensorimotrice de l'agent avec son environnement sont exprimées par des *schémas* (dérivé de la notion de schème), structures qui représentent la prédiction des effets d'une action dans un contexte spécifique. Un schéma est un triplet *Contexte-Action-Résultat*, où le contexte et le résultat sont définis comme un ensemble d'*items* binaires, qui représente l'ensemble des perceptions (voir Figure 2.8).

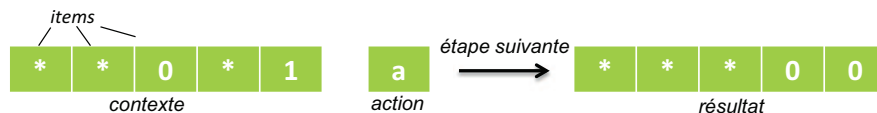


FIGURE 2.8 – Schéma

L'un des intérêts de ce modèle est qu'il repose sur un couplage "situation, action→prédiction", par opposition à un couplage "situation→action". C'est-à-dire que l'apprentissage ne représente pas quelle action il faut effectuer dans chaque situation mais plutôt quelles pourraient être les conséquences de chaque action dans chaque situation. La représentation construite n'est pas un modèle du monde ou un comportement cible, mais plutôt une représentation des capacités d'interaction de l'agent avec son environnement. La représentation n'est pas fonction d'un but qui est externe au système d'apprentissage, elle met seulement en relation des éléments de son expérience (sauf en considérant que ce but externe est celui de faire des prédictions). Mais comme on l'a vu précédemment, tout porte à croire que la capacité de prédiction est au cœur des mécanismes régissant la cognition.

2.4.4.2 Amorçage de l'apprentissage sensorimoteur

Un tel apprentissage sans connaissances à priori n'est pas sans présenter quelques difficultés. Premièrement, la variabilité des effets d'une action dans des circonstances différentes, rend le résultat d'une action difficile à prévoir. Deuxièmement, un même résultat peut avoir des causes diverses. Troisièmement, suite à l'exécution d'une action, le résultat peut être perçu parmi un ensemble d'évènements proches dans le temps mais

sans rapport avec l'action en question. De ce fait, des résultats même très pertinents peuvent être très difficiles à isoler tant que l'on ne connaît pas les circonstances correspondantes (le contexte de l'action). D'autre part, ces circonstances elles-mêmes ne peuvent être identifiées sans savoir préalablement à quel résultat elles correspondent. Ainsi on se trouve face à un problème d'amorçage (*bootstrapping problem*) explicité par [Drescher91] (mentionné comme problème de la poule et l'œuf).

Drescher propose une solution possible à ce problème, par le mécanisme de *marginal attribution*. Dans cette optique l'angle de vue utilisé consiste à regarder une action et l'état de l'environnement qui suit immédiatement l'action, afin de détecter des schémas. L'idée générale est de distinguer deux notions pour évaluer un schéma : sa *pertinence* (*relevance*) et sa *fiabilité* (*reliability*). Plus précisément, un résultat est pertinent pour une action si ce résultat apparaît beaucoup plus souvent lorsque l'action est effectuée que lorsqu'elle ne l'est pas, et ce indépendamment de la fréquence de ce résultat, même lorsque l'action est effectuée. Il s'agit bien sûr d'une approche statistique qui permet de détecter un lien entre l'action et le résultat, même si le résultat est rare car ne survenant que sous certaines conditions très précises. Cela permet également d'évincer les co-occurrences fortuites, en demandant une différence de fréquence importante.

Ayant identifié un résultat pertinent, l'étape suivante est de chercher sous quelles conditions on peut le rendre fiable. Il s'agit pour cela de trouver le contexte permettant de distinguer les cas de succès des cas d'échecs. Notons que cette stratégie, est similaire à la stratégie employée par [Aztiria09] dans son modèle *PUBS*. Il s'agit dans un premier temps de détecter des corrélations, sur la base de différences de fréquence dans les cooccurrences de certaines perceptions, puis de chercher dans un second temps à améliorer ce début de représentation en cherchant à spécifier les conditions nécessaires à une prédiction plus fiable.

Le mécanisme de *marginal attribution* pour l'amorçage de l'apprentissage de schémas, nécessite l'entretien de statistiques sur chaque couple action/résultat possibles et donc un coût en ressources (calculs et mémoire) important. Cette méthode est possible car la proximité temporelle entre une action et ses conséquences est induite par le modèle, le temps étant ici géré de manière discrète, découpé en étapes de la boucle sensorimotrice. Le schéma est un élément représentationnel ou *invariant sensorimoteur*. Sa structure est définie à priori mais informe sur la nature de la représentation élémentaire, et pas sur l'environnement en lui-même.

Au départ, le système possède une population de schémas primitifs vierges, un pour chaque action primitive. Le système commence à collecter de l'expérience en activant ces schémas primitifs aléatoirement. Au fur et à mesure de l'activité, de nouveaux schémas apparaissent, se spécialisent ou se généralisent en fonction de l'expérience. Le contexte expérimental est celui d'un robot dans un environnement simulé, analogue à celui d'un enfant assis, jouant avec des objets posés devant lui. L'objectif étant bien sûr l'apprentissage de schémas sensorimoteurs mettant en relation actions et perceptions. Par exemple, il obtient des schémas permettant à l'agent de coordonner le déplacement

du bras articulé en fonction de la position de l'objet à saisir dans le champ visuel et de la position du bras.

2.4.4.3 Capacité d'abstraction et de généralisation

L'autre aspect fondamental de ce type d'apprentissage est l'invention de concept et la montée en abstraction. C'est-à-dire la capacité à définir des concepts radicalement nouveaux, parce que désignant des entités fondamentalement différentes de ce qui est déjà représenté, par exemple dépassant le spectre de la description sensorimotrice permise par un seul schéma.

Dans le *schema mechanism* de Drescher, une solution proposée consiste en l'ajout d'*items synthétiques* pour étendre le contexte des schémas. Ce mécanisme permet au système d'augmenter son ontologie (l'ensemble des variables ou items) par la création d'une nouvelle notion, lorsque les mécanismes mis en place ne suffisent pas à distinguer deux situations (c'est-à-dire : un schéma est pertinent mais on ne parvient pas à le rendre plus fiable). Les items synthétiques représentent les conditions nécessaires pour qu'un certain schéma soit valide. Ils ne sont pas liés aux perceptions, c'est pourquoi au départ leur valeur n'est connue que rétroactivement à la prise de l'action en fonction du succès ou de l'échec du schéma. Mais au fil de l'expérience, les conditions qui en déterminent la valeur sont apprises progressivement en relation avec l'activation d'autres schémas. Ainsi ces items acquièrent un sens du point de vue du système, par l'utilité qu'ils ont à améliorer sa capacité de prédiction. Un autre mécanisme permettant la montée en abstraction consiste à créer des actions composites, qui sont formées de plusieurs schémas pour atteindre un but.

[Perotto07] reprend le modèle proposé par Drescher pour vérifier que le système est capable d'apprendre des propriétés cachées d'un environnement partiellement observable. L'auteur souligne l'importance de la transition entre l'intelligence sensorimotrice et l'intelligence symbolique, et explique que les travaux de Piaget rendent compte de la façon dont l'humain affronte ce problème lors de son développement. Selon lui, un agent doit pouvoir outrepasser les limites de ses perceptions, en utilisant des concepts de plus haut niveau que des schémas sensorimoteurs, ainsi il ne doit pas être enfermé dans un vocabulaire représentationnel rigide. Pour cela les items synthétiques offrent une piste pertinente. Perotto applique le *schema learning* à des problèmes simples, et montre que le mécanisme des items synthétiques est capable de représenter des notions nouvelles par rapport au cadre sensorimoteur des schémas telles que : une propriété cachée de l'environnement (hors spectre des perceptions), une étape dans une séquence ou une propriété abstraite.

2.4.5 La robotique développementale

2.4.5.1 Présentation des principaux concepts

La robotique développementale est une discipline qui s'inscrit dans le paradigme constructiviste de l'apprentissage. Naturellement, ce domaine est confronté à la gestion des environnements complexes pour les robots réels. C'est pourquoi, il est intéressant de passer en revue les techniques développées dans ce domaine pour gérer ce problème et considérer les applications possibles pour un système ambiant. La robotique développementale prend en effet en compte le fait que les organismes biologiques intelligents passent par une phase de développement avant d'atteindre leur forme adulte. L'idée générale est d'utiliser les connaissances acquises sur le développement ontogénétique des organismes vivants pour la conception de robots. Dans son article, Lungarella [Lungarella03] propose une analyse détaillée de ce paradigme, ses justifications et ses perspectives, ainsi qu'un état de l'art. Les problématiques évoquées par ce thème de recherche sont très proches de celles auxquelles nous sommes confrontés pour l'AmI. Ceci provient du fait qu'un environnement intelligent est comme un robot constitué d'un ensemble de perceptions et d'actions, et confronté à la complexité et la dynamique du monde réel. Potentiellement un ensemble d'objectifs de nature très différente peuvent être demandés au système dans une infinité de situations possibles. C'est pourquoi le système doit être fortement adaptatif et ceci passe par un apprentissage que l'on ne peut qu'envisager comme autonome, c'est-à-dire ne reposant pas sur une représentation prédéfinie mais au contraire construite au fil de l'expérience. Dans une approche développementale le concepteur doit ainsi être le plus possible absent du système, en créant seulement une architecture de développement, qui guidera le robot pour construire sa propre représentation de lui-même et de son environnement. Les modèles de sensations et d'actions n'ont pas besoin d'être spécifiés par l'ingénieur, mais ils émergent de la propre expérience sensorimotrice du robot. Une des problématiques intéressantes réside dans la faculté du robot à percevoir les conséquences de ses propres actes. Le robot doit pour cela apprendre à isoler certaines de ses perceptions dans le temps. Un des intérêts de l'approche développementale est que l'apprentissage est facilité par le découpage structuré des compétences et par l'augmentation graduelle de la complexité. Cette augmentation graduelle de la complexité peut être due à des contraintes particulières pouvant "freiner" le développement à des moments précis. Par exemple au début de sa croissance l'enfant a des capacités motrices et sensorielles limitées, ainsi que des limitations au niveau du système nerveux. Ceci peut à première vue apparaître comme un défaut, mais on peut également considérer que c'est un avantage du point de vue de l'apprentissage, car cela permet de diminuer ou prévenir la surcharge d'information. Si l'agent (ex. : l'enfant) est soumis à une trop grande complexité précocement, il pourrait être submergé d'information et cela nuirait à son développement. Ainsi une gestion des limitations peut être bénéfique voire nécessaire pour l'apprentissage.

2.4.5.2 Motivation intrinsèque

Un robot autonome doit être doté d'un ensemble initial de valeurs et de motivations afin d'interagir avec l'environnement pour amorcer le processus de développement. Le rôle de ces prérequis est également de guider l'apprentissage et promouvoir l'exploration. Les principales questions sont de savoir d'une part à quel point ces prérequis doivent être spécifiques ou généraux ; d'autre part que doit-on prédéfinir et qu'est-ce qui doit être acquis ? Dans la théorie de Piaget nous avons vu que les mécanismes de réaction circulaire innée, permettaient d'illustrer l'activité initiale de l'enfant lui permettant de découvrir les premiers schèmes primaires.

La question qui se pose ici est : qu'est-ce qui pousse l'agent à agir et à apprendre ? On peut considérer qu'un objectif d'un processus développemental est que le système puisse toujours se placer dans une situation où il est proche de savoir quelque chose, puis de l'apprendre. Aucune tâche ou objectif n'est prédéfinie. Le robot doit donc être *intrinsèquement* motivé à choisir ses propres actions. Cette idée de *motivation intrinsèque* est opposée à la définition d'un objectif externe qui conditionne l'apprentissage du robot. [Oudeyer07] définit ce concept et passe en revue différentes implémentations possibles. Par exemple, le robot peut être doté d'une certaine curiosité ou soif d'apprentissage qui s'exprime par sa tendance à se focaliser sur les "niches de progression". Cette idée peut correspondre à des situations qui ne sont ni trop prédictibles ni pas assez. [Barto04] propose ainsi un système de motivation intrinsèque qui se base sur les événements imprévus. Comme nous l'avons montré précédemment, la capacité à faire des prédictions est une partie du mécanisme de développement, à partir duquel la cognition est construite. Des fonctions cognitives de haut niveau comme la planification peuvent être interprétées comme une façon de prédire les conséquences d'une série d'évènements. Le processus de prédiction peut être envisagé comme tâche de fond permanente, qui se concrétise par des actions lorsque les prédictions concordent avec la motivation intrinsèque de l'agent.

On ne peut néanmoins se contenter de spécifier une évaluation de la qualité de la prédiction comme seul moyen de guidage de l'activité du système. En effet dans ce cas, on peut être confronté au problème suivant : l'agent a intérêt à ne pas agir du tout afin de maximiser son score de prédiction (*dark room problem*). Il faut par exemple intégrer une forme de motivation incitant à une recherche de nouveauté.

Un autre exemple est celui de [Steels04] qui propose un système de motivation intrinsèque basé sur le *principe autotélique*. Ce principe repose sur l'hypothèse que la motivation d'un sujet à exécuter une tâche pour elle-même provient du fait que la complexité de cette tâche est adaptée à ses compétences. C'est à dire que la tâche est suffisamment complexe pour ne pas être ennuyante, mais suffisamment facile pour ne pas paraître insurmontable. Ceci explique pourquoi le sujet paraît apprendre plus facilement lorsqu'il exécute des tâches qu'il apprécie. La motivation intrinsèque repose dans ce cas sur le maintien d'un équilibre entre les compétences de l'agent et la difficulté des

tâches qu'il s'attribue. C'est pourquoi lorsque le système a appris une tâche, il tendra vers une tâche un peu plus complexe. Cette approche pose cependant des difficultés de modélisation car l'évaluation de la complexité des tâches et de la compétence de l'agent n'est pas évidente.

Oudeyer observe que malgré une grande diversité d'approches différentes, elles convergent toutes sur une même idée : la motivation intrinsèque est définie en fonction de la relation entre le flot de données sensorimotrices d'une part, et les connaissances et compétences de l'agent d'autre part.

"Each of the described models defines a certain interpretation of intrinsic motivation in terms of properties of the flow of sensorimotor values and of its relation to the knowledge and know-how of the system independently of the meaning of the sensorichannels that are involved" ([Oudeyer07]).

2.4.6 Applications des approches constructiviste ou développementales aux environnements continus

Certains travaux se placent dans la continuité du travail de Drescher en essayant de le faire évoluer pour limiter l'importante charge en ressources, et ouvrir la voie à des applications concrètes. En effet le modèle proposé initialement par Drescher reposait sur un environnement très simple. Un contexte était défini par un vecteur de valeurs binaires (les *items*), et le temps était géré pas-à-pas (les actions ayant des conséquences instantanées).

L'architecture CLASM de [Chaput03] permet d'améliorer le *schema mecanism*, notamment grâce à l'utilisation de *cartes auto-organisatrices* (SOM) hiérarchiques pour compléter le mécanisme de *marginal attribution*. Il montre que l'utilisation de cartes auto-organisatrices pour exécuter le processus de marginal attribution est plus efficace tout en permettant d'arriver aux mêmes résultats que Drescher pour le même contexte expérimental. Même si dans cette expérience, l'environnement est discret et le temps est géré pas-à-pas, l'auteur pense que cette alternative permet d'envisager d'appliquer le *schema learning* à des problèmes du monde réel, tel que celui du robot mobile. On peut donc conserver l'idée générale du mécanisme de marginal attribution pour amorcer la construction de schéma sensorimoteur, mais différentes implémentations sont possibles.

[Holmes05] proposent une version étendue du *schema learning* de Drescher en ajoutant trois améliorations notables. Premièrement les *items* peuvent avoir différentes valeurs discrètes arbitraires. Deuxièmement, l'environnement peut être modélisé par un POMDP (Partially Observable Markov Decision Process). Troisièmement, la gestion des *items* synthétiques est améliorée grâce à un mécanisme de prédiction. Des résultats satisfaisants sont présentés pour l'application de cette méthode d'apprentissage à un problème de modélisation de la parole. Les auteurs soulèvent deux problèmes majeurs qui restent ouverts.

- La gestion des variables continues.
- La gestion des actions dont les effets surviennent après une durée indéterminée, et pas seulement l'étape suivante de la simulation.

[Provost06] proposent un système (SODA : *Self-Organizing Distinctive-state Abstraction*) qui utilise un processus de discrétisation basé sur les SOM couplé à un apprentissage par renforcement dans un problème de navigation de robot. Cette fois les données proviennent directement d'un environnement continu. Le processus d'apprentissage se déroule en trois étapes :

- Apprentissage d'un ensemble d'états *distinctifs* de l'environnement par une SOM alors que le robot explore l'environnement grâce à des actions primitives.
- Construction d'un ensemble d'actions de haut niveau pour passer d'un état distinctif à un autre.
- Apprentissage de politique liée à une tâche de haut niveau (par renforcement) dans l'espace des perceptions et actions abstraites ainsi générées.

Dans le même esprit, [Linaker01] 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. L'extraction d'évènements est utilisée pour gérer le problème de la réponse à une action après une durée indéterminée (*delayed response task problem*). Pour cela ils utilisent plusieurs niveaux de traitement de l'information.

- Le niveau le plus bas travaille avec les données brutes, et utilise un mécanisme de classification non supervisé, pour classer ces données.
- Au second niveau, l'algorithme d'extraction d'évènements (transitions entre les classes) travaille sur ces données classifiées.
- Le troisième niveau n'est appelé que lorsque un évènement intervient, il ne dépend donc plus de l'échelle de temps des niveaux plus bas. Il y a un filtrage des évènements pertinents.

La discrétisation de données brutes en évènements permet d'apprendre une politique d'action à un plus haut niveau (celui de l'apprentissage par renforcement), donc il est clair que les résultats du mécanisme d'extraction d'évènements vont drastiquement influencer le reste du système.

On remarque que dans toutes ces approches, la gestion de l'environnement passe par un traitement en deux étapes. Premièrement une étape de discrétisation est réalisée, fournissant ainsi une représentation discrète reposant sur des états ou des évènements. Deuxièmement, une technique d'apprentissage est appliquée à cet environnement discret ainsi généré. Le fait que la discrétisation ne soit pas réalisée à priori à la conception, mais effectuée par un processus automatique est un étape importante dans la réalisation de l'apprentissage sensorimoteur pour des agents en environnements continus. Néanmoins, on pourrait regretter que le processus de classification à bas niveau soit unique et défini à priori. En effet, pour une même variable, les classes d'évènements générées peuvent être différentes en fonction des motifs 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. Cependant si l'on ne définit pas complètement le processus de discrétisation, l'apprentissage ne repose sur rien. Il s'agit là d'un problème d'amorçage que nous présentons dans le paragraphe suivant.

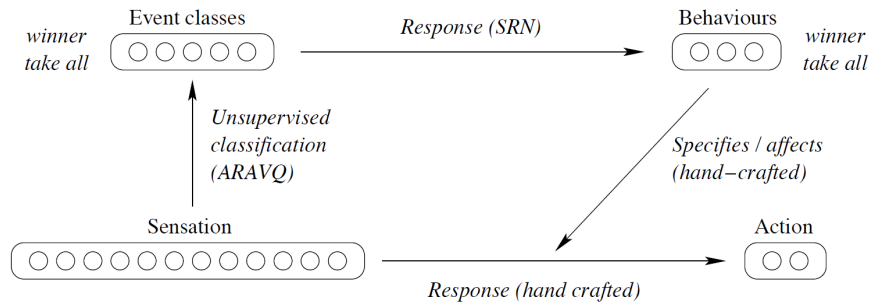


FIGURE 2.9 – L'architecture proposée par [Linaker01] pour le robot mobile confronté au *delayed response task problem*

[Nolfi99] proposent quant à eux un modèle d'apprentissage par prédiction utilisant des réseaux de neurones (RNA) qui sont entraînés à prédire l'état suivant des capteurs d'un robot se déplaçant entre deux pièces. Il s'agit d'une approche développementale dans laquelle l'apprentissage se déroule en plusieurs phases, et sur plusieurs couches. Durant la première phase, la première couche apprend à reconnaître des régularités de bas niveau (murs, coins, corridor). Ensuite, la deuxième couche étant connectée en sortie de la première, elle apprend des régularités de plus haut niveau tel que "le mur gauche de la grande pièce". Les auteurs proposent deux pistes intéressantes d'amélioration de leur modèle :

- Premièrement, étudier différents moyens de compresser l'information temporelle.
- Deuxièmement, introduire un mécanisme de feedback provenant des couches hautes pour influencer les couches basses.

2.4.7 Le problème d'amorçage

2.4.7.1 Définition

S'il existe des régularités dans l'interaction système-environnement, et qu'un processus d'apprentissage est capable de construire une représentation interne de ces régularités, comment ce processus commence-t-il à former les régularités ? L'idée de construction suppose un processus incrémental qui part d'une étape initiale, quelle est-elle ?

Pour illustrer ce problème, on pourrait comparer le flot constant de perceptions à une pelote de fils entremêlés. Cet ensemble peut paraître extrêmement complexe à considérer dans sa globalité. Mais en considérant un "*point de vue*" particulier (ex. on regarde seulement les fils bleus), on peut s'apercevoir facilement d'une régularité (ex. : tous les fils bleus sont noués à un fil rouge). On peut donc effectuer une prédiction. Les *points de vue* nécessaires peuvent être variés suivant les différents types de régularités à apprendre (ex. tous les couples de fils noués deux fois entre eux sont noirs et blancs), et il peut y en avoir une infinité.

Dans le cas de l'apprentissage de régularités entre variables continues, un *point de vue* possible pour considérer une variable en termes d'évènement pourrait être de chercher des variations de forte amplitude peu fréquentes. Des *points de vue* pour associer des évènements pourraient être : la proximité temporelle entre des évènements ; la proximité spatiale ; la séquentialité d'évènements, etc.

Une question se pose si l'on veut apprendre des régularités à partir d'une expérience sous la forme d'un flot de données brutes sans indication préalable sur la nature des concepts recherchés : faut-il envisager une infinité d'angles de vue et leurs combinaisons possibles ? Ou bien y-a-t-il un mécanisme qui permet d'apprendre toutes sortes de régularités à partir d'un nombre fini d'angles de vue initiaux ?

La discrétisation des données brutes en termes d'évènements détermine ce qu'il va être possible d'apprendre. D'un autre côté, en sachant ce que l'on veut apprendre, on pourrait guider le processus de discrétisation pour que celui-ci fournisse les éléments les plus adaptés à la représentation. Ce problème téléologique est caractéristique de l'étude des systèmes vivants. Une solution possible à ce problème est d'avoir une réelle dynamique entre les deux niveaux d'apprentissage, c'est-à-dire de permettre au mécanisme de discrétisation d'évoluer, guidé par une boucle de rétroaction provenant de l'apprentissage à plus haut niveau. Les deux processus s'influencent mutuellement. Pour les organismes biologiques, ce double apprentissage est probablement réalisé 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. Cette position est soutenue par [Kuipers15] :

"It is tempting to try to escape this problem by assuming that the foundational representations are innate when the individual is born, and so need not be learned. But this only pushes the learning problem onto the species, which must learn this knowledge over evolutionary time. We believe that in many ways developmental learning and evolutionary learning are similar, except that search is depth-first in one and breadth-first in the other. So we pretend that all learning is done by the individual, and postpone the decision of where to place the evolutionary/developmental boundary."
([Kuipers15])

On retrouve ce problème de la dépendance entre le processus de discrétisation de l'expérience et l'apprentissage exprimé différemment dans [Clark97]. Les auteurs pro-

posent de différencier les problèmes d'apprentissage de régularités à partir de données brutes en deux catégories :

- Apprentissage de *type-1*¹ : La régularité est directement présente, statistiquement parlant, dans la base de données. C'est-à-dire que l'on peut trouver la régularité en examinant la matrice des probabilités conditionnelles.
- Apprentissage de *type-2* : Il faut d'abord opérer un traitement spécifique des données brutes. La régularité est perceptible en considérant les relations entre les éléments de plus haut niveau issus de ce traitement.

La plupart du temps la présentation adéquate des données effectuée d'une manière ou d'une autre par un concepteur, permet de se ramener au *type-1*. Dans le cas du *type-2*, l'espace des traitements possibles des données brutes est infini, c'est l'espace des fonctions calculables. Ceci implique une différence de complexité considérable entre les deux types de problèmes. La solution passe donc par une recherche de ces traitements (ou "discrétisation") habituellement réalisée par le concepteur si besoin, qui fournissent des représentations intermédiaires rendant possible cet apprentissage.

"[...] a variety of existing learning algorithms tend to rely predominantly (and in some cases exclusively) on the extraction of a specific type of regularity from a body of input data. This type of regularity lies close to the surface of the training data, in the form of pronounced frequency effects and is thus fairly straightforwardly extracted by a variety of direct sampling methods. Some appearances to the contrary, the extraction of these (type-1) regularities is really all we currently know how to achieve - and no wonder, once the size of the search space for the other form is appreciated. The extraction of the more opaque type-2 regularities is not, however, impossible. The crucial maneuver in such cases is somehow to trade achieved representation (or perhaps on occasion real-world action) against computational search. Achieved representational states act as a kind of filter or feature detector allowing a system to recode an input corpus in ways that alter the nature of the statistical problem it presents to the learning device. Thus are type-2 tigers reduced to type-1 kittens." ([Clark97])

Par ailleurs, les auteurs indiquent qu'il serait intéressant que le système n'ait pas simplement des processus de traitement figés, mais puisse recoder ses représentations de façon endogène. Par exemple, si le processus de discrétisation s'adapte à un objectif d'apprentissage, les représentations ainsi créées sont disponibles pour un autre objectif. Ainsi on considère qu'une boucle de rétroaction de l'apprentissage vers le processus de discrétisation peut ouvrir la voie à une nouvelle possibilité d'apprentissage. Ce genre de mécanisme reflète bien l'idée d'un apprentissage développemental et incrémental.

En résumé, le *problème d'amorçage* est caractéristique de l'apprentissage des motifs sensorimoteurs en environnement continu à partir des données brutes de capteurs in-

1. Notation employée par [Clark97]

connus [Kuipers06]. Ce problème peut être exprimé de différentes façons dans la littérature, mais la plupart du temps il est ignoré puisqu'on s'arrange pour que les données soient adaptées aux algorithmes d'apprentissage que l'on souhaite utiliser. Cela est dû au fait que pour beaucoup de problèmes, l'effort de présentation des données est faible. Dans un domaine tel que celui de l'AmI, nous avons montré que cet effort de modélisation est possible si l'on se restreint à des sous-objectifs avec un système relativement stable, mais impossible si l'on considère le paradigme dans son expression la plus générale. C'est-à-dire le cas d'un système constitué d'une population de capteurs/actionneurs inconnus, dans un environnement inconnu, avec des objectifs d'apprentissage non prédéterminés, intervenant sur des échelles de temps hétérogènes. Nous considérons qu'une solution de ce problème passe tout d'abord par une modélisation systémique de la résolution de deux tâches d'apprentissage interdépendantes : la discrétisation de l'environnement d'une part, la création d'une représentation comme structure prédictive d'autre part.

2.4.7.2 Exemples de travaux répondant au problème d'amorçage

Mugan propose un processus similaire au système de *marginal attribution*, mais capable de gérer des variables continues ([Mugan07a, Mugan07b, Mugan12]). La discrétisation de l'expérience est réalisée par la création de *bornes*, pour transformer les variables continues en variables discrètes qui représentent un intervalle sur le domaine des valeurs possibles de la variable. Un changement de valeur de la variable discrète, c'est-à-dire le passage d'un intervalle à un autre, est considéré comme un "événement". Pour chaque variable continue, deux variables discrètes sont créées. Une pour l'*amplitude* et une pour le *sens de variation* de la variable continue. Le temps est géré de manière discrète, par étapes de temps successives. L'apprentissage consiste ensuite à prédire un événement à partir de l'occurrence d'un autre sous forme de règles, puis de fiabiliser cette prédiction en spécifiant un contexte d'application de la règle. On retrouve ici également la méthode employée par Drescher présentée précédemment. Le cadre expérimental utilisé (robot + environnement, voir Figure 2.10) est similaire à celui employé par [Drescher91] et [Chaput03], à l'exception des variables continues.

L'association entre deux événements repose initialement sur une condition d'ordre temporel : les deux événements doivent être proches dans le temps (contingents). La contingence d'un événement par rapport à un autre peut être de deux types :

- Contingence prospective : le fait qu'un événement A survienne après un événement B, un peu plus souvent que le hasard
- Contingence rétrospective : le fait qu'un événement A survienne avant un événement B, un peu plus souvent que le hasard

Une règle est définie par : $\langle C : E_1 \rightarrow E_2 \rangle$. C est un contexte composé d'un ensemble de variables discrètes de type amplitude. E_1 est un événement se produisant "juste avant" E_2 . La notion de proximité temporelle est exprimée par un nombre maximal de pas de temps à ne pas dépasser entre les deux événements.

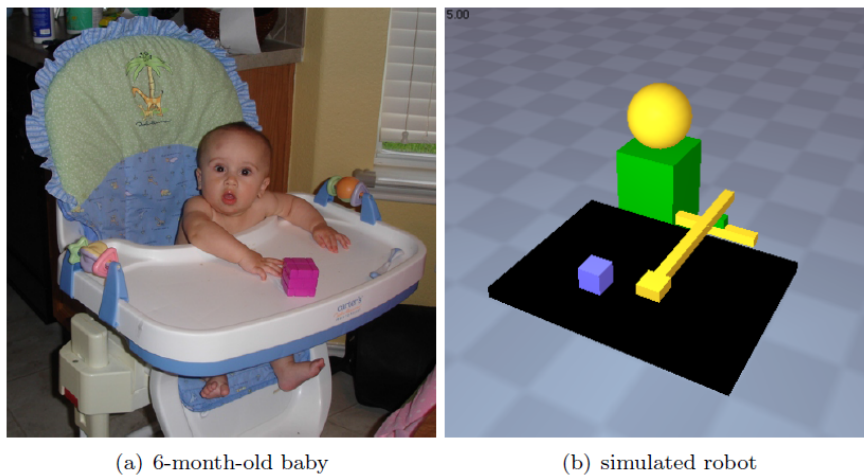


FIGURE 2.10 – L'environnement simulé du robot nourrisson ([Mugan07a])

L'obstacle pour détecter les contingences, est qu'il faut d'abord trouver les distinctions pour caractériser un évènement. Un aspect intéressant est qu'il n'y a pas de distinction fondamentale entre action et perception, et donc la structure représentationnelle élémentaire est plus générique qu'un schéma classique. Certaines variables sont néanmoins définies comme des variables "motrices", ce qui permettra à l'agent d'agir lors de l'activation des règles apprises pour lesquelles elles apparaissent dans la partie gauche. Le robot commence par effectuer des actions aléatoires (*motor babbling*) pour apprendre les premiers schémas. Ensuite un apprentissage actif est mis en place afin que le robot se focalise plus rapidement sur des zones intéressantes de son espace de recherche. Pour cela, des objectifs sont spécifiés : une variable doit par exemple atteindre une certaine valeur. Le robot va alors choisir un objectif et essayer d'utiliser les règles connues pour l'atteindre.

L'aspect le plus intéressant de ce modèle est le feedback qui est mis en place entre : le processus d'évaluation des prédictions et le processus de discrétisation (voir Figure 2.11). Ceci permet en effet à un agent d'affiner son apprentissage, en fonction de ce qu'il a déjà appris. Concrètement, cela est fait en favorisant la création de nouveaux états (en créant des bornes supplémentaires), pour apprendre de plus précis et de plus nombreux évènements et par conséquent de nouvelles règles (prédictions). Cette création de bornes supplémentaire illustre l'influence que va avoir l'apprentissage de haut niveau sur le processus de discrétisation. Deux raisons différentes peuvent entraîner cette création de bornes.

- La création d'une nouvelle borne permet d'améliorer le score d'une règle.
- La borne correspond à une valeur remarquable liée à un évènement. Pour chaque évènement, on entretient les histogrammes des valeurs de toutes les variables. Si l'histogramme représentant la distribution des valeurs d'une variable au mo-

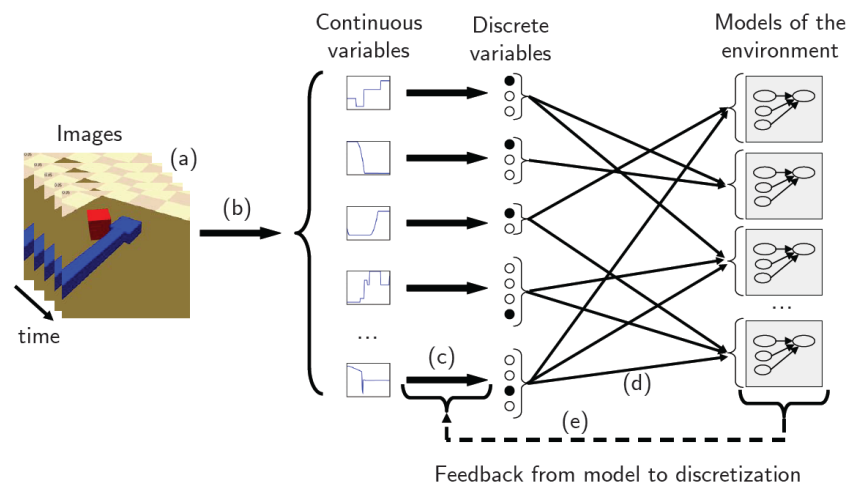


FIGURE 2.11 – Processus de discrétisation de l'expérience ([Mugan07b])

ment de l'évènement est radicalement différent par rapport à la distribution de ses valeurs en général, cela suscite la création d'une nouvelle borne.

[Mugan12] compare cette méthode à une méthode d'apprentissage supervisé. Ils montrent que leur méthode est plus efficace pour gérer des tâches complexes sur des intervalles de temps longs. Et également, que les performances de leur méthode ne sont pas détériorées lorsqu'ils augmentent la complexité de l'environnement en ajoutant des objets "distracteurs" dans l'environnement, alors que c'est le cas pour les méthodes d'apprentissage supervisé.

2.4.8 Les approches constructivistes dans le cadre de l'AmI

A notre connaissance il existe encore peu de travaux dans ce domaine.

Nous pouvons néanmoins citer [Najjar13] qui présentent le modèle *Nestor* basé sur l'architecture CLASM proposée par [Chaput03]. Dans ce modèle, l'apprentissage est séparé en deux phases successives, la *phase d'éducation* puis la *phase de travail*. Durant la première phase (deux semaines dans un environnement simulé de maison intelligente) le système apprend des schémas en classifiant grâce aux SOM les contextes qui suivent immédiatement une action (à t , $t + 1$, $t + 2$ et $t + 3$). Durant la deuxième phase, le système interagit avec l'utilisateur pour renforcer sa représentation et construire un langage commun. Pour cela l'utilisateur est invité à confirmer les propositions d'actions suscitées par l'activation des schémas et à nommer les schémas proposés par le système. Les schémas ainsi confirmés par l'utilisateur, peuvent acquérir un indice de confiance plus élevé et ont plus de chances d'être utilisés pour créer des motifs de plus haut niveau à partir de l'association de plusieurs schémas.

L'établissement d'une communication entre le système et l'utilisateur est un aspect indispensable des approches développementales. En effet, le système n'ayant pas d'objectif à priori, il faut d'une manière ou d'une autre communiquer avec le système pour susciter des comportements désirés. La "co-construction" d'un langage commun entre l'humain et l'agent peut être une piste envisageable. L'établissement d'une communication entre l'humain et le système est une problématique intéressante mais complexe que nous n'abordons pas pour l'instant. Dans le cas où ses structures internes sont explicites, on peut néanmoins interpréter directement la représentation de l'agent développemental. C'est un peu comme si on pouvait déchiffrer directement le contenu du cerveau d'un individu et le modifier sans avoir besoin de communiquer avec lui. Les structures de types schémas, même si elles sont de bas niveau permettent une interprétation plus ou moins aisée des motifs qu'elles représentent.

3

Proposition d'un modèle

▷

Les chapitres précédents ont situé la thèse dans son contexte théorique et applicatif et décrit le positionnement choisi pour l'élaboration d'un modèle d'apprentissage constructiviste pour un système d'intelligence ambiante. Dans ce chapitre, nous présentons plus précisément le contexte industriel de la thèse, et présentons en détail le modèle proposé. Tout d'abord nous décrivons le fonctionnement général du modèle, puis nous en proposons une implémentation qui permettra de réaliser des expériences présentées dans le chapitre suivant.

◁

Plan du chapitre

3.1	Contexte : un modèle d'apprentissage intégré à un système d'intelligence ambiante	81
3.1.1	Présentation du système HEMIS	81
3.1.2	Le rôle de l'apprentissage	86
3.1.3	Intégration du système d'apprentissage	88
3.2	Présentation générale du Modèle	89
3.2.1	Aperçu global	90
3.2.2	Perception à partir d'évènements	90
3.2.3	L'organisation du système multi-agent	94
3.2.4	L'évaluation des organisations	99
3.2.5	Utilisation d' <i>espaces de marquage</i> pour l'interaction inter-agents	104
3.2.6	Modèle étendu et propositions	107
3.3	Présentation d'une implémentation du modèle	110
3.3.1	Élément de représentation	110
3.3.2	Comportement générique des agents	111
3.3.3	Les agents D	114
3.3.4	Les agents A	117
3.3.5	Les agents S	120

3.1 Contexte : un modèle d'apprentissage intégré à un système d'intelligence ambiante

Le système d'apprentissage que nous proposons est conçu pour s'intégrer dans un système d'intelligence ambiante (HEMIS, proposé par la société *ubiant*) qui possède son propre fonctionnement. Le système d'apprentissage intervient donc comme un module qui est connecté aux données du système qu'il intègre et qui partage ses résultats avec celui-ci. Comme nous l'avons présenté dans la partie 1.3, il s'agit d'un système d'apprentissage constructiviste dont l'objectif est de construire des motifs sensorimoteurs à partir de données brutes non interprétées. Ces données peuvent être toutes les variables manipulées par le système d'intelligence ambiante, les variables provenant du matériel (ex. capteurs) comme les variables internes. Avant de présenter plus précisément le rôle du module d'apprentissage et la façon dont il s'intègre dans le système d'AmI dans la partie 3.1.2, nous allons dans la partie suivante présenter succinctement le système *HEMIS*.

3.1.1 Présentation du système HEMIS

3.1.1.1 Fonctions principales

Comme présenté dans la partie 1.1, le domaine de l'intelligence ambiante se distingue par la multiplicité et l'hétérogénéité des appareils connectés qui doivent être gérés pour et par l'utilisateur. Ainsi, le premier besoin auquel répond le système HEMIS concerne la gestion du matériel. On peut noter une similitude avec le *système d'exploitation* dans le monde informatique, comme illustré Figure 3.1. Un système d'exploitation est un ensemble de programmes qui mettent en relation les ressources matérielles de l'ordinateur et les applications de l'utilisateur. Il fournit aux applications une abstraction du matériel et facilite leur développement par les développeurs. Le système HEMIS a été élaboré pour assurer l'équivalent de ces tâches pour le bâtiment. Il est l'interface entre les applications et le matériel installé dans le bâtiment. Il facilite l'ajout et la suppression des capteurs et actionneurs, leur manipulation par les applications et les utilisateurs. Il gère les applications et leur assure un environnement d'exécution hautement sécurisé et robuste.

D'autre part, la deuxième fonction assurée par HEMIS est de gérer le bon déroulement des tâches réalisées par le système, de maintenir les objectifs décidés par l'utilisateur et de réagir aux aléas. Le système HEMIS garantit le lien entre la couche applicative et la couche matérielle. Pour cela, HEMIS s'appuie sur un modèle de base faisant abstraction des instances de matériel par des interfaces qui permettent aux utilisateurs la manipulation des applications et du matériel. Il est conçu pour faciliter la manipulation de *scénarios* interagissant avec des objets intelligents (regroupement d'actionneurs/moniteurs) installés dans un environnement très évolutif, que l'on peut dé-



FIGURE 3.1 – Principe : le système HEMIS en tant que système d'exploitation de l'habitat.

couper en *zones*. Les scénarios peuvent être activés et désactivés à tout moment par l'utilisateur, les objets manipulés par le système peuvent être déplacés d'une zone à une autre ou supprimés du système, de nouveaux objets peuvent apparaître et disparaître à tout moment. Afin de supporter toute cette dynamique et maintenir les objectifs et la stabilité du système, HEMIS s'adapte aux changements en temps réel.

3.1.1.2 Architecture

HEMIS adopte le paradigme des systèmes multi-agents (SMA) pour gérer les appareils connectés au système ambiant et répondre à différents objectifs. On peut plus précisément distinguer deux grandes parties : un SMA réactif qui gère la partie exécutive, et un SMA cognitif pour les tâches de plus haut niveau tel que la planification des objectifs, la gestion de scénarios et l'interfaçage avec l'utilisateur. Le noyau HEMIS est décomposé en quatre composants essentiels, comme présenté sur la Figure 3.2.

Une architecture en couches

Les API (*Application Programming Interface*) regroupent l'ensemble des fonctions permettant une interaction structurée et facile avec le système HEMIS. Les données collectées de l'environnement et celles générées par le système sont toutes stockées dans une base de données accessible par les différents composants du noyau grâce à une couche dédiée à l'accès aux données (DAO). Un système multi-agent cognitif gère les concepts de haut niveau manipulables par les utilisateurs (objectif, contrainte, facteur environnemental, etc.). Un système multi-agent réactif plus proche du monde physique est le lieu d'exécution concret des applications. Finalement, la connexion aux différents objets est assurée par différents contrôleurs matériel qui implémentent différents protocoles de communication et qui sont accessibles de façon transparente à partir de la couche d'agents réactive.

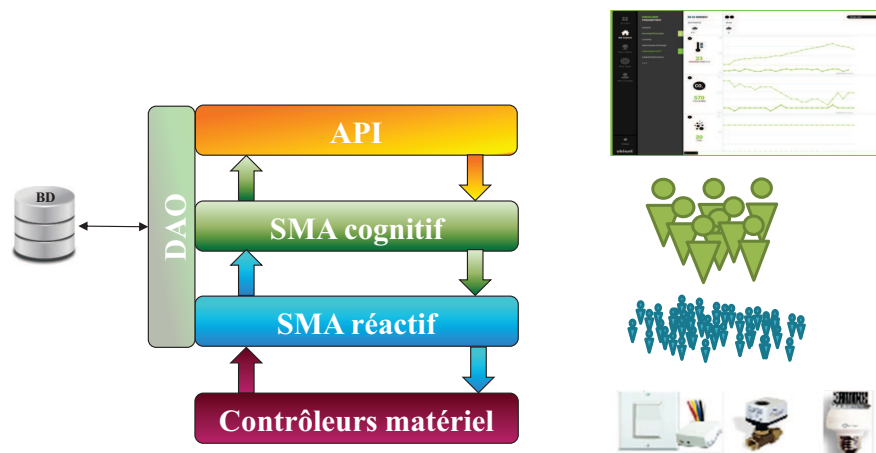


FIGURE 3.2 – Architecture globale du système HEMIS.

Un système interopérable

HEMIS se veut extensible sans limitation de type de matériel. Le problème récurrent dans l'industrie du bâtiment est l'absence de standard unique de communication permettant de contrôler les actionneurs et de collecter les données des capteurs. Le système HEMIS définit une interface commune abstraite de communication entre la couche hardware et le noyau. Différents systèmes de communication associés à des gammes de produit hardware sont implémentés et respectent cette couche abstraite commune. Des passerelles implémentent l'ensemble des primitives de base, définies dans la couche commune, comme la récupération des dernières valeurs pour les capteurs, l'ajout d'un nouveau matériel, ou l'envoi d'ordre à un composant hardware. Ainsi le noyau peut supporter simultanément l'existence de plusieurs passerelles et communiquer en toute transparence avec chacune d'elles.

3.1.1.3 Notions principales du modèle HEMIS

Le modèle du système HEMIS est basé sur quatre concepts majeurs : les *facteurs environnementaux*, les *scénarios*, les *contraintes* et les *objectifs*.

Les facteurs environnementaux

HEMIS introduit le concept générique de *facteur environnemental* afin de gérer les nombreux capteurs et actionneurs intervenant dans la résolution des objectifs. Un facteur environnemental est défini pour une modalité ciblée (ex. température) dans une zone précise du bâtiment (ex. chambre). Tout capteur relève les valeurs détectées pour un facteur environnemental, et tout actionneur agit sur un ou plusieurs facteurs environnementaux. Les objets physiques que sont les actionneurs et les capteurs impactent

directement le/les facteurs environnementaux qui lui/leur sont associés. Si par nécessité de redondance des capteurs, il y a plusieurs capteurs de température dans une même zone, le facteur environnemental synthétisera ces différentes informations.

Par exemple, dans le cas de la modélisation de la dynamique d'une pièce, la température peut être modifiée par les équipements de chauffage ou de climatisation classique mais d'autres sources de variation peuvent participer à cette modification. Ainsi, l'apport calorifique externe des rayons de soleil recueilli en ouvrant les volets d'une fenêtre influence également le facteur environnemental température de la pièce. L'actionneur permettant de contrôler les volets devient ainsi un actionneur pour le facteur environnemental "température dans la chambre" mais aussi pour le facteur environnemental de la luminosité dans cette zone. De même, un appareil spécifique comme le sèche-linge ou le four électrique va avoir une influence sur la puissance électrique mais aussi la température de la pièce dans laquelle il est situé. Le même mécanisme est reproduit pour tous les facteurs environnementaux du système. Ce n'est pas l'objet physique qui est important mais plutôt les rôles qu'il incarne à travers différents facteurs environnementaux.

Notons ici qu'au sein du système, un matériel est associé à un facteur environnemental particulier, sur la base de connaissances expertes (des informations concernant ce matériel sont stockées dans une base de données). Par exemple, un capteur de température identifié comme tel, sera associé au facteur environnemental température de la zone dans laquelle il est placé. Cependant, dans le cas général un appareil communicant pourrait être connu (on possède des informations sur la nature des valeurs qu'il produit ou sur la nature des grandeurs qu'il affecte), ou bien inconnu (on n'a pas d'information ou des informations incomplètes sur ce matériel). Le système d'apprentissage pourra cependant exploiter ce capteur et éventuellement proposer de compléter les métadonnées du système HEMIS.

Les scénarios

Atteindre une température ambiante dans une pièce donnée quand il y a une présence et à partir d'une certaine heure de la journée, ou réduire la consommation générale de la maison quand on reçoit une demande d'ajustement sont des exemples de scénarios facilement réalisés dans le modèle HEMIS.

Ainsi nous proposons des scénarios composés essentiellement de deux concepts les *contraintes* et les *objectifs*. Un scénario possède un ou plusieurs objectifs et un état qui change selon la validité de ses contraintes. Le scénario peut avoir un ou plusieurs groupes de contraintes et il n'est exécutable que si au moins un des groupes de contraintes est valide. Quand il est exécutable, tous les objectifs composant le scénario sont alors exécutables. Cette définition d'un scénario est suffisamment générique pour pouvoir exprimer grand nombre de besoins, tout en assurant un environnement d'exécution simple.

Les objectifs

Nous distinguons dans HEMIS deux types d'objectifs. Les objectifs concernant un facteur environnemental permettent de modéliser une intention d'atteindre une valeur souhaitée pour le facteur environnemental ciblé (ex. 20°C dans la cuisine). Les objectifs fonctionnels permettent quant à eux d'étendre le système avec des fonctions non forcément modélisables en termes d'objectifs sur un facteur environnemental (par exemple un objectif d'affichage). HEMIS vérifie que les contraintes de ce type d'objectif sont valides et les exécute à travers une méthode générique existante dans tous les objectifs fonctionnels.

Les contraintes

Dans le modèle HEMIS, nous distinguons deux types de contraintes, celles qui portent sur les facteurs environnementaux et les contraintes temporelles. Les contraintes sur un facteur environnemental se modélisent par une valeur minimale et une valeur maximale. La contrainte est satisfaite si la valeur réelle de l'environnement mesurée par les capteurs se situe entre les bornes définies dans la contrainte. D'autre part la contrainte sur des facteurs particuliers tels que la présence sont une spécialisation des contraintes sur un facteur environnemental. Ce type de contrainte permet de définir des conditions sur une présence quelconque ou sur une ou plusieurs présences identifiées. Afin de valider l'exécutabilité des scénarios, les contraintes sont vérifiées en continu par le système.

Plusieurs scénarios sont présents dans le système simultanément. Les scénarios peuvent concerner par exemple le confort thermique, lumineux, la gestion énergétique ou encore la sécurité. Un concept de priorité combiné à un mécanisme d'apprentissage dédié permet de résoudre les conflits entre les différents objectifs concurrents et exécutables en même temps. Les objectifs peuvent être de différents types selon leurs durées et avoir différents modes d'exécution.

3.1.1.4 Le SMA cognitif

La couche multi-agent cognitive permet une manipulation aisée des concepts de haut niveau pour exprimer les besoins des utilisateurs. Ainsi, chaque scénario créé par l'utilisateur est représenté par un agent cognitif dans HEMIS. Cet agent supervise l'exécution de ses objectifs et reporte de façon continue leur état d'avancement, échec ou succès. Il est aussi responsable d'établir le lien avec le reste des agents cognitifs et de transférer ses objectifs vers la couche réactive qui les exécutera concrètement.

D'autre part chaque facteur environnemental est lui-même un agent cognitif qui maintient la liste d'actionneurs et de moniteurs qui le concernent. C'est cet agent qui

gère en continu la correspondance entre les objectifs des scénarios et les actionneurs capables de les exécuter.

Un *objet communicant* (regroupement de capteurs et actionneurs) ajouté dans le système est aussi représenté par un agent cognitif. Ce dernier crée dans la couche réactive les agents réactifs correspondants à ses capteurs et actionneurs et établit les relations avec les différents facteurs environnementaux intéressés par un ou plusieurs de ses actionneurs ou capteurs.

L'interaction entre les différents agents cognitifs est continue, afin de permettre l'auto-organisation du système. En effet, un scénario peut être irréalisable à cause de l'absence d'actionneurs capables d'exécuter ses objectifs ou l'absence de capteurs capables de valider ses contraintes. Dès l'ajout d'un nouvel objet intelligent dans le système, chaque agent facteur environnemental concerné va mettre à jour sa liste de capteurs/actionneurs et informera les scénarios qui lui sont liés de l'existence de nouveaux capteurs ou actionneurs. Le scénario pourra alors devenir réalisable si un hardware manquant vient d'être ajouté grâce au nouvel objet intelligent. Le même principe d'auto-organisation est appliqué quand un objet intelligent est supprimé.

3.1.1.5 Le SMA réactif

Les objets physiques manipulés par le système (actionneurs et capteurs) sont représentés par des agents réactifs. Les données collectées et les ordres envoyés au matériel passent par cette couche réactive. L'interaction entre les agents réactifs et les objets physiques se fait grâce à une passerelle implémentant différents protocoles de communication. Dans cette couche, les facteurs environnementaux ainsi que les contraintes sont modélisés par le concept d'*espace de marquage*. Les objectifs sont portés par des agents réactifs qui marquent ces espaces. Les agents actionneurs sont reliés à ces espaces et recherchent continuellement à atteindre ces objectifs. Pour un objectif à atteindre, concernant un facteur environnemental donné, chaque actionneur impactant ce facteur participe à l'accomplissement de l'objectif. L'effort fourni par chaque agent actionneur est proportionnel à sa capacité et à sa disponibilité (par rapport à d'autres objectifs auxquels il peut participer). Les agents actionneurs vont réitérer leur participation à l'exécution de l'objectif, tant que ce dernier n'est pas atteint.

3.1.2 Le rôle de l'apprentissage

Le système HEMIS est donc un système d'AmI total, visant à gérer l'ensemble des problématiques du domaine. Il repose sur une architecture décentralisée à base de système multi-agent hybride, similaire à l'approche présentée dans la partie 2.4. Grâce à cela le système est capable de s'auto-adapter pour assurer la gestion de multiples scénarios possiblement concurrents, et l'ajout ou la suppression de matériel en temps réel tout en maintenant ses objectifs. Dans ce contexte, les problématiques d'apprentissage

sont multiples et peuvent intervenir à différents niveaux du système. Dans le cadre du système HEMIS, nous avons identifié plusieurs problématiques d'apprentissage principales, sur trois niveaux : *l'apprentissage des capacités sensorimotrices, la reconnaissance d'évènements complexes, et l'apprentissage des objectifs et des préférences des utilisateurs*. Le travail présenté dans ce mémoire contribue à répondre aux deux premiers niveaux présentés.

3.1.2.1 Bas niveau : Apprentissage des capacités sensorimotrices

Chaque actionneur a un effet particulier sur son environnement, et chaque capteur peut interpréter son environnement d'une manière particulière. Tout d'abord, un actionneur peut être inconnu du système. Il s'agit d'un cas extrême dans lequel HEMIS n'a pas d'information sur le matériel connecté. Par exemple c'est le cas du contrôle d'un appareil indéterminé par une prise communicante. Le système connaît la prise mais pas forcément l'appareil branché dessus.

De plus, l'actionneur peut avoir différents effets suivant l'endroit ou le type d'environnement dans lequel il est placé. Par exemple : un contrôleur d'ouverture de porte, suivant qu'il est sur une porte d'entrée ou bien une porte de communication, n'aura pas les mêmes effets concernant le facteur température ou bien la capacité d'aération et n'affectera pas les objectifs de sécurité de la même manière. Enfin, l'effet d'un actionneur est quantitativement différent en fonction de son contexte. Par exemple l'activation d'une lampe peut grandement contribuer à augmenter la luminosité si la pièce est faiblement éclairée, mais n'a pas quasiment aucun effet si la pièce baigne dans la lumière du jour. D'autre part les capteurs et actionneurs possèdent des temps de latence et des fréquences de mise à jour différentes. Tout ceci illustre le fait que le système doit s'approprier ses moyens d'agir et de percevoir de manière précise, dans leurs dimensions spatiales et temporelles. Cet apprentissage ne peut se faire qu'à partir de l'expérience. Du point de vue de l'apprentissage constructiviste, il s'agit d'apprendre des motifs sensorimoteurs simples (cf. schèmes primaires).

3.1.2.2 Niveau intermédiaire : Reconnaissance d'activités, de situations, d'évènements complexes

Pour prendre une décision intelligente, un système d'intelligence ambiante doit pouvoir reconnaître les activités en cours dans la maison (ex. : dormir, manger, regarder la télé, ou seulement détecter la présence d'une personne dans une zone). Il ne s'agit pas ici de caractériser à priori ces activités sur la base d'une représentation experte, mais plutôt de permettre au système de reconnaître ces activités s'il a les moyens matériels de les identifier dans son expérience sans supposer la façon d'y parvenir. Le système doit en effet pouvoir différencier cette situation du reste de l'expérience de façon fiable en recoupant les différentes informations dont il dispose. Nous supposons

que le système d'apprentissage doit créer le motif correspondant à une activité en fonction de son propre besoin. Par exemple, la reconnaissance de l'activité "petit-déjeuner" pourrait permettre de prédire efficacement l'évènement "quitter le logement", davantage que chacun des sous-événements qui le constituent. Cette façon de considérer les motifs de niveau abstrait est similaire à la façon dont sont créés les items synthétiques dans l'apprentissage de schémas, c'est-à-dire en fonction d'un besoin de modélisation pour l'amélioration de la capacité prédictive du système, autrement dit pour une raison intrinsèque au système. Cependant nous pouvons aussi imaginer un guidage de l'apprentissage d'activités pour des raisons pratiques, par exemple en indiquant au système les moments où il se passe quelque chose d'intéressant et éventuellement les facteurs concernés. Du point de vue de l'apprentissage constructiviste, cette étape correspond à la construction de motifs complexes à partir des motifs sensorimoteurs primaires (cf. schèmes secondaires).

3.1.2.3 Haut niveau : Apprentissage des objectifs et des préférences utilisateurs

Les utilisateurs peuvent régler leurs préférences sous forme de scénarios grâce aux interfaces et applications, mais les scénarios pourraient également être inférés automatiquement sur la base de l'observation et de la communication avec l'utilisateur. D'autre part, les scénarios peuvent présenter des exceptions difficilement programmables dans l'interface, ils peuvent être contextuels ou simplement être périmés. Cette gestion des scénarios peut être simplifiée grâce à un apprentissage impliquant un retour important de l'utilisateur. Ce troisième point suppose une prise en compte de concepts de haut niveau de la couche cognitive (scénarios, objectifs, contraintes) ainsi que la mise en place d'une capacité de communication avec l'utilisateur. Une modélisation de ce type d'apprentissage dans le cadre constructiviste utilisé pour les deux autres catégories est difficile et constitue une étape future de développement du modèle.

Dans ce travail nous nous intéressons aux deux premières catégories.

3.1.3 Intégration du système d'apprentissage

Nous considérons les *facteurs environnementaux* ainsi de que les *états des actionneurs* (couche réactive) comme des variables continues qui varient au cours du temps. Ces variables constituent l'expérience pour le système d'apprentissage, qui ne possède aucune information supplémentaire. Il n'est cependant pas exclu de "brancher" le module d'apprentissage sur d'autres éléments au sein de HEMIS, par exemple dans la couche cognitive, mais dans le cadre de cette étude nous laissons de côté ces possibilités.

Comme nous l'avons présenté, l'apprentissage constructiviste repose 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 donc modélisée par un ensemble de variables continues (Figure 3.3) qui représente l'ensemble des cap-

teurs et des actionneurs : $V = \{v_1; v_2 \dots v_n\}$. Ce flot continu de données brutes peut être "interprété" (ou discrétisé) par le système, par exemple comme une séquence d'évènements. Par la suite nous appelons *expérience* $E = \{e_1; e_2 \dots e_n\}$ l'ensemble des évènements construits par l'agent. Nous faisons l'hypothèse qu'il existe des régularités au sein de cette expérience. Une régularité est un motif récurrent que le système devra être capable d'isoler et d'identifier en explorant son espace de recherche. Du point de vue du système proposé, il n'y a donc pas de différence entre les variables représentant les capteurs et les actionneurs. Par conséquent les motifs appris peuvent concerner indifféremment des combinaisons de capteurs et actionneurs.

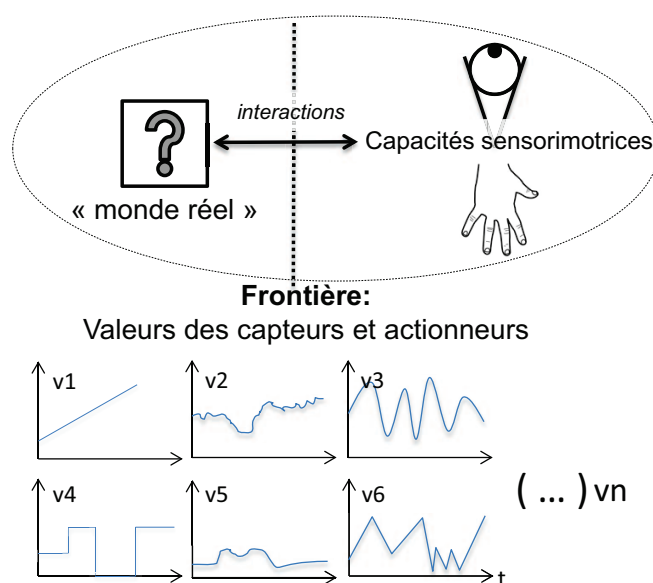


FIGURE 3.3 – Du point de vue théorique : frontière agent-environnement comme un ensemble de variables continues

La manière dont le système HEMIS récupère et utilise les résultats du module d'apprentissage, et les mécanismes pour éventuellement orienter l'apprentissage ne sont pas abordés dans le cadre de cette thèse.

3.2 Présentation générale du Modèle

Dans cette partie, nous présentons tout d'abord le principe de fonctionnement du modèle proposé relatif à l'amorçage de l'apprentissage sensorimoteur. Pour cela nous décrivons une structure de représentation basée sur la notion d'évènement, puis introduisons le processus d'apprentissage décentralisé en tant que recherche d'interprétations prédictives de l'expérience. Ensuite, nous présentons plus précisément l'archi-

itecture multi-agent proposée pour réaliser ce modèle. Pour finir, nous présentons une architecture plus générale d'apprentissage constructiviste dans laquelle se situe le modèle proposé.

3.2.1 Aperçu global

L'architecture générale du modèle est représentée par la Figure 3.4. Le processus de discrétisation interprète l'expérience pour créer des structures représentationnelles qui permettent de compresser et mémoriser les données brutes. Initialement ces structures sont "primitives" dans le sens où ce sont des propositions de représentation non encore confirmées par leur intervention utile pour la prédiction (il s'agit des événements et associations évalués intrinsèquement intéressants). Donc le *feedback* vers ce processus provient d'une part de l'évaluation de ces structures en elles-mêmes par la méthode de la mesure d'intérêt (feedback transitionnel), et d'autre part il provient des *prédictions* réalisées grâce à elles (feedback prédictif). Lorsque des motifs sont confirmés par leur succès en matière de prédiction, on peut envisager d'approfondir l'apprentissage dans leur voisinage au sein de l'espace de recherche. On peut ainsi utiliser des motifs pour construire une représentation plus abstraite (*généralisation*), ou bien plus spécifique pour caractériser plus précisément l'interaction (*spécialisation*). On appelle *motifs primaires* les associations d'événements primaires, directement issus de la phase d'amorçage à partir des variables de l'environnement.

Dans la suite nous détaillerons les composantes de cette architecture, en commençant par le problème principal traité dans ce travail, à savoir la phase d'amorçage de l'apprentissage sensorimoteur.

3.2.2 Perception à partir d'évènements

Pour concevoir un système d'apprentissage constructiviste, une question fondamentale est de décider quelle est la brique de base à partir de laquelle la représentation de l'agent va être construite. Le rôle de la *structure élémentaire de représentation* est d'exprimer la régularité ou le *motif* 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 Uexküll09]. Notons ici qu'une autre approche possible afin de s'affranchir du problème d'amorçage est de considérer que le schéma, ou toute autre structure élémentaire du même type, constitue une structure élémentaire indivisible intrinsèque au système. Par exemple [Georgeon13] 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 sensorimotrices de l'agent. La même remarque s'applique à l'approche de [Brooks91] avec l'architecture de subsomption pour la robotique où les comportements de bas niveau sont préalablement codés.

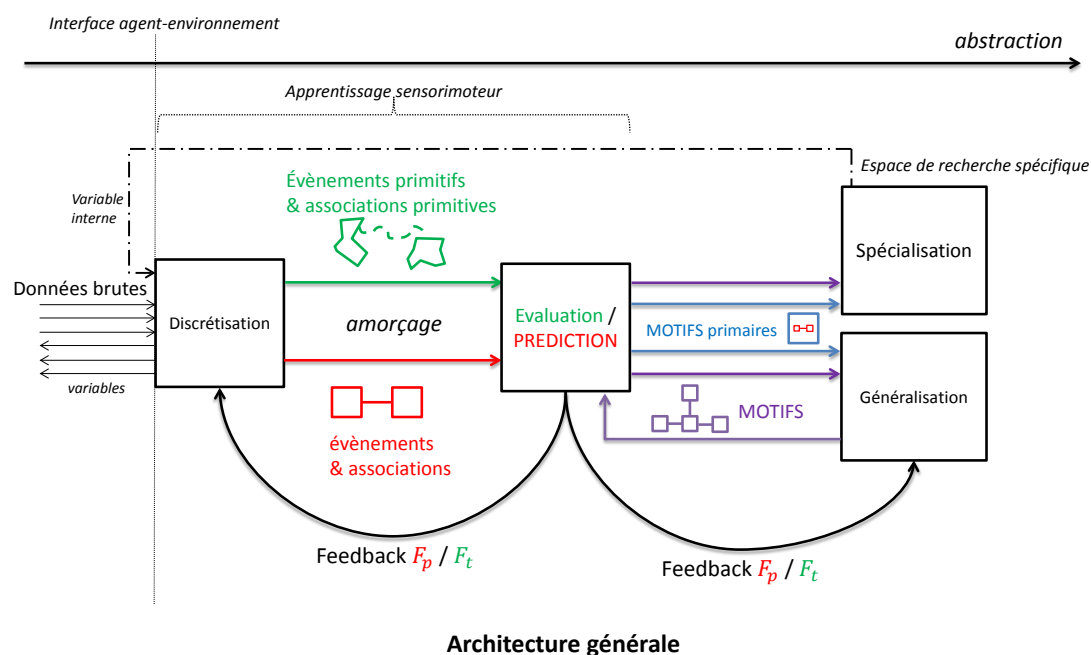


FIGURE 3.4 – Modèle général d'apprentissage constructiviste.

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.). Mais lorsque l'expérience est modélisée par un ensemble de variables continues, cela suscite de nombreuses questions. Par exemple, comment définir dans ce cas les contextes, actions, et résultats ? Comment définir les durées de ces éléments et quelles sont leurs relations temporelles ? 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 dès lors que le concepteur laisse tout ou partie de ce travail de discrétisation et d'abstraction à la charge de l'agent. Bien sûr, ce problème est évacué si le concepteur du système réalise lui-même l'abstraction nécessaire pour discrétiser l'environnement.

3.2.2.1 Définition d'un événement

Une représentation sous forme de schémas est limitée dans le sens où elle oblige le système à apprendre des régularités du type "perception-action-perception" ce qui constitue une contrainte assez forte. En effet cela suppose que l'on a défini à priori ce

qu'est une action d'une part, et d'autre part une perception –telle que défini dans un modèle d'apprentissage de schémas "classique"– est un contexte qui contient les valeurs de toutes les variables à un instant t . Dans un environnement complexe continu, cela implique une discrétisation des valeurs et une discrétisation du temps à priori. De plus, des régularités d'un niveau d'abstraction équivalent mais non restreintes au format "perception-action-perception" devraient pouvoir également être apprises : par exemple le type perception-perception (ex. « à chaque fois que la porte s'ouvre il y a du mouvement avant ou après ») ou action-action (ex. « j'allume la lampe, puis je baisse le volet roulant »). Comme nous l'avons dit, dans un système d'intelligence ambiante, une certaine variable (un actionneur) peut être à la fois considérée comme une action et comme une perception, suivant que c'est l'utilisateur qui modifie la variable ou bien le système. Que ce soit un actionneur ou un capteur qui produit un évènement, on peut généraliser cela en disant que c'est une variable qui varie, et des régularités peuvent être apprises indifféremment entre ces variables de la même façon.

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. 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.

Nous nommons "*instance d'évènement*" la représentation d'un évènement particulier qui s'est produit et "*concept d'évènement*" (ou classe d'évènement) la représentation d'un évènement récurrent identifié en général.

Les différentes implémentations possibles de ces éléments de représentation et des moyens de les produire constituent donc un *espace de recherche global* de la discrétisation de l'expérience continue. Cet espace de recherche est donc l'ensemble des façons de découper l'expérience en évènements, et l'ensemble des façons d'associer ces 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.

3.2.2.2 Remarques concernant les actions

Nous avons dit que le système considèrerait indifféremment les actions et les perceptions dans l'élaboration des motifs. Il faut cependant que des actions soient effectivement exécutées pour pouvoir apparaître dans des patterns sensorimoteurs. 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 tel que proposé dans [Mugan07a] 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. Ainsi, nous considérons comme contrainte de conception que chaque action pouvant 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*¹, de sorte que si cette action est impliquée dans un motif, l'agent, l'ayant apprise, soit capable de la réaliser par lui-même plus tard. [Najjar13] 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. Dans notre cas, le couplage avec le système HEMIS entre également en jeu puisque des mécanismes propres au système HEMIS vont être à l'origine d'actions. Par conséquent le problème de la motivation intrinsèque présenté dans la partie 2.4.5.2 est sensiblement atténué, et nous n'aborderons pas ce problème davantage dans ce travail. On peut se demander néanmoins s'il suffit que le système fasse l'expérience des actions ou si l'intention de les produire est nécessaire pour construire un comportement ayant un sens pour l'agent. Dans cette optique, les actions – bien qu'elles puissent avoir un effet perceptible dans l'environnement – ne se différencieraient pas des perceptions et le système serait dans le cas d'un "*chaton passager*" de l'expérience de [Held63] présentée en partie 1.2.3. Cependant, dans cette expérience les individus passifs ne faisaient pas l'expérience de l'action du tout. Nous partons du principe que l'action suffit, et laissons de côté cette question de l'intention qui est un vaste sujet en soi.

3.2.2.3 Évaluation des structures internes

L'objectif du modèle est que la construction de motifs soit basée sur la prédiction, de manière à ce que la représentation soit conditionnée par son utilité pour le système ce qui est conforme à une interprétation constructiviste de l'élaboration de la représentation en IA. Un motif est donc jugé intéressant s'il permet de relier des événements de manière prédictive. Ainsi les processus de discrétisation ayant permis la construction de ce motif bénéficient d'un feedback positif. Cependant, l'espace de recherche énorme que constitue cette recherche de régularités n'autorise pas l'utilisation exclusive de ce

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

moyen d'évaluation en un temps acceptable. D'ailleurs comme nous l'avons dit, la réponse au problème d'amorçage est réalisée, dans la nature, de manière bien plus complexe puisqu'entrent certainement en jeu des mécanismes d'adaptation à l'échelle phylogénétique. Une *estimation de l'intérêt* de la représentation est proposée afin de guider le processus de discrétisation. Cette évaluation porte sur l'activité des agents générant les évènements et les associations avant que l'on puisse réaliser des prédictions, en examinant le potentiel intérêt de la discrétisation obtenue. Cette mesure d'intérêt qui sera présentée dans la partie 3.2.4.1 permet de limiter la création de motifs et de prédictions à tester. D'autre part, l'utilisateur devrait aussi intervenir dans l'évaluation des motifs car certains motifs peuvent ne pas être "vérifiables" aisément par leur éventuel rôle dans une prédiction. A condition de développer une interface de communication intuitive, l'utilisateur pourrait de plus guider l'apprentissage vers certains motifs intéressants, à l'instar du modèle de [Najjar13]. Ainsi nous distinguons trois sortes de feedback possibles au sein du système :

- Feedback prédictif (F_p) : Ce feedback repose sur l'évaluation de la qualité des prédictions réalisées.
- Feedback transitionnel (F_t) : Ce feedback est relatif à la phase d'amorçage, il est basé sur une évaluation de la qualité intrinsèque d'une représentation (estimation).
- Feedback externe (F_e) : Le feedback provenant de l'extérieur. Il peut venir de l'utilisateur, ou d'un autre système (ex. HEMIS) pour évaluer l'utilité de la représentation relativement à des objectifs externes.

3.2.3 L'organisation du système multi-agent

Notre proposition est basée sur l'utilisation d'un système multi-agent (SMA) pour réaliser l'amorçage et l'apprentissage de motifs sensorimoteurs à partir de l'expérience continue. La Figure 3.5-(a) illustre la notion de motif récurrent entre des évènements interprétés (ex. des variations) à partir de variables dans le domaine de l'AmI. Le système multi-agent proposé se compose de trois populations d'agents distinctes qui exécutent des activités complémentaires et interagissent afin de construire la représentation. Ils sont guidés par des boucles de rétroaction provenant de l'évaluation de la construction réalisée. En effet, pour obtenir les structures élémentaires de représentation présentées précédemment, nous définissons trois *opérations élémentaires*, tel qu'illustré sur la Figure 3.5-(b).

- Il faut tout d'abord "*découper*" l'expérience continue des variables en des moments finis que l'on peut considérer comme des évènements.
- Ensuite on doit "*associer*" ces évènements pour créer des évènements plus complexes (*évènements associations*).
- Enfin il est nécessaire de pouvoir "*comparer*" deux éléments de représentation, par exemple pour reconnaître différentes instances d'un même évènement.

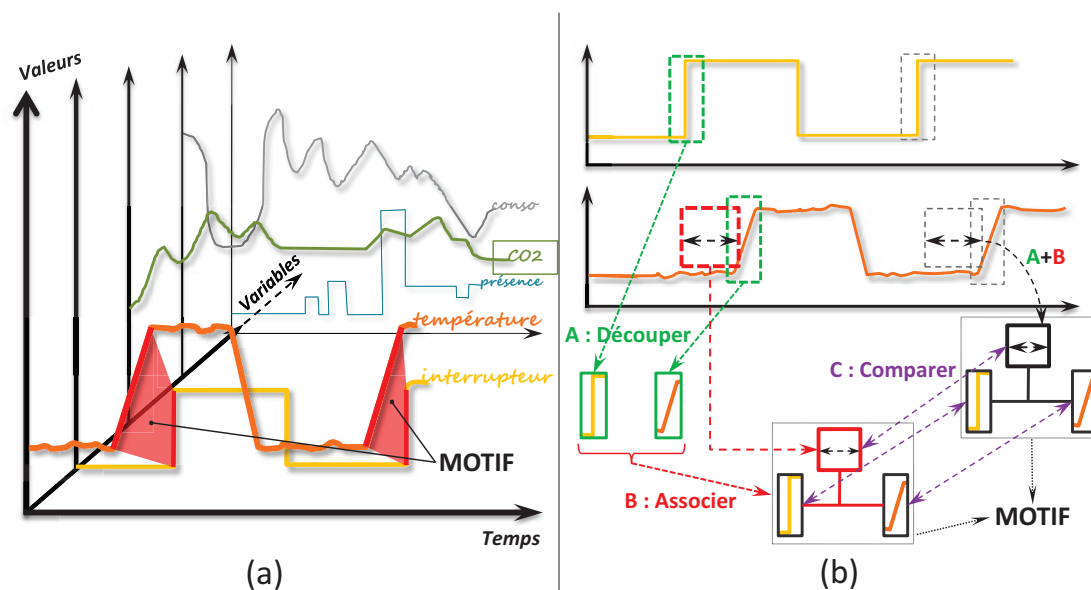


FIGURE 3.5 – (a) L'expérience continue modélisée comme un ensemble de variables. Un motif est une régularité récurrente dans l'évolution des variables.

(b) Les trois types d'opérations élémentaires composantes de la construction d'un motif sensorimoteur.

A : Découper : proposer une discrétisation de l'expérience (création d'un évènement à partir de l'évolution continue).

B : Associer : proposer des corrélations entre évènements (création d'une association).

C : Comparer : évaluer la similarité entre des éléments de représentation comparables.

L'interprétation d'une variable continue (ou le choix d'un *point de vue* (voir partie 2.4.7)) passe donc par un découpage spatio-temporel de son évolution. Par exemple, le découpage temporel (sur l'axe du temps) peut permettre de créer des évènements d'une durée finie ; et le découpage spatial (sur l'axe des valeurs) d'identifier des états stables ou des variations. D'autre part l'interprétation de l'expérience passe aussi par le choix d'un *point de vue* englobant plusieurs variables. Par exemple en créant des associations entre des évènements de différentes variables. Un *agent* au sein du SMA implémente 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 mul-

tiples qui constituent l'espace de recherche global. L'évaluation des solutions oriente la recherche et renforce les zones intéressantes. La Figure 3.6 offre une vue synthétique sous forme de diagramme d'activité de ce processus de construction décentralisé. Un *type d'agent* désigne l'implémentation particulière d'un rôle, c'est-à-dire des agents réalisant une même fonction.

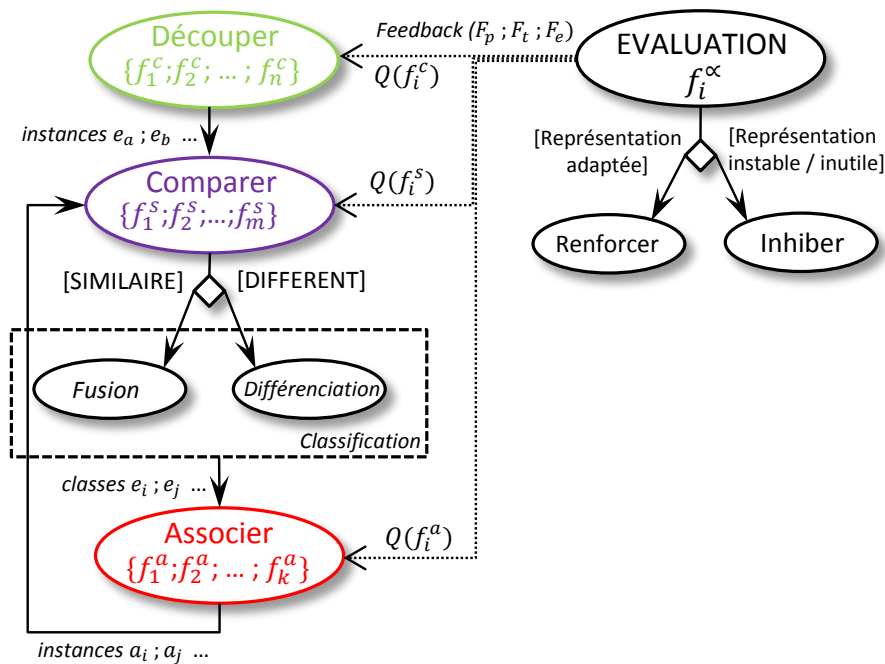


FIGURE 3.6 – Activité globale du système.

3.2.3.1 Les rôles

Ces 3 opérations sont modélisées au sein du SMA en spécifiant 3 rôles d'agents qui peuvent être implémentés par des *fonctions*. Une fonction est une implémentation particulière pour réaliser le comportement défini par le rôle. Il en existe un nombre infini pour chaque rôle. Nous décrivons ci-dessous les rôles joués par les agents qui implémentent ces différents types de fonctions, et leurs interactions comme présenté sur La Figure 3.6.

Les agents "producteurs" (D & A)

Ces agents interprètent l'expérience, discrétisent et compressent les données brutes pour produire des éléments de représentation (événements et association). Pour cela ils sont connectés à une variable de l'environnement ou bien à une variable interne au système.

- 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 définie 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 D*. Chaque agent D est connecté à une variable de l'environnement dont il doit explorer les interprétations. Par exemple, une implémentation possible est un agent qui utilise des fenêtres glissantes avec une durée paramétrable pour créer des évènements. L'exploration possible liée à cette fonction pourrait être la recherche d'une taille de fenêtre pertinente, adaptée à la variable et aux éventuels motifs concernés. Une autre fonction possible pourrait reposer sur l'utilisation de SOM ou une autre technique de classification.

- Deuxièmement l'opération "associer" permet de « relier » les évènements existants. On peut également considérer qu'il s'agit d'un "découpage" de l'expérience (ou choix d'un point de vue) dans le sens où l'on sélectionne un nombre restreint de variables et de structures ciblées à interpréter. Une *fonction association* ($F^a = \{f_1^a; f_2^a; \dots; f_n^a\}$) peut être définie par :

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

$$E \times E \rightarrow E$$

Tout comme un agent D concerne une variable, un agent A s'occupe d'un évènement en particulier (*évènement de référence*) qu'il essaie de relier à la représentation. Une *variable interne* permet de modéliser les occurrences de l'élément de référence de l'agent A. Par exemple une implémentation possible de cette fonction est un *agent A* qui, à partir de l'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 entre les éléments. Une autre implémentation possible pourrait être une fonction "pendant" qui génère une représentation basée sur le fait que e_1 se déroule pendant une occurrence de e_2 ou non.

Les agents similarité (S)

- Le troisième type d'opération, "compare" est une 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 S va dépendre des structures de représentation utilisées pour les éléments de représentation. Par exemple on peut utiliser des histogrammes, et une implémentation possible 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.

3.2.3.2 Couple d'agents : Producteur + Similarité

Le traitement opéré par les agents S est de trier les instances d'évènements produits par les deux autres types d'agents et de sélectionner 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 3.7 qui montre les formes d'organisations possibles sous la forme d'un diagramme Agent-Groupe-Rôle ([Ferber04]). Un couple d'agent est donc composé d'un agent producteur (D ou A) et d'un agent similarité (S).

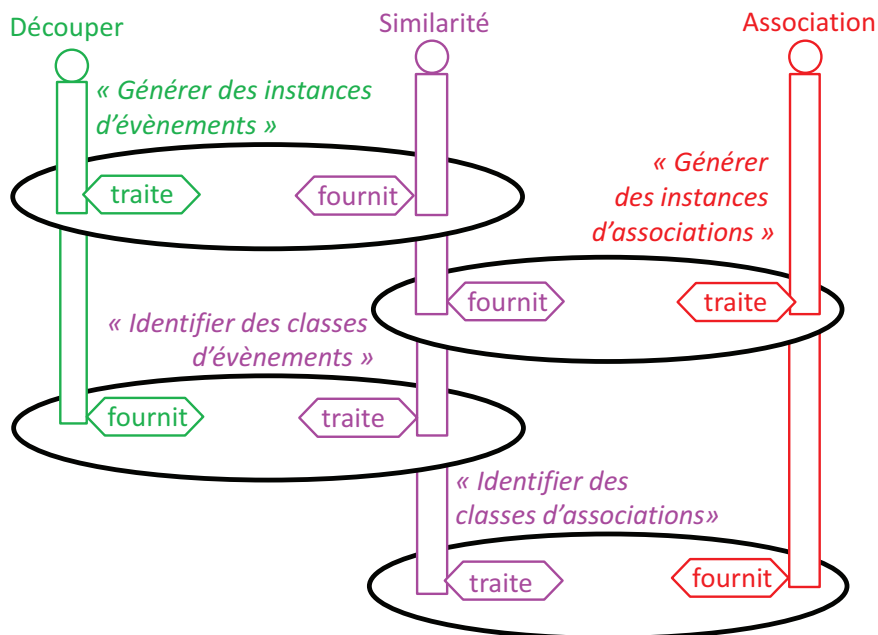


FIGURE 3.7 – Organisations possibles entre les types d'agents.

Au sein d'un *couple*, un agent producteur reçoit des données sélectionnées à un niveau inférieur, et les traite pour créer un nouvel évènement. Le niveau le plus bas étant

la variable, la variable joue le rôle de fournisseur (en remplacement de S) pour initier le processus et permettre la création des premiers évènements. Les nouveaux évènements sont classés et sélectionnés par l'agent S, et diffusés à leur tour dans le système sous forme de variables internes. Comme illustré sur la Figure 3.8, les agents peuvent appartenir à différentes organisations, dans lesquelles ils jouent différents rôles, à différents niveaux.

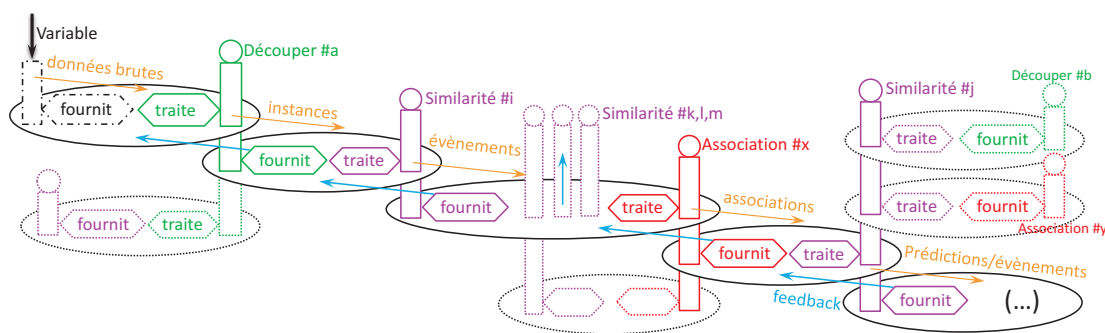


FIGURE 3.8 – Exemples d'interactions entre instances d'agents.

Comme on peut le voir sur la Figure 3.6, 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 précédentes, et est fusionné avec un élément similaire ou constitue 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 sensorimoteurs.

3.2.4 L'évaluation des organisations

Un agent est donc défini par un *rôle* (l'une des 3 opérations qu'il réalise), un *type* (la fonction qu'il implémente pour exécuter ce rôle) et un *paramètre* (module le comportement de la fonction). L'évaluation permet de guider le système vers les combinaisons utiles de types d'agents et de paramètres en fonction des variables. Pour guider l'exploration des agents (types + paramètres) en fonction de cette évaluation, un marquage est effectué dans un *espace de marquage* partagé à chaque nouvelle évaluation d'une structure de représentation. La position du marquage dans cet espace correspond aux paramètres des différentes fonctions employées pour construire la structure évaluée. Le choix de modélisation de ces informations d'évaluation sous forme d'espaces de marquage présente certains avantages. Tout d'abord les données manipulées par le système

sont aisément interprétables visuellement, grâce à la spatialisation de l'information. Ensuite cette représentation spatiale permet d'envisager d'utiliser les propriétés qui en découlent (ex. la notion de distance à l'instar des SOM) pour exploiter les données sous des angles différents, tout en gardant un aperçu intuitif des phénomènes observés.

3.2.4.1 Feedback transitionnel (F_t)

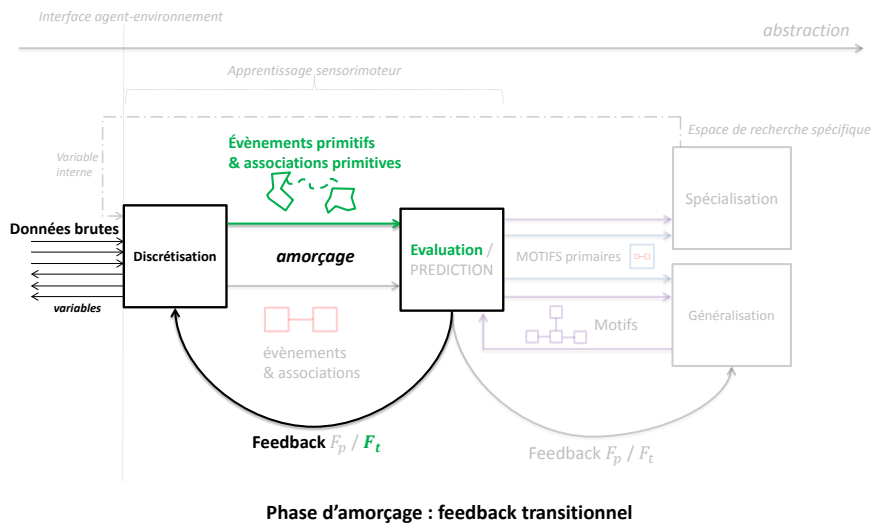


FIGURE 3.9 – Le feedback transitionnel permet de guider à priori la discrétisation de l'expérience vers des structures stables et potentiellement intéressantes.

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 proposons un feedback transitionnel afin d'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 une mesure d'intérêt basée sur leur *spécificité* et leur *poids*. Cette modélisation repose sur une formulation intuitive du type de discrétisation qui semble efficace pour détecter des événements qui se "démarchent" du reste de l'expérience par contraste (ex. un bruit soudain, un mouvement ample, un changement de couleur, etc.). Si les processus de construction sont trop précis, une grande quantité d'événements très spécifiques ayant un poids faible risque d'être générée (ex. considérer une variation de température de $0,1^\circ\text{C}$ comme non similaire à une variation de $0,15^\circ\text{C}$). Si au contraire, le processus de discrétisation est trop général, peu d'éléments seront générés. Ces éléments auront probablement un poids fort mais ne seront pas spécifiques, c'est-à-dire qu'ils ne se démarqueront pas suffisamment du reste de l'expérience (ex. considérer une variation de 5°C comme similaire à

une variation de 0,1°C). Cette mesure repose donc sur l'idée de trouver un compromis entre précision et stabilité, les représentations intéressantes étant celles qui génèrent des éléments spécifiques ayant néanmoins un poids fort.

La spécificité d'un élément peut être exprimée comme la différence de similarité entre cet élément et la classe d'élément la plus générale possible, appelée la *référence*. Par exemple, pour une association qui contient les durées entre les occurrences de 2 événements e_1 et e_2 , la référence est l'association qui contient toutes les durées entre les occurrences de e_1 et des événements aléatoires e_i . La *spécificité* d'un événement 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 événement est défini en fonction de la répartition des occurrences de cet événement par rapport aux autres événements 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 événement est

$i(e) = s(e) * w(e)$. Ainsi, une classe d'évènement obtient une mesure d'intérêt élevée lorsque les deux valeurs s et w sont proches de 1. Autrement dit : d'une part son profil est suffisamment éloignée de la distribution normale de la variable observée (spécificité élevée), et d'autre part cette caractéristique n'est pas partagée uniformément avec les autres classes spécifiques (poids élevé), comme ce devrait être le cas si cette caractéristique dépendait du hasard. La spécificité élevée signifie donc que ce profil d'évènement est peu fréquent, et le poids élevé signifie que parmi les événements peu fréquents, celui-ci se démarque en étant anormalement plus fréquent que les autres. Plus cette asymétrie est forte, plus l'évènement est considéré comme intéressant.

La Figure 3.10 montre deux exemples qui illustrent ce comportement. Dans l'exemple A, certaines variations aléatoires générées par le bruit se distinguent de la référence. Cependant, en moyenne, aucune des classes qui leur correspond (ex. $c1$ et $c2$) n'a une probabilité d'apparition nettement supérieure aux autres. Au contraire dans l'exemple B, deux classes correspondant à des variations très spécifiques sont créées. La classe $c3$ est une variation récurrente, alors que $c4$ est suscitée par un aléas. Ainsi $c3$ présente une probabilité d'apparition nettement supérieure à $c4$ et bénéficie d'un poids fort.

Nous proposons cette mesure d'intérêt comme une possibilité afin d'implémenter le feedback transitionnel. Il existe cependant de nombreuses méthodes permettant d'évaluer la qualité intrinsèque d'une classification et également des outils de la théorie de l'information, qui pourraient être adaptés ici. L'étude de cet aspect du modèle sera un élément important pour les travaux futurs.

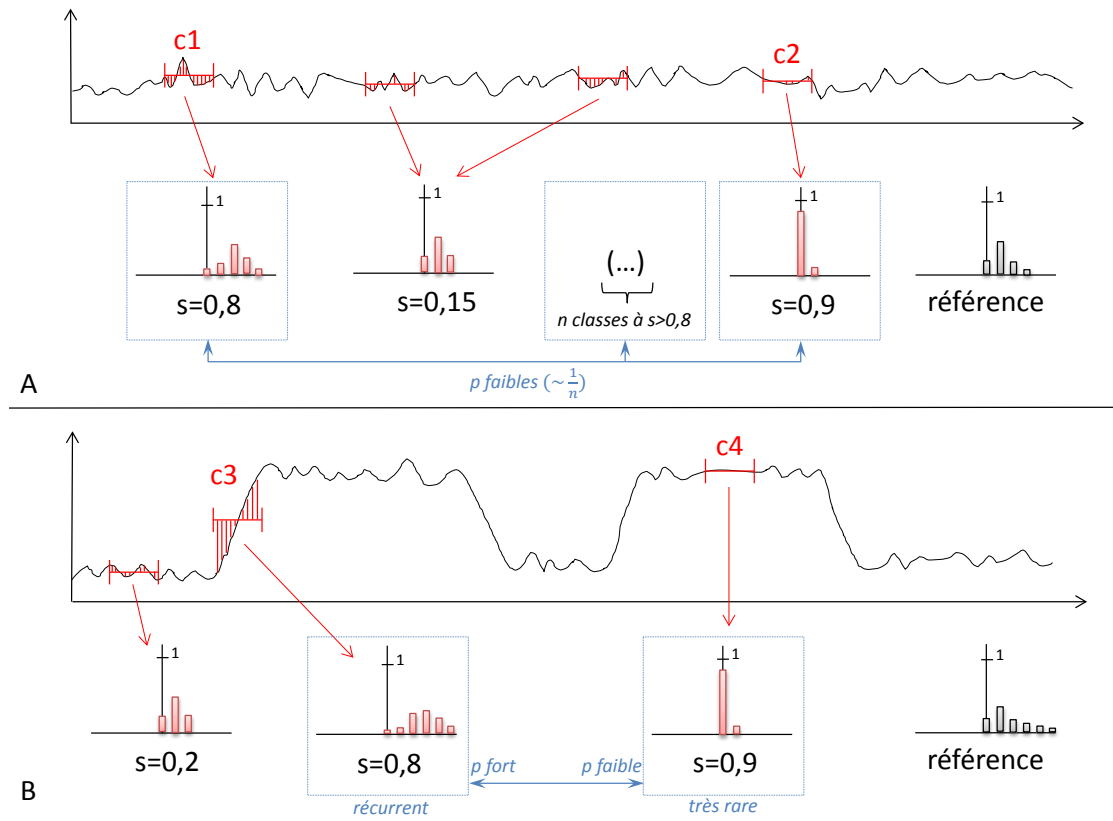


FIGURE 3.10 – Illustration du rôle des notions de "spécificité" et de "poids" dans la mesure d'intérêt posée.

3.2.4.2 Feedback prédictif (F_p)

L'espace de recherche global du système est l'ensemble des façons de discrétiser l'expérience, soit l'ensemble des fonctions possibles $F = F^D \cup F^A \cup F^S$. L'évaluation des motifs peut correspondre à leur faculté à produire des prédictions fiables, comme présenté dans [Mugan07a]. La pertinence d'une fonction peut donc être évaluée d'après sa participation à la construction de motifs prédictifs, qui se fait nécessairement en collaboration avec des fonctions correspondant à une opération différente. Nous pouvons représenter ce feedback par une évaluation $Q : f \in F \rightarrow [0, 1]$ qui s'applique aux structures générées, et influence les agents participants. Comme illustré sur la Figure 3.12, la collaboration des trois rôles d'agents permet de construire un motif dont l'évaluation fournit un feedback vers les agents participants. Cela permet par exemple de renforcer le paramètre $P1$ utilisé par l'agent $S1$. Ainsi le but du système est de trouver les combinaisons de fonctions qui permettent de construire des motifs pertinents, en maximisant

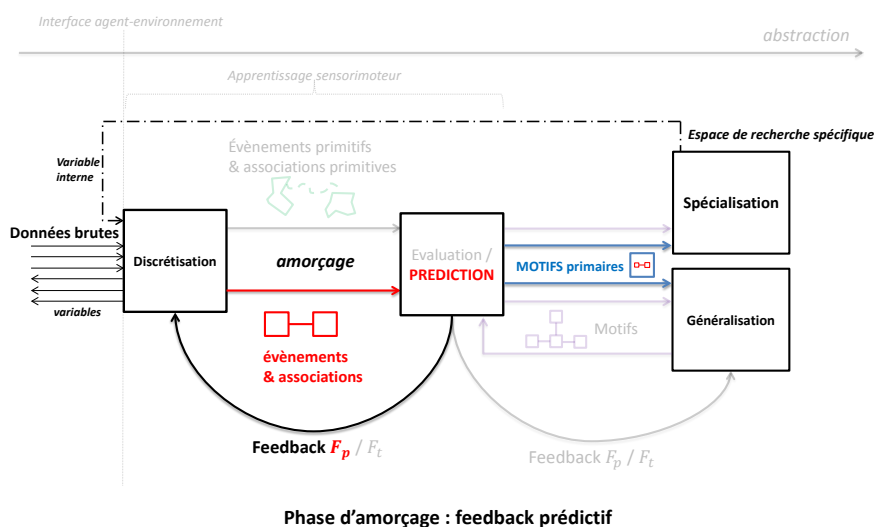


FIGURE 3.11 – Le feedback prédictif permet d'évaluer l'intérêt utile des structures produites.

l'évaluation. Cet ensemble peut être défini comme $F' \subset F = \{f' \in F \mid Q(f') > T_Q\}$ où T_Q est un seuil pour conserver uniquement les structures ayant le meilleur score.

Une fois qu'un motif est identifié comme *intéressant* (grâce à l'évaluation transitionnelle présentée ci-dessus), il peut être considéré comme une structure prédictive entre un événement e_1 et un événement e_2 , qui peut être évaluée de manière classique par exemple en fonction de son taux de succès et de sa précision. Nous définissons pour cela un score $s = acc * rel$. La confiance est définie comme le rapport du nombre de prédictions réussies sur le nombre d'événements e_1 observés : $rel = nb(predictions) / nb(e_1)$. Soit tol la tolérance de la prédiction, qui est l'écart type de la durée entre e_1 et e_2 lors des prédictions réussies. La précision 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 .

Ces deux types d'évaluation (prédiction et intérêt) sont effectués régulièrement lorsque les agents possèdent une quantité conséquente d'informations nouvelles, et sont utilisés pour "marquer" un même *espace de marquage* afin de guider le processus d'exploration/exploitation des agents. Lorsqu'un événement est suffisamment intéressant (dépassement d'un seuil), une variable interne est créée et un ou plusieurs agents A y sont connectés. Lorsqu'un événement association est suffisamment intéressant, le motif peut passer en "*mode prédiction*", c'est-à-dire que le deuxième mode d'évaluation est activé.

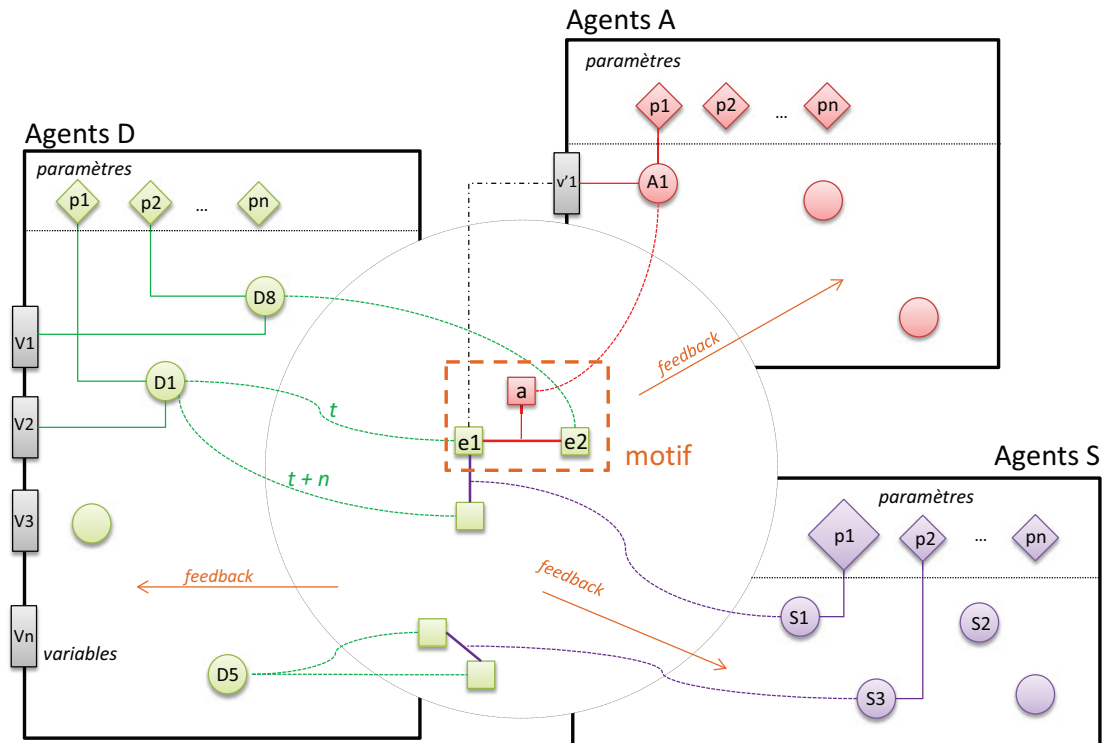


FIGURE 3.12 – Illustration de la construction d'un motif par la collaboration des trois rôles d'agents. Les agents sont représentés par des ronds. Le motif est composé d'une association reliant deux évènements issus des variables : e_1 et e_2 . e_1 est l'évènement de référence de l'agent A_1 , comme modélisé par la variable interne v'_1 à laquelle il est attaché.

3.2.5 Utilisation d'espaces de marquage pour l'interaction inter-agents

On s'intéresse ici à la façon dont les agents peuvent partager l'expérience acquise lors de l'exploration de l'espace de recherche par combinaisons et paramétrage de leurs comportements. Le principe de ce partage d'expérience est d'une part de répartir l'effort de recherche de façon équilibrée, d'autre part de partager les combinaisons efficaces trouvées pour des configurations différentes (ex. capteurs différents) mais qui pourraient néanmoins présenter des similitudes dans la manière de les discrétiser (ex. combinaisons de types d'agent et paramétrages proches).

3.2.5.1 Présentation des espaces de marquage

Lorsqu'une combinaison particulière d'agents génère un motif qui est évalué, le résultat de cette évaluation est inscrit dans un *espace de marquage* qui représente une portion de l'espace de recherche global. Un espace de marquage est une structure que peuvent percevoir certains agents et sur laquelle ils peuvent agir, constituant ainsi un moyen de communication indirect entre les agents. Ce procédé est inspiré du mécanisme de "marquage" de l'environnement par une phéromone et de la perception de cette phéromone chez les fourmis, et des propriétés d'organisation qui en découlent, comme présenté dans la partie 2.2.1. Le marquage effectué peut donc être utilisé pour guider les comportements d'exploration et d'exploitation des agents.

Concrètement, un espace de marquage est créé pour chaque combinaison de types d'agent utilisée dans un couple (D+S ou A+S). Pour rappel, un type d'agent correspond à l'implémentation particulière d'un rôle. Par exemple, un espace de marquage peut être créé pour un couple (agent A_D^i implémentant la fonction f_d^i) + (agent A_S^j implémentant la fonction f_s^j). Il existe donc un certain nombre d'espaces de marquage différents, en fonction des différentes combinaisons d'agents possibles. Un espace de marquage ainsi défini consiste en une matrice à 3 dimensions :

- Variable
- Paramètre de l'agent producteur
- Paramètre de l'agent similarité

L'ensemble des paramètres possibles pour chaque agent est spécifié par une échelle préétablie. L'espace est donc découpé en un nombre fini de positions possibles pour un couple d'agents. Chaque "case" de cette espace peut donc être marquée par une valeur d'évaluation. La qualité du marquage diffère selon qu'il provient d'une mesure d'intérêt (qualité moindre) ou d'une évaluation de prédiction (qualité plus élevée). Lorsqu'un agent met à jour une case déjà marquée, il écrase ce marquage si son évaluation est de meilleure qualité ou il le modifie en faisant une moyenne pondérée à qualité équivalente ou inférieure. De plus, l'évaluation reposant sur la mesure d'intérêt qui est comprise entre 0 et 1 peut être pondérée pour être réduite à une valeur entre 0 et 0.5 par exemple. La Figure 3.13 donne un aperçu d'un espace de marquage, et du "déplacement" exploratoire possible d'un couple lorsque les agents le composant font varier leur paramètre.

Etant donné que ce sont les motifs qui sont évalués et non les agents individuellement, le *problème de l'attribution de la récompense* se pose. Comme nous l'avons mentionné dans la partie 2.2.2.1, on peut imaginer différentes façons d'attribuer la récompense. On choisit ici la stratégie de répartir uniformément la récompense à tous les agents participant à la création du motif évalué. Etant donné que l'exploration est toujours possible autour des zones d'intérêt, les éventuels agents ayant été surévalués seront remplacés lorsque de meilleures solutions seront trouvées dans le voisinage.

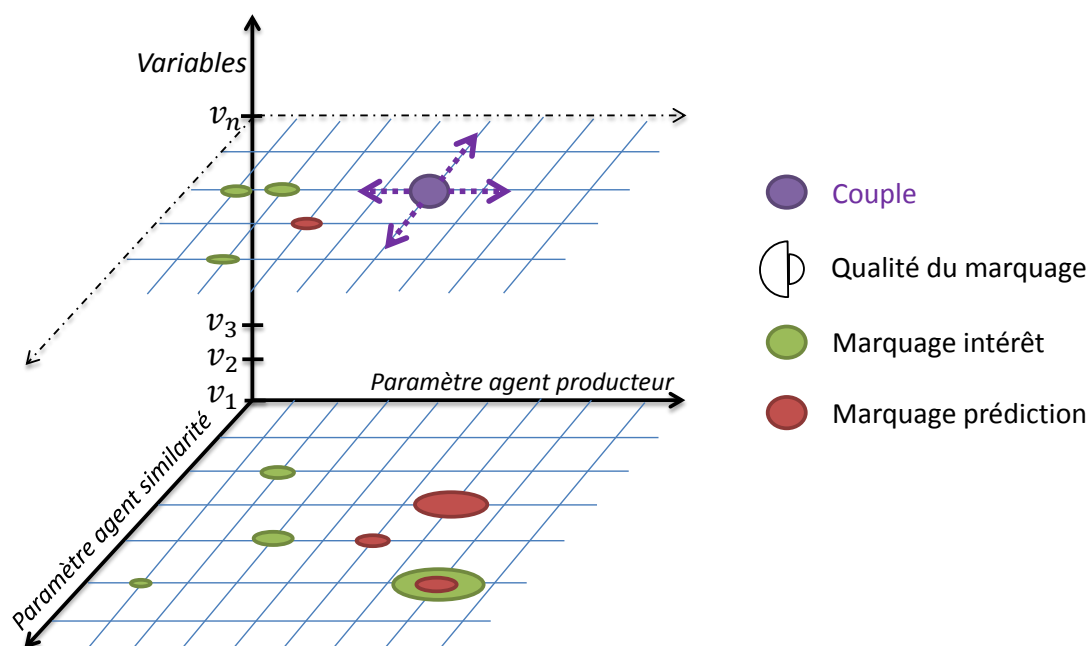


FIGURE 3.13 – Représentation d'un espace de marquage

3.2.5.2 Principe d'exploration et remarques générales

L'exploration des agents est en priorité influencée par le marquage existant pour la variable à laquelle ils sont connectés, c'est-à-dire la matrice à deux dimensions qui correspond à cette variable. Mais un agent peut aussi être influencé par des marquages réalisés pour d'autres variables que la sienne. C'est notamment le cas lorsqu'une nouvelle variable inconnue est intégrée au système. L'axe "variable" n'est pas à priori ordonné comme les deux autres axes. Cependant on peut imaginer que les variables les plus similaires soient proches les unes des autres sur cet axe, afin de faire bénéficier le mécanisme d'exploration de cette information. Pour cela un processus pourrait trier les variables en fonction par exemple de la connectivité entre les variables dans les motifs, ou par rapport à d'autres critères (ex. informations provenant du système HEMIS sur la nature des variables, leurs positions dans l'espace, etc.). Enfin, on pourrait également permettre à un couple d'être influencé par d'autres espaces de marquage, par exemple ceux qui possèdent un type d'agent en commun.

Pour aller plus loin, on peut considérer que le partage de ces espaces de marquage entre différents systèmes d'AmI (ex. l'ensemble des systèmes HEMIS) pourrait permettre de créer une expérience collective qui soit accessible à chaque système individuel. Ainsi un système particulier pourrait rapidement converger vers les méthodes

de discrétisation les plus adaptées à sa configuration propre. Cela est possible en utilisant les métadonnées du matériel disponibles dans HEMIS, qui correspondent à des variables au sein du système d'apprentissage. Cette perspective est intéressante comme alternative aux longues échelles de temps pour obtenir la grande quantité d'informations nécessaires à l'apprentissage. En effet, la collecte des données d'apprentissage serait en quelque sorte parallélisée sur la communauté des systèmes similaires qui fonctionnent en parallèle. Bien sûr, chacun des systèmes posséderait ses propres motifs particuliers, la connaissance partagée collectivement étant une métaconnaissance sur la façon de traiter les appareils connectés pour la réalisation de l'apprentissage constructiviste.

Pour conclure, cette méthode d'exploration stochastique et itérative doit permettre de gérer la combinatoire extrêmement importante de la recherche d'interprétations des données brutes au sens large. D'autre part, on peut notamment initialiser le système avec des agents cherchant les régularités les plus "évidentes" ou remarquables, et se servir par la suite de la connaissance acquise pour réduire l'espace de recherche. C'est en effet une des idées de l'approche développementale de l'apprentissage pour aborder la complexité de ce type de problème, que de commencer par apprendre une représentation élémentaire pour la complexifier progressivement. Cela est possible car le processus est conçu comme un apprentissage en continu tout au long de la "vie" du système. A ce titre, des "processus de régulation" seront nécessaires afin de réguler le nombre et l'activité des agents explorateurs au sein du module d'apprentissage afin que celui-ci soit maintenu à un niveau d'utilisation des ressources constant.

3.2.6 Modèle étendu et propositions

Au delà du problème d'amorçage de l'apprentissage sensorimoteur en environnement continu, le modèle est conçu pour que l'apprentissage se poursuive sur le même principe pour des niveaux d'abstraction supérieurs. Certains aspects comme la spécialisation par la création de variables internes sont implémentés et ont été présentés précédemment. Les capacités de *spécialisation* et de *généralisation* font partie de ce processus. Ces termes génériques recouvrent diverses pistes que nous proposons de présenter dans cette partie.

3.2.6.1 Généralisation

Sur la Figure 3.14 est illustrée la capacité de créer de nouveaux motifs en utilisant un autre motif comme composant. La montée en abstraction permet de créer de nouveaux motifs plus complexes à partir de motifs existants. On appelle motif complexe un motif qui est composé à partir d'autres motifs déjà existants, et non des événements primaires uniquement. Ce type de motif est une première forme d'abstraction car il permet de représenter un concept unique à partir de plusieurs événements. Par exemple,

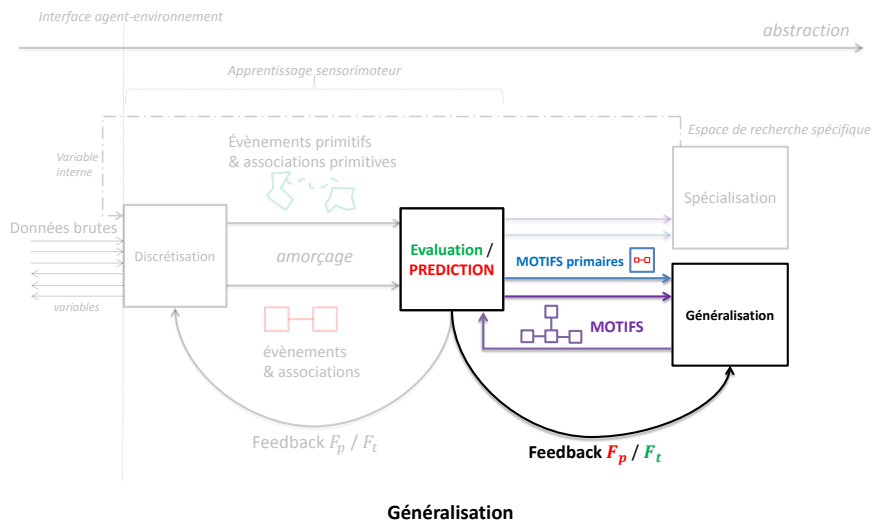


FIGURE 3.14 – L'étape de généralisation

soit un premier motif M exprimant l'entrée ou la sortie du logement par l'occupant : {variation du capteur : *Ouverture/fermeture de porte* dans la zone *hall* + variation du capteur : *Présence* dans la zone *hall*}. Le motif M peut être considéré comme un événement et être associé à un autre événement E par exemple : {variation du capteur CO_2 dans la zone *global*}. L'association de ces deux événements $M + E$ peut permettre de faire la différence entre l'arrivée dans le logement ou bien le départ, suivant que le taux de CO_2 augmente ou diminue, mais aussi de renforcer le concept de départ ou d'arrivée. En effet, l'occupant peut très bien franchir la porte pour des raisons diverses sans pour autant partir ou rester longtemps dans le lieu. Par exemple, la personne sort les poubelles et revient immédiatement ou encore, la personne entre pour chercher un objet qu'elle a oublié et repart aussitôt. La combinaison de différents motifs faisant intervenir plusieurs modalités différentes peut ainsi permettre à la fois d'exprimer des motifs plus complexes décrivant des situations abstraites, mais aussi de renforcer la capacité d'identifier et de distinguer des motifs.

La capacité de *généralisation* peut s'exprimer par différents mécanismes. Ainsi, la généralisation peut passer également par la reconnaissance de régularités internes au système. Par exemple déterminer les analogies entre deux motifs différents fonctionnant sur le même principe. On peut ainsi imaginer un motif décrivant les interactions liées à un radiateur dans la zone *chambre*, et un autre similaire dans la zone *salon*. Un motif générique pourrait être créé à partir de ces motifs, pour exprimer une régularité de plus haut niveau. La création d'un mécanisme comparable à celui des *items synthétiques* présenté par [Drescher91] peut également être envisagée. Il s'agit de

créer des *variables internes* liées aux motifs appris qui sont intégrées au système suscitant un apprentissage de régularité à un niveau plus abstrait portant sur le système lui-même. Cette deuxième forme d'abstraction est plus difficile à mettre en place, et nous la laisserons de côté pour le moment.

3.2.6.2 Spécialisation

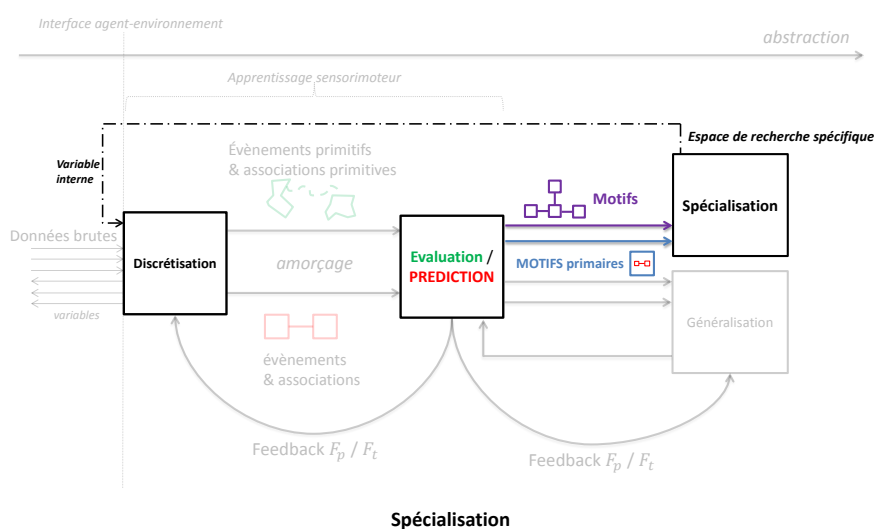


FIGURE 3.15 – L'étape de spécialisation

La réduction de l'espace de recherche autour d'un motif permet donc l'exploration de nouveaux moyens plus spécifiques d'interpréter les données. Comme illustré sur la Figure 3.15 cette réduction peut être modélisée au sein du système par la création de nouvelles variables (internes) qui sont issues d'une interprétation des variables initiales préalablement acquise.

Par exemple, une fois qu'un lien entre un actionneur A et un capteur C est établi, il s'agit de déterminer plus précisément les liens entre ces deux variables. Ainsi, le motif initial peut exprimer le fait qu'une variation de A est suivie dans 80% des cas par une variation de C avec un laps de temps compris entre 30 secondes et 1 minute. Plus spécifiquement, on peut déterminer que lorsque la variation de A est une augmentation, celle de C est aussi une augmentation et inversement. Pour cela on peut chercher une nouvelle façon d'interpréter les données brutes, mais en restreignant le périmètre de ces données dans l'espace (seulement deux variables concernées) et dans le temps (les données sont prises uniquement durant les événements). Ensuite, on peut trouver le lien entre l'amplitude de la variation de A et celle de C . On peut aussi trouver que

lorsqu'une tierce variable X est dans l'état y , le taux de réussite de la prédiction passe à 90% et ainsi de suite.

Ainsi, lorsqu'un motif est confirmé parce qu'il permet de faire des prédictions fiables, il ouvre de nouvelles possibilités d'apprentissage. En effet, on peut considérer que le motif se situe dans une zone d'intérêt de l'espace de recherche global. Le système a trouvé une combinaison parmi toutes les variables et les diverses façons de les discrétiser et de les associer, mais il y a probablement d'autres combinaisons possibles qui impliquent les mêmes variables. Le système peut ainsi sélectionner deux variables particulières sur la base d'un motif primaire appris. Une fois que l'on a ainsi réduit l'espace de recherche, on peut réaliser un apprentissage plus spécifique, dans un espace de recherche plus restreint.

3.3 Présentation d'une implémentation du modèle

Dans cette partie nous présentons une implémentation pour chaque composante du modèle générique présenté précédemment. Cette implémentation constitue une version élémentaire du système, à partir de laquelle on peut expérimenter. L'objectif est tout d'abord de tester la dynamique du modèle, c'est pourquoi nous choisissons de donner une version basique de chaque composante en essayant de garder des implémentations aussi simples que possible. De plus il ne faut pas perdre de vue que chaque fonction exécutée par un agent particulier n'a pas pour vocation d'être adaptée à l'apprentissage de tout type de régularités. Au contraire, le système doit s'adapter pour apprendre ce qui est possible avec les moyens qu'il possède, et c'est la multiplicité et la diversité des agents que l'on intégrera au sein du système qui permettra d'apprendre toute sorte de régularités.

3.3.1 Élément de représentation

L'*histogramme* est la structure de donnée choisie pour représenter les événements et les associations. Cette structure de donnée est notamment beaucoup utilisée dans les systèmes de vision artificielle. L'histogramme permet de représenter de manière intuitive la distribution de valeurs d'une source. Par exemple, la variance sur une fenêtre temporelle, ou la durée entre deux événements. Il s'agit d'une structure de donnée assez générique permettant de facilement compresser l'information et d'effectuer simplement et efficacement certaines opérations. Il existe d'ailleurs des modules matériels (puces) dédiés aux calculs sur les histogrammes (ex. [Pirim13]). Il est possible de fusionner les données de deux histogrammes pour obtenir une nouvelle répartition, qui s'en trouve en quelque sorte renforcée. En effet plus on ajoute de valeurs, plus la répartition est stable car l'ajout d'une nouvelle valeur a moins d'impact. Un aspect intéressant est que la discrétisation de l'histogramme peut être modulée pour obtenir différents types de classes.

3.3.2 Comportement générique des agents

Tous les agents quel que soit leur rôle et leur type possèdent un comportement global commun qui peut être défini comme suit :

```
while active do
  if updateChanges(); /* vérification de l'état de l'agent au
    sein du système */
  then
    performTask(); /* réalisation de la tâche correspondant
      au rôle de l'agent */
    evaluate(); /* évaluation et comportement
      exploration/exploitation */
  end
end
```

FIGURE 3.16 – Méthode *execute()* d'un agent.

La méthode *updateChanges()* permet principalement de vérifier que l'agent appartient au moins à un couple actif, et de maintenir à jour une liste de couples auxquels participe l'agent. D'autre part, pour les agents producteurs, il faut vérifier que la variable à laquelle ils sont connectés est toujours disponible dans le système.

La méthode *evaluate()* permet d'évaluer les couples auxquels participe l'agent pour marquer l'environnement, et de décider du comportement d'exploration ou d'exploitation. L'espace de marquage est une "grille" sur laquelle les agents se déplacent lorsqu'ils modifient leurs paramètres. De ce fait les couples d'agents se déplacent également sur cette grille (nous donnerons plus de précisions à ce sujet dans la partie 3.2.5). Deux *sous-rôles* pouvant s'appliquer à n'importe quel rôle sont spécifiés afin de faciliter la tâche d'exploration et d'exploitation.

- Les *agents exploreurs* se "déplacent" plus rapidement sur la grille (variations de paramètres plus importantes) et à chaque occasion. Les représentations qu'ils génèrent ne servent qu'à l'évaluation (pas de variable interne créée, pas de prédiction activée).
- Les *agents exploiters* se déplacent lentement pour explorer mais se "téléportent" pour aller sur une meilleure position connue. Ils ne se déplacent que s'ils ne participent pas encore à une prédiction. Contrairement aux agents exploreurs, ils peuvent "exploiter" une position en générant des éléments de représentation disponibles pour le système. Les représentations générées sont utilisées pour créer des motifs et évaluer des prédictions.

L'algorithme suivant résume la procédure effectuée par la méthode *evaluate()*.

```

foreach Couple c do
  if readyForEvaluation(); /* on vérifie si le couple possède
  suffisamment de nouvelles informations, s'il est stable,
  etc. */
  then
    interest ← evaluateRepresentationInterest(); /* mesure d'intérêt
    du couple basé sur l'intérêt des éléments de
    représentation générés */
    markSpace(); /* marquage intérêt */
    if isExplorator() then
      explore(); /* à partir de la meilleure position, se
      déplace vers des paramètres non explorés ou peu */
    else
      if Possède des motifs en mode prédiction then
        foreach Prédiction p do
          if readyForPrediction(); /* vérification que
          suffisamment de nouvelles occurrences ont eut
          lieu */
          then
            evaluatePrediction(); /* évaluation de la
            prédiction. Si score trop faible,
            désactivation du mode prédiction. */
            markSpace(); /* marquage prédiction */
          end
        end
      else
        if interest ≥  $\epsilon$  then
          if Couple.agentProducteur : Agent D then
            | L'évènement génère une variable interne (exploit())
          end
          if Couple.agentProducteur : Agent A then
            | L'association passe en mode prédiction (exploit())
          end
        else
          explore(); /* à partir de la meilleure position,
          explorer les paramètres proches */
        end
      end
    end
  end
end

```

FIGURE 3.17 – Méthode *evaluate()* d'un agent.

3.3.2.1 Exploration

La méthode *explore()* code le comportement exploratoire des agents de manière assez simple. En partant d'une position intéressante (une des positions connues ayant le meilleur score dans l'espace de marquage) les agents explorateurs se déplacent aléatoirement jusqu'à trouver une position libre (non occupée par d'autres agents pour une même variable) et non explorée ou peu explorée comparée aux autres. Les agents exploitateurs explorent quant à eux une zone d'intérêt en s'éloignant en spirale autour d'une position initiale choisie aléatoirement parmi plusieurs positions intéressantes jusqu'à trouver une position libre.

L'exploration de l'espace de recherche par les agents est donc un processus stochastique, à l'image des algorithmes évolutionnistes par exemple. A l'échelle d'un type particulier d'agent, l'activité de recherche est en partie aléatoire mais itérativement guidée par les résultats obtenus. Les agents explorateurs couvrent rapidement l'espace de recherche pour cibler les zones intéressantes, et augmenter les chances que d'autres agents les explorent plus en détails.

A l'échelle du système, certains types d'agents (exécutant des fonctions très générales) peuvent cibler plus rapidement les régularités évidentes et permettre ainsi un apprentissage plus spécifique dans un espace de recherche plus restreint. Par exemple, en apprenant d'abord à reconnaître des événements "*variations*" et les régularités associées, puis des événements "*augmentation*" et "*diminution*" et de nouvelles régularités plus précises associées.

3.3.2.2 Exploitation

La méthode *exploit()* (réservée donc aux agents exploitateurs) se traduit différemment dans le cas des agents *D* et des agents *A* même si dans les deux cas, il s'agit de rendre disponible dans le système l'élément produit :

- Dans le cas des agents *D*, une classe d'évènements est rendue disponible pour la recherche d'association, par la création d'une variable interne. Notons que si la classe d'évènement disparaît car elle est fusionnée ou si l'agent décide d'explorer une autre position, cette variable interne sera alors détruite ainsi que tous les agents associations connectés.
- Dans le cas des agents *A*, une association est sélectionnée entre deux évènements et le "mode prédiction" est activé. C'est-à-dire que l'on va évaluer le motif composé de cette association et des deux évènements qu'elle relie, grâce à la mesure de prédiction.

Dans tous les cas, à chaque fois que la méthode *evaluate()* est appelée, un agent exploitation a une chance d'exploiter un élément en fonction de l'intérêt de cet élément (tirage aléatoire pondéré : plus la position est intéressante, plus l'agent exploitateur risque de l'exploiter si il est toujours en phase d'exploration). Si cet élément perd de l'intérêt,

l'agent peut arrêter de l'exploiter et lorsqu'il n'exploite plus aucun élément sur une position, alors il reprend une phase d'exploration.

3.3.3 Les agents D

3.3.3.1 Comportement général

Cet agent réalise principalement trois tâches à l'appel de la méthode *performTask()* :

- L'agent découpe l'expérience en éléments de représentation arbitraires que l'on nommera *instances* et les diffuse aux différents couples auxquels il appartient.
- Maintenir la référence en fusionnant toutes les instances
- En mode prédiction : Produire les évènements appartenant à des motifs.

3.3.3.2 Implémentations proposées

Agent D : "fenêtre glissante" (AD_{FG})

Cet agent collecte les données brutes sur une fenêtre de temps avec une certaine fréquence (ex. 20 prélèvements par fenêtre) et les stocke dans un histogramme qui représente donc une *instance d'évènement*. La fenêtre se déplace avec un certain pas (ex. un cinquième de la durée de la fenêtre). Pour cet agent la construction de l'élément de référence se fait simplement par la fusion de toutes les instances d'évènement créées. La durée de la fenêtre est définie par le paramètre de l'agent. Lorsque l'agent explore son espace de recherche, la taille de la fenêtre varie. Pour une variable telle que la température qui évolue lentement, on s'attend à ce qu'une taille de fenêtre importante (ex. de l'ordre de plusieurs dizaines de minutes) soit adaptée. Pour une variable telle que la consommation électrique, des variations rapides peuvent survenir et l'on s'attend à ce qu'une taille de fenêtre réduite soit plus adaptée. Pour certaines variables il est moins évident de deviner a priori quelle échelle de temps est pertinente. D'autre part, plusieurs échelles de temps différentes peuvent être adaptées pour une même variable dans des concepts différents. Par exemple des variations fines de température peuvent participer à la caractérisation d'une activité (ex. faire la cuisine) alors que d'autres plus amples peuvent correspondre à des périodes de présence ou d'absence sur plusieurs jours.

Agent D : "fenêtre glissante variations" (AD_{FGV})

Comme nous l'avons mentionné dans l'état de l'art, le cerveau n'enregistre vraisemblablement pas des contextes figés, mais plutôt des variations importantes : par exemple l'écart entre les notes d'une mélodie plutôt que le son des notes. D'autre part, l'attention est attirée par des variations, comme par exemple l'arrêt soudain d'un bruit de fond, alors que ce bruit n'était pas perçu jusqu'alors. Les évènements primitifs qui entrent en jeu dans les motifs les plus élémentaires de l'apprentissage sensorimoteur

sont probablement des variations importantes facilement identifiables et très générales. Ainsi, au lieu de collecter directement les données brutes sur la fenêtre temporelle, l'agent AD_{FGV} collecte, pour chaque point, l'écart de la valeur avec la moyenne glissante sur la fenêtre temporelle, comme illustré sur la Figure 3.18. Si cet agent possède la taille de fenêtre adaptée, il peut donc distinguer les variations des états stables, même dans un contexte bruité. Les variations importantes et suffisamment rares devraient ainsi être considérées comme des événements grâce à une spécificité et un poids élevés dans la mesure d'intérêt.

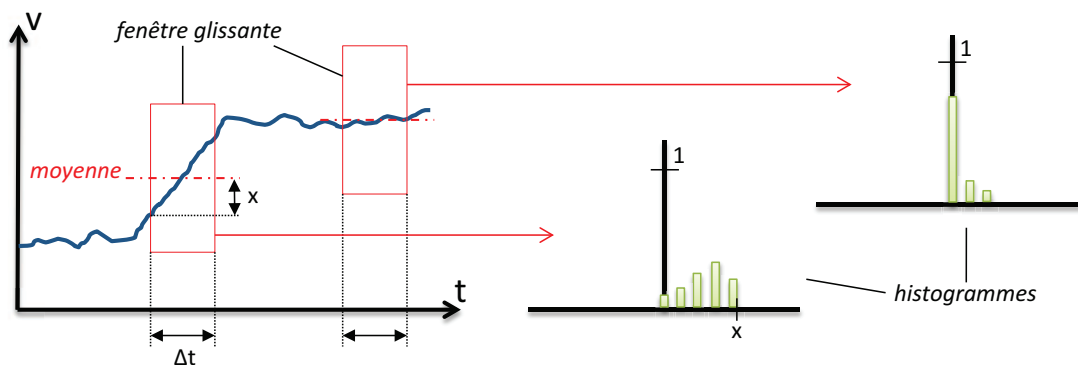


FIGURE 3.18 – Agent D : variations sur fenêtre glissante

En général les variations importantes ou franches sont les variations qui sont pertinentes pour être considérées en tant qu'évènements et associées à d'autres évènements. Néanmoins nous ne voulons pas créer un critère de décision basé sur l'amplitude des variations, car cela peut changer en fonction des variables et nous souhaitons un modèle plus générique. Nous faisons l'hypothèse que ces variations sont moins fréquentes que les états stables. Donc pour une taille constante de fenêtre glissante, les fenêtres représentant une variation apparaîtront comme spécifiques en comparaison de la référence. A l'inverse pour une variable qui varie en permanence, les états stables seront considérés comme des évènements. La référence permet ainsi de créer un modèle de comparaison de ce qui est "normal" à partir de l'expérience, et évite de définir ce modèle a priori (par exemple en définissant un critère pour juger de l'amplitude d'une variation).

Pour vérifier et illustrer cette façon d'interpréter, une expérience de principe a été effectuée sur 24 heures, avec des données réelles (bureaux). Pour la température, 4 fenêtres différentes sont testées : 5 minutes, 10 minutes, 30 minutes et 1 heure. Une fonction de test est à 1 si le cumul de la variance sur la fenêtre temporelle passée est supérieur à deux fois la moyenne des cumuls sur les 24 heures, à 0 sinon. La moyenne des cumuls sur les 24 heures joue ici le rôle de la référence par rapport à laquelle on peut

comparer une fenêtre temporelle particulière et, pour simplifier, le critère de supériorité remplace la mesure de spécificité. On peut voir sur la Figure 3.19 que pour une même procédure, on peut cibler des variations plus ou moins précises en faisant seulement varier la taille de la fenêtre.

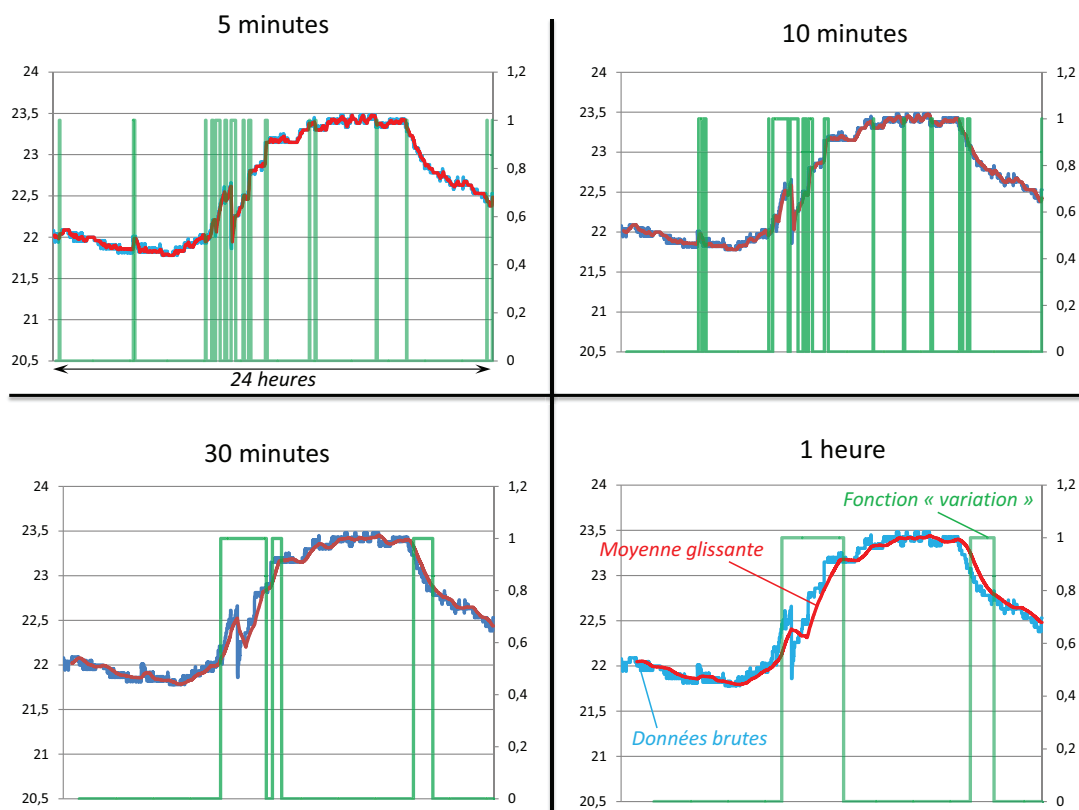


FIGURE 3.19 – Test de différentes fenêtres temporelles pour l'interprétation d'événements en tant que variations

La Figure 3.20 montre la même procédure appliquée à d'autres grandeurs sur les mêmes 24 heures, par exemple le CO₂ et la consommation. Concernant le CO₂ on voit qu'avec une fenêtre de 1 heure, on peut nettement identifier 3 événements pertinents de la journée. Pour la consommation, l'utilisation d'une fenêtre temporelle réduite permet de déceler un événement pour chaque activation ou désactivation d'un appareil consommateur. Dans une étape de spécification on pourrait ensuite déceler des régularités prenant en compte l'amplitude de ces variations pour distinguer le rôle d'un appareil particulier dans un motif.

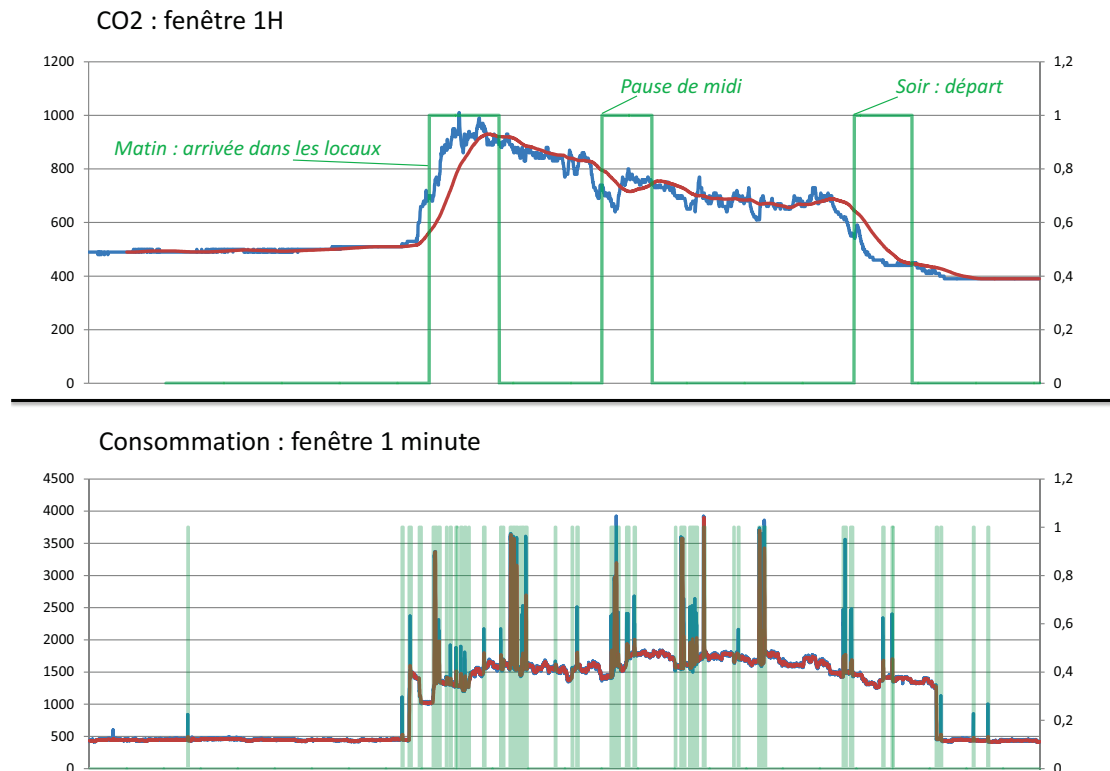


FIGURE 3.20 – Interprétation d'évènements en tant que variations pour le CO2 et la luminosité sur 24 heures de données réelles

3.3.4 Les agents A

3.3.4.1 Comportement général

Cet agent réalise aussi trois tâches principales dans méthode *performTask()* :

- L'agent tente d'associer toutes les nouvelles occurrences de l'élément de référence avec des évènements relatifs parmi l'ensemble des évènements intéressants dans le système. Les éléments choisis ainsi que la manière d'associer dépendent de la fonction implémentée par l'agent. Pour chaque élément relatif, une "instance" d'association est générée et diffusée aux couples auxquels cet agent appartient.
- Maintenir la référence en fusionnant toutes les instances
- En mode prédiction : Produire les associations entre les occurrences des évènements du motif.

3.3.4.2 Implémentations proposées

Agent association "binaire durée relative" (AA_{BDR})

L'agent proposé est l'*agent binaire durée* (AA_{BDR}) dont le principe est le suivant : interpréter les occurrences des événements deux à deux en fonction des durées relatives (par rapport à leurs fréquences respectives) qui les séparent en moyenne. A chaque nouvelle occurrence de l'*événement de référence* modélisée par la variable interne, l'agent récupère la durée qui sépare cette occurrence des occurrences les plus proches de chaque autre événement intéressant (*événement relatif*). Le paramètre de cet agent est binaire : soit il mesure la distance à un événement relatif antérieur (régularité de type rétro-prédiction), soit la distance aux événements relatifs qui lui succèdent (régularité de type prédiction). Cette durée est transformée en durée relative en la multipliant par la fréquence de l'événement relatif, de manière à homogénéiser les représentations entre les différents événements et à pouvoir les comparer. Cette durée relative est l'information qui est stockée dans l'histogramme, comme l'illustre la Figure 3.21. La référence est l'histogramme obtenu en fusionnant les données de tous les événements relatifs.

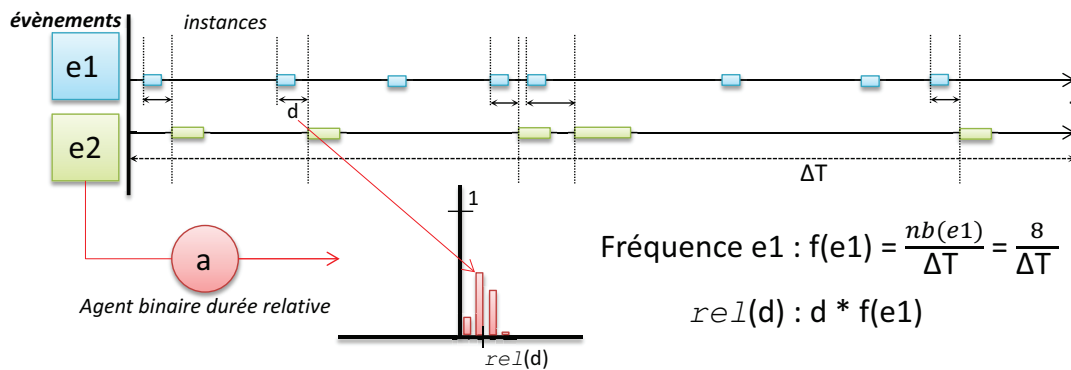


FIGURE 3.21 – Agent association : binaire durée relative (l'agent est connecté à l'événement $e2$, et génère une représentation de l'association avec l'événement relatif $e1$, à partir des occurrences successives)

L'idée générale est que la durée entre deux événements constitue une régularité si elle se distingue du hasard. La référence offre ainsi une représentation du hasard qui ne provient pas d'un modèle défini a priori mais qui est générée par l'expérience. Après un nombre suffisant d'exemples, l'histogramme représentant les durées relatives entre deux événements corrélés devrait ainsi être spécifique par rapport à la référence. Une hypothèse est qu'il est plus facile de repérer les régularités entre événements proches. Mais lorsque les événements impliqués sont peu fréquents, cette notion

de proximité temporelle est relative. C'est pourquoi nous utilisons cette représentation tenant compte de la fréquence des événements. Comme pour l'agent AD_{FGV} nous effectuons une expérience de principe sur les mêmes données réelles pour illustrer ce comportement. La Figure 3.22 montre les événements obtenus sur une période de 24 heures pour les facteurs environnementaux suivants : Luminosité (BRI) ; Température (TMP) ; CO₂ ; Consommation (POW) et Humidité (HUM) dans une zone *bureau*.

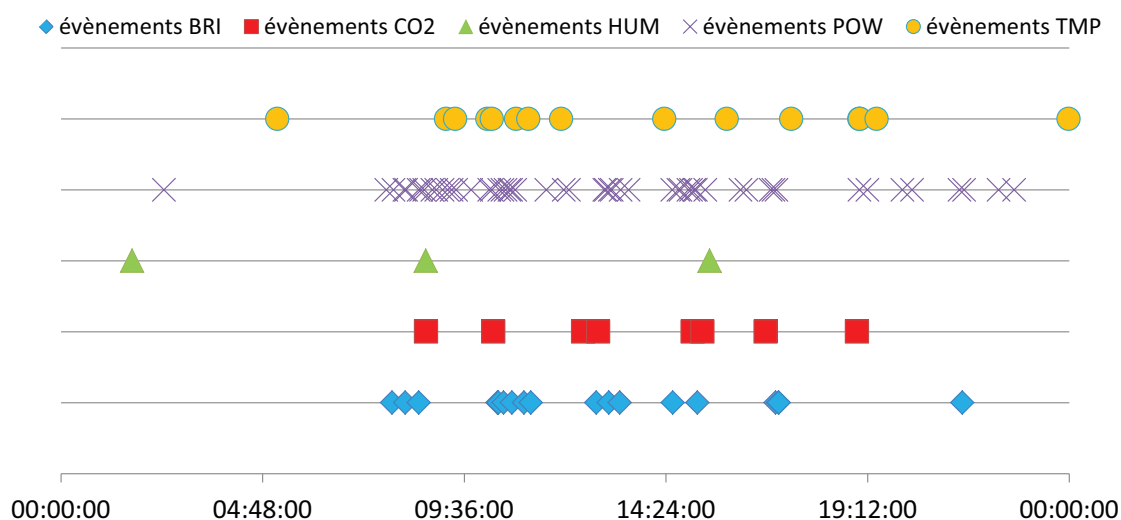


FIGURE 3.22 – Les événements de plusieurs facteurs environnementaux générés par le mécanisme de variation sur 24 heures

Pour simplifier nous testons seulement la notion de proximité temporelle relative en remplacement de la notion de spécificité par rapport à une référence. Pour cela, une fonction sélectionne les durées des événements les plus proches qui sont nettement inférieures (d'après un critère α) à la moyenne des durées relatives pour tous les événements. La Figure 3.23 montre les associations créées grâce à cette procédure.

Le nombre d'associations obtenues dépend ici de la contrainte du critère α . Le but de ce processus n'est pas de détecter directement les associations pertinentes, mais de proposer un ensemble d'associations "réalistes", pouvant être mises à l'épreuve de la prédiction. Les associations obtenues dans cette expérience semblent intuitivement correspondre à cet objectif, si l'on cherche des associations exprimant une proximité temporelle relative. Par exemple, les variations de luminosité associées aux variations de consommation en début de journée pourraient appartenir à un motif identifiable. De même pour les variations d'humidité et de CO₂ le matin qui sont provoquées par l'arrivée des occupants dans les bureaux.

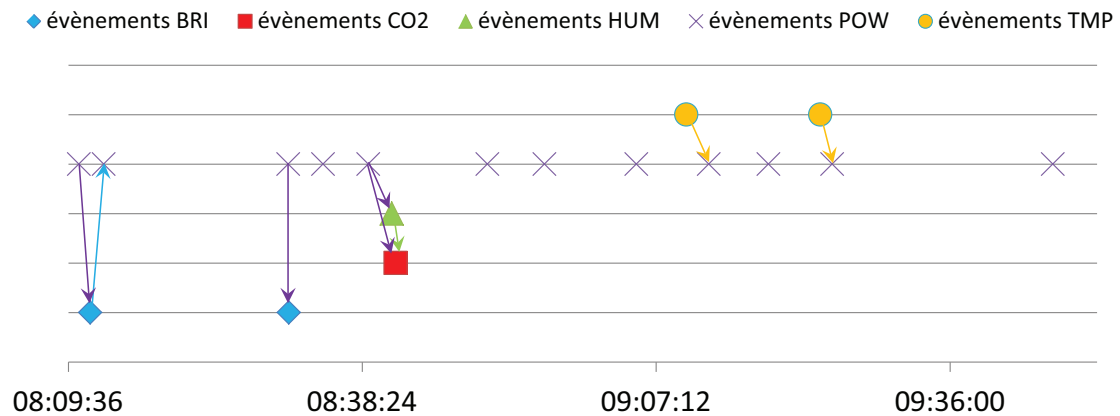


FIGURE 3.23 – Les associations générées entre 8 heures et 10 heures

3.3.5 Les agents S

3.3.5.1 Comportement général

Durant l'appel de la méthode *performTask()*, l'agent effectue des opérations différentes suivant le rôle de l'agent producteur du couple :

```

foreach Couple c do
  if Couple.agentProducteur : Agent D then
    classifyNewInstances(); /* Pour chaque nouvelle instance, la
    fusionner avec la classe d'évènement la plus similaire
    ou créer une nouvelle classe */
  end
  if Couple.agentProducteur : Agent A then
    selectAssociations(); /* Trier les instances pour ne
    conserver que celles que l'on peut considérer comme
    des associations */
  end
end

```

FIGURE 3.24 – Méthode *performTask()* d'un agent S.

La méthode *compare()* est l'opération essentielle de l'agent similarité. Elle permet de comparer deux éléments de représentation du couple, par exemple une instance avec une classe d'évènement comme dans la méthode *classifyNewInstances()*. Outre

son utilisation dans l'exécution de la méthode *performTask()*, la méthode *compare()* est nécessaire pour réaliser d'autres opérations sur les structures de représentation, comme par exemple l'évaluation de la mesure d'intérêt. C'est pourquoi l'agent S peut exécuter *compare()* en dehors de sa méthode *execute()* à l'initiative d'autres agents.

3.3.5.2 Implémentations proposées

Agent "histogramme simple intersection" (AS_{HSI})

L'implémentation des agents similarité dépend naturellement de la structure de données choisie, en l'occurrence l'histogramme. La fonction de comparaison utilisée est l'intersection comme illustrée par la Figure 3.25. Cette valeur est complétée par une mesure d'*entrelacement* qui évalue à quel point les classes des deux histogrammes sont alternées. On pourrait également utiliser un algorithme de comparaison dédié à ce genre d'opérations de comparaison sur les histogrammes, tel que par exemple l'algorithme appelé *Dynamic Time Warping* (DTW) (voir [Al-Naymat09]). Cela permet d'améliorer la comparaison par intersection qui peut être un peu biaisée dans certains cas. On parle ici d'*histogramme simple* pour signifier le fait que les classes sont de taille homogène et réparties de manière fixe sur les données par rapport à l'amplitude des valeurs de la référence. La taille des classes de l'histogramme dépend du paramètre de l'agent. Ce paramètre varie entre 0 et 0.5, valeurs correspondant respectivement à la taille minimale (une classe pour chaque valeur) et à la taille maximale (2 classes). Donc plus ce paramètre est élevé, moins la comparaison sera "précise".

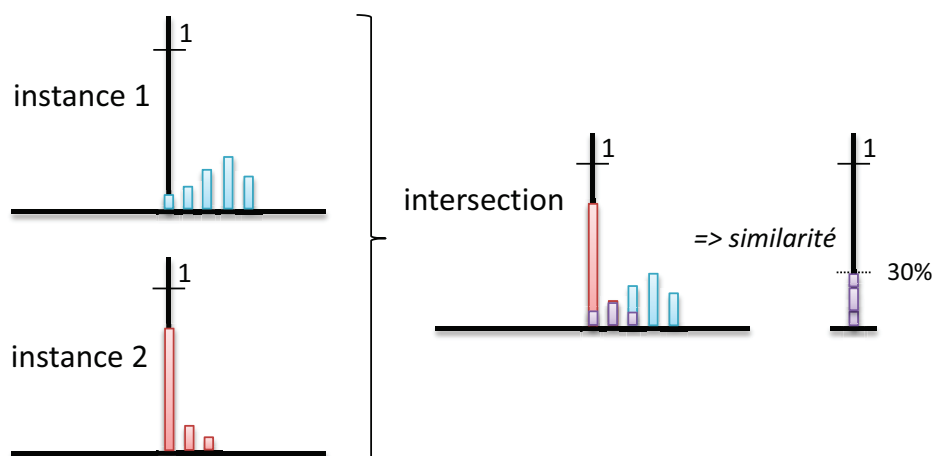


FIGURE 3.25 – Agent similarité : intersection

4

Application et résultats

▷ Dans le chapitre précédent, nous avons présenté un modèle général d'apprentissage, ainsi que les cadres théoriques et applicatifs dans lesquels il s'inscrit. Nous avons enfin présenté une implémentation possible de ce modèle. Dans cette partie nous présentons et commentons les expérimentations réalisées avec ce système. ◁

Plan du chapitre

4.1	Présentation des Expérimentations	125
4.2	Amorçage et apprentissage de motifs élémentaires sur simulateur . .	126
4.2.1	Expériences concernant la mesure d'intérêt	126
4.2.2	Création de motifs sensorimoteurs primaires	132
4.2.3	Temps d'apprentissage : influence du nombre d'agents et du partage des espaces de marquage	139
4.3	Expérimentations en environnement réel	142
4.3.1	Commentaires sur l'exploitation des données réelles et sur les performances	143
4.3.2	Création de motifs sensorimoteurs primaires	144
4.3.3	Bilan des expériences en environnement réel	150

4.1 Présentation des Expérimentations

Un problème important dans le domaine de l'AmI, du point de vue expérimental, est l'implication d'échelles de temps variées, potentiellement longues. Même si certaines régularités se produisent sur des temps très courts (ex. variations de luminosité, de consommation énergétique), beaucoup d'autres impliquent des temps longs (ex. température, CO₂). Il est clair que dans une démarche de programmation itérative expérimentale, il s'agit d'une contrainte majeure. Le même problème se pose en robotique développementale, c'est pourquoi l'utilisation de la simulation est fréquente, préalablement aux tests sur un robot physique. Nous proposons donc tout d'abord de tester le système sur des données simulées, à l'aide d'un simulateur que nous programmons.

Dans une deuxième partie, nous présentons des résultats montrant le système directement en interaction avec le système d'AmI HEMIS qui est installé dans les bureaux de l'entreprise. Le module d'apprentissage est en effet conçu pour fonctionner en temps réel, ce qui demande des temps d'expérimentation très longs. Néanmoins sur des temps d'expérimentation courts, certains motifs très fréquents et rapides peuvent apparaître. Théoriquement, le système d'apprentissage de motifs tel que nous l'avons défini est indépendant de l'échelle, donc si un motif peut être appris sur une échelle de temps courte, un motif proportionnellement identique sur une échelle de temps plus longue sera reconnu de la même façon.

Pour finir, on peut également tester des temps d'apprentissage plus longs grâce à une méthode dédiée. Il s'agit de faire évoluer le système beaucoup plus rapidement qu'une exécution en temps réel, en utilisant des données brutes préalablement acquises et en synchronisant l'horloge au rythme où le système les traite. Cette façon de procéder permet de vérifier les résultats de l'apprentissage sur des périodes longues de données réelles en un temps relativement court. Pour cela il faut disposer de données réelles brutes que l'on a préalablement exhaustivement enregistrées. Cette méthode suppose néanmoins que le système d'apprentissage soit passif au sens où il ne peut donc pas influencer HEMIS pour la prise d'action durant la phase d'apprentissage.

Nous pouvons donc effectuer trois catégories d'expériences, suivant l'origine des données connectées au module d'apprentissage (i.e. les variables) :

- Simulateur
- HEMIS temps réel
- HEMIS temps accéléré (simulation du temps réel)

L'ensemble des expériences est réalisé sur une machine disposant d'un processeur 4 cœurs (8 threads) à 2.2 GHz¹ et de 8Go de mémoire RAM.

1. Intel®Core™ i7-2720QM Processor

4.2 Amorçage et apprentissage de motifs élémentaires sur simulateur

Le simulateur consiste à générer en temps réel les valeurs d'un certain nombre de variables dont on peut spécifier les caractéristiques d'évolution (ex. bruit, variations), et les interactions (liens temporels entre variations de différentes variables). Les variables sont bruitées avec une amplitude paramétrée et on peut spécifier des variations franches qui apparaissent régulièrement. On peut choisir que des variations surviennent les unes après les autres avec un délai aléatoire borné. Ces activations sont répétées après un intervalle de temps dont la durée est aléatoire et bornée. La Figure 4.1 illustre l'ensemble des paramètres du simulateur permettant de spécifier l'activité des variables et certaines régularités. Ces paramètres sont explicités dans le Tableau 4.2.

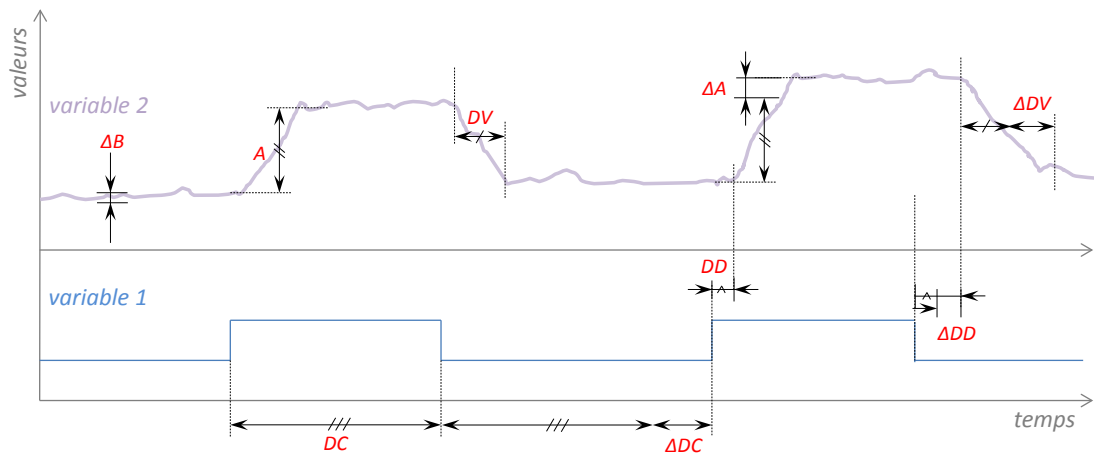


FIGURE 4.1 – Les différents paramètres du simulateur pour caractériser l'activité des variables (légende présentée dans le Tableau 4.2).

4.2.1 Expériences concernant la mesure d'intérêt

La *mesure d'intérêt* est une fonction d'évaluation cruciale dans le processus d'amorçage afin de guider l'exploration dans l'immense espace de recherche de l'ensemble des fonctions de discrétisation de l'expérience. Nous avons proposé une évaluation fondée sur la notion de compromis entre une discrétisation spécifique mais instable et une discrétisation générale mais peu expressive (se référer à la section 3.2.4). Pour cela, chaque couple d'agents produisant une représentation (ex. ensemble de classes d'évènements) est capable de l'évaluer et de partager cette information dans un *espace de marquage* accessible aux autres agents afin de guider l'exploration vers les zones intéressantes de

ΔB	bruit	marge d'erreur de la mesure autour de la valeur "réelle"
A	amplitude	amplitude de la variation
ΔA	variance de l'amplitude	différence d'amplitude possible des variations successives par rapport à l'amplitude de base (A)
DV	durée de la variation	ex. instantané (0) pour la variable 1 sur la Figure 4.1
ΔDV	variance de la durée de variation	la durée de variation est plus ou moins régulière par rapport à DV
DD	durée du délai	délai entre évènements corrélés (variations de variables différentes)
ΔDD	variance de la durée du délai	les évènements sont séparés par un délai plus ou moins régulier par rapport à DD
DC	durée du cycle	délai entre deux évènements d'une même variable
ΔDC	variance de la durée du cycle	le cycle de régularités se répète à un rythme plus ou moins régulier par rapport à DC

FIGURE 4.2 – Les différents paramètres du simulateur pour caractériser l'activité des variables (voir Figure 4.1).

l'espace de recherche. Nous proposons dans cette partie d'étudier le comportement de la mesure d'intérêt proposée à travers quelques expériences.

4.2.1.1 Adaptation de la mesure à différentes échelles de temps.

L'agent S utilisé dans cette expérience est l'agent AS_{HSI} . La mesure d'intérêt appliquée aux agents AD_{FGV} devrait permettre d'identifier les zones de l'espace de recherche qui rendent possible la discrétisation des variables afin d'obtenir des évènements correspondant aux variations importantes. Par conséquent, la durée de la fenêtre glissante employée par un agent devrait s'adapter à l'ordre de grandeur temporel des variations de la variable qu'il observe. Afin d'analyser ce comportement, nous proposons une expérience dans laquelle nous faisons varier uniquement la durée de variation d'une variable dans le simulateur, toutes choses étant égales par ailleurs. Seules la variance de la durée de variation ainsi que la durée de cycle évoluent proportionnellement à cette durée de variation. La configuration du simulateur est donnée dans le Tableau 4.3. Les différentes durées de variation testées sont : instantannée, 1 seconde, 10 secondes et 1 minute.

Les résultats sont présentés sur la Figure 4.4, qui montre une représentation en 3 dimensions de l'espace de marquage. L'axe x représente le paramétrage de l'agent AS_{HSI} ,

qui influe sur la précision de la comparaison effectuée (nombre de classes des histogrammes). On constate que les zones les plus intéressantes (intérêt maximum, repéré par un rond) de l'espace de marquage se "déplacent" le long de l'axe y correspondant à la durée de la fenêtre glissante, lorsque la durée de la variation simulée augmente (de 0 à 1 minute). Il existe un gradient de la mesure d'intérêt autour des zones les plus intéressantes, pouvant ainsi permettre une stratégie d'exploration stochastique telle que celle proposée, utilisant les voisinages. Ces graphiques montrent les résultats d'une seule simulation pour chaque expérience mais ils illustrent parfaitement le comportement moyen observé. Chaque simulation possède ses propres aléas et les agents privilégient des solutions différentes en fonction de l'ordre des positions explorées et des occurrences d'évènement générées par le simulateur. Il n'y a pas une solution optimale, c'est-à-dire une durée de fenêtre glissante idéale et un nombre de classes idéal, les deux types d'agents pouvant adapter leur paramétrage pour "compenser" celui de l'autre. On observe dans le cas de la variation d'une minute (en bas à droite sur la Figure 4.4) un deuxième "pic" de la mesure d'intérêt aux alentours de la position (3 classes, 150 ms). Cette position correspond à un découpage trop précis qui est néanmoins parvenu à obtenir un haut score pour la mesure d'intérêt, par exemple en identifiant un évènement dans le bruit de la variable. Lorsqu'un marquage non approprié est effectué, il risque d'être atténué d'une part par une mise à jour de l'évaluation de cette position, et d'autre part par l'échec de création d'associations liées à l'évènement correspondant.

Expérience	1	2	3	4
ΔB	2	-	-	-
A	100	-	-	-
ΔA	1	1	1	1
DV	0	1s	10s	1 min
ΔDV	0	100ms	1s	6s
DC	5s	10s	100s	40min
ΔDC	1s	2s	20s	10min

FIGURE 4.3 – Paramétrage du simulateur. Dans cette expérience seule la durée de variation (DV) change et les autres paramètres de durée sont ajustés proportionnellement.

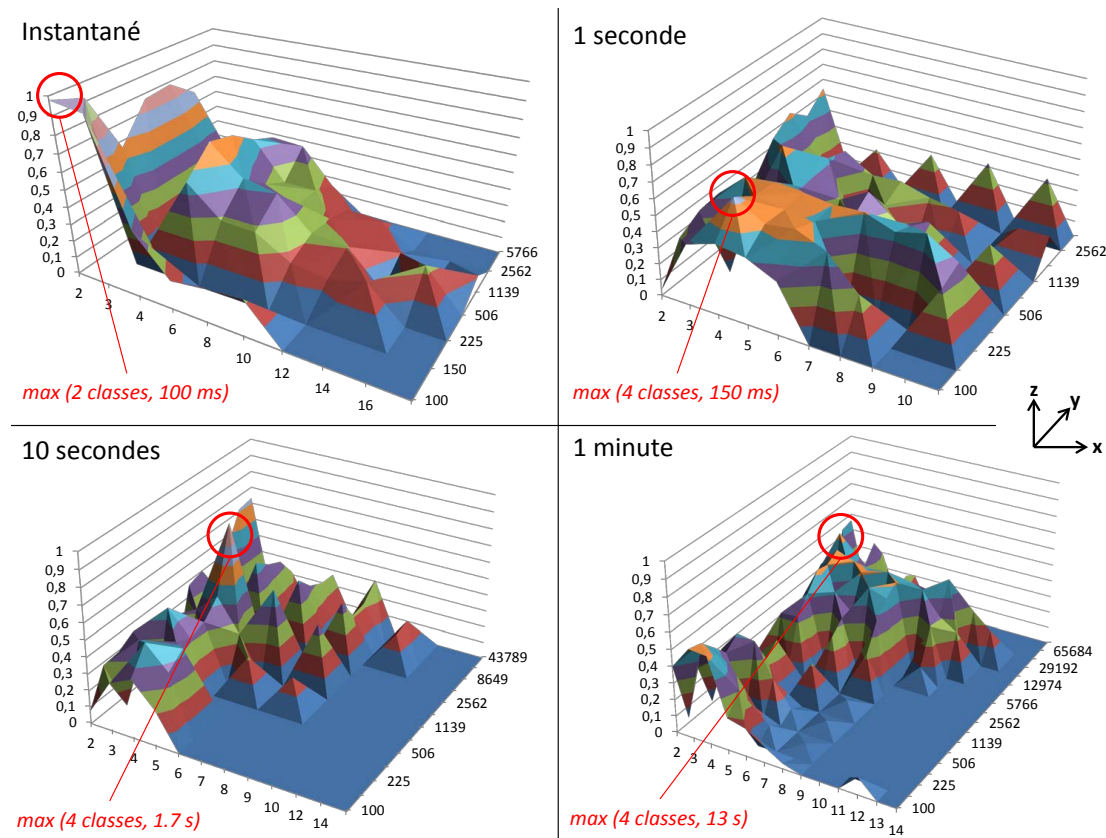


FIGURE 4.4 – Visualisation de l’espace de marquage de la mesure d’intérêt en fonction de la durée de variation
 (x : nombre de classes de l’agent AS_{HSI} ; y : durée de la fenêtre de l’agent AD_{FGV} en ms ; z : mesure d’intérêt)

4.2.1.2 Influence du bruit sur la qualité de la mesure d’évaluation de l’intérêt.

Afin de visualiser l’effet du bruit sur la perception des variations en tant qu’évènements, nous effectuons différentes simulations pour lesquelles seule l’amplitude du bruit (ΔB) varie. L’amplitude de la variation reste stable à 100, donc si ΔB vaut 2, il s’agit d’un bruit de 2 % par rapport à l’amplitude de variation. Les paramètres du simulateur des différents tests effectués sont donnés par le Tableau 4.5.

La Figure 4.6 montre les résultats de cette expérience. La courbe affichée en arrière-plan permet de visualiser le bruit sur les données brutes utilisées par rapport à l’amplitude des variations. On remarque tout d’abord que plus le bruit est important, plus l’espace de solution potentiellement intéressantes est réduit et plus la valeur moyenne

d'évaluation de ces solutions diminue. Lorsque le bruit n'est pas très important, des solutions à partir de fenêtres temporelles longues (ex. plus longues que la durée de variation) sont possibles, alors que ce n'est pas le cas si le bruit est important. Cela s'explique par le fait que l'augmentation de la variance au moment de la variation a moins d'impact sur la distribution de la variance pour une fenêtre longue, plutôt que pour une fenêtre courte, ciblée sur le moment de la variation. Cela est vrai en général, mais encore davantage lorsque il y a peu de différence entre la variance moyenne lors d'une variation et la variance moyenne à l'état stable, comme c'est le cas lorsque le bruit est important. Par conséquent, le bruit restreint le périmètre des solutions possibles et atténue le score de l'évaluation des meilleures solutions. La difficulté accrue de perception liée au bruit est prévisible si l'on considère l'expérience humaine. Malgré cela, nous observons que le système conserve tout de même un gradient de marquage vers une solution acceptable, et qu'il est donc susceptible de s'adapter à un bruit relativement important pour l'extraction d'évènements de type "*variations*".

Expérience	1	2	3	4
ΔB	2	25	50	75
A	100	-	-	-
ΔA	1	-	-	-
DV	1s	-	-	-
ΔDV	50ms	-	-	-
DC	16s	-	-	-
ΔDC	4s	-	-	-

FIGURE 4.5 – Dans cette expérience seule l'amplitude du bruit varie.

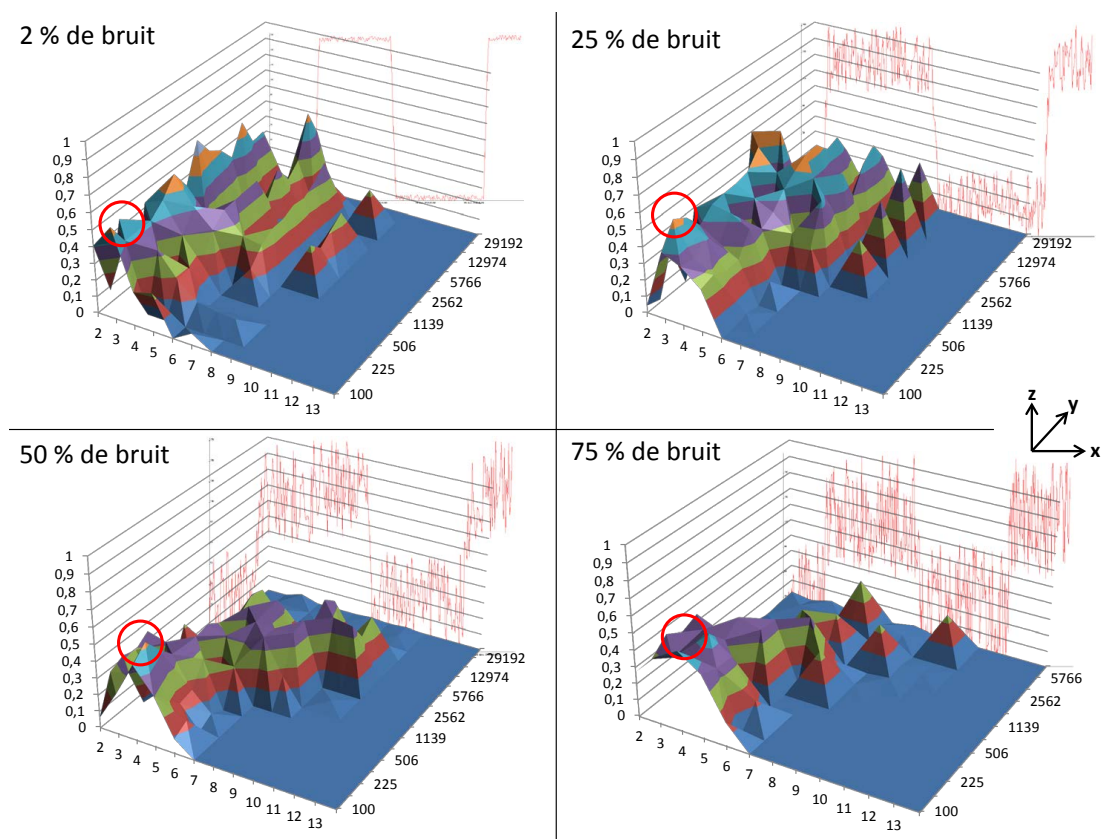


FIGURE 4.6 – Visualisation de l’espace de marquage de la mesure d’intérêt en fonction de l’amplitude du bruit par rapport à l’amplitude de variation (en %)
 (x : nombre de classes de l’agent AS_{HSI} ; y : durée de la fenêtre de l’agent AD_{FGV} en ms; z : intérêt)

4.2.1.3 Discussion sur les expériences liées à la mesure d’intérêt

Les résultats de ces expériences sont bien entendu liés aux implémentations choisies, mais ils permettent d’illustrer le processus d’exploration générique de la discrétisation par l’évolution des paramètres des agents D et S. De même, la mesure d’intérêt proposée peut sûrement être améliorée ou remplacée par d’autres méthodes. L’idée générale est que cette étape permette une présélection des solutions de discrétisation avant de mettre à l’épreuve de la prédiction les événements qui en résultent.

Pour compléter cette partie, on peut observer que la mesure d’intérêt se comporte de manière appropriée également avec des variables réelles comme l’illustre par exemple la Figure 4.7 qui montre l’espace de marquage obtenu pour la luminosité dans une pièce

lors d'une expérience en système réel.

Ces expériences permettent donc de confirmer le comportement adéquat de la mesure d'intérêt proposée pour guider le processus d'amorçage à travers une pré-évaluation des structures générées par différentes possibilités de discrétisation.

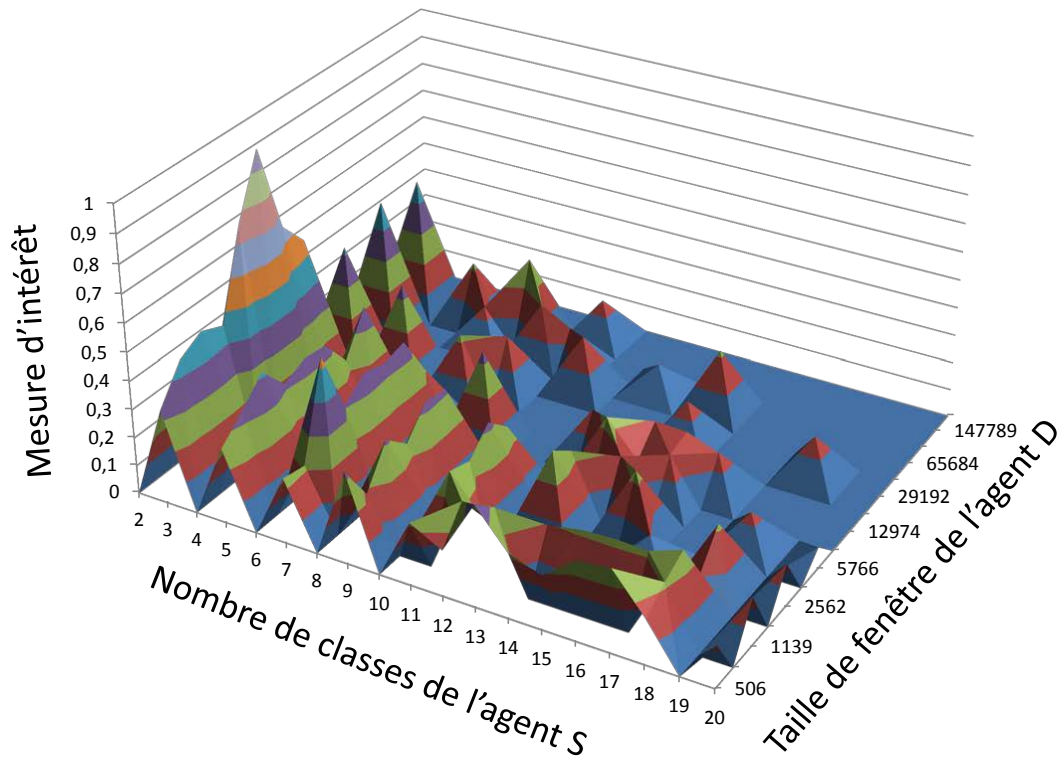


FIGURE 4.7 – Visualisation de l'espace de marquage de la mesure d'intérêt pour la luminosité (expérience en système réel).

4.2.2 Création de motifs sensorimoteurs primaires

Dans cette expérience, nous créons dans le simulateur 8 variables $V1; \dots; V8$, certaines étant liées par des régularités et d'autres non, comme illustré par la Figure 4.8. Le paramétrage du simulateur utilisé est donné par le tableau 4.9. L'activation ou la désactivation de $V1$ entraîne des variations de $V2$ et $V3$ après un certain laps de temps. Il en va de même pour $V6$ et $V7$ avec des délais et des amplitudes différentes, tandis que $V4$, $V5$ et $V8$ varient fréquemment mais à des moments aléatoires, sans lien avec aucune autre variable.

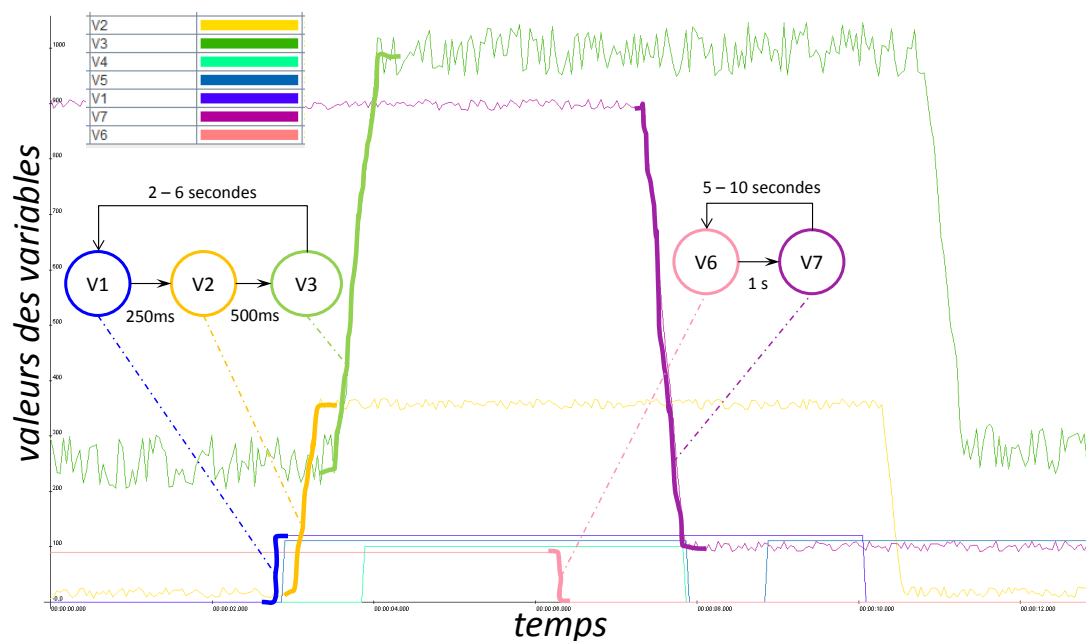


FIGURE 4.8 – Simulation de régularités entre variables continues (voir Tableau 4.9).

Régularité	V1-V2-V3			V4	V5	V6-V7		V8
Variables	V1	V2	V3	V4	V5	V6	V7	V8
ΔB	0	2	10	0	0	0	50	2
A	1	20	50	1	1	1	500	50
ΔA	0	1	2	0	0	0	25	1
DV	0	200ms	700ms	0	0	0	1s	2s
ΔDV	0	10ms	10ms	0	0	0	100ms	100ms
DD	100ms & 300ms			x	x	1s		x
ΔDD	10ms & 30ms			x	x	20ms		x
DC	5s			5s	16s	8s		12s
ΔDC	1s			3s	15s	2s		2s

FIGURE 4.9 – Simulation de régularités entre variables continues (voir Figure 4.8).

4.2.2.1 Analyse globale de l'activité du système

L'objectif de cette expérience est 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* ([Dutot07]). Cela permet de suivre l'évolution de l'état du système au cours de l'activité d'exploration de l'espace de discrétisation et de construction de motifs, comme illustré par la Figure 4.10 qui montre trois étapes différentes de ce processus². Dans cette expérience le système est initialement composé de trois agents D (AD_{FGV}) *explorateurs* par variable et des agents S (AS_{HSI}) qui sont générés à la demande. Les agents A (AA_{BDR}) quant à eux, apparaissent lorsque un événement créé est suffisamment intéressant. Par ailleurs, les agents *exploiteurs* quel que soit leur rôle, sont générés lorsque une zone d'intérêt suffisante a été découverte. Ils sont ici limités à deux par variable.

L'état initial est représenté sur la Figure 4.10-A. Lorsqu'une structure (événement, association) est évaluée avec une mesure d'intérêt suffisamment élevée, les agents exploitent cette "position" de l'espace de recherche : des événements ou des associations sont générés et partagés dans le système (Figure 4.10-B). Enfin, éventuellement les prédictions sont activées pour une association suffisamment intéressante (Figure 4.11). Lorsque la prédiction atteint un certain score, elle est confirmée (motif). 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 système³.

Un motif est donc constitué par un ensemble d'agents (représentés par un rond) et un ensemble d'éléments de représentation (événements et associations, représentés par un carré). Ces éléments contiennent des données qui décrivent précisément la nature de cette régularité. En l'occurrence ces données sont des histogrammes, comme nous l'avons présenté dans la partie précédente. Les associations peuvent être utilisées sous forme de prédiction lorsqu'elles ont un intérêt suffisant. Lorsque le score de cette prédiction est supérieur à un seuil fixé (ex. 0.5) cette prédiction est confirmée, pouvant ainsi être diffusée dans le système (ex. pour un apprentissage de motifs complexes).

Comme le montre la Figure 4.11, on constate que les variables qui sont liées par des régularités dans le simulateur, sont bien reliées par des prédictions. Les autres variables ($V4$, $V5$ et $V8$) sont quant à elles isolées.

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

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

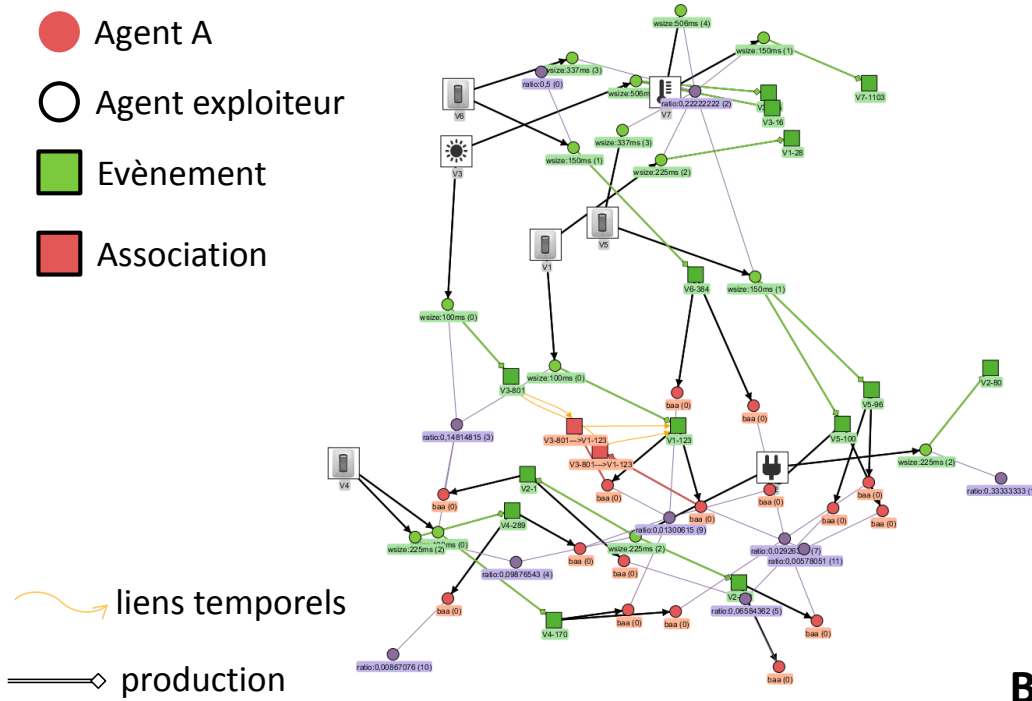
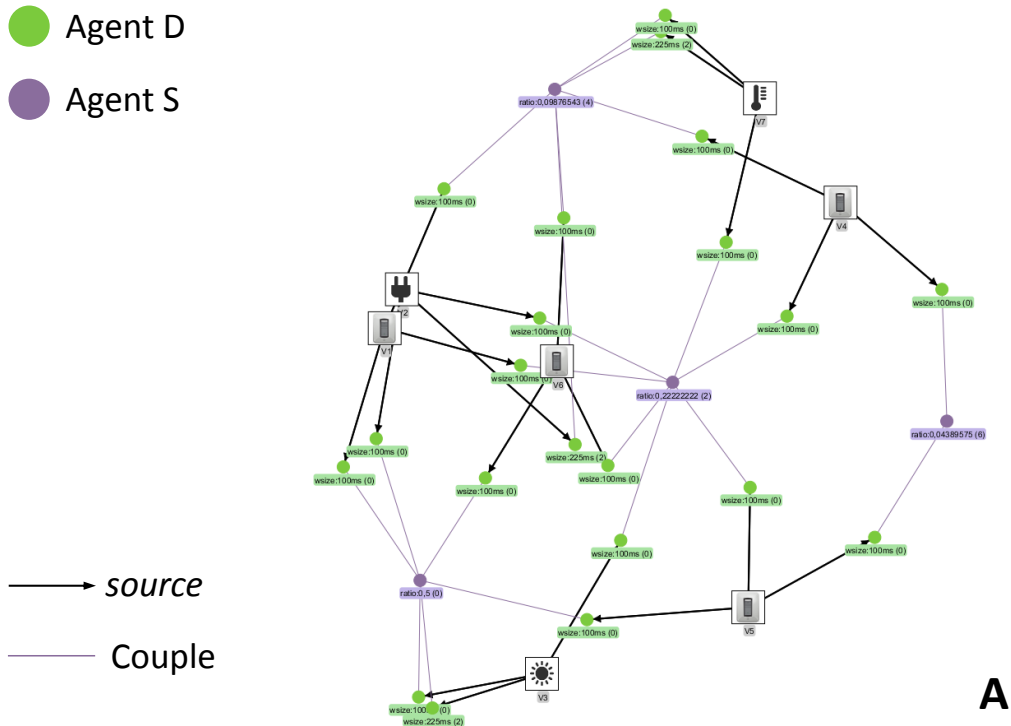


FIGURE 4.10 – A : Etat initial. B : Création d'évènements et d'associations. Un agent explorateur est représenté par un rond et un agent exploiteur par un rond cerclé de noir.

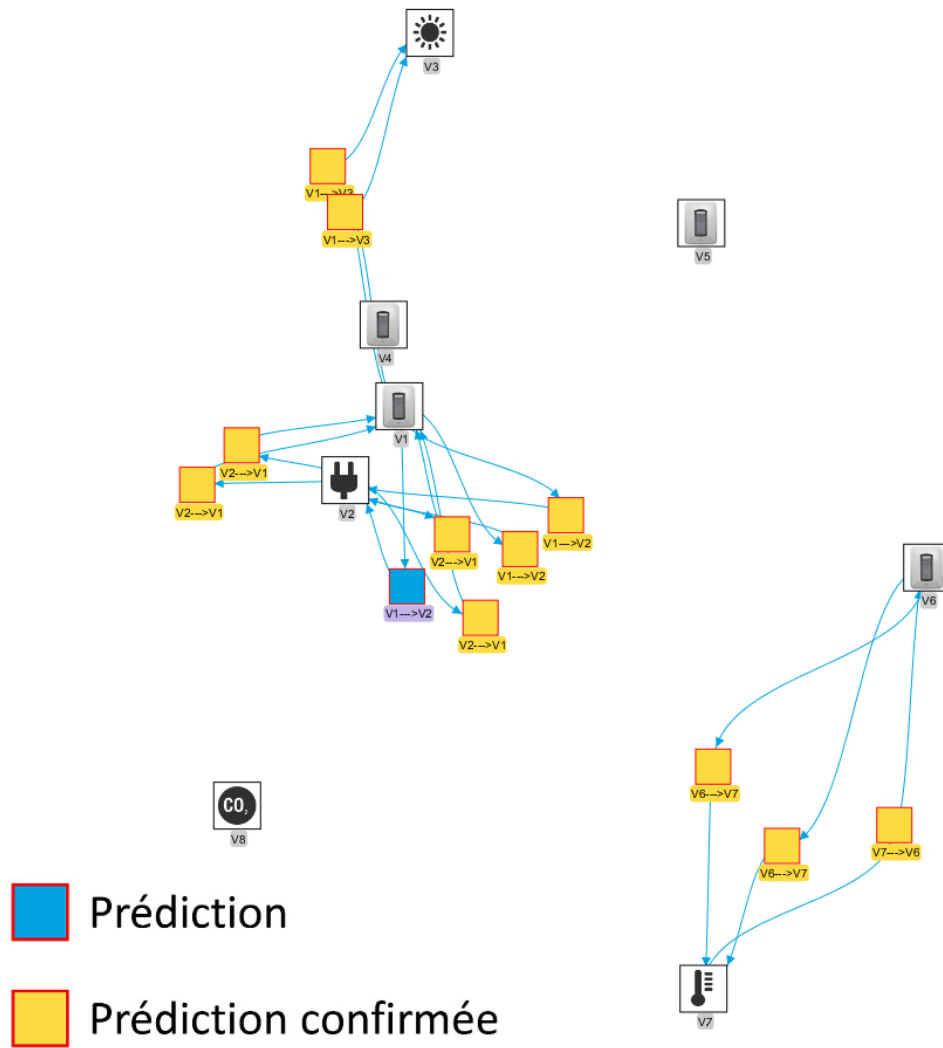


FIGURE 4.11 – Un exemple d'état stable des motifs produits par le système. Visualisation simplifiée des liens temporels entre les variables (événements cachés)

4.2.2.2 Analyse des motifs appris

Les Figures 4.12 et 4.14 associées respectivement aux Tableaux 4.13 et 4.15 montrent les détails de plusieurs motifs primaires obtenus. Sur les figures, de gauche à droite, on visualise l'évènement initial e_1 , l'évènement association, et l'évènement prédit e_2 . Pour les évènements e_1 et e_2 les histogrammes montrent la répartition de la variance sur les fenêtres temporelles sélectionnées par rapport à la répartition en général de toutes les fenêtres (*référence*). Ces évènements ont été détectés comme intéressants car ils sont spécifiques et suffisamment récurrents. Mais, à la fin, c'est seulement leur participation à un motif permettant effectivement de réaliser des prédictions, qui autorise à les considérer comme des évènements pertinents. L'histogramme de l'évènement association, montre quant à lui la répartition des durées observées entre les instances de e_1 et e_2 par rapport à la répartition des durées entre e_2 et d'autres évènements choisis aléatoirement (*référence*).

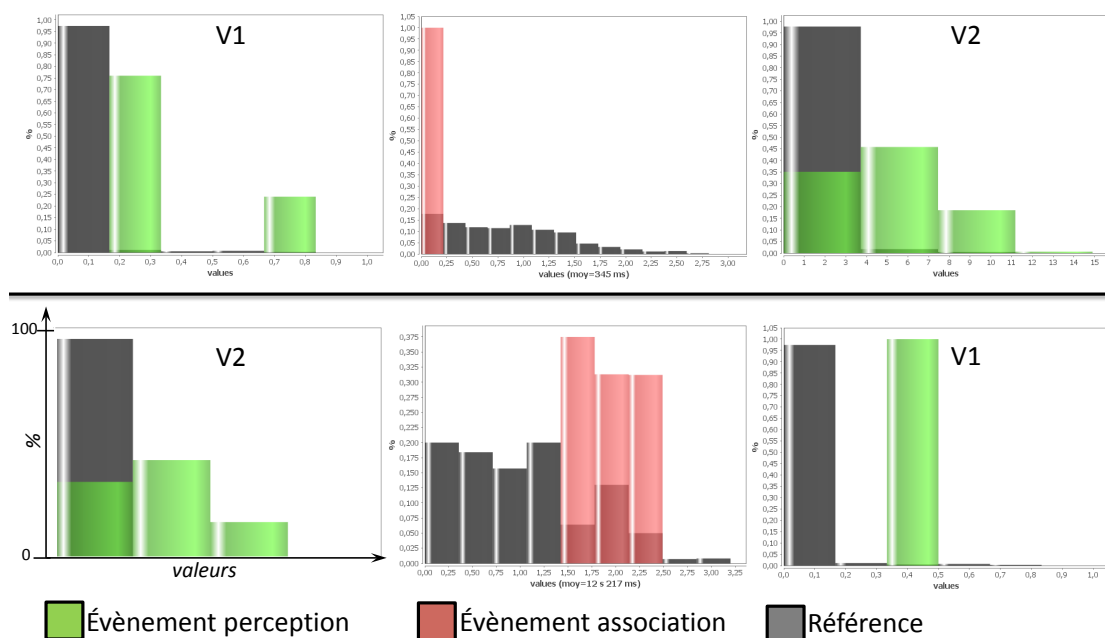


FIGURE 4.12 – Visualisation de 2 motifs primaires entre V1 et V2. Les histogrammes représentent (de la gauche vers la droite) : le modèle du premier évènement, le modèle de l'association (durée entre les 2 évènements) et le modèle du deuxième évènement (voir Tableau 4.13).

Nous constatons donc que le système est capable de retrouver les régularités qui existent entre les variables, comme on peut par exemple le vérifier en comparant les régularités définies dans le simulateur pour V6 et V7 (Tableau 4.1) et les informations

Régularité	V1-V2		V2-V1	
Variables	V1	V2	V2	V1
durée moyenne évènement ($\Delta DV \pm \Delta DV$)	294ms	347ms	345ms	547ms
période évènement ($DC \pm \Delta DC$)	6.5s	6.3s	6.2s	6.8s
délag prédiction ($DD \pm \Delta DD$)	228ms \pm 45ms		6.1s \pm 481ms	
précision prédiction	0.99		0.92	
confiance prédiction	1		0.89	
score prédiction	0.99		0.77	

FIGURE 4.13 – Caractéristiques des 2 motifs primaires entre V1 et V2 (voir Figure 4.12).

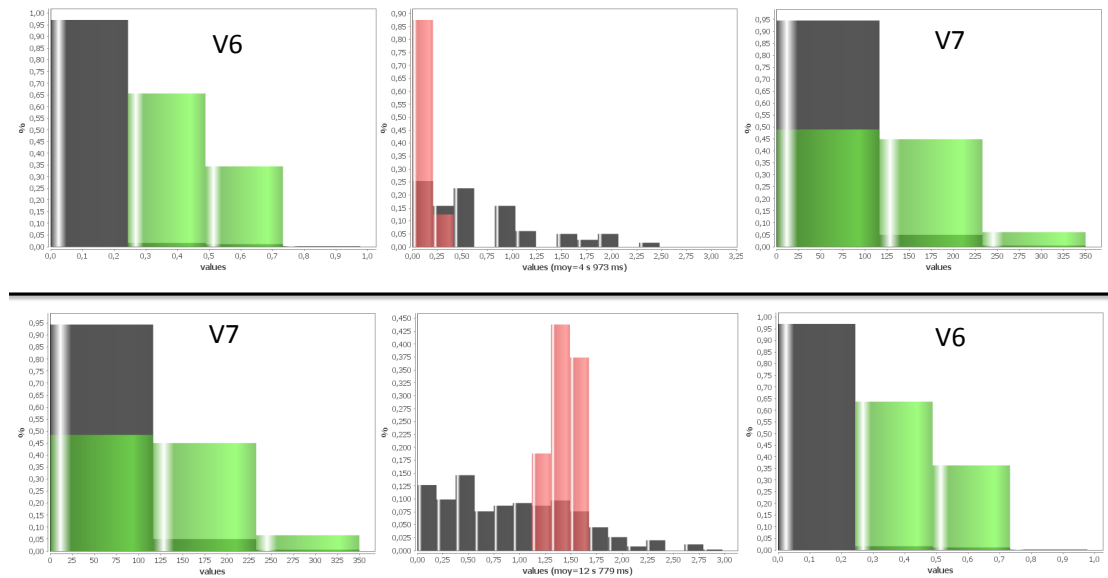


FIGURE 4.14 – Visualisation de 2 motifs primaires entre V6 et V7. Les histogrammes représentent (de la gauche vers la droite) : le modèle du premier évènement, le modèle de l'association (durée entre les 2 évènements) et le modèle du deuxième évènement (voir Tableau 4.15).

obtenues pour le motif V6-V7 (Tableau 4.15). Le score des prédictions est de 0.99 (sur un maximum de 1) pour les motifs prédisant les évènements sur V7 et V2 à partir respectivement des évènements sur V6 et V1. Ce score maximal s'explique d'une part par une confiance de 1, c'est-à-dire qu'à chaque fois que le système a perçu e_1 il a ensuite perçu e_2 dans un temps correspondant au modèle défini par l'association. D'autre part, la prédiction est très précise puisque le système prédit –par exemple pour l'évènement

Régularité	V6-V7		V7-V6	
Variables	V1	V2	V2	V1
durée moyenne évènement ($\Delta DV \pm \Delta DV$)	633ms	1.7s	1.6s	627ms
période évènement ($DC \pm \Delta DC$)	9.7s	10.2s	10s	11.1s
délat prédition ($DD \pm \Delta DD$)	1s \pm 98ms		6.1s \pm 481ms	
précision prédition	0.99		0.89	
confiance prédition	1		0.84	
score prédition	0.99		0.67	

FIGURE 4.15 – Caractéristiques des 2 motifs primaires entre V6 et V7 (voir Figure 4.14).

sur V7– un évènement ayant lieu toutes les 6.3s à 45ms près en moyenne. D'autre part, rares sont les régularités non existantes dans la simulation (*faux positifs*⁴) qui apparaissent en tant que motifs. Cela correspond généralement à des coïncidences qui sont apprises mais qui ne peuvent être renforcées et disparaissent rapidement car elles ne permettent pas de prédire efficacement. On remarque par contre que le système a appris à prédire le début du cycle suivant à partir de l'évènement de fin de la régularité. C'est par exemple le cas de la prédiction de V1 à partir de V2 et de V6 à partir de V7. Ces prédictions sont moins fiables car elles correspondent à une régularité plus aléatoire (la durée du cycle étant comprise entre 4 et 6 secondes pour V1-V2 et entre 6 et 10 secondes pour V6-V7). Cela se traduit par une précision et une confiance amoindries et un score plus bas (scores de 0.77 et 0.67) mais néanmoins suffisant pour que le motif soit confirmé (si on définit un seuil de 0.5).

4.2.3 Temps d'apprentissage : influence du nombre d'agents et du partage des espaces de marquage

Le temps d'apprentissage peut être évalué par rapport au nombre d'observations nécessaires à l'obtention d'une prédiction confirmée. Il est important de mentionner que certains paramètres internes du système (ex. seuils d'intérêts, critères de stabilité, etc.) peuvent également avoir une influence importante, notamment sur le compromis entre la rapidité d'apprentissage et la permissivité de l'apprentissage (limiter le risque de faux positifs, ou de *fausses pistes*).

Dans cette partie nous analyserons l'influence du nombre d'agents, et du partage des données des espaces de marquage sur le temps d'apprentissage. Comme la diffi-

4. Remarque : il ne peut réellement y avoir de "faux positifs" au sens classique dans ce type d'apprentissage, dans le sens où l'agent n'apprend pas un concept cible mais des régularités dans son expérience. Si une régularité est issue d'une série de coïncidences, elle est "vraie" jusqu'à ce que l'expérience la contredise. Nous appelons donc "faux positifs" des motifs temporaires qui peuvent être générés par le système mais qui ne semblent pas correspondre à une régularité spécifiée dans le simulateur

culté intrinsèque d'apprentissage d'une régularité (ex. en fonction de l'importance des aléas : bruit sur les variables et les délais) peut influencer le temps d'apprentissage, ces expériences portent sur l'obtention d'une même régularité ciblée V1-V2 avec les paramètres du simulateur identiques.

4.2.3.1 Influence du nombre d'agents sur le temps d'apprentissage

Lorsque le nombre d'agents par variable augmente, le système est en principe capable d'explorer davantage de possibilités en parallèle. Néanmoins cela demande un supplément de calcul et de mémoire, surtout dans un système vaste qui comprend beaucoup de variables. Ainsi il peut être intéressant d'évaluer le gain –en termes de temps d'apprentissage– de l'augmentation du nombre d'agents afin de trouver le meilleur compromis entre rapidité d'apprentissage et consommation excessive en ressources de calcul (mémoire, CPU). Dans cette expérience, seul le nombre d'agents par variable varie, toutes choses étant égales par ailleurs. La Figure 4.16 montre les résultats de cette expérience sur les 3 boîtes à moustaches de gauche. La distribution des temps d'apprentissage est créée à partir de 10 exécutions (échecs exclus). On considère une exécution comme un échec s'il apparaît à l'appréciation de l'observateur que tous les agents susceptibles de créer le motif cible sont "piégés" dans un minimum local ou dans un comportement en boucle. L'augmentation du nombre d'agents de <2 explorateurs, 1 exploiteur> à <4 explorateurs, 2 exploiters> permet de diminuer considérablement le temps moyen d'apprentissage mais aussi la variabilité des temps d'apprentissage. D'autre part, cela permet de diminuer le nombre d'échecs. En effet, avec un seul agent exploiteur il y a un risque que celui-ci ne parvienne pas à une position adaptée à cause d'un comportement inapproprié, et donc, qu'aucun évènement pertinent ne soit trouvé pour cette variable. Par exemple, ce cas se produit si le seul agent capable de créer des évènements à partir de la variable V1 alterne entre deux positions non pertinentes. L'augmentation du nombre d'agents de <4 explorateurs, 2 exploiters> à <8 explorateurs, 4 exploiters> permet ainsi de limiter encore davantage le nombre d'échecs (aucun échec pour les 10 essais), mais ne semble cependant pas permettre de diminuer drastiquement le temps d'apprentissage. D'ailleurs, si le temps médiant semble diminuer sensiblement, la variabilité semble par contre augmenter. Nous analysons cela comme l'expression du comportement réactif simple et homogène des agents. En effet, comme nous l'avons expliqué dans la partie 3.3.2.1 les agents sont attirés par un certain nombre de positions qui ont été bien évaluées et ils cherchent une position libre aux alentours en s'écartant progressivement. Le problème est que si tous les agents ont un comportement similaire, ils sont susceptibles de converger dans les mêmes zones, et de ne pas mettre à profit l'avantage du nombre, risquant au contraire de se "gêner". C'est pourquoi nous pensons qu'il faut améliorer cet aspect en complexifiant un peu le comportement des agents. Pour cela deux pistes sont possibles. Premièrement en ajoutant un comportement de *répulsion* entre les agents : cela peut être implémenté par exemple en diminuant la probabilité qu'un agent explore à proximité d'un autre agent,

la dimension de la zone "interdite" étant dépendante de l'intérêt de la zone en question. Deuxièmement en forçant une hétérogénéité au sein de la population d'agents afin que leurs comportements diffèrent sensiblement. Il a été en effet montré que l'introduction d'une variabilité des paramètres comportementaux des agents permettait d'améliorer la stabilité et l'efficacité des phénomènes d'auto-organisation au sein d'un SMA, malgré un comportement qualitativement similaire de chaque agent, par exemple dans [Campbell11].

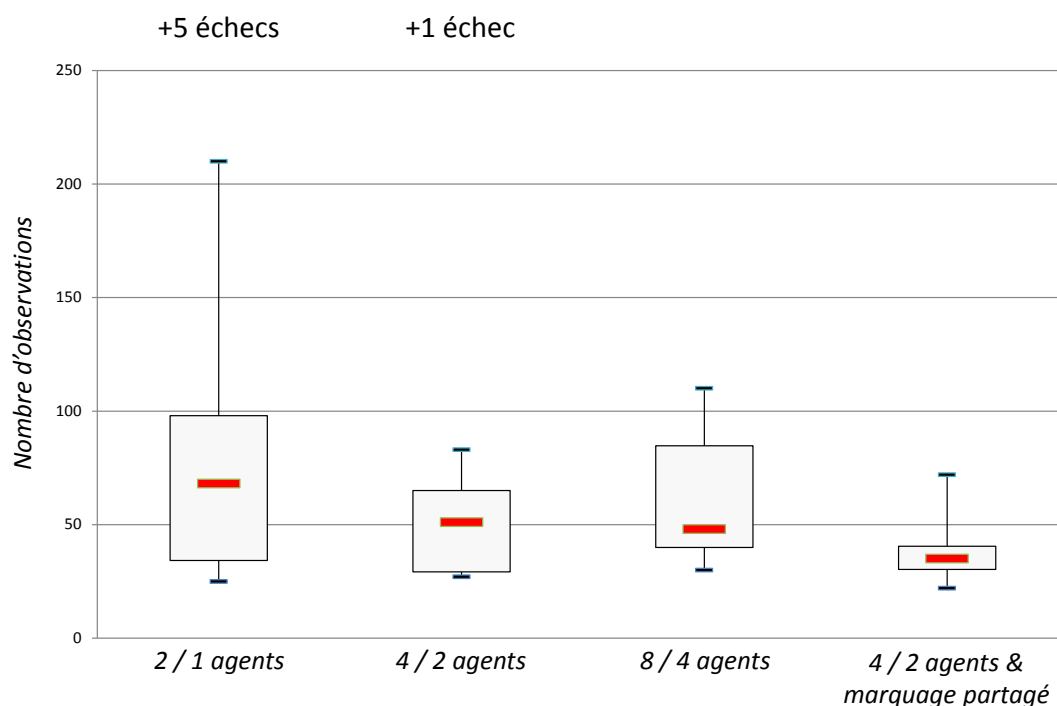


FIGURE 4.16 – Distribution du temps d'apprentissage (en nombre d'observations) par rapport au nombre d'agents par variable (explorateur/exploiteur) sur 10 exécutions.

4.2.3.2 Remarques sur le temps d'apprentissage

Comme nous l'avons dit en introduction, le temps d'apprentissage moyen dépend de divers critères et paramètres du système que nous ne détaillerons pas dans ce mémoire. Ainsi les valeurs absolues des temps d'apprentissage présentées dans les expériences suivantes sont à relativiser. Le paramétrage du système utilisé pour ces expériences est plutôt "strict", c'est-à-dire que le système a tendance à attendre une confirmation conséquente pour exploiter une découverte. A titre indicatif, la médiane pour

<4/2 agents> qui est au alentours de 50 peut être ramenée entre 20 et 30 avec un paramétrage différent. Cependant, avec un paramétrage plus souple, davantage de *pistes* sont explorées et même si elles ne donnent pas de faux positifs (étant refoulées par l'épreuve de la prédiction), elles suscitent une importante activité du système. De plus amples expériences seront nécessaires afin de trouver les paramètres adéquats pour minimiser le temps d'apprentissage sans nuire à la stabilité du système. Une perspective intéressante pour la suite, serait d'automatiser le réglage du système afin qu'il soit plus ou moins rapide/permisif en fonction de son environnement et des ressources dont il dispose par exemple.

4.2.3.3 Influence du partage d'un espace de marquage sur le temps d'apprentissage

Comme introduit dans la partie 3.2.5.2, le nombre d'observations nécessaires à l'apprentissage d'un phénomène peut être réduit si on partage les espaces de marquage entre différents systèmes : de cette façon chaque système devrait converger plus rapidement vers des choix de discrétisation adaptés. Cette hypothèse est possible si l'on admet que les différents bâtiments équipés du système d'AmI (ex. HEMIS) possèdent des similitudes dans les régularités potentiellement présentes dans l'environnement et dans le matériel utilisé, ce qui est vraisemblablement le cas, par exemple pour le logement des particuliers. Dans cette expérience, nous examinons l'influence qu'a le partage d'un espace de marquage entre deux systèmes identiques : même simulateur, même paramétrage du système et même nombre d'agents (<4/2 agents>). Un système S2 débute son exécution avec un espace de marquage provenant d'un autre système S1 plus avancé dans le développement, car ayant déjà extrait des motifs. La Figure 4.16 montre sur la droite de l'image la distribution du temps d'apprentissage pour le système S2 sur 10 exécutions. Comme on peut s'y attendre le temps d'apprentissage ainsi que sa variabilité sont considérablement réduits. En effet, même si le système S2 doit apprendre par lui-même ses propres motifs (potentiellement différents de ceux de S1), les agents vont pouvoir discrétiser plus rapidement les variables de manière adaptée. Ainsi, une partie de l'apprentissage sensorimoteur concernant l'ajustement de la discrétisation est en quelque sorte parallélisée entre différents systèmes. Dans cette expérience nous utilisons le nom des variables afin de réaliser ce partage d'espaces de marquage. Dans le cas réel des systèmes ambiants, c'est vraisemblablement les métadonnées du matériel qui permettront de faire correspondre les variables entre différents systèmes (ex. type de capteur).

4.3 Expérimentations en environnement réel

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 de nos bureaux (zones nommées : *R&D*, *prod* et *cuisine*). Pour cette expérience, une configuration préalable des plages de paramètres des agents en fonction des variables nous permet de borner les espaces de recherche et d'accélérer le processus d'apprentissage (il est en effet inutile de permettre au système de chercher des variations de température de l'ordre d'une seconde, ou des activations d'interrupteur de 30 min). Nous présentons ici les premiers résultats de ces expériences en environnement réel.

4.3.1 Commentaires sur l'exploitation des données réelles et sur les performances

Les temps d'apprentissage pouvant être très longs pour des régularités de l'environnement réel, nous utilisons le mode *temps accéléré* à partir de données brutes préalablement enregistrées. Malgré cela, en l'état actuel, le prototype implémenté nécessite plusieurs minutes d'activité (temps réel) pour calculer une heure d'activité (temps accéléré). Ce temps de calcul tend à augmenter au fur et à mesure de l'activité du système, car le nombre d'agents augmente lorsque de nouvelles structures apparaissent dans le système (moins d'une minute pour une heure au départ, 5 minutes pour une heure au bout de deux jours simulés). C'est pourquoi nous ne présentons ici que des expériences montrant les résultats de quelques jours d'apprentissage. Cette augmentation peut cependant être prévenue par plusieurs mécanismes consistant à brider l'activité du système :

- On constate qu'il y a de la redondance dans les motifs produits. Par conséquent, un certain nombre d'agents exécutent un traitement similaire. Donc premièrement on peut par exemple limiter l'exploitation pour une même variable et un même type d'agent dans une zone ciblée de l'espace de recherche.
- Deuxièmement, il n'y a pas de limite temporelle définie pour la recherche. On peut envisager qu'à partir d'un certain stade où aucune régularité n'est détectée dans une zone suffisamment explorée, on annule la recherche ou on limite le nombre d'agents alloués à cette zone. Le problème étant qu'on ne peut jamais être complètement certain qu'une partie donnée de l'espace de recherche est inexploitable.

La Figure 4.17 montre l'utilisation des ressources de calcul durant l'exécution du mode accéléré. L'utilisation importante du CPU est normale dans cette expérience étant donné qu'on cherche à exploiter au maximum les capacités de la machine pour "accélérer le temps". Ce qui est intéressant, c'est que l'utilisation de la mémoire est constante, malgré le fait que l'activité du système s'accroît avec le temps. En effet, le système ne garde pas les données brutes en mémoire, et les informations mémorisées (la connaissance acquise) le sont sous des formes économes en mémoire (espaces de marquages, histogrammes). Le principal usage qui est fait de la mémoire est l'utilisation de *buffers* d'évènements récents, correspondant à une mémoire à court terme, ce qui explique la variabilité de l'utilisation mémoire que l'on peut voir sur la Figure 4.17.



FIGURE 4.17 – Utilisation du CPU (en %, en haut) et de la mémoire (en MB, en bas) au cours de l'activité en mode "temps accéléré".

4.3.2 Création de motifs sensorimoteurs primaires

L'objectif de cette expérience est de vérifier que l'activité du système dans un contexte réel permet de faire émerger des motifs de type sensorimoteur, de la même manière que ceux obtenus dans l'expérience sur simulateur. La principale différence est que nous n'avons, en revanche, aucune garantie que de telles régularités, perceptibles par le système, existent. D'autre part, si des régularités sont certainement présentes, elles peuvent être beaucoup moins "marquées" que les régularités spécifiées en simulateur et donc beaucoup plus difficiles à distinguer et à stabiliser pour le système.

La Figure 4.18 présente une vue des motifs obtenus au bout d'une trentaine d'heures d'activité.

On remarque tout d'abord qu'il est plus difficile d'obtenir des motifs confirmés (ex. score prédiction $> 0,5$) que dans le cas de la simulation. De manière générale, les motifs construits obtiennent de moins bons scores et sont moins stables, c'est-à-dire qu'ils disparaissent et apparaissent fréquemment. Cela est principalement dû au fait que les régularités réelles sont plus variables et plus complexes que dans les simulations. Ces régularités nécessiteraient par exemple de caractériser un contexte par l'intermédiaire d'un évènement supplémentaire ou d'une information temporelle qui conditionnerait les cas de succès et les cas d'échec. Dans l'architecture proposée, ce type de régularités correspond à des motifs complexes et non à des motifs primaires. Les motifs sensorimo-

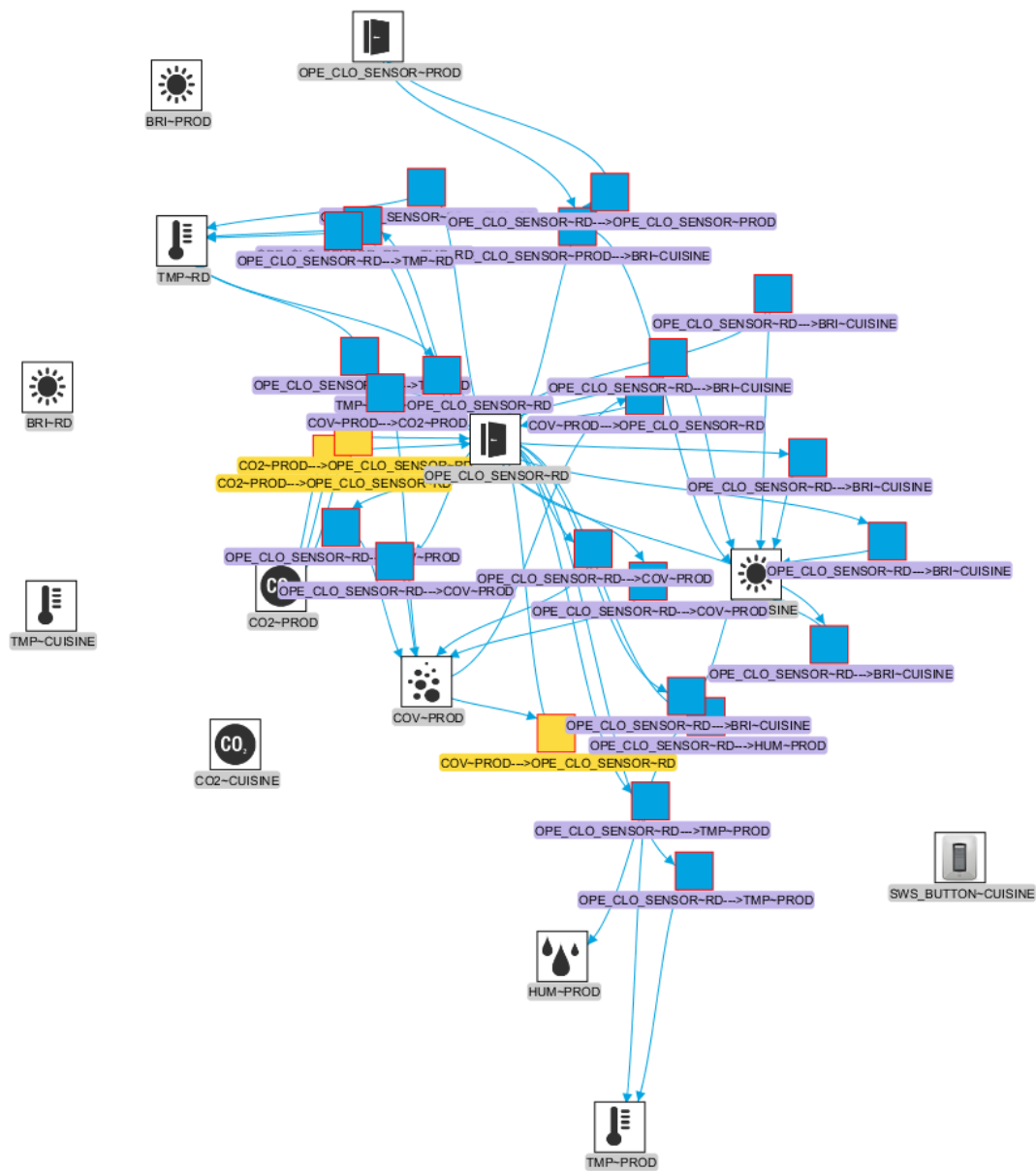


FIGURE 4.18 – Visualisation des motifs obtenus au bout d’une trentaine d’heures.

teurs primaires doivent être suffisamment simples et fiables, ils correspondent en théorie à la perception relativement immédiate de la conséquence d’une action dans l’environnement. Or, au sein du système ambiant utilisé pour ces expériences, la plupart des variables proviennent des capteurs de grandeurs physiques (température, luminosité,

consommation, CO₂, COV⁵, humidité), qui correspondent plutôt à des perceptions. Le système dispose néanmoins de certaines informations correspondant à des actions dans l'environnement (ouverture de porte, interrupteur, *smartplug*⁶). On remarque d'ailleurs que beaucoup de motifs qui apparaissent sur la Figure 4.18 sont liés à ce type de variables, par exemple le capteur d'ouverture de porte nommé *OPE_CLO_SENSOR~RD*.

Il s'avère que les "meilleurs" motifs obtenus concernent des régularités courtes concernant une variable correspondant à une action. La Figure 4.19 montre deux exemples de tels motifs dont les caractéristiques sont donnés dans le tableau 4.20. En particulier le motif permettant d'exprimer le fait que l'activation/désactivation de la *smartplug* entraîne une variation de la luminosité, obtient un score de prédiction de 0.99. Il peut être surprenant de constater que la durée du délai entre les deux événements est de l'ordre d'une vingtaine de secondes. Dans l'expérience humaine, la variation de luminosité perçue est immédiate après l'activation d'un interrupteur. Le système ambiant possède quant à lui un capteur de luminosité auto-alimenté (*energy harvesting*) qui ne renvoie des valeurs qu'à un rythme assez lent de façon à économiser l'énergie de ses batteries. Par conséquent la variation de la luminosité se produit en moyenne 23 secondes après l'activation de la *smart-plug* dans l'expérience du système. Ce motif plutôt contre-intuitif illustre trivialement la différence entre le monde propre du système et la "réalité" de l'humain et ainsi la possible difficulté à définir a priori les régularités que le système doit apprendre, en fonction de notre expérience humaine.

Le motif concernant deux portes présenté en haut de la Figure 4.19 présente quant à lui un lien entre deux variables de type "action". Il possède néanmoins un score moins important que le précédent notamment parce que la confiance est de 0.5, c'est-à-dire que le motif échoue une fois sur deux. Ce motif correspond vraisemblablement à une régularité dans les déplacements des employés dans les bureaux, qui ne sont pas entièrement prévisibles sans informations supplémentaires.

La Figure 4.21 montre également un lien entre des variables correspondant à une action et des variables correspondant à une perception. En l'occurrence, il s'agit de l'influence que peut avoir l'ouverture d'une porte sur la température dans les zones adjacentes. Une infime variation de la température semble perceptible par le système quelques minutes après l'activation du capteur d'ouverture. Si on lui accorde davantage de temps, on peut assez sûrement présager que le système découvre des motifs similaires entre le capteur d'ouverture installé sur les fenêtres et la température par exemple (CO₂ et humidité également). Dans ce cas les variations de température, notamment, risquent d'être alors plus importantes que celles des motifs présentés ici. Cependant l'obtention de tels motifs nécessite certainement des temps d'apprentissage de plusieurs semaines, afin d'avoir un nombre suffisant d'observations, étant donné que l'ouverture/fermeture d'une fenêtre est un événement peu fréquent.

On constate que le système est susceptible d'apprendre également d'autres formes

5. Composé organique volatil. Entre en compte dans l'évaluation de la qualité de l'air.

6. Prise communicante actionnable dotée d'un capteur de consommation

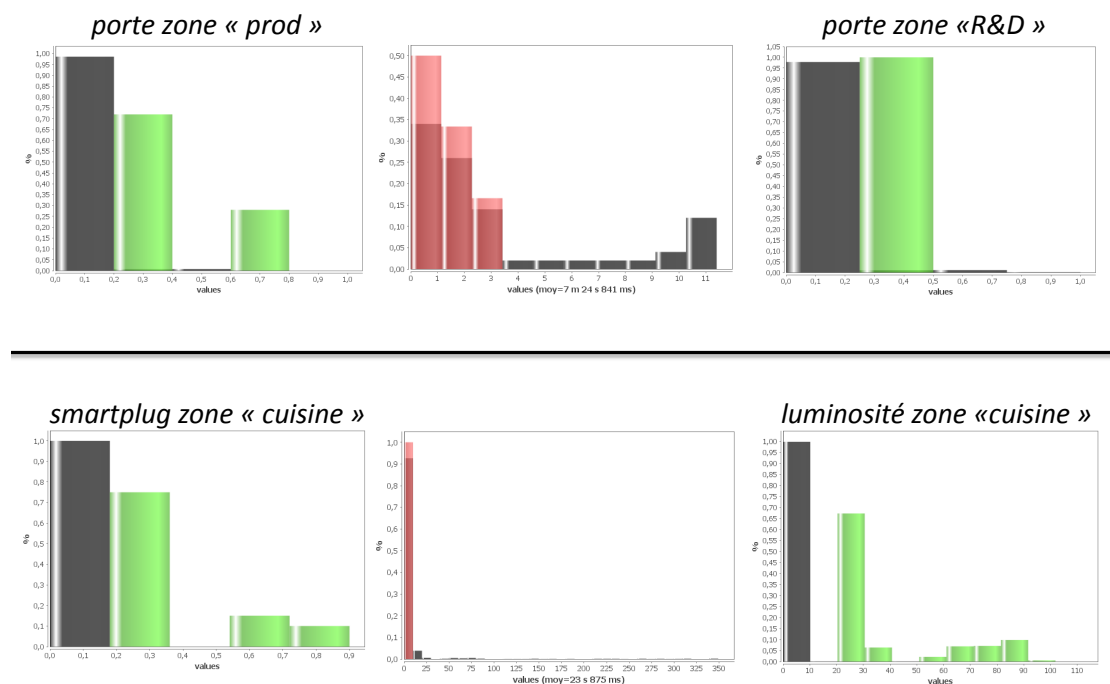


FIGURE 4.19 – Visualisation de 2 motifs primaires. En haut : une régularité entre l'ouverture des portes de deux pièces. En bas, le lien entre l'activation d'une *smartplug* et la luminosité (voir Tableau 4.20).

Régularité	porte R&D-porte prod		<i>smartplug</i> -luminosité	
Variables	porte R&D	porte prod	<i>smartplug</i>	luminosité
durée moyenne évènement ($\Delta DV \pm \Delta DV$)	16.2s	18.6s	522ms	3.4s
période évènement ($DC \pm \Delta DC$)	10min	5.2min	17min	16min
délai prédiction($DD \pm \Delta DD$)	2min49s \pm 1min20s		25s \pm 3s	
précision prédiction	0.75		0.99	
confiance prédiction	0.5		1	
score prédiction	0.28		0.99	

FIGURE 4.20 – Caractéristiques des 2 motifs primaires entre : d'une part "porte R&D" et "porte prod" dans les zones "R&D" et "prod", d'autre part "*smartplug*" et "luminosité" dans la zone "cuisine" (voir Figure 4.19).

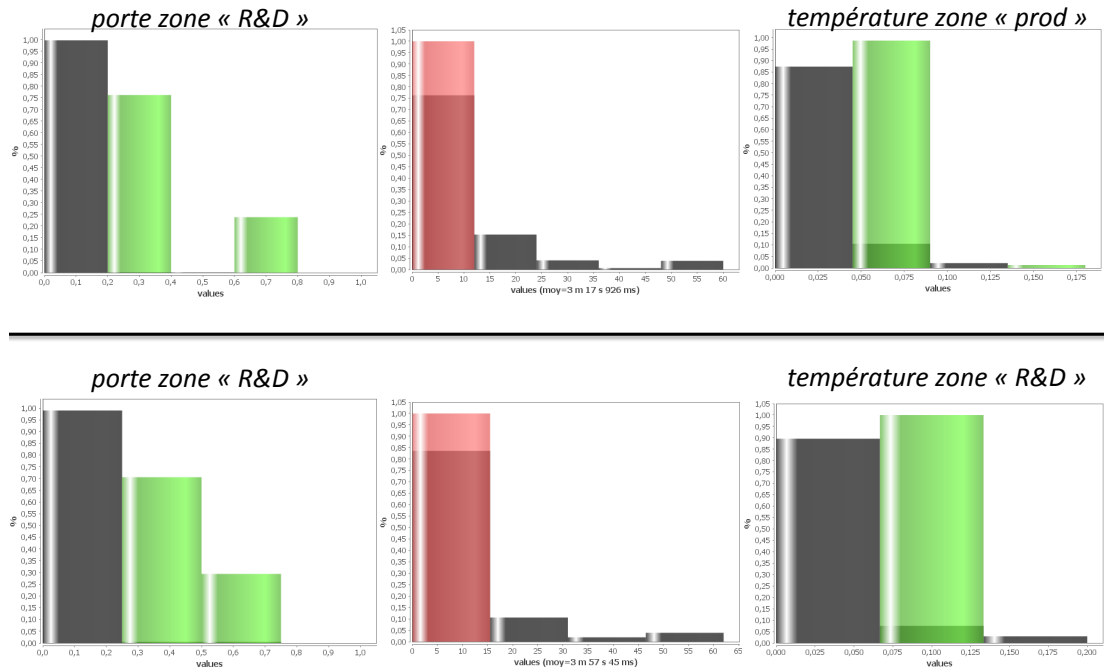


FIGURE 4.21 – Visualisation de 2 motifs primaires. L'ouverture d'une porte donnant sur l'extérieur provoque une légère variation de la température dans deux zones adjacentes.

de régularités du type "perception-perception" ou "perception-action", tels que les exemples présentés sur la Figure 4.22 et dont les détails sont donnés dans le Tableau 4.23. Le motif du haut concerne une variation du taux d'humidité dans une zone, suivie par une variation du taux de COV, après un délai de 7 minutes ($\pm 5min$). Vraisemblablement, ce type de motif est associé à la présence d'une personne à proximité des capteurs. Le motif du bas indique quant à lui une prédiction de l'ouverture de la porte, 3 minutes après une variation de COV constatée dans la zone "prod". L'interprétation de ce motif n'est pas évidente mais est probablement liée à des régularités dans les déplacements des personnes dans les locaux.

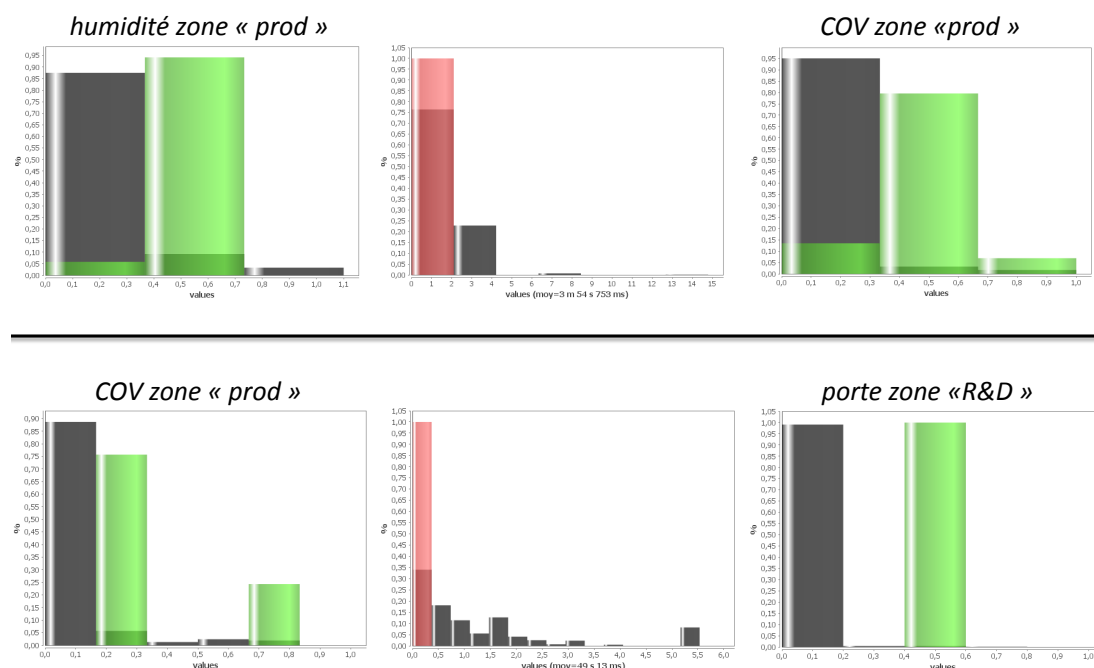


FIGURE 4.22 – Visualisation de 2 motifs primaires. En haut : une régularité entre une variation d'humidité et une variation de COV dans une même zone. En bas, une prédiction d'une ouverture de porte à partir d'une variation de COV (voir Tableau 4.23).

Régularité Variables	humidité-COV		COV-porte R&D	
	humidité	COV	COV	porte R&D
durée moyenne évènement ($\Delta DV \pm \Delta DV$)	12min	3min	4min14s	4.2s
période évènement ($DC \pm \Delta DC$)	1h35min	1h01min	11min	4min32s
délai prédiction ($DD \pm \Delta DD$)	7min32s \pm 5min49s		3min49s \pm 1min17s	
précision prédiction	0.9		0.71	
confiance prédiction	0.58		0.87	
score prédiction	0.47		0.44	

FIGURE 4.23 – Caractéristiques des 2 motifs primaires entre : d'une part "humidité" et "COV" dans la zone "prod", d'autre part "COV" dans la zone "prod" et porte R&D (voir Figure 4.22).

4.3.3 Bilan des expériences en environnement réel

De manière générale le système montre un comportement conforme aux attentes et aux observations effectuées en simulation. Le mécanisme d'apprentissage mis en place permet d'obtenir des motifs sensorimoteurs à partir de la seule observation des données brutes continues. Certaines régularités sont parfaitement adaptées à une modélisation par un *motif primaire*, comme c'est le cas de l'activation d'un actionneur et de la perception de ses conséquences immédiates dans l'environnement (ex. activation d'une lampe). Ce type de régularités est facilement appris par le système et très stable car présentant des scores d'évaluation très importants. D'autres régularités sont moins évidentes et demandent à être précisées par des informations contextuelles pour être rendues plus fiables. Il s'agit alors de mettre en place la création de motifs complexes par les processus de spécialisation et de généralisation. La plus grande difficulté rencontrée au cours de ces expériences réside dans les temps longs d'exécution nécessaires spécifiquement dans ce domaine des systèmes ambiants. Le "mode accéléré" proposé nous a permis de contourner ce problème. Néanmoins cette solution est pour l'instant limitée parce que le prototype n'est pas suffisamment rapide, notamment à cause du problème de l'augmentation du temps de calcul au cours de l'exécution. Par conséquent, les travaux futurs concernant l'application du système aux environnements réels s'orientent autour de trois principaux axes :

- Effectuer des expériences avec un système plus riche en "actions" afin de permettre l'apprentissage d'un plus grand nombre de motifs primaires (ex. capteurs de présence, lampes, chauffage, volets, ventilateurs, etc.)
- Optimisation du système pour permettre d'une part de gérer un plus grand nombre de variables, et d'autre part d'effectuer des simulations plus rapidement en mode accéléré.
- Développer les parties plus avancées du modèle, notamment les étapes de spécification et généralisation.

5

Conclusion et perspectives

▷ Nous synthétisons tout d'abord dans ce chapitre le positionnement choisi, l'analyse des problématiques et le modèle proposé. Ensuite nous commentons les expériences réalisées et les résultats obtenus. Enfin nous présentons les travaux futurs et indiquons notre vision des perspectives offertes par cette approche. ◁

Plan du chapitre

5.1	Conclusion	153
5.1.1	Synthèse	153
5.1.2	Bilan des expérimentations	155
5.1.3	Application des résultats actuels : Autocalibrage d'actions dans HEMIS	156
5.2	Perspectives	157
5.2.1	Développement du modèle et améliorations	157
5.2.2	Problématiques et pistes de recherche	159

5.1 Conclusion

5.1.1 Synthèse

5.1.1.1 Positionnement

Parmi les multiples problématiques du domaine de l'intelligence ambiante, nous avons choisi de nous intéresser particulièrement à celle de la reconnaissance d'activités et/ou de situations dans l'environnement. Cette capacité cognitive d'un système d'AmI est considérée par beaucoup comme une condition nécessaire au développement d'un système ambiant réellement intelligent. Un tel système pour être autonome et proactif sans perturber les habitants/usagers, doit en effet être capable de percevoir et manipuler des concepts de haut niveau d'abstraction associés à l'activité humaine. Nous avons mis en évidence la difficulté de plus en plus grande de concevoir des systèmes experts pour des environnements aussi complexes, qui demandent un effort de modélisation extrêmement important. Il est par conséquent difficilement envisageable de doter un système d'intelligence ambiante d'une représentation de ce genre de concepts à priori. Cela est d'autant plus vrai que les systèmes ambiants possèdent d'une part la propriété d'être indéterminés dans leur composition (qui est, de plus, évolutive), et d'autre part la propriété d'être potentiellement très hétérogènes d'un environnement d'application à l'autre. C'est pourquoi nous envisageons d'aborder cette problématique d'IA dans une optique d'apprentissage. L'analyse des caractéristiques du problème nous amène à généraliser la problématique au delà du domaine de l'intelligence ambiante, à l'apprentissage de concepts à partir de l'expérience d'un agent artificiel incarné, disposant uniquement d'informations brutes non interprétées. Naturellement, ce positionnement se rapproche d'un problème de robotique, où un robot doit apprendre à interagir avec son environnement de manière autonome, et apprendre une représentation à partir de sa propre expérience. Le domaine de la robotique développementale est à ce titre une source d'inspiration très intéressante. Parmi diverses influences ce paradigme s'inscrit notamment dans l'approche de l'IA incarnée et le constructivisme. Dans ce mémoire nous avons proposé une analyse des possibilités d'application de ce type d'approche au domaine de l'intelligence ambiante et des problématiques rencontrées. En particulier, nous abordons le problème de l'amorçage de motifs sensorimoteurs à partir de données brutes inconnues. Ce problème de type auto-référentiel est fondamental dans l'application des approches constructivistes aux systèmes réels en environnement continu. Il s'agit en pratique de faire co-évoluer le processus de discrétisation de l'expérience produisant les éléments de représentation, et le processus d'apprentissage –ou autrement dit de construction d'une représentation– utilisant ces éléments, l'un permettant et influençant l'autre.

5.1.1.2 Modèle

Le modèle proposé consiste en une architecture décentralisée reposant sur une modélisation multi-agents pour répondre à ce problème d'amorçage. Les principales motivations expliquant ce choix sont les suivantes :

- Une conception théorique de la cognition s'appuyant sur un processus d'auto-organisation décentralisé.
- La volonté de modéliser les mécanismes impliqués de manière systémique, à un haut niveau d'abstraction.
- Les aspects pratiques liés au paradigme de programmation des SMA (modularité, généralité, interprétabilité, robustesse, etc.).
- Le contexte applicatif (HEMIS) reposant sur une technologie multi-agent.

Le modèle proposé permet de décrire le processus d'amorçage et de construction de la représentation par l'interaction de plusieurs types d'agents représentant les opérations génériques élémentaires nécessaires à cette tâche. Les rôles de ces différents types d'agents sont décrits de manière abstraite, pour permettre de multiples instanciations possibles des agents. C'est également le cas des structures élémentaires de représentation créées et manipulées par le système. D'autre part, ces éléments de représentation doivent être aisément interprétables, pour offrir des possibilités de communication avec un système externe et potentiellement un utilisateur humain (*intelligibilité*).

Le modèle définit les interactions entre les processus de discrétisation et d'apprentissage par des mécanismes de feedback reposant sur l'évaluation des structures de représentation produites. Ce feedback permet un retour d'information vers l'ensemble des agents ayant participé à l'élaboration de la structure évaluée. Concrètement, ces mécanismes de feedback sont mis en place au sein du SMA à travers des échanges d'informations sur des espaces de marquage qui influencent le comportement d'exploration/exploitation des agents. Le système est conçu de telle sorte, qu'une régularité perçue dans l'environnement donne lieu, dans le système, à une structure de représentation stable issue de l'activité des agents, tant que cette régularité existe. Si la régularité disparaît dans l'environnement, l'organisation ayant donné lieu au motif correspondant dans le système se désagrège. Un motif peut être composé d'une variété d'éléments de représentation issus de diverses variables, permettant ainsi un apprentissage multimodal de concept.

Le problème d'amorçage est géré grâce à la création initiale par les agents de structures primitives, sorte de "propositions" de discrétisation. Nous pouvons comparer cela par exemple aux solutions pouvant être purement aléatoires, proposées à l'état initial d'un algorithme évolutionniste. Cependant, afin de faire converger le système plus rapidement, ces structures primitives sont évaluées par un mécanisme de présélection visant à guider le système vers des solutions de discrétisation adaptées à une représentation prédictive sous forme de motif. Cette présélection est effectuée par l'évaluation d'une mesure d'intérêt, et un mécanisme de feedback transitionnel.

Ce processus d'amorçage est illustré au cours de ce mémoire à travers l'exemple des motifs sensorimoteurs primaires, c'est-à-dire l'apprentissage des éléments de plus bas niveau de la représentation. Néanmoins, ce mécanisme peut être généralisé et étendu aux niveaux supérieurs de la représentation. Le modèle proposé présente ainsi une vision plus large de ce processus à travers les mécanismes de spécialisation et de généralisation.

5.1.2 Bilan des expérimentations

Des expériences menées grâce à un environnement simulé par un ensemble de variables continues, nous ont permis de confirmer le comportement attendu du système d'apprentissage. D'une part, le mécanisme de la mesure d'intérêt et du feedback transitionnel semble permettre de guider le système vers des choix de discrétisation adaptés. D'autre part, l'activité de construction de motifs sensorimoteurs fait émerger les structures qui correspondent précisément aux régularités de l'environnement. Ces structures sont stables tant que la régularité existe, et différentes qualités de prédiction peuvent être exprimées et apprises. Par ailleurs, ces régularités peuvent être obtenues à partir d'une vingtaine d'observations en fonction du paramétrage du système. De plus, nous avons montré que le partage d'informations à travers les espaces de marquage entre différents systèmes pouvait permettre de rendre l'apprentissage plus efficace, ce qui offre des perspectives d'apprentissage à l'échelle d'un ensemble de systèmes ambiants.

Les tests en environnements réels permettent de confirmer les résultats observés en simulation. Malgré l'aspect plus intermittent des régularités, on constate que le système est capable de construire des motifs intéressants. Notamment, les motifs impliquant une action et la perception de ses conséquences dans l'environnement sont appris aussi facilement et précisément que les régularités issues des simulations. Par ailleurs, il s'avère que dans un lieu possédant une activité riche tels que des bureaux, une journée d'apprentissage peut suffire pour apprendre ce genre de motifs. Les régularités se déroulant sur des temps plus longs (ex. évènements ayant lieu une fois par jour) impliquant des temps d'apprentissage de l'ordre de plusieurs semaines, nécessitent des expériences supplémentaires. Premièrement, nous pouvons envisager l'exécution en temps réel du système d'apprentissage sur une longue période de temps. Deuxièmement, on peut aussi procéder à l'exécution du mode accéléré sur des données préalablement enregistrées. Pour cela, il faudra soit optimiser le prototype actuel du système d'apprentissage, comme il sera fait dans une phase d'industrialisation prochaine ; soit utiliser par exemple une grille de calcul ou un supercalculateur afin de pouvoir obtenir les résultats de plusieurs mois d'activité en un temps acceptable.

5.1.3 Application des résultats actuels : Autocalibrage d'actions dans HEMIS

En l'état actuel, le système d'apprentissage offre une piste intéressante pour l'exploitation des résultats dans le cadre industriel du système HEMIS. En effet, une problématique rencontrée concernant l'activité de résolution des scénarios est celle de l'*autocalibrage* des actionneurs. Le problème est que l'on ne connaît pas précisément à priori quel est l'effet d'une action sur un facteur environnemental ciblé. Par exemple, si un objectif indique qu'une température de 22°C est souhaitée dans une zone, les agents actionneurs concernés au sein d'HEMIS (ex. plusieurs chauffages électriques différents) vont agir afin d'atteindre cet état. Le mécanisme d'adaptation mis en place implique que les agents évaluent chacun une "quantité d'action" à exécuter (ex. puissance utilisée), en fonction de l'écart entre la valeur courante du facteur environnemental et la valeur souhaitée. Cette décision initiale se fait d'après un paramétrage expert. Si l'action effectuée est trop importante ou insuffisante, un nouvel écart est alors calculé et le comportement des agents est ajusté. Ainsi, après quelques itérations le système converge plus ou moins rapidement vers la valeur souhaitée. Cependant, ce phénomène d'ajustement peut être perceptible lorsqu'il s'agit par exemple du facteur environnemental "luminosité", avec lequel on peut parfois observer un phénomène d'oscillation de la luminosité avant que celle-ci ne se stabilise. Cette phase d'ajustement pourrait être supprimée ou atténuée pour éviter d'une part qu'elle soit perceptible par l'utilisateur, d'autre part pour une question d'optimisation.

Une solution possible pour autocalibrer le système est d'apprendre par l'expérience les effets des actions sur les différents facteurs environnementaux influencés, c'est-à-dire typiquement une forme d'apprentissage sensorimoteur. Par exemple, on pourrait ainsi caractériser assez précisément l'amplitude de la variation de luminosité induite par l'activation d'un appareil ainsi que l'intervalle de temps nécessaire à la perception d'un retour d'information. Nous pouvons ainsi tirer parti du fait que le système d'apprentissage proposé produit des structures de représentation intelligibles, en exploitant les informations des motifs sensorimoteurs primaires obtenus, directement dans HEMIS. Concrètement, ces informations seraient prises en compte pour la décision initiale de la "quantité d'action" à attribuer à chaque agent en fonction de l'objectif à atteindre. Ainsi, grâce à ces informations du module d'apprentissage acquises par l'expérience, le système pourrait converger plus rapidement vers son objectif et atténuer considérablement la phase d'ajustement.

Cette application proposée ici est une utilisation particulière du système d'apprentissage à partir d'une sous-partie de ses résultats. Cela illustre l'intérêt d'avoir une approche modulaire et itérative ainsi que l'avantage d'une représentation intelligible. Ces travaux d'interface entre HEMIS et le module d'apprentissage sont en cours à l'heure de la rédaction du présent mémoire et les premiers résultats indiquent que cette méthode est efficace.

5.2 Perspectives

Le travail présenté dans ce mémoire nous a permis de poser les bases d'un modèle générique offrant de nombreuses opportunités pour la poursuite du travail de recherche et d'implémentation. Nous en donnons ici les principales idées, sans rentrer dans les détails outre mesure. Tout d'abord nous présentons certaines améliorations et développements sur lesquels nous souhaitons travailler prochainement, puis des problématiques et des perspectives de recherche plus générales associées au modèle proposé.

5.2.1 Développement du modèle et améliorations

5.2.1.1 Enrichissement du comportement des agents au sein du système

Comme nous l'avons notamment analysé dans la partie 4.2.3.1 le comportement des agents au sein du système est en l'état actuel assez basique et peut être amélioré afin d'augmenter notamment la rapidité du processus d'exploration. Nous avons par exemple mentionné un mécanisme de "répulsion" entre les agents qui permettrait de prévenir la redondance en forçant les agents à explorer des zones différentes afin de maximiser l'intérêt de la parallélisation des traitements. Dans la même optique, il est possible d'introduire de légères variations dans les comportements des agents par l'intermédiaire de paramètres comportementaux. Cela permettrait d'assurer une hétérogénéité au sein de la population d'agents comme expliqué également dans la partie 4.2.3.1. De manière générale, nous avons proposé une version minimale des agents du SMA dans lequel nous pensons qu'il existe de multiples opportunités pour créer des comportements d'agents permettant d'optimiser l'activité du système. Ces comportements évolués constituent un sujet d'étude intéressant pour la suite, que la littérature du domaine des SMA nous permettra certainement d'approfondir.

5.2.1.2 Enrichissement des interactions afin de favoriser l'auto-organisation

En lien avec le point précédent, nous pensons que l'ensemble des interactions proposées dans ce modèle constituent une base qui pourrait être étendue afin d'augmenter les capacités d'auto-organisation du système. Concrètement, il s'agirait de spécifier de nouveaux modes d'interaction entre les agents et les structures au sein du SMA. Par exemple, nous avons mentionné dans la partie 3.2.5.2 la possibilité de définir des interactions entre des agents appartenant à des espaces de marquage différents (ex. différents types d'agents, agents travaillant sur des variables différentes). Par exemple, deux variables différentes peuvent correspondre à un matériel similaire. Si ces équipements sont similaires il y a des chances pour que les variables soient impliquées dans des motifs équivalents. Par conséquent on peut utiliser une information sur la similarité entre

différentes variables pour élargir le périmètre d'interaction des agents. Cette information peut provenir de connaissances expertes, ou bien de l'analyse de motifs obtenus à partir de ces variables sur d'autres systèmes. Un autre type d'interaction pourrait être défini en lien avec les informations spatiales à propos des variables, par exemple en exploitant la notion de zone. Ainsi la recherche de régularités entre variables d'une même zone pourrait être privilégiée. De manière générale, le fait d'orienter la recherche vers certaines zones de l'espace de recherche (par exemple en allouant plus d'agents) est un mécanisme à développer notamment afin d'exploiter des connaissances expertes ou des connaissances issues de la communauté pour améliorer l'efficacité du système.

Développement de méthodes de contrôle / stabilisation

A l'inverse, il faudrait aussi pouvoir freiner le système dans son activité. Comme nous l'avons vu dans les parties 3.2.5.2 et 4.3.1, une problématique importante de ce modèle concerne la gestion de l'activité d'exploration. En effet, en l'état actuel le système ne possède pas de mécanisme permettant d'arrêter ou de freiner la recherche de régularités. Or, il est clair que certaines variables par exemple ne permettront en aucun cas de trouver une quelconque régularité d'aucune sorte. Ainsi il serait préférable que le système décide au bout d'un certain temps qu'il vaut mieux brider l'activité de recherche pour cette variable. La question à élucider est de savoir quels sont les critères permettant d'en juger et de définir les moyens de contrôle à mettre en place pour effectuer cette régulation. En effet, pour un vaste système comprenant un grand nombre de variables, la multiplication d'efforts de recherche inutiles par un grand nombre d'agents dédiés, induirait certainement un large impact en termes d'utilisation des ressources de calculs. Là encore, il serait utile de pouvoir gérer l'utilisation de connaissances expertes afin, par exemple, de "condamner" certaines zones de l'espace de recherche.

5.2.1.3 Implémentation d'une plus large variété d'agents

Un des intérêts principaux du modèle consistant en la définition d'opérations génériques, on pourra expérimenter diverses façon d'implémenter ces opérations à travers des agents différents. L'objectif est ainsi d'autoriser la création d'un système disposant d'un maximum d'expressivité. De la sorte, une plus grande variété de motifs peut être obtenue, au delà de ceux présentés dans le présent mémoire. Il sera intéressant de voir comment ces différentes méthodes de discrétisation peuvent se combiner pour apprendre des motifs complexes variés.

5.2.1.4 Développement des mécanismes de spécialisation et de généralisation

La création de nouveaux types d'agents est directement liée aux mécanismes de spécialisation et de généralisation. En effet, certains types d'agents ne doivent pas être disponibles immédiatement dans le système, à l'instar des agents A qui interviennent seulement lorsque des évènements suffisamment intéressants sont générés. Comme

nous l'avons expliqué dans la partie 3.15 l'apprentissage d'un motif assez général peut donner lieu à un apprentissage plus spécifique des caractéristiques de ce motif. Ainsi un apprentissage qui n'aurait pas été possible abruptement dans l'espace de recherche le plus vaste, peut devenir possible dans un espace de recherche restreint, ciblé sur un motif préalablement acquis. Pour illustrer cela, nous avons par exemple mentionné le cas de l'apprentissage du lien entre deux événements de type "variation" que l'on peut ensuite caractériser comme une "augmentation" et une "diminution", puis distinguer des sous-cas comme le rapport entre l'amplitude de l'augmentation de la variable 1 et la durée de la diminution de la variable 2.

La généralisation est le processus complémentaire de montée en abstraction. On peut considérer la création de motifs complexes comme une forme de généralisation, puisqu'un motif à un niveau n , devient un composant d'un autre motif à un niveau $n + 1$. La création de motifs complexes peut être réalisée à partir des agents A proposés (AA_{BDR}), mais on peut assurément créer de multiples façons d'associer les événements. Enfin la notion d'association n'est pas limitée à un usage de description temporelle, permettant ainsi d'envisager un apprentissage plus abstrait sur les motifs (ex. analogies). Pour finir, une piste intéressante à envisager est celle des *items synthétiques* proposée par Drescher, présenté dans la partie 2.4.4.3, qu'il s'agirait d'adapter dans ce modèle.

5.2.2 Problématiques et pistes de recherche

5.2.2.1 Etudier les possibilités d'apprentissage commun intersystèmes

Comme nous l'avons expliqué dans les parties 3.2.5.2 et 4.2.3.3, il est possible de partager des informations entre différents modules d'apprentissage exécutés au sein de systèmes ambiants différents. Cela est possible par exemple en partageant les espaces de marquage à travers ces systèmes en tenant compte de la nature des variables. L'intérêt principal est de mettre à profit les solutions trouvées par les différents systèmes, à disposition de l'ensemble pour paralléliser davantage la recherche. Au delà de cet exemple concernant le partage des espaces de marquage, les possibilités d'apprentissage liées à la mise en commun des informations d'une grande quantité de systèmes offre de nombreuses perspectives que nous aurons à étudier.

5.2.2.2 Feedback utilisateur

Dans le cadre des approches constructivistes de l'apprentissage, un problème fondamental est celui de l'établissement d'une communication entre le système artificiel et l'humain en vue de l'exploitation des connaissances acquises par le système, à des fins utiles pour l'humain. S'il est possible de créer une interface directe avec les structures internes du système lorsque celles-ci sont intelligibles, il est aussi intéressant de considérer le rôle actif de la communication lors de la phase d'apprentissage, comme

introduit dans la partie 3.2.2.3. En effet, il est évident qu'une partie de nos facultés cognitives sont directement liées à notre capacité de communiquer avec d'autres individus pour échanger des informations, et notamment au langage. De la même façon, le dialogue de l'agent avec l'humain au cours de la phase d'apprentissage est certainement un point essentiel des approches constructivistes et développementales pour la création d'agents incarnés intelligents.

5.2.2.3 Apprentissage actif

Même si nous avons présenté les raisons qui nous permettent d'envisager un apprentissage constructiviste "passif" dans le cadre de l'AmI dans la partie 3.2.2.2, il est néanmoins extrêmement intéressant d'aborder l'apprentissage actif et la question de la motivation intrinsèque qui y est directement liée. Dans le domaine de l'AmI, l'apprentissage actif peut être sensible par rapport à l'impact que peuvent avoir des actions inappropriées sur les habitants. Néanmoins, on peut aussi envisager que ces actions interviennent dans le cadre d'interactions dédiées avec l'humain (cf. le point précédent).

A

Annexes



Annexes

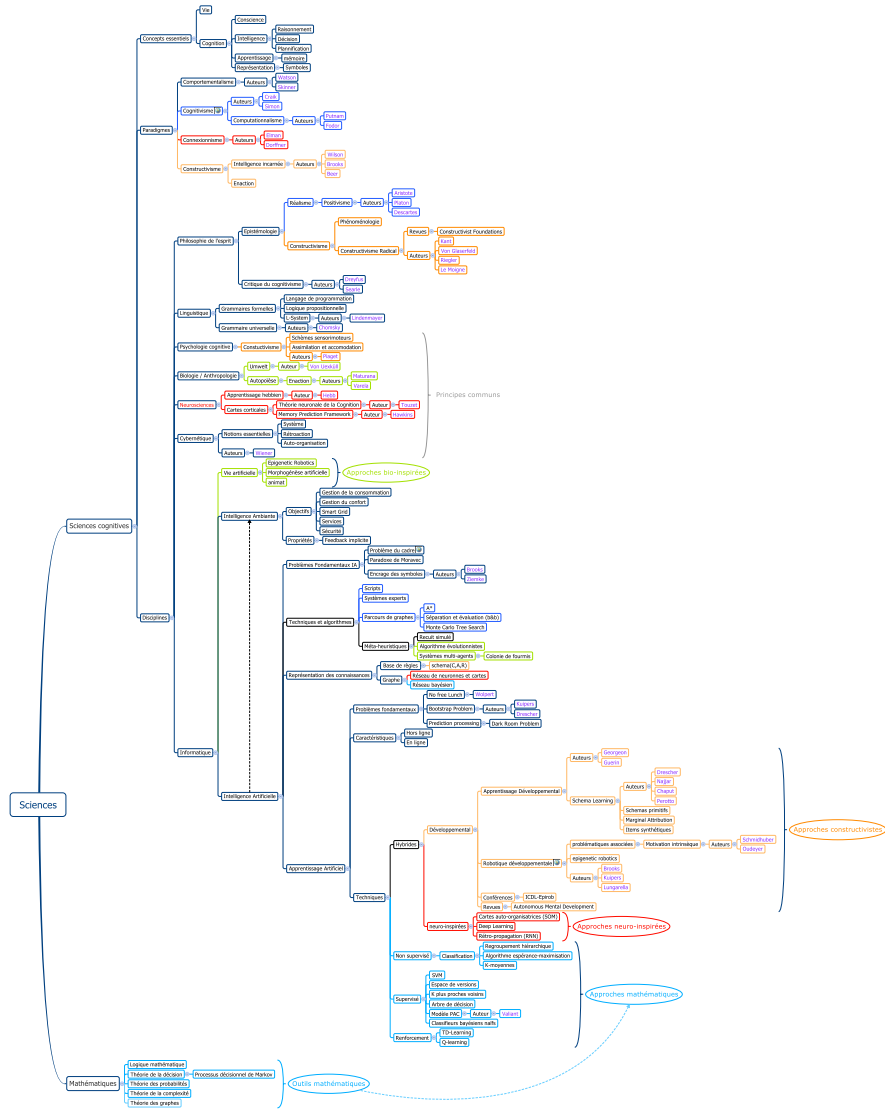


Plan du chapitre

A.1	Arbre des concepts et paradigmes	163
A.2	Publications	164

A.1 Arbre des concepts et paradigmes

Cet arbre (prévu uniquement pour une visualisation de la version numérique de cette thèse) propose une vue d'ensemble de certains concepts et paradigmes abordés dans cette thèse.



A.2 Publications

Publications

- [1] Sébastien Mazac, Frédéric Armetta, and Salima Hassas. Bootstrapping sensori-motor patterns for a constructivist learning system in continuous environments. In *ALIFE 14 : the 14th International Conference on the Synthesis and Simulation of Living Systems*, July 2014.
- [2] Sébastien Mazac, Frédéric Armetta, and Salima Hassas. Approche décentralisée pour un apprentissage constructiviste en environnement continu : application à l'intelligence ambiante. In *Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*, Rennes, France, June 2015.
- [3] Saber Mansour, Nicolas Wiest, Olivier Lefevre, and Sébastien Mazac. Hemis: Hybrid Multi-agent architecture for energy management and home automation, September 2012. SASO 2012 Sixth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SaSo 2012).
- [4] Saber Mansour, Olivier Lefevre, and Sébastien Mazac. HEMIS : un système multi-agent hybride pour la gestion énergétique des bâtiments., October 2014. Journées Francophones sur les Systèmes Multi-Agents (JFSMA).

Bibliographie

- [Abrás08] Shadi Abrás, Stéphane Ploix, Sylvie Pesty & Mireille Jacomino. *A Multi-agent Home Automation System for Power Management*. In Juan Andrade Cetto, Jean-Louis Ferrier, José Miguel Costa dias Pereira & Joaquim Filipe, éditeurs, Informatics in Control Automation and Robotics, volume 15 of *Lecture Notes Electrical Engineering*, pages 59–68. Springer Berlin Heidelberg, 2008. 49
- [Abrás09a] Shadi Abrás. *Système domotique Multi-Agents pour la gestion de l'énergie dans l'habitat*. Doctorat, Grenoble INP, 2009. xiii, 45, 49, 50, 51
- [Abrás09b] Shadi Abrás, S. Pesty, Stéphane Ploix & Mireille Jacomino. *Apport d'une approche Multi-agents pour la résolution d'un problème de gestion de l'énergie dans l'habitat*. In Journée Franco-phones des systèmes Multi-Agents, pages –, Lyon, France, September 2009. 49
- [Al-Naymat09] Ghazi Al-Naymat, Sanjay Chawla & Javid Taheri. *SparseDTW : A Novel Approach to Speed Up Dynamic Time Warping*. In Proceedings of the Eighth Australasian Data Mining Conference - Volume 101, AusDM '09, pages 117–127, Darlinghurst, Australia, Australia, 2009. Australian Computer Society, Inc. 121
- [Alonso01] Eduardo Alonso, Mark d'Inverno, Daniel Kudenko, Michael Luck & Jason Noble. *Learning in multi-agent systems*. The Knowledge Engineering Review, vol. 16, no. 03, pages 277–284, 2001. 45
- [Augusto07] J.C. Augusto. *Ambient intelligence : the confluence of ubiquitous/pervasive computing and artificial intelligence*. Intelligent Computing Everywhere, pages 213–234, 2007. xiii, 4
- [Aztiria09] A. Aztiria, J. Augusto, A. Izaguirre & D. Cook. *Learning accurate temporal relations from user actions in intelligent environments*. In 3rd Symposium of Ubiquitous Computing and

- Ambient Intelligence 2008, pages 274–283. Springer, 2009. xiii, 41, 42, 66
- [Aztiria10] A. Aztiria, A. Izaguirre & J.C. Augusto. *Learning patterns in ambient intelligence environments : a survey*. Artificial Intelligence Review, vol. 34, no. 1, pages 35–51, 2010. ii, 39, 41
- [Aztiria12] A. Aztiria, J.C. Augusto, R. Basagoiti, A. Izaguirre & D.J. Cook. *Discovering frequent user-environment interactions in intelligent environments*. Personal and Ubiquitous Computing, vol. 16, no. 1, pages 91–103, 2012. 41
- [Bach-y Rita03] Paul Bach-y Rita & Stephen W. Kercel. *Sensory substitution and the human-machine interface*. Trends in Cognitive Sciences, vol. 7, no. 12, pages 541 – 546, 2003. 54
- [Barto04] Andrew G. Barto. *Intrinsically motivated learning of hierarchical collections of skills*. pages 112–119, 2004. 69
- [Bengio09] Yoshua Bengio. *Learning Deep Architectures for AI*. Found. Trends Mach. Learn., vol. 2, no. 1, pages 1–127, January 2009. 57
- [Brdiczka05] Oliver Brdiczka, Patrick Reignier & James L. Crowley. *Supervised Learning of an Abstract Context Model for an Intelligent Environment*. In Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence : Innovative Context-aware Services : Usages and Technologies, sOcEUSAI '05, pages 259–264, New York, NY, USA, 2005. ACM. 39
- [Brooks91] Rodney Brooks. *Intelligence Without Representation*. Artificial Intelligence, vol. 47, pages 139–159, 1991. 16, 90
- [Brooks98] R.A. Brooks, C. Breazeal, R. Irie, C.C. Kemp, M. Marjanovic, B. Scassellati & M.M. Williamson. *Alternative essences of intelligence*. In Proceedings of the national conference on artificial intelligence, pages 961–968. John Wiley & sons ltd, 1998. 11, 19
- [Campbell11] A. Campbell, C. Riggs & A.S. Wu. *On the impact of variation on self-organizing systems*. In Self-Adaptive and Self-Organizing Systems (SASO), 2011 Fifth IEEE International Conference on, pages 119–128. IEEE, 2011. 141
- [Chaput03] Harold H Chaput, Benjamin Kuipers & Risto Miikkulainen. *Constructivist learning : A neural implementation of the schema mechanism*. In Proceedings of the Workshop on Self-Organizing Maps (WSOM03), 2003. 21, 70, 75, 77

- [Clark97] Andy Clark & Chris Thornton. *Trading spaces : Computation, representation, and the limits of uninformed learning*. Behavioral and Brain Sciences, vol. 20, no. 01, pages 57–66, 1997. 73, 74
- [Colorni91] Alberto Colorni, Marco Dorigo, Vittorio Maniezzo & others. *Distributed optimization by ant colonies*. In Proceedings of the first European conference on artificial life, volume 142, pages 134–142. Paris, France, 1991. 44
- [Conte03] G. Conte & D. Scaradozzi. *Viewing home automation systems as multiple agents systems*. Multi-agent system for industrial and service robotics applications, RoboCUP2003, Padova, Italy, 2003. 48
- [Cornuéjols11] Antoine Cornuéjols & Laurent Miclet. *Apprentissage artificiel : concepts et algorithmes*. Editions Eyrolles, 2011. 31
- [Davidsson00] P. Davidsson & M. Boman. *Saving energy and providing value added services in intelligent buildings : A MAS approach*. Agent Systems, Mobile Agents, and Applications, pages 79–143, 2000. 48
- [De Loor09] Pierre De Loor, Kristen Manac’h & Jacques Tisseau. *Enaction-Based Artificial Intelligence : Toward Co-evolution with Humans in the Loop*. Minds and Machines, vol. 19, no. 3, pages 319–343, 2009. 19
- [Delahaye14] Jean-Paul Delahaye. *Une seule intelligence ?* Pour la Science, no. 446, 2014. 58
- [Doncieux13] S. Doncieux. *Transfer learning for direct policy search : A reward shaping approach*. In Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on, pages 1–6, August 2013. 39
- [Dorffner97] Georg Dorffner. *Radical connectionism-a neural bottom-up approach to AI*. Neural Networks and a New Artificial Intelligence, pages 93–132, 1997. 15, 21
- [Drescher87] Gary L. Drescher. *A Mechanism for Early Piagetian Learning*. In Proceedings of the Sixth National Conference on Artificial Intelligence - Volume 1, AAAI’87, pages 290–294, Seattle, Washington, 1987. AAAI Press. 65
- [Drescher91] G. L. Drescher. *Made-Up Minds : a constructivist approach to artificial intelligence*. MIT Press, 1991. 20, 21, 65, 66, 75, 108
- [Dreyfus92] Hubert L Dreyfus. *What computers still can’t do : a critique of artificial reason*. MIT press, 1992. 15

- [Dutot07] Antoine Dutot, Frédéric Guinand, Damien Olivier & Yoann Pigné. *GraphStream : A Tool for bridging the gap between Complex Systems and Dynamic Graphs*. In Emergent Properties in Natural and Artificial Complex Systems. Satellite Conference within the 4th European Conference on Complex Systems (ECCS'2007), Dresden, Germany, October 2007. 134
- [Elman90] Jeffrey L. Elman. *Finding Structure in Time*. Cognitive Science, vol. 14, no. 2, pages 179–211, 1990. 52
- [Ferber99] Jacques Ferber. *Multi-agent systems : an introduction to distributed artificial intelligence*, volume 1. Addison-Wesley Reading, 1999. 44
- [Ferber04] Jacques Ferber, Olivier Gutknecht & Fabien Michel. *From Agents to Organizations : An Organizational View of Multi-agent Systems*. In Paolo Giorgini, JörgP. Müller & James Odell, editeurs, Agent-Oriented Software Engineering IV, volume 2935 of *Lecture Notes in Computer Science*, pages 214–230. Springer Berlin Heidelberg, 2004. 98
- [Fondation Jean Piaget15] Fondation Jean Piaget. *Intelligence sensori-motrice*. http://www.fondationjeanpiaget.ch/fjp/site/oeuvre/index_notions_nuage.php?NOTIONID=126, 2015. 61
- [Friedewald05] Michael Friedewald, Olivier Da Costa, Yves Punie, Petteri Alahuhta & Sirkka Heinonen. *Perspectives of ambient intelligence in the home environment*. Telematics and Informatics, pages 221–238, 2005. i, 5
- [Georgeon13] Olivier Georgeon & David Aha. *The radical interactionism conceptual commitment*. Journal of Artificial General Intelligence, vol. 4, no. 2, pages 31–36, 2013. 90
- [Glaserfeld97] Ernst von Glasersfeld. *Homage to Jean Piaget (1896–1982)*. The Irish Journal of Psychology, vol. 18, no. 3, pages 293–306, 1997. 62
- [Guerin11] Frank Guerin. *Learning like a baby : a survey of artificial intelligence approaches*. The Knowledge Engineering Review, vol. 26, no. 02, pages 209–236, 2011. ii, 21, 64
- [Harnad03] Stevan Harnad. *Categorical Perception*. In Encyclopedia of Cognitive Science, volume LXVII. MacMillan : Nature Publishing Group, 2003. 18
- [Hawkins05] J. Hawkins & S. Blakeslee. *On intelligence*. Owl Books, 2005. 53

- [Held63] Richard Held & Alan Hein. *Movement-produced stimulation in the development of visually guided behavior*. Journal of comparative and physiological psychology, vol. 56, no. 5, page 872, 1963. 17, 93
- [Holmes05] Michael P. Holmes & Charles Jr. *Schema Learning : Experience-Based Construction of Predictive Action Models*. In L. K. Saul, Y. Weiss & L. Bottou, editeurs, Advances in Neural Information Processing Systems 17, pages 585–592. MIT Press, 2005. 70
- [Jakkula07] V.R. Jakkula, A.S. Crandall & D.J. Cook. *Knowledge discovery in entity based smart environment resident data using temporal relation based data mining*. In Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on, pages 625–630. IEEE, 2007. 40, 42
- [Kaelbling96] Leslie Pack Kaelbling, Michael L Littman & Andrew W Moore. *Reinforcement learning : A survey*. Journal of artificial intelligence research, pages 237–285, 1996. 32
- [Kohonen90] T. Kohonen. *The self-organizing map*. Proceedings of the IEEE, vol. 78, no. 9, pages 1464–1480, September 1990. 36
- [Koos13] S. Koos, J.-B. Mouret & S. Doncieux. *The Transferability Approach : Crossing the Reality Gap in Evolutionary Robotics*. Evolutionary Computation, IEEE Transactions on, vol. 17, no. 1, pages 122–145, February 2013. 35
- [Kuipers05] Benjamin Kuipers. *Consciousness : Drinking from the Firehose of Experience*. In Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3, AAAI'05, pages 1298–1305, Pittsburgh, Pennsylvania, 2005. AAAI Press. 8, 25
- [Kuipers06] B.J. Kuipers, P. Beeson, J. Modayil & J. Provost. *Bootstrap learning of foundational representations*. Connection Science, vol. 18, no. 2, pages 145–158, 2006. ii, 22, 74
- [Kuipers15] B. Kuipers. *Bootstrap Learning of Foundational Representations*. <http://web.eecs.umich.edu/~kuipers/research/ssh/bootstrap-learning.html>, 2015. 73
- [Leahu08] Lucian Leahu, Phoebe Sengers & Michael Mateas. *Interactionist AI and the Promise of Ubicomp, or, How to Put Your Box in the World Without Putting the World in Your Box*. In Proceedings of the 10th International Conference on Ubiquitous Computing, UbiComp '08, pages 134–143, New York, NY, USA, 2008. ACM. 12, 13, 23

- [Lefort10] Mathieu Lefort, Yann Boniface & Bernard Girau. *Self-organization of neural maps using a modulated BCM rule within a multimodal architecture*. In Brain Inspired Cognitive Systems 2010, page 26, Madrid, Spain, July 2010. 56
- [Lefort12] Mathieu Lefort. *Apprentissage spatial de corrélations multimodales par des mécanismes d'inspiration corticale*. Theses, Université Nancy II, July 2012. 11
- [Lefort13] Mathieu Lefort, Yann Boniface & Bernard Girau. *SOMMA : Cortically Inspired Paradigms for Multimodal Processing*. In International Joint Conference on Neural Networks, Dallas, United States, August 2013. 56
- [Lefort14] Mathieu Lefort, Thomas Kopinski & Alexander Gepperth. *Multimodal space representation driven by self-evaluation of predictability*. In ICDL-EPIROB - The fourth joint IEEE International Conference on Development and Learning and on Epigenetic Robotics, Gênes, Italy, October 2014. 57
- [Linaker01] Fredrik Linaker & Henrik Jacobsson. *Mobile Robot Learning of Delayed Response Tasks Through Event Extraction : A Solution to the Road Sign Problem and Beyond*. In Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'01, pages 777–782, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. xiii, 71, 72
- [Lungarella03] M. Lungarella, G. Metta, R. Pfeifer & G. Sandini. *Developmental robotics : a survey*. Connection Science, vol. 15, no. 4, pages 151–190, 2003. 22, 68
- [Marc Lavielle09] Marc Lavielle. *Algorithme stochastique et recherche optimale...* <http://images.math.cnrs.fr/Algorithme-stochastique-et.html>, 2009. 24
- [Mazac14] Sébastien Mazac, Frédéric Armetta & Salima Hassas. *Bootstrapping sensori-motor patterns for a constructivist learning system in continuous environments*. In M. I. T. Press, editeur, ALIFE 14 : the 14th International Conference on the Synthesis and Simulation of Living Systems, pages 160–167, July 2014. ii
- [McGee05] Kevin McGee. *Enactive Cognitive Science. Part 1 : History and Research Themes*. Constructivist Foundations, vol. 1, no. 1, pages 19–34, 2005. 19
- [McGee06] Kevin McGee. *Enactive Cognitive Science. Part 2 : Methods, Insights, and Potential*. Constructivist Foundations, vol. 1, no. 2, pages 73–82, 2006. 19

- [McIntosh00] A. R. McIntosh. *Towards a network theory of cognition*. Neural Networks, vol. 13, no. 8-9, pages 861–870, 2000. 52
- [Meeden06] Lisa A. Meeden & Douglas S. Blank. *Introduction to developmental robotics*. Connection Science, vol. 18, no. 2, pages 93–96, 2006. 20
- [Miller56] George A Miller. *The magical number seven, plus or minus two : some limits on our capacity for processing information*. Psychological review, vol. 63, no. 2, page 81, 1956. 11
- [Minsky91] M.L. Minsky. *Logical versus analogical or symbolic versus connectionist or neat versus scruffy*. AI magazine, vol. 12, no. 2, page 34, 1991. 25, 26, 37
- [Mitchell82] T.M. Mitchell. *Generalization as search*. Artificial intelligence, vol. 18, no. 2, pages 203–226, 1982. 31
- [Mountcastle78] V. Mountcastle. *An organizing principle for cerebral function : the unit model and the distributed system*. In G. Edelman & V. Mountcastle, editeurs, *The Mindful Brain*. MIT Press, Cambridge, Mass., 1978. 53
- [Mozer98] M.C. Mozer. *The Neural Network House : An Environment hat Adapts to its Inhabitants*. In Proc. AAAI Spring Symp. Intelligent Environments, 1998. 58
- [Mugan07a] J. Mugan & B. Kuipers. *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, 2007. xiii, 75, 93, 102
- [Mugan07b] J. Mugan & B. Kuipers. *Learning to Predict the Effects of Actions : Synergy between Rules and Landmarks*. In Proceedings of Sixth IEEE International Conference on Development and Learning (ICDL-07), pages 253–258, 2007. xiii, 75, 76
- [Mugan12] J. Mugan & B. Kuipers. *Autonomous Learning of High-Level States and Actions in Continuous Environments*. Autonomous Mental Development, IEEE Transactions on, vol. 4, no. 1, pages 70–86, March 2012. 75, 77
- [Murray Shanahan09] Murray Shanahan. *The Frame Problem*. <http://plato.stanford.edu/entries/frame-problem/#EpiFraPro>, 2009. 18
- [Nagel74] Thomas Nagel. *What is it like to be a bat ?* The philosophical review, pages 435–450, 1974. 18
- [Najjar13] Amro Najjar & Patrick Reignier. *Constructivist Ambient Intelligent Agent for Smart Environments*. In PerCom - IEEE In-

- ternational Conference on Pervasive Computing and Communications, San Diego, États-Unis, March 2013. 23, 77, 93, 94
- [Nolfi99] S. Nolfi & J. Tani. *Extracting regularities in space and time through a cascade of prediction networks : The case of a mobile robot navigating in a structured environment*. *Connection Science*, vol. 11, no. 2, pages 125–148, 1999. 71
- [Oudeyer07] Pierre-Yves Oudeyer & Frederic Kaplan. *What is Intrinsic Motivation ? A Typology of Computational Approaches*. *Frontiers in Neurorobotics*, vol. 1, page 6, 2007. 69, 70
- [Panait05] Liviu Panait & Sean Luke. *Cooperative Multi-Agent Learning : The State of the Art*. *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 3, pages 387–434, November 2005. 31, 47
- [Panzoli08] David Panzoli. *Proposition de l'architecture "Cortexionist" pour l'intelligence comportementale de créatures artificielles*. Doctorat, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2008. 52
- [Perotto07] Filippo Studzinski Perotto, Jean-Christophe Buisson & Luis Otávio Alvares. *Constructivist Anticipatory Learning Mechanism (CALM) - dealing with partially deterministic and partially observable environments*. *Cognitive Development*, pages 117–124, 2007. 11, 21, 67
- [Perotto10] F.S. Perotto. *Un Mécanisme Constructiviste d'Apprentissage Automatique d'Anticipations pour des Agents Artificiels Situés*. Doctorat, Institut National Polytechnique de Toulouse-INPT, 2010. 63, 64
- [Piaget54] Jean Piaget. *The Construction of Reality in the Child*. Basic Books, 1954. 21, 60
- [Piaget73] Jean Piaget. *Biologie et connaissance*. Gallimard idées, 1973. 61
- [Pirim13] Patrick Pirim. *Generic Bio-inspired Chip Model-Based on Spatio-temporal Histogram Computation : Application to Car Driving by Gaze-Like Control*. In NathanF. Lepora, Anna Mura, HolgerG. Krapp, PaulF.M.J. Verschure & TonyJ. Prescott, éditeurs, *Biomimetic and Biohybrid Systems*, volume 8064 of *Lecture Notes in Computer Science*, pages 228–239. Springer Berlin Heidelberg, 2013. 110
- [Provost06] Jefferson Provost, Benjamin J. Kuipers & Risto Miikkulainen. *Developing navigation behavior through self-organizing*

- distinctive-state abstraction*. Connection Science, vol. 18, no. 2, pages 159–172, 2006. 70
- [Rennard02] Jean-Philippe Rennard & Daniel Mange. Vie artificielle : où la biologie rencontre l’informatique : illustré avec Java. Vuibert, 2002. 44
- [Riegler01] A. Riegler. *Towards a radical constructivist understanding of science*. Foundations of Science, vol. 6, no. 1, pages 1–30, 2001. 20
- [Searle80] John R. Searle. *Minds, brains, and programs*. Behavioral and Brain Sciences, vol. 3, no. 03, pages 417–424, 1980. 13
- [Sen99] S. Sen & G. Weiss. *Learning in Multiagent Systems*. In Multiagent systems : A modern approach to distributed artificial intelligence, page 259. 1999. 29, 30, 45
- [Shani05] Guy Shani & Ronen I. Brafman. *Resolving Perceptual Aliasing In The Presence Of Noisy Sensors*. In L. K. Saul, Y. Weiss & L. Bottou, éditeurs, Advances in Neural Information Processing Systems 17, pages 1249–1256. MIT Press, 2005. 18
- [Shannon01] Claude Elwood Shannon. *A mathematical theory of communication*. ACM SIGMOBILE Mobile Computing and Communications Review, vol. 5, no. 1, pages 3–55, 2001. 43
- [Sims94] Karl Sims. *Evolving Virtual Creatures*. In Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’94, pages 15–22, New York, NY, USA, 1994. ACM. 36
- [Steels04] Luc Steels. *The Autotelic Principle*. In Fumiya Iida, Rolf Pfeifer, Luc Steels & Yasuo Kuniyoshi, éditeurs, Embodied Artificial Intelligence, volume 3139 of *Lecture Notes in Computer Science*, pages 231–242. Springer Berlin Heidelberg, 2004. 69
- [Stewart01] J. Stewart. *Radical constructivism in biology and cognitive science*. Foundations of science, vol. 6, no. 1, pages 99–124, 2001. 20
- [Strack88] Fritz Strack, Leonard L Martin & Sabine Stepper. *Inhibiting and facilitating conditions of the human smile : a nonobtrusive test of the facial feedback hypothesis*. Journal of personality and social psychology, vol. 54, no. 5, page 768, 1988. 17
- [Touzet13] Claude Touzet. *Why Neurons are Not the Right Level of Abstraction for Implementing Cognition*. In Biologically Inspired Cognitive Architectures 2012, pages 317–318. Springer, 2013. 56

- [Tsymbal04] Alexey Tsymbal. *The Problem of Concept Drift : Definitions and Related Work*. Rapport technique, 2004. 18
- [Varela92] Francisco J. Varela, Evan T. Thompson & Eleanor Rosch. *The Embodied Mind : Cognitive Science and Human Experience*. The MIT Press, November 1992. Published : Paperback. 21
- [Varela93] Francisco J Varela, Evan Thompson, Eleanor Rosch & Véronique Havelange. *L'inscription corporelle de l'esprit : sciences cognitives et expérience humaine*. Éd. du Seuil, 1993. 17
- [Von Glasersfeld84] E. Von Glasersfeld. *An introduction to radical constructivism*. *The invented reality*, pages 17–40, 1984. 20, 62
- [Von Uexküll09] Jakob Von Uexküll. *A stroll through the worlds of animals and men : A picture book of invisible worlds*. *Semiotica*, vol. 89, no. 4, page 319, 2009. 18, 21, 90
- [Weiser91] M. Weiser. *The computer for the 21st century*. *Scientific American*, vol. 265, no. 3, pages 94–104, 1991. i, 3
- [Weng01] Juyang Weng, James McClelland, Alex Pentland, Olaf Sporns, Ida Stockman, Mriganka Sur & Esther Thelen. *Autonomous Mental Development by Robots and Animals*. *Science*, vol. 291, no. 5504, pages 599–600, 2001. 63
- [Wiener65] Norbert Wiener. *Cybernetics or Control and Communication in the Animal and the Machine*, volume 25. MIT press, 1965. 43
- [Wikipédia15a] Wikipédia. *Jean Piaget*. https://fr.wikipedia.org/w/index.php?title=Jean_Piaget&oldid=114400011, April 2015. 60
- [Wikipédia15b] Wikipédia. *Réaction circulaire*. https://fr.wikipedia.org/w/index.php?title=R%C3%A9action_circulaire&oldid=114377882, April 2015. 61
- [Wilson86] StewartW. Wilson. *Knowledge Growth in an Artificial Animal*. In KumpatiS. Narendra, editeur, *Adaptive and Learning Systems*, pages 255–264. Springer US, 1986. 35
- [Zaidenberg09] Sofia Zaidenberg, Patrick Reignier & JamesL. Crowley. *Reinforcement Learning of Context Models for a Ubiquitous Personal Assistant*. In JuanM. Corchado, DantelI. Tapia & José Bravo, editeurs, *3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008*, volume 51 of *Advances in Soft Computing*, pages 254–264. Springer Berlin Heidelberg, 2009. 40

[Ziemke01]

T. Ziemke. *The construction of "reality" in the robot : Constructivist perspectives on situated artificial intelligence and adaptive robotics*. *Foundations of Science*, vol. 6, no. 1, pages 163–233, 2001. ii, 52