

Mémoire d'habilitation à diriger des recherches

Spécialité : Informatique

GAUTHIER PICARD

Université Jean Monnet, Saint-Étienne
École nationale supérieure des Mines de Saint-Étienne

Systemes multi-agents adaptatifs

Ingénierie et utilisation dans le cadre de la résolution de problèmes

soutenu le 4 décembre mai 2014 devant le jury composé de :

- Rapporteurs** — René MANDIAU, Professeur, Université de Valenciennes
Philippe MATHIEU, Professeur, Université de Lille
Juan PAVÓN MESTRAS, Professeur, Universidad Complutense Madrid
- Examineurs** — Marie-Pierre GLEIZES, Professeur, Université de Toulouse
Abderrafaa KOUKAM, Professeur, Université de Technologie de Belfort-Montbéliard
Rodolphe LE RICHE, DR CNRS, MINES Saint-Etienne
Juan Antonio RODRÍGUEZ-AGUILAR, Tenured Scientist, IIA-CSIC
Wamberto VASCONCELOS, Senior Lecturer, University of Aberdeen
- Directeur** — Olivier BOISSIER, Professeur, Mines Saint-Étienne



Table des matières

I	Problématique et cadre de travail	1
1	Introduction	3
1.1	Approche et objectif	3
1.2	Axes de recherche	4
1.3	Structure du document	6
2	De l'adaptation dans les systèmes multi-agents	9
2.1	Notion d'agent	9
2.2	Notion d'organisation	10
2.3	Adaptation d'organisations	12
2.4	Réorganisation et auto-organisation	17
II	Systèmes multi-agents et adaptation dans la résolution de problèmes	19
3	Résolution de problèmes et multi-agents : un positionnement	21
3.1	Résolution <i>par</i> ou <i>pour</i> les systèmes multi-agents	21
3.2	Cadres de modélisation des problèmes	22
3.3	Stratégies d' <i>agentification</i>	24
3.4	Adaptation dans la résolution	26
4	Adaptation dans l'exploration par population d'agents	29
4.1	Optimisation et métamodélisation	29
4.2	Diversification de métamodèles	31
4.3	Partitionnement auto-adaptatif	33
4.4	Affinement de métamodèles	37
4.5	Synthèse et analyse	39
5	Auto-adaptation dans la résolution de problèmes	41
5.1	Processus d'auto-adaptation	41
5.2	Approche par coordination via l'environnement	42
5.3	Approche par négociation	46
5.4	Synthèse et analyse	50

III	Ingénierie des systèmes multi-agents adaptatifs	53
6	Ingénierie orientée multi-agent : un positionnement	55
6.1	Modèles et architectures d'agents	55
6.2	Programmation et architectures de multi-agents	57
6.3	Les méthodes orientées agents et multi-agents	59
6.4	L'adaptation dans l'ingénierie multi-agent	60
7	Approche organisationnelle de la réorganisation	63
7.1	Approche <i>MOISE</i> de l'adaptation	63
7.2	Organisation de la réorganisation	65
7.3	Synthèse et analyse	68
8	Architectures pour l'adaptation par auto-réorganisation	71
8.1	Architecture de gouvernance adaptative de systèmes	71
8.2	Architecture de gestion de la confiance dans les communautés virtuelles	75
8.3	Synthèse et analyse	79
IV	Conclusion et perspectives	81
9	Synthèse des travaux de recherche	83
9.1	Contributions	83
9.2	Domaines d'application et projets	85
10	Perspectives et projet(s) de recherche	87
10.1	Résolution multi-agent de problèmes	87
10.2	Ingénierie et théorie de l'auto-réorganisation	89
10.3	Applications	90
V	Références	93
11	Publications	95
11.1	Chapitres de livres	95
11.2	Éditions de livre	96
11.3	Journaux et revues	96
11.4	Conférences internationales (avec comité de lecture)	97
11.5	Workshops internationaux (avec comité de lecture)	99
11.6	Conférences nationales (avec comité de lecture)	103
11.7	Autres communications (avec comité de lecture)	104
11.8	Thèses et mémoires	105
	Bibliographie	107

VI	Annexes	115
A	Activités d'enseignement	117
A.1	Responsabilités pédagogiques	117
A.2	Interventions pédagogiques	118
A.3	Responsabilités administratives	120
B	Activités de recherche	121
B.1	Parcours professionnel	121
B.2	Responsabilités scientifiques et de valorisation	121
B.3	Responsabilités administratives	123
C	Encadrements de thèses et de masters	125
C.1	Thèses soutenues	125
C.2	Thèses en cours	126
C.3	Masters recherche	126
C.4	Jurys de thèses	127
D	Programmes nationaux et internationaux	129
D.1	Projets nationaux	129
D.2	Projets internationaux	130
D.3	Coopérations diverses	130

Première partie

Problématique et cadre de travail

1 Introduction

Ce manuscrit dresse une synthèse de mes travaux de recherche afin d'obtenir l'*Habilitation à Diriger des Recherches* (HDR). Ces travaux ont été menés lors de post-doctorats à l'IRIT, de 2004 à 2007, puis en poste de maître-assistant à l'École des Mines de Saint-Etienne, à partir de 2007. Depuis ma thèse, soutenue en 2004 et s'intéressant principalement à l'ingénierie de systèmes adaptatifs tels que des systèmes robotiques collectifs ou des logiciels de gestion d'emploi du temps, mes travaux se sont diversifiés et ont touché un plus large éventail de sujets théoriques et applicatifs — mais toujours dans le but de concevoir des systèmes pour lesquels les capacités d'adaptation sont la préoccupation principale. Ces travaux ont été menés dans le cadre de divers projets nationaux et internationaux, au travers de l'encadrement d'étudiants de Master, de doctorants, ou tout simplement lors de coopérations avec des collègues académiques ou industriels. Ce travail n'est donc pas un travail solitaire, sa synthèse en devient d'autant plus nécessaire, compte tenu de la richesse des influences et des collaborations y ayant participé.

1.1 Approche et objectif

Mes travaux de recherche se situent dans le domaine de l'informatique, de l'ingénierie logicielle et de l'intelligence artificielle en général et les *systèmes multi-agents* en particulier. Les systèmes multi-agents sont une approche de modélisation et de développement d'applications dans lesquelles les décisions sont décentralisées par nature, au sein d'agents. Ces agents autonomes interagissent entre eux directement ou indirectement par l'intermédiaire d'un espace partagé (appelé environnement) pour atteindre leurs objectifs individuels et collectifs. Les recherches sur les systèmes multi-agents sont diverses et peuvent se focaliser sur des mécanismes de négociation, des modèles de coordination ou des méthodes de développement, par exemple. Cette approche permet d'aborder les verrous et défis des applications actuelles comme la décentralisation, la complexité, ou l'ouverture. Les domaines d'application sont nombreux — robotique collective, contrôle distribué (manufacturier, par exemple), intelligence ambiante, villes et territoires intelligents, conception de produits complexes, etc.

Adaptation comme objectif

Mes travaux ont toujours eu comme objectif de fournir des outils (méthodes, modèles, etc.) et des mécanismes (algorithmes, protocoles, etc.) de mise en place de processus d'adaptation dans des systèmes logiciels. L'adaptation est donc ici une propriété que nous cherchons à fournir (voire à garantir)

à nos systèmes afin de répondre à une dynamique endogène ou exogène. En effet, les systèmes actuels sont à la fois (i) composés de multitudes de composants (parfois non conçus initialement pour fonctionner ensemble) ayant leurs propres dynamiques internes (dynamique endogène), et (ii) en interaction avec des environnements fortement dynamiques et incertains — notamment en raison d’une augmentation de l’échelle de déploiement et de la forte intégration de l’utilisateur dans le fonctionnement nominal des systèmes (dynamique exogène).

Approche par systèmes multi-agents

L’approche multi-agent est un excellent paradigme de modélisation et d’implantation de systèmes possédant de telles propriétés, notamment au travers du concept d’*organisation* (que celle-ci soit explicite ou non). En effet, l’organisation est le concept porteur des fonctionnalités du système à adapter — *changer d’organisation pour adapter le système*. Nous utilisons donc cette approche dans nos travaux comme *moyen* de mise en place des capacités d’adaptation au niveau systémique — notamment par l’usage de mécanismes d’auto-organisation. Ici, nous mettons donc en place une *adaptation par SMA*. Il est bien sûr possible, de manière complémentaire, d’aborder le problème de l’adaptation par le changement des comportements des agents, au sein d’une organisation, ce que nous aborderons dans certains travaux présentés dans ce document.

Le domaine des SMA est également source de nombreux défis, et est un *sujet d’étude* à part entière. Nous pensons notamment aux besoins de développer des méthodes de modélisation et de développement propres à l’*adaptation dans les SMA*, ou bien aux problèmes de coordination ou de respect de la *privacités*¹ dans l’usage de telles techniques.

De l’adaptation par auto-organisation dans les systèmes multi-agents

Mes travaux de recherche visent à répondre à la question scientifique transverse « *Comment mettre en place des processus d’auto-organisation permettant l’émergence de comportements cohérents et souhaités afin de développer des systèmes adaptatifs ?* »

Nous nous posons une question centrale dans la compréhension et la modélisation de systèmes artificiels complexes composés de multiples parties interagissant dans des schémas dynamiques non-linéaires. La question d’ingénierie soulevée lorsque l’on s’attaque à un tel problème est de savoir comment garantir certaines propriétés macroscopiques du système conçu en ne développant que des entités (et leurs comportements) à un niveau microscopique. Dans cette approche, nous utilisons les systèmes multi-agents pour modéliser et concevoir ces systèmes complexes adaptatifs et utilisons des outils d’analyse statistique, par exemple, pour vérifier, par simulation, la bonne conduite du système.

Tous les travaux présentés dans ce mémoire contribuent donc, à leur manière, à répondre à cette question, déclinée en deux sous-axes présentés ci-après.

1.2 Axes de recherche

De manière transverse, mes travaux s’intéressent à l’adaptation dans les systèmes complexes, à l’auto-organisation et l’émergence, dans les systèmes multi-agents. Ces recherches se structurent principalement en deux axes, qui y contribuent chacun à leur manière : la *résolution multi-agent de problèmes* et l’*ingénierie orientée multi-agent*. Ces deux axes se justifient par mon parcours scientifique. En effet, ma thèse s’intéressait aux aspects méthodologiques pour les SMA, tout en illustrant les outils et méthodes proposés dans le cadre de la résolution de problèmes réels. Il a donc été naturel pour moi de continuer à aborder les aspects ingénierie (notamment lors de mon intégration dans l’équipe de l’École

1. Nous faisons le choix de traduire le mot *privacy* par *privacités*, au lieu d’utiliser confidentialité, étant plus réducteur.

des Mines), mais également de creuser les techniques multi-agents de résolution de problèmes, en tant que telles, dans le cadre de projets de recherche et de collaborations nationales et internationales.

1.2.1 Adaptation des systèmes multi-agents et résolution de problèmes

Cet axe cherche à répondre à la question scientifique : « *Quel est l'apport de l'adaptation des systèmes multi-agents à la résolution de problèmes dynamiques?* ». Par résolution de problèmes nous entendons ici des cadres formels de modélisation de problèmes réels, comme en intelligence artificielle ou en recherche opérationnelle (e.g. problèmes de satisfaction de contraintes, problèmes d'optimisation). Cette question concerne la résolution de problèmes à travers le prisme multi-agent, et invoque l'intérêt d'une telle approche lorsque le contexte est dynamique. Cependant, avant de répondre à cette question il est important de s'interroger sur les moyens et les techniques. Cette question concerne donc également la résolution de problèmes intrinsèquement présents dans les systèmes multi-agents, comme les problèmes de coordination ou de coopération, dynamiques par nature.

Afin de résoudre des problèmes naturellement ou conceptuellement distribués (e.g. constitution d'emplois du temps multi-usagers, coordination en robotique collective), les systèmes multi-agents apparaissent comme un paradigme de modélisation riche et pertinent. Notamment, les systèmes multi-agents auto-organiseurs proposent des mécanismes de résolution de conflits, d'adaptation, de décentralisation, et de stabilisation qui peuvent être vus comme autant de moyens de résoudre des problèmes distribués et dynamiques. Dans le cadre de projets collaboratifs, j'ai principalement développé des recherches autour de problèmes combinatoires, et de l'aide à la conception de produits complexes.

Dans le cadre de mes recherches j'ai abordé la résolution de *problèmes combinatoires* sous contraintes comme l'affectation de fréquences, le contrôle manufacturier ou la gestion d'emplois du temps, par auto-organisation. L'idée est dans ce cas de développer des agents — responsables de l'atteinte d'un objectif micro, e.g. trouver une valeur pour une variable — et les mécanismes de coordination et de négociation pour atteindre, grâce à la dynamique du système, un état stable, c'est-à-dire une solution au problème macro. L'avantage de cette approche est de fournir un système capable de résoudre le problème dans sa globalité, mais également de réparer des solutions lorsque l'état du système change. Par exemple, dans un système de contrôle manufacturier, il est important de proposer des solutions alors que le système se dégrade (e.g. pannes machines, indisponibilités d'opérateurs) ou lorsque les données d'entrée changent (e.g. ajout d'une commande prioritaire), sans toutefois reprendre le processus de résolution à partir de zéro.

Au-delà des problèmes combinatoires, j'ai également appliqué les techniques multi-agents à la résolution de *problèmes de conception de produits complexes* par des processus d'optimisation faisant appel à des simulateurs coûteux (e.g. l'*optimisation multidisciplinaire*), qui nécessitent une extension des techniques classiques de métamodélisation et d'exploration de l'espace de recherche par ajout de capacités d'adaptation par auto-organisation (voire d'apprentissage) au solveur en cours de fonctionnement. La question de l'adaptation dans cet axe est principalement *algorithmique* et se pose de deux manières (chacune étant abordée dans un chapitre de ce document) :

Adaptation pour la modélisation de l'espace de recherche (chapitre 4) : dans le cadre de la conception de produits complexes, il est nécessaire de fournir des méthodes de compréhension des modèles étudiés (souvent coûteux), notamment par métamodélisation. La capacité d'un système multi-agent à s'adapter à l'espace de recherche est une technique proche des techniques de métamodélisation utilisées en optimisation.

Adaptation à la dynamique environnementale (chapitre 5) : lors de la résolution d'un problème complexe dynamique, les *données/paramètres/modèles* d'entrée peuvent changer (e.g. le concepteur change les objectifs en cours de résolution), voire les caractéristiques du solveur (e.g. le concepteur change les comportements des agents pour explorer de nouvelles solutions). Ce genre de dynamique implique une capacité réelle d'adaptation du système à ces perturbations. Dans nos travaux, cette capacité sera déléguée aux agents composant le système (*auto-adaptation*).

1.2.2 Ingénierie des systèmes multi-agents adaptatifs

Cet axe cherche à répondre à la question scientifique « *Comment mettre en place l'adaptation dans des systèmes (multi-agents)?* » Cette question positionne les systèmes multi-agents comme *sujet*. La question soulève le problème de l'ingénierie de l'adaptation dans les systèmes multi-agents. Les travaux conduits ici peuvent être déclinés suivant un point de vue centré *agent* (ACPV) et un point de vue centré *organisation* (OCPV), dans lesquels la notion centrale de dynamique des organisations est considérée. Dans ACPV, cette notion correspond à des phénomènes ascendants et émergents qui sont regroupés sous le terme général d'*auto-organisation*. Dans OCPV, le concept d'organisation nourrit un large panel de travaux relatifs à la *réorganisation* formelle et descendante d'organisations préexistantes installées dans le système multi-agent.

Le but de ces recherches est de fournir des méthodes et des outils pour la modélisation et la conception de systèmes multi-agents. En intégrant l'équipe ISCOD de l'École des Mines de Saint-Étienne, experte dans la conception de systèmes multi-agents suivant une approche organisationnelle normative, nous avons pu étudier l'apport mutuel des approches centrées agent et des approches centrées organisation. En effet, alors que l'approche organisationnelle préconise une modélisation de la structure du système multi-agent, au travers de structure d'organisation, de plans et de normes, l'approche auto-organisationnelle se focalise sur les comportements des agents et les protocoles de coordination à mettre en œuvre pour atteindre un *objectif* commun. Nos travaux ont été l'occasion d'explorer le développement des systèmes multi-agents suivant ces deux approches de manière complémentaire, notamment sous le prisme de l'adaptation des systèmes par réorganisation ou auto-organisation. La mise en place de processus de contrôle ascendants et descendants (boucle micro-macro) est un des verrous scientifiques majeurs auquel nous nous attaquons dans ces travaux.

Pour résumer, la question de l'ingénierie des systèmes multi-agents adaptatifs dans cet axe est principalement *méthodologique et architecturale*, et se pose de trois manières (dont deux seront détaillées dans le document) :

Approche auto-organisationnelle : la conception de systèmes adaptatifs consiste en la mise en place de processus d'auto-organisation (organisation implicite). Cette question a été le principal sujet de mes travaux de thèse et ne seront donc pas détaillés plus avant dans ce document.

Approche réorganisationnelle (chapitre 7) : la conception du système adaptatif fait appel explicitement à la notion d'organisation, qui est manipulable par les agents grâce à un environnement et des primitives organisationnelles.

Approche auto-réorganisationnelle (chapitre 8) : la conception fait à la fois appel à des notions d'auto-organisation et de réorganisation, par la mise en place d'une boucle micro-macro sur l'objet organisation.

1.3 Structure du document

La partie I présente le cadre de travail de ce document, à savoir l'*adaptation dans les systèmes multi-agents*, dans le chapitre suivant (chapitre 2). Nous présenterons ici les notions importantes à la base de nos travaux (agents, organisation, etc.), la vision *multi-agent* que nous adoptons, ainsi que ce que nous entendons par *adaptation* dans la suite de ce manuscrit.

La partie II correspond au premier axe de recherche présenté dans la section 1.2.1, à savoir l'adaptation des systèmes multi-agents et la résolution de problèmes. Le chapitre 3 présente le cadre de la résolution de problèmes. Le chapitre 4 synthétise les travaux s'intéressant à la mise en place de capacités d'adaptation dans des solveurs de problèmes, et les illustre au travers d'exemples issus du projet ANR ID4CS (ANR-09-COSI-005). Enfin, le chapitre 5 complète cette partie sur la résolution de problèmes, par la présentation de travaux proposant des modèles d'adaptation pour la résolution de problèmes combinatoires dynamiques.

La partie **III** compile les contributions à l'ingénierie de systèmes multi-agents adaptatifs — axe présenté en section 1.2.2. Le premier chapitre résume la problématique de l'ingénierie des systèmes multi-agents et de l'adaptation, au travers de différents modèles, méthodologies et approches. Le chapitre 7 s'intéresse à la première facette de ces travaux, à savoir une approche organisationnelle de la réorganisation par programmation multi-agent. Dans le chapitre 8, l'ingénierie de systèmes adaptatifs passe cette fois par la définition d'architectures de systèmes suivant une approche organisationnelle et des comportements locaux de régulation mis en œuvre par des agents. Ces travaux seront illustrés par des applications dans les domaines des communautés virtuelles, des réseaux M2M et de l'intelligence ambiante. Ce chapitre discute ainsi de la mise en place conjointe de techniques de réorganisation et d'auto-organisation.

La partie **IV** dresse un bilan de nos travaux et des différentes contributions et projets, dans le chapitre 9, puis propose plusieurs perspectives pour chacun des deux axes de recherche, ainsi que d'autres pistes de domaines d'application, qui constitueront mon projet de recherche futur, dans le chapitre 10.

Une bibliographie complète se trouve en fin de manuscrit.

De l'adaptation dans les systèmes multi-agents

Ce chapitre discute de notre vision de l'adaptation dans les systèmes multi-agents— avant de détailler nos travaux en résolution de problèmes (voir partie II) et en ingénierie (voir partie III). Nous présentons le cadre général des systèmes multi-agents considérés, notamment au travers des concepts clés d'*agent* et d'*organisation* qui structureront notre approche. Nous posons la question du caractère implicite ou explicite de l'organisation, ou bien du lieu de définition de la coordination et de l'adaptation (dans les agents, ou dans l'organisation).

Nous n'allons cependant pas reprendre toutes les notions issues du domaine des systèmes multi-agents, et redirigeons donc le lecteur vers une bibliographie plus fondamentale [FERBER, 1995 ; WOOLDRIDGE, 2002], pour des définitions plus détaillées.

2.1 Notion d'agent

Au cœur de la conception d'un système multi-agent se trouvent les *agents*. Ces entités autonomes sont munies de capacités de décision, et ne possèdent qu'une vision locale de leur environnement. Ce sont généralement des abstractions de concepts existants (e.g. un agent représentant un robot autonome), mais ils peuvent également être de purs artefacts de conception, fournissant des fonctionnalités particulières au système (e.g. un agent représentant une variable dans un problème d'optimisation). Dans nos travaux, nous retrouvons ces deux types d'abstractions.

Plusieurs architectures pour concevoir des agents coexistent, mais aucune n'a vraiment pris le dessus sur les autres, compte tenu de leur complémentarité. En fonction des besoins, et des capacités de décision des agents, des architectures peuvent être plus appropriées que d'autres. Par exemple, pour concevoir des agents pouvant mettre en œuvre des raisonnements simples, une architecture d'agent logique délibératif conviendra [GENESERETH et al., 1987], alors que pour développer des agents nécessitant des capacités de planification, une architecture BDI sera plus adéquate [RAO et al., 1995]. C'est pour cela que, dans nos travaux, nous n'avons pas fait de choix précis d'architecture pour le développement de nos agents.

Dans ce manuscrit, plusieurs modèles d'agents se côtoient. Dans de nombreux travaux, de simples agents dit *réactifs* sont utilisés, et équipés de mécanismes de *coordination* directs (e.g. protocole de négociation) [PICARD et al., 2007b] ou indirects (e.g. interactions médiées par l'environnement physique) [CLAIR et al., 2008b]. Souvent ces agents ne font pas appel à des connaissances de haut niveau, et interagissent par échange de valeurs numériques. Dans d'autres travaux, nous avons recouru à une

conception plus cognitive, en utilisant des agents BDI ou des agents faisant appel à des concepts de haut niveau comme la confiance [YAICH et al., 2013] ou une structuration ontologique de leurs connaissances [SORICI, BOISSIER, PICARD et ZIMMERMANN, 2013 ; GILLANI, LAFOREST et al., 2014].

2.2 Notion d'organisation

Les agents évoluent ensemble dans un système (multi-agent), et peuvent faire partie d'une organisation ayant des objectifs différents des leurs. C'est sur cette notion d'organisation que va s'articuler notre vision de l'adaptation. En effet, une organisation, qu'elle soit explicite ou non, reflète la fonctionnalité du système. Si cette organisation n'est pas adéquate (fonctionnellement, par exemple), il faut la changer, l'*adapter* (directement/explicitement ou indirectement/implicitement). Nous allons voir qu'en fonction de l'implantation de l'organisation dans le système, les méthodes d'adaptation pourront varier.

Il n'y a pas de définition consensuelle de la notion d'*organisation*. Cependant il est possible de dresser un paysage synthétique des différentes approches reposant sur ce concept [J. F. HÜBNER, 2003 ; PICARD et al., 2009b]. Toutes ces visions ne sont pas mutuellement exclusives et ont donné lieu à différentes approches dans le domaine des systèmes multi-agents. Comme dans [BOISSIER, HÜBNER et al., 2007], nous nous focalisons sur quelques caractéristiques afin de construire une vision synthétique. Tout d'abord, nous prendrons en compte le processus de définition de l'organisation des agents, puis nous considérerons sa représentation au sein des agents. Nous nous intéressons ici aux moyens de manipuler et d'adapter une organisation, et non à sa forme. Une analyse des formes d'organisations se retrouve dans [FERBER, 1995].

2.2.1 Vue centrée agent vs. vue centrée organisation

Le premier axe de la grille (Figure 2.1) est une extension des points de vue centré agent et centré organisation initialement proposés dans [LEMAÎTRE et al., 1998].

Le point de vue *centré agent* considère les agents comme étant le « moteur » de l'organisation. Les organisations n'existent qu'en tant que phénomènes émergents observables qui établissent une vision, globale, objective et ascendante des schémas de coopération entre les agents (voir première ligne dans la Figure 2.1-a-b). Par exemple (cas (a)), dans une colonie de fourmis aucune contrainte comportementale et organisationnelle n'est explicitement et directement définie dans les fourmis [DROGOUL et al., 1995]. L'organisation est le résultat d'un comportement collectif émergent conséquence de la façon pour les agents d'agir et d'interagir dans un environnement dynamique commun [MÜLLER, 1998]. Un point de vue similaire peut être considéré dans les différentes approches réactives de la littérature [PICARD et GLIZE, 2006]. D'un point de vue plus cognitif (cas (b)), les études sur la formation de coalitions définissent des mécanismes (au sein des agents, comme le raisonnement social [J. S. SICHTMAN et al., 1994], par exemple), pour construire les schémas de coopération, structurant et aidant les agents dans leurs activités collaboratives, lors d'un processus ascendant.

Le point de vue *centré organisation* est orienté vers la direction opposée : les organisations existent en tant qu'entités explicites du système (voir la deuxième ligne dans la Figure 2.1-c-d). Il souligne l'importance d'une dimension supra-individuelle et l'usage de primitives différentes de celles des agents. Un schéma de coopération est fixé par le concepteur (ou par les agents eux-mêmes) et installé de manière descendante afin de contraindre ou de définir les comportements des agents. Notons que l'observateur du système peut obtenir une description de l'organisation du système. Par exemple, dans une école, on dispose de documents décrivant comment cet établissement est organisé. Bien sûr, en plus de la description explicite de l'organisation, l'observateur peut également observer l'organisation réelle de l'école qui est possiblement différente de l'organisation formellement définie.

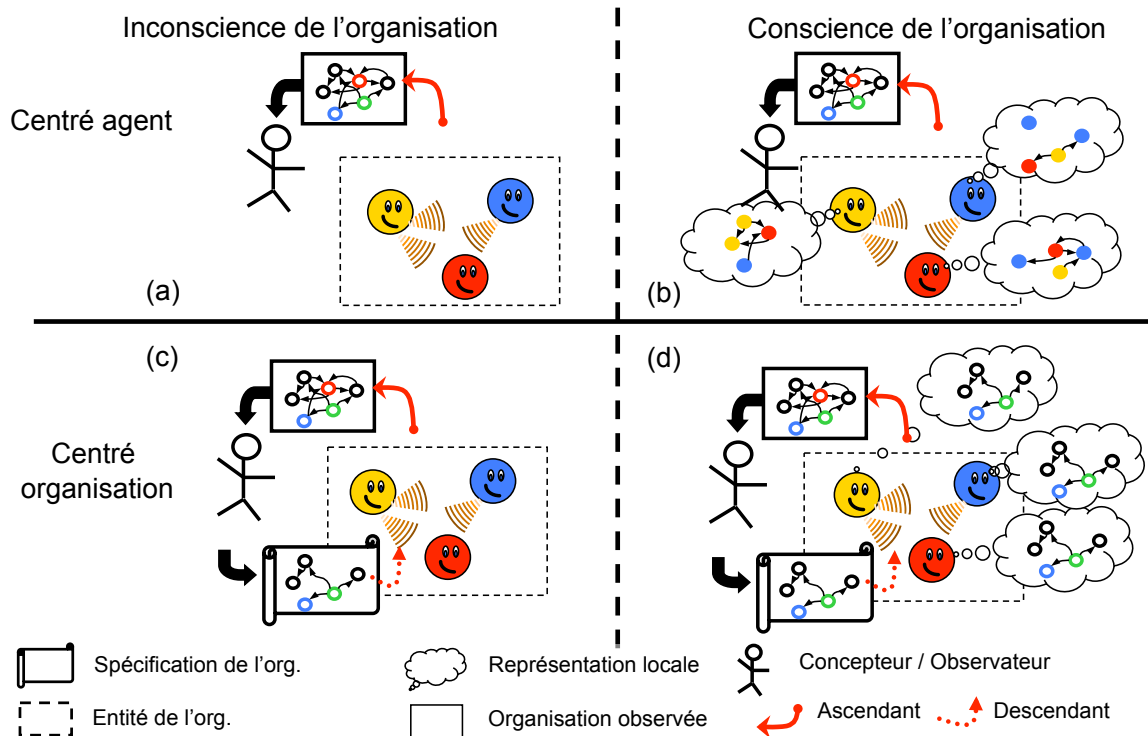


FIGURE 2.1 – Vue synthétique [J. F. HÜBNER, 2003 ; PICARD et al., 2009b] : (a) SMA émergents; (b) SMA basés sur les coalitions; (c) Ingénierie orientée agent; (d) SMA orientés organisation. Le concepteur/observateur peut être le développeur/utilisateur (cas exogène) ou un ensemble d'agents (cas endogène).

2.2.2 Organisation implicite vs. explicite

En considérant l'architecture des agents, nous pouvons affiner ces deux points de vue en suivant l'axe orthogonal sur la capacité des agents à se représenter et à raisonner sur l'organisation, et à l'expliquer.

Dans la première colonne de la Figure 2.1, les agents ne connaissent rien à propos de l'organisation. Dans le cas (a) les agents ne représentent pas l'organisation, bien que l'observateur puisse identifier une organisation émergente des interactions entre agents. Ils peuvent néanmoins avoir une représentation locale, interne et partielle des autres agents. Dans le cas (c), l'organisation existe comme un schéma spécifié et formalisé, construit par le concepteur des agents, mais les agents ne savent rien de cette organisation et ne peuvent donc raisonner sur ce sujet. Ils s'y conforment simplement comme si les contraintes organisationnelles étaient codées *en dur* en eux — par exemple, le système multi-agent résultant d'un déroulement d'une méthode de développement orientée agent où le code des agents est généré à partir de spécifications organisationnelles.

Dans la seconde colonne, nous considérons les cas où les agents ont une *représentation locale explicite de l'organisation* dans laquelle ils évoluent. Dans le cas (b), chaque agent possède une représentation des schémas de coopération qu'il suit afin de décider que faire — par exemple, les réseaux de dépendances pour la formation de coalitions [J. S. SICHMAN et al., 1994]. Cette représentation locale est obtenue par les perceptions, les communications ou des raisonnements explicites car il n'y a, a priori, dans une vue centrée agent, aucune représentation globale explicite de l'organisation définie. Les agents sont capables de raisonner sur l'organisation et de l'utiliser afin d'initier des coopérations avec d'autres agents.

Toutefois, les frontières entre ces différentes catégories sont floues et perméables et les manipulations ascendantes ou descendantes d'une organisation peuvent être réalisées soit de manière *endogène*

(par les agents appartenant à l'organisation) ou de manière *exogène* (par une entité, humain ou agent, hors du système).

2.2.3 Positionnement des travaux multi-agents

À des fins d'illustration, nous pouvons positionner quelques systèmes multi-agents concrets et approches existantes dans cette grille.

SMA à organisation émergente (en haut à gauche) : ici les agents sont incapables de raisonner sur l'organisation puisqu'elle n'est pas modélisée. Les agents s'auto-organisent de pair-à-pair ou en utilisant des communications indirectes via l'environnement comme dans [PARUNAK et S. A. BRUECKNER, 2004]; les agents peuvent aussi avoir une représentation locale d'autres agents comme dans certains systèmes multi-agents adaptatifs (à la AMAS) [GEORGÉ, EDMONDS et al., 2004].

SMA basés sur les coalitions (en haut à droite) : ici les agents sont toujours incapables de raisonner sur l'organisation globale mais peuvent construire des modèles propres, plus ou moins détaillés, des interactions et des relations de dépendance avec leur voisinage en respectant des règles sociales et des schémas prédéfinis. Des exemples classiques de telles approches sont les systèmes multi-agents basés sur le *contract net protocol* [SMITH, 1980]. De manière plus élaborée, les coalitions multi-agents [J. S. SICHTMAN et al., 1994] raisonnent directement sur des schémas de coopération.

Ingénierie orientée agent (en bas à gauche) : cette catégorie d'approches considère l'organisation durant la conception (*organisation by design*). Les organisations sont spécifiées avant l'implémentation des agents. Des exemples de telles approches se retrouvent principalement dans la communauté AOSE (*Agent Oriented Software Engineering*) qui propose de nombreuses méthodes orientées (multi-)agent se focalisant sur les modèles organisationnels comme dans MaSE [DELOACH, 2004], INGENIAS [PAVÓN et al., 2003] ou bien ASPECT pour les approches holoniques basées sur CRIO [COSSENTINO, GAUD, GALLAND et al., 2007].

SMA orientés organisation (en bas à droite) : ces approches sont influencées à la fois par l'ingénierie orientée agent et le raisonnement social. En effet, les organisations sont manipulées (i) par le concepteur pour spécifier le système à concevoir et (ii) par les agents qui ont une représentation de cette organisation et peuvent donc effectuer des actes organisationnels et ainsi potentiellement modifier l'organisation. AGR [FERBER et GUTKNECHT, 1998], TAEMS [V. LESSER et al., 2004], STEAM [TAMBE, 1997], MOISE⁺ [HÜBNER et al., 2002] ou ISLANDER [ESTEVA et al., 2001] en sont de bons exemples.

Nous placerons dans cette grille les travaux présentés dans la suite de ce document.

2.3 Adaptation d'organisations

Comme nous avons pu l'aborder plus tôt, nous envisageons l'adaptation d'un système via l'adaptation de son organisation. Nous analysons ici les différents aspects concernant les deux principales approches pour l'adaptation des organisations — la réorganisation et l'auto-organisation — au travers des questions *quoi, qui, pourquoi, quand et comment*.

2.3.1 Qu'est-ce qui est changé?

Changer l'organisation peut simplement impliquer des changements au sein du système à différents niveaux (agents, coalitions, topologie, etc.). Ces changements peuvent dépendre fortement du point de vue adopté (ACPV ou OCVP) et sur les capacités organisationnelles des agents (étant conscients ou non de l'organisation).

Dans les *SMA à organisation émergente* (Fig.2.1-a), l'organisation observée peut être changée en terme de : configuration spatiale, voisinages, différenciation/spécialisation des individus, etc. Dans de

nombreuses approches, la configuration spatiale contraint fortement les capacités et le potentiel que le système peut exhiber. Par conséquent, l'adaptation du système s'effectue en changeant cette configuration de telle sorte que le système se comporte de manière plus adaptée à la pression de l'environnement perçue ou subie. Par exemple, dans un mécanisme auto-construit, les composants mécaniques (agents) changent eux-mêmes leurs connexions à d'autres composants afin d'optimiser une fonction de trajectoire, qui peut changer en cours de fonctionnement, sans être conscients de la forme globale, donc de l'organisation, du mécanisme [CAPERA, M. GLEIZES et al., 2004]. Une telle adaptation structurelle est également proposée dans [KOTA et al., 2012], dans le contexte de l'allocation de ressources. Ce genre d'adaptation se retrouve en robotique collective où la position spatiale des agents est fortement liée aux plans qu'ils doivent exécuter [DE WEERDT et al., 2009]. L'organisation observée peut également être modifiée en changeant les partenaires des agents ou leur manière d'interagir entre-eux. On retrouve de tels mécanismes de régulation dans les systèmes épidémiques (*gossip*), les marchés (création-destruction) ou bien les communautés d'intérêts basées sur la confiance. D'autres exemples de mécanismes auto-organiseurs sont également présentés dans [DI MARZO SERUGENDO et al., 2006 ; HASSAS et al., 2006]. Adapter un système peut aussi être effectué à travers des changements de tâches, de buts à atteindre. Dans les approches bio-inspirées, l'auto-différenciation met en place une allocation de rôles (du point de vue de l'observateur) ascendante et implicite [DROGOUL et al., 1995].

Dans les SMA basés sur les coalitions, la topologie de l'organisation est exprimée en terme de configurations sociales locales : les voisinages, les coalitions exprimées en termes de pouvoirs, de dépendances, ou d'engagements sociaux. L'organisation peut ainsi être modifiée en changeant les partenaires avec lesquels les agents coopèrent ou leur manière d'interagir. Par exemple, les agents modifient leur façon d'interagir en fonction d'une évaluation de la *croyance* qui est calculée en utilisant des expériences passées et des opinions provenant d'autres agents [CASTELFRANCHI, 2000]. Ce mécanisme local de régulation guide le système vers un ordre social. Dans des solveurs distribués de problèmes sous contraintes, une coalition est définie/modifiée à partir d'une élection dépendant de l'état courant du voisinage (par exemple, l'agent le plus contraint est élu pour proposer une solution) ou de l'avancement dans la résolution [MAILLER et V. LESSER, 2004 ; PICARD et al., 2007b].

Pour les SMA orientés organisation, comme il existe une spécification explicite, le changement d'organisation peut être considéré à deux niveaux : (i) modification de la définition même de l'organisation et (ii) modification de l'allocation des rôles aux agents. Un exemple du premier cas se retrouve dans les travaux de [J. HÜBNER, J. SICHMAN et al., 2004], dans lesquels les agents sont capables d'évaluer leur organisation, d'identifier que le problème est causé par la définition ou spécification courante, et ainsi de décider de modifier la spécification. Ce genre de changements implique normalement des changements dans l'allocation des rôles. Dans le second cas, la réorganisation peut consister en la reproduction et la composition d'agents afin de réassigner les rôles joués par les agents [KAMBOJ et al., 2007]. Les raisons pour changer l'organisation proviennent de la volonté d'obtenir une structure adaptée à l'environnement et que les tâches soient adaptées aux agents présents. Un autre exemple se retrouve dans [GLASER et al., 1997] où l'organisation est changée par l'entrée de nouveaux agents jouant des rôles particuliers. Un nouvel agent est accepté seulement lorsqu'il augmente l'utilité du système dans son intégralité. La génération de structures organisationnelles a également été proposée, comme le résultat de perspectives locales [GASSER et al., 1991], globales [CORKILL et al., 1983], et hybrides.

D'un point de vue *ingénierie*, le concepteur peut changer le *modèle* du système à plusieurs niveaux. A un bas niveau, l'agentification peut être changée, et ainsi l'organisation. A un plus haut niveau, le modèle de l'organisation peut également être modifié en ajoutant de nouveaux rôles, groupes, tâches, etc.

2.3.2 Quand et qui change l'organisation ?

Une organisation peut être modifiée à différents moments du cycle de vie du système. Ce processus peut être initié par différents acteurs. La décision d'initier le processus peut être soit statique, soit dynamique [J. HÜBNER, J. SICHMAN et al., 2004]. Dans le premier cas, le processus est démarré en fonc-

tion de critères prédéfinis lors de la spécification de l'organisation. Dans le second cas, les processus de réorganisation est une conséquence du fonctionnement du système. Cela signifie que si les agents ne vérifient pas certains critères (but, performance, etc.) l'organisation est changée.

Le processus est statique et réalisé par le concepteur et c'est uniquement *lors de la conception* que les organisations sont adaptées. Seuls les concepteurs peuvent détecter, en utilisant des outils de *model checking*, de prototypage rapide ou de simulation leur permettant d'exhiber des comportements globaux non désirés. Les agents ne sont alors pas acteurs de cette adaptation de l'organisation, mais seulement une implémentation du modèle organisationnel que le concepteur spécifie.

Le processus est dynamique quand l'organisation est modifiée en *cours de fonctionnement*. Cette modification peut être réalisée par une entité extérieure au SMA (concepteur, autre système, etc.) qui agit sur les agents en train de s'exécuter ou par les agents eux-même, afin de concevoir des systèmes autonomes. Par exemple, les agents au sein d'un système auto-organisateur changent indirectement l'organisation en réponse à un *changement environnemental* détecté au niveau des agents. Ainsi, dans l'exemple d'un mécanisme physique auto-construit, les composants mécaniques peuvent changer leurs pondérations (tension, longueur, etc.) lorsqu'ils reçoivent un *feedback* extérieur positif ou négatif (par le biais de mécanismes de propagation), concernant la distance à la trajectoire-objectif [CAPERA, M. GLEIZES et al., 2004]. Une coalition réagit également aux changements détectés au niveau de la coalition, par *pression sociale* (par exemple, une coalition a reçu une opinion négative concernant son leader), et ainsi change l'organisation (e.g. en changeant de leader) [LONG et al., 2007].

2.3.3 Pourquoi et comment l'organisation change ?

Alors que nous pouvons identifier les différentes sortes d'*objets ou artefacts modifiés* en fonction du modèle organisationnel utilisé, nous pouvons également identifier certains types de *processus de changement* (pourquoi/comment) qui seront détaillés dans la suite de la section : (i) les changements prédéfinis, (ii) les changements contrôlés et (iii) les changements émergents. L'organisation doit être modifiée dès lors que l'organisation n'aide pas à l'atteinte du but social. En d'autres mots, l'organisation courante contraint les comportements des agents à des comportements qui ne permettent pas d'atteindre le but global. De telles situations peuvent survenir, par exemple : lorsque l'environnement, le but du système multi-agent et/ou les exigences de performances ont changé ; lorsque les agents ne sont pas capables de bien jouer leurs rôles ; lorsque une nouvelle tâche arrive alors que l'organisation n'est pas appropriée, etc. [DIGNUM et al., 2004] définissent le « quand » se réorganiser comme étant lié à l'utilité de l'organisation (succès des interactions, des rôles et de la structure) et à l'utilité des agents (différente pour chaque agent, dépendant de ses buts et production/consommation de ressources). Dans les AMAS [GEORGÉ, EDMONDS et al., 2004], par exemple, l'organisation doit changer dès que le système n'est plus en adéquation fonctionnelle avec son environnement. Cependant, ce changement d'organisation peut également avoir un coût à prendre en compte lors de la décision, et pose la question du compromis entre adéquation fonctionnelle et coût de l'adaptation — d'où la proposition d'un service dédié de réorganisation conscient de ce coût dans [ALBEROLA et al., 2012].

Processus générique de changement d'organisation

De manière générale, afin de modifier une organisation, nous pouvons identifier un processus générique qui sera implémenté différemment en fonction de l'approche suivie. Ce processus est composé de deux phases — la surveillance (ou *monitoring*), et la réparation. Cette dernière phase suivant le type de système, peut être constituée des phases de conception (ou *design*), de sélection et d'exécution (ou d'implémentation) [So et al., 1993] ou uniquement des phases de sélection et d'exécution. Il peut s'inscrire dans un bouclage environnement-système : perception, processus de changement, action, et ainsi de suite. Les problèmes inhérents à ce processus de changement et à l'approche choisie sont détaillés dans la suite de cette section.

1. La phase de surveillance ou *monitoring* est destinée à détecter les problèmes d'inadéquation entre le système, l'organisation ou les agents, et l'environnement. Quelle que soit l'entité responsable de cette surveillance (agent, organisation, entité externe), et donc quel que soit le niveau d'abstraction de cette détection (niveau macro ou niveau micro), il convient de définir des situations de non adaptation.
2. Une fois qu'un besoin de modification est détecté lors du monitoring, la phase de *réparation* met en œuvre un processus pour que l'organisation retrouve un fonctionnement le plus optimal possible, et ceci en cours de fonctionnement du système.
 - (a) Pour cela, la phase *design* vise à définir et à développer un ensemble d'alternatives possibles pour l'organisation courante, de manière ascendante ou descendante.
 - (b) La phase de *sélection* détermine une des alternatives pour modifier l'organisation. Le principal problème est donc la définition de *critères d'évaluation* des propositions les plus prometteuses.
 - (c) La phase d'*exécution* ou d'*implémentation* correspond à la mise en œuvre de l'alternative choisie précédemment.

La phase de *monitoring* identifie ici une situation où l'*organisation actuelle n'est pas adaptée* et ne satisfait pas les besoins du système multi-agent. Le principal problème lors de cette phase est d'*identifier si le but global ne sera pas atteint (ou dans de mauvaises conditions) à cause d'une organisation courante inappropriée*. De nombreuses autres raisons peuvent impliquer que le SMA n'accomplisse pas sa tâche globale (par exemple, le but est simplement impossible à atteindre). Dans certains cas, changer l'organisation n'est pas une solution. Même dans le cas où le problème est connu pour être résolu par un processus de réorganisation, le nouveau problème est d'*identifier quelle partie de l'organisation a causé le problème* afin de modifier l'organisation au bon niveau. La sous-partie de l'organisation responsable du problème peut être soit sa spécification (par exemple, l'ensemble des rôles jouables), soit l'instantiation courante de cette même spécification (par exemple, l'affectation des rôles à des agents). La *réparation* nécessite ensuite d'exécuter les phases *design*, de *sélection* et d'*exécution*. La phase de *design* vise ensuite à développer un ensemble d'alternatives possibles pour l'organisation courante. La conception de cet ensemble peut être (i) basée sur la recherche dans une bibliothèque d'organisations prédéfinies ou (ii) construite à la demande. Dans le premier cas, le problème est d'identifier quelle organisation prédéfinie est appropriée pour réparer l'échec survenu. Dans le second cas, il faut gérer la grande taille de l'espace de recherche (défini par le modèle organisationnel) pour une nouvelle spécification organisationnelle. Lors de l'*exécution*, il faut établir les moyens de changer l'organisation en cours d'exécution sans causer d'échecs. Par exemple, comment un agent gèrera-t-il le fait que le rôle qu'il jouait jusqu'alors soit retiré de la nouvelle spécification organisationnelle? Que fera-t-il des engagements qu'il avait pris pour ce rôle obsolète? Dans la limite de nos connaissances, il n'existe actuellement pas de travaux répondant à ces problèmes.

Changement prédéfini de l'organisation

Ici les changements sont déjà planifiés et exprimés avec, entre autres, des modèles organisationnels temporels [CARRON et al., 2001] par le concepteur. Par exemple, une équipe de football a prévu de changer de formation au bout de 30 minutes de jeu [STONE et al., 1999]. Dans cette approche, l'exécution du processus de changement est assez immédiat. Le monitoring est effectué par les agents eux-mêmes ou une entité extérieure (par exemple, le coach de l'équipe de football consulte un timer pour connaître le temps pour la condition de déclenchement du changement). La phase de conception, qui détermine toutes ces conditions de déclenchement, n'est pas réalisée en cours de fonctionnement, puisque les conditions de déclenchement doivent être prédéfinies. Les phases de sélection et d'exécution sont immédiates, car les conditions de déclenchement vont de pair avec des actions prédéfinies, exécutables à la volée.

Ici, le système ne sait pas quand l'organisation changera, mais connaît les conditions nécessaires de déclenchement d'un changement, qui sera effectué en suivant une procédure connue (par exemple, une équipe a un expert qui contrôle le processus de réorganisation). Ce processus peut être conduit de deux manières : (i) une approche *endogène* où un agent particulier (centralisation) ou une partie des agents eux-mêmes (de manière décentralisée et coordonnée) prendront en charge la réorganisation ; (ii) une approche *exogène* où l'utilisateur du système multi-agent, ou un système extérieur, contrôle le processus. Durant l'exécution d'une instance du système spécifié, l'entité responsable du changement de l'organisation (concepteur ou agent) peut détecter que l'organisation n'est pas adaptée pour cause de *performances non adéquates*, et peut ainsi modifier le modèle et la spécification pour améliorer les performances en *programmant* une organisation plus adaptée.

MOISE⁺ est un bon exemple d'une telle approche réorganisationnelle [J. HÜBNER, J. SICHMAN et al., 2004]. Ces travaux considèrent la structure (rôles, groupes) et le fonctionnement de l'organisation. Ils se concentrent sur le contrôle du processus de réorganisation présenté précédemment. De leur point de vue, la réorganisation est un processus coopératif en lui-même qui est effectué de manière endogène et décentralisée. Ce processus peut lui-même être le sujet d'une organisation spécifique composée d'une hiérarchie de rôles dédiés à la gestion de la réorganisation. Un autre exemple est [HORLING et al., 2001], où un processus de réorganisation centralisé est utilisé, basé sur le langage de modélisation TAEMS et sur le diagnostic d'un sous-système expert chargé de détecter les défauts de l'organisation et d'assister à la création d'une solution. Des exemples d'auto-organisation contrôlée peuvent également se rencontrer dans certains solveurs SMA, dans lesquels des rôles prédéfinis sont jouables par les agents en fonction de l'avancement du processus de résolution [MAILLER et V. LESSER, 2004 ; PICARD et al., 2007b]. Les agents peuvent jouer différents rôles — *médiateur* [MAILLER et V. LESSER, 2004] ou *élu* [PICARD et al., 2007b] — prédéfinis par le concepteur, à différents moments de la résolution.

Changement émergent de l'organisation

Dans de ce dernier cas, comme pour le changement contrôlé de l'organisation, le moment où l'organisation doit changer n'est pas prédéfini par le concepteur. Les raisons de ce besoin de changement sont identiques aux cas précédents : le comportement du système est inadéquat dans son environnement. La caractéristique essentielle du processus de modification de l'organisation est qu'il n'est pas dirigé par une entité extérieure au système.

La phase de monitoring est réalisée de manière endogène, par un (ou plusieurs) agent(s) du système. Le concepteur leur donne les moyens de détecter au niveau local que l'organisation au niveau global n'est pas adaptée. Pour cela, les agents sont capables de savoir qu'*ils ne sont pas adaptés*. Ensuite, l'agent ayant participé à la détection du problème va exécuter la phase de réparation. Elle consiste en une phase de sélection d'une ou de plusieurs actions parmi les actions possibles, suivie d'une phase d'exécution. L'agent choisit et exécute l'action qu'il juge la plus appropriée selon un critère local – par exemple, la coopération dans [PICARD et GLIZE, 2006]. Cette action est déjà définie et implémentée ou bien elle a été apprise par l'agent [DROGOUL et al., 1995]. C'est toujours en cours de fonctionnement que cette phase est réalisée. Il *réagit* pour changer sa position dans l'organisation/topologie afin d'adapter le système dans sa globalité, ou changer sa spécialisation comportementale [DROGOUL et al., 1995]. Un agent peut décider de manière autonome et isolée, de l'action qui va changer l'organisation – par exemple, en disparaissant du système. Un agent peut aussi décider d'agir de manière coopérative c'est-à-dire en tenant compte des agents qu'il connaît [PICARD et GLIZE, 2006]. Dans un mécanisme auto-construit, un composant recevant un *feedback* négatif doit changer sa pondération afin de changer sa fonction et ses interactions avec son voisinage de manière plus adaptée et ainsi réduire le retour négatif. Les coalitions modifient leurs organisations pour cause de *relations inadaptées entre agents*. De tels changements sont le résultat d'un *processus de raisonnement* basé sur des concepts sociaux, comme les pouvoirs ou les dépendances, manipulés par les agents. Par exemple, dans une coalition de drones de surveillance, les agents changent de leader après que le précédent ait démontré des manques de puissance de calcul ou de communication, afin de maintenir un collectif adapté à la mission [LONG et al.,

		Organisation implicite	Organisation explicite
Quoi?	ACPV	Topologie, Voisinages Poids, influences Différentiation	Dépendances Engagements Pouvoirs
	OCPV	Modèle de conception Agentification	Spécification de l'organisation Affectation des rôles
Qui?	ACPV	Agents, par pression environnementale	Agents, par pression sociale
	OCPV	Concepteur	Concepteur & Agents
Pourquoi?	ACPV	Agents non adaptés	Relations inter-agents non adaptées
	OCPV	Performances non adéquates	Organisation non adaptée
Comment?	ACPV	En réagissant, indirectement sur l'organisation, directement sur l'environnement	En raisonnant, indirectement sur l'organisation, directement sur les schémas de coopération
	OCPV	En programmant, directement l'organisation, directement l'environnement	En organisant, directement l'organisation directement les schémas de coopération

TABLE 2.1 – Aspects de l'adaptation des organisations

2007].

2.4 Réorganisation et auto-organisation

La table 2.1 fournit une vision synthétique structurée suivant la portée du processus d'adaptation (*quoi, qui, pourquoi* et *comment*), la conscience de l'organisation (implicite vs. explicite) et la différenciation entre les deux points de vue ACPV et OCPV. Nous pouvons ainsi proposer une définition des concepts de réorganisation et d'auto-organisation, dans le contexte de la vision synthétique de la Figure 2.1 et de la Table 2.1.

Définition 1 (Réorganisation). La *réorganisation* est un processus endogène ou exogène, qui concerne les systèmes dans lesquels l'organisation est explicitement manipulée au travers de spécifications, de contraintes ou autres moyens. La réorganisation assure un comportement global adéquat, lorsque l'organisation n'est pas adaptée. Les agents étant conscients de l'organisation, ils sont capables de manipuler des primitives afin de modifier leur environnement social. Ce processus peut être à la fois initié par une entité externe au système ou par les agents eux-mêmes, en raisonnant directement sur l'organisation (rôles, spécification organisationnelle) et sur les schémas de coopération (dépendances, engagements, pouvoirs).

Ce processus apparaît ainsi dans le côté droit du quadrant, et concerne donc principalement les systèmes orientés organisation, et, dans une moindre mesure, les coalitions. Nous verrons, dans la suite de ce document, des exemples détaillés de cette approche appliquée à la gouvernance de réseau M2M, ainsi qu'à l'intelligence ambiante (voir chapitres 7 et 8).

Définition 2 (Auto-organisation). L'*auto-organisation* est un processus endogène ascendant concernant les systèmes dans lesquels seules des informations et représentations locales sont manipulées par les agents inconscients de l'état de l'organisation dans sa globalité, afin d'adapter le système à la pression environnementale en modifiant indirectement l'organisation, donc en changeant directement la configuration du système (topologie, voisinages, influences, différenciation), ou l'environnement du système, par des interactions et propagations locales.

Ce processus apparaît en haut du cadran, et concerne donc les SMA à organisation émergente, et dans une moindre mesure les coalitions. Nous présentons également des exemples d'application de cette approche dans nos travaux en résolution de problèmes (voir chapitres 4 et 5).

Nous pouvons donc identifier un continuum entre les systèmes auto-organiseurs et les systèmes réorganiseurs, passant par les systèmes basés sur les coalitions. Réorganisation et auto-organisation sont également des implémentations distinctes d'un même processus générique d'adaptation des organisations : détection et réparation nécessaires à l'adaptation. Du côté réorganisation, ce processus peut être centralisé ou décentralisé, mais toujours explicite et directement réalisé par des entités (concepteur ou agent) manipulant des primitives organisationnelles. Du côté auto-organisation, ce processus est décentralisé, implicite et endogène, laissé à la responsabilité des agents et souvent initié par une pression environnementale. Nous qualifierons d'*auto-réorganisateur* un processus mettant en œuvre réorganisation et auto-organisation.

Deuxième partie

Systemes multi-agents et adaptation dans la résolution de problèmes

Résolution de problèmes et multi-agents : un positionnement

Ce chapitre présente un panorama succinct des approches multi-agents pour la résolution de problèmes, ainsi que le positionnement de nos travaux et contributions, détaillées dans les chapitres suivants, vis-à-vis de l'existant. La section 3.1 précise ce qu'est la résolution de problèmes, et différencie les approches de résolution de problèmes *par* systèmes multi-agents (*moyen*), et la résolution de problèmes issus de la communauté multi-agent (*sujet*). La section 3.2 aborde les principaux cadres et modèles de résolution de problèmes multi-agents, et la section 3.3 présente les différents modes de décomposition de problèmes envisagés. Enfin, la section 3.4 met en évidence les besoins d'adaptation dans ces approches, ainsi que certaines techniques mises en œuvre pour y répondre.

3.1 Résolution *par* ou *pour* les systèmes multi-agents

Les systèmes multi-agents, de part leur décentralisation et leur dimension sociale, sont à la fois (i) un paradigme de modélisation de problèmes exhibant de telles propriétés — donc un *moyen* —, et (ii) un champ d'application avec ses problèmes spécifiques — donc un *sujet* —, pour lesquels des solutions centralisées ne se montrent pas adaptées.

SMA comme moyen. Résoudre un problème consiste à modéliser ce problème dans un cadre (formel ou non) adapté aux spécificités de ce problème, puis à concevoir des techniques de résolution (e.g. des algorithmes ou des protocoles d'interaction) afin de trouver une solution à ce problème, dans ce cadre de modélisation. Les techniques provenant des *mathématiques appliquées* ou de la *recherche opérationnelle* sont les premiers outils vers lesquels se tourner lorsque l'on est face à un problème pour lequel nous cherchons une solution *optimale*, de manière systématique et reproductible. Cependant, ce ne sont pas les seules techniques; *a fortiori* si l'on cherche à résoudre des problèmes distribués, ou pour lesquels la notion d'optimalité n'est pas évidente à formaliser au premier abord — qu'est-ce qu'un comportement optimal? Ainsi, lorsqu'un problème à résoudre exhibe des propriétés de distribution, de décentralisation, ou de dynamique, les systèmes multi-agents peuvent se révéler être une alternative efficace [MÜLLER, 1998 ; M.-P. GLEIZES, 2004 ; YOKOO, 2001 ; SHOHAM et al., 2009]. Dans ce cas, la modélisation va souvent consister à considérer certaines variables, données, critères ou contraintes du problème comme des agents, en les dotant d'autonomie et de capacités d'interaction. C'est ce que nous appelons la phase d'*agentification*.

SMA comme objet. De manière complémentaire, certaines techniques de résolution de problèmes

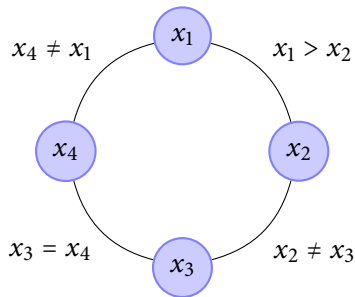


FIGURE 3.1 – Un exemple de DCSP représenté sous forme de graphe variables-contraintes.

ont été conçues à l'origine pour répondre à des problèmes spécifiques à la communauté multi-agent. Nous pensons notamment aux problèmes de formation de coalitions [SHEHORY et al., 1998], d'affectation de ressources [CHEVALEYRE et al., 2005] ou bien à des problèmes de planification multi-agent [DE WEERDT et al., 2009]. Les contributions dans ce domaine vont également se baser sur des travaux provenant de la recherche opérationnelle ou de l'intelligence artificielle classique, et vont la plupart du temps reposer sur des cadres formels éprouvés comme l'optimisation sous contraintes, ou la théorie des jeux [SHOHAM et al., 2009]. Dans ces approches, la phase d'agentification est plus immédiate, car les agents existent *per se*.

Positionnement (Résolution *par* ou *pour* les systèmes multi-agents). Dans nos travaux, nous nous sommes davantage intéressés à proposer des solutions dans lesquelles les systèmes multi-agents sont un *moyen* de résoudre des problèmes réels (en développant des *solveurs* ou des *optimiseurs*), comme le contrôle manufacturier ou l'affectation de fréquences [PICARD et al., 2007b ; CLAIR et al., 2008b]. Nous avons également proposé des cadres génériques de résolution de problèmes distribués et dynamiques [PICARD et GLIZE, 2006 ; GLIZE et PICARD, 2011]. Nos travaux se positionnent ainsi plus du côté *moyen* que *sujet*.

3.2 Cadres de modélisation des problèmes

Les cadres de modélisation offerts par les mathématiques appliquées et la recherche opérationnelle pour la résolution de problèmes consistent essentiellement en la définition d'*objectifs globaux* à atteindre. Ces objectifs sont le plus souvent exprimés comme une maximisation ou une minimisation de certaines propriétés du système (e.g. trouver la configuration d'un avion qui minimise sa masse, ou trouver le plus court chemin). Ces données peuvent également être sujettes à des physiques propres au domaine du problème, qui seront par exemple traduites par des *contraintes* sur les valeurs définissant l'état du système. Nous pouvons alors nous placer dans des cadres formels bien établis comme les *problèmes de satisfaction de contraintes* ou les *problèmes d'optimisation sous contraintes* en fonction de la nature du problème à résoudre [DECHTER, 2003 ; YOKOO, 2001]. Dans le cas de problèmes dont les contraintes sont binaires (ou peuvent être transformées en contraintes binaires) il est possible de représenter les problèmes comme des graphes dont les nœuds sont les agents, et les arêtes sont les contraintes liant ces agents, comme l'illustre la figure 3.1. Il est intéressant de noter qu'*un problème peut ainsi correspondre à une topologie, et donc une organisation* (cf. section 2.2). Les agents interagissent avec les agents avec lesquels ils partagent des contraintes.

Algorithmes pour les problèmes distribués. La communauté multi-agent a principalement proposé les modélisations par *distributed constraint satisfaction problems* (DCSP) et par *distributed constraint optimisation problems* (DCOP) [MODI et al., 2005] comme cadre de résolution de problèmes distribués. Les algorithmes proposés dans ces approches consistent essentiellement en

des protocoles synchrones ou asynchrones implantant des stratégies de recherche dans les espaces d'états issues de la recherche opérationnelle ou de l'intelligence artificielle (e.g. *backtracking*, *breakout*) [YOKOO, 2001]. À titre d'exemple, voici une formalisation de DCOP, dont les algorithmes de résolution (complets ou approchés) célèbres sont ADOPT [MODI et al., 2005], DPOP [PETCU, 2007], MGM [MAHESWARAN et al., 2004] ou DSA [ZHANG et al., 2005].

Exemple. Un DCOP peut être défini comme un tuple $\langle A, V, \mathcal{D}, f, \alpha, \eta \rangle$, où :

- A est un ensemble d'agents;
- V est un ensemble de variables $\{v_1, v_2, \dots, v_{|V|}\}$;
- \mathcal{D} est un ensemble de domaines de valeurs $\{D_1, D_2, \dots, D_{|V|}\}$, où chaque $D \in \mathcal{D}$ est un ensemble fini contenant les valeurs auxquelles peuvent être assignées les variables ;
- f est une fonction

$$f : \bigcup_{s \in \mathfrak{P}(V)} \sum_{v_i \in S(\{v_i\} \times D_i)} \leftarrow \mathbb{N} \cup \{\infty\}$$

qui associe un coût à chaque assignation de variable possible ;

- α est une fonction $\alpha : V \rightarrow A$ associant les variables à leur agent. $\alpha(v_i) \mapsto a_j$ signifie que c'est l'agent a_j qui a la responsabilité de trouver une valeur pour la variable v_i
- η est un opérateur qui agrège tous les coût individuels f pour toutes les affectations possibles. C'est généralement une somme :

$$\eta(f) \mapsto \sum_{s \in \bigcup_{v_i \in S(\{v_i\} \times D_i)} \mathfrak{P}(V)} f(s).$$

L'objectif d'un DCOP est de trouver une affectation de valeur pour chaque agent qui minimise ou maximise $\eta(f)$.

Ce cadre a été la source de nombreux travaux, mais il fait une *hypothèse souvent difficile à garantir* : l'objectif global est une agrégation (souvent une simple somme) des états des agents. De plus, dans sa formalisation même, ce cadre présuppose une décomposition du problème par variables (cf. section 3.3). Ainsi, bien qu'offrant un cadre formel et des solutions efficaces, la phase de modélisation, de passage d'un problème réel à un DCOP, peut impliquer l'injection d'hypothèses de travail trop restrictives. Il peut alors être intéressant de se tourner vers des méthodes de résolution approchées, par émergence [M.-P. GLEIZES, 2004 ; PICARD et GLIZE, 2006], si le problème attaqué rentre difficilement dans de cadre DCOP.

Coordination via l'environnement. D'autres approches, basées sur une coordination via un environnement virtuel, ont été développées pour résoudre des problèmes distribués. Prenant inspiration des approches par stigmergie [DORIGO et al., 2004], ces méthodes transforment le problème de résolution en un problème de coordination ; ceci par le biais d'un environnement et de primitives de modification de cet environnement. Les agents agissent sur cet environnement pour assigner des valeurs et interagissent via cet environnement afin de garantir la cohérence de la solution. Des approches comme ERA pour les DCSP [LIU et al., 2002] ou ETTO pour les emplois du temps distribués [PICARD, BERNON et M.-P. GLEIZES, 2005a] en sont de bons exemples. Dans ERA, les agents évoluent dans une grille représentant l'espace de recherche et déposent des marqueurs pour signaler les conflits éventuels avec les valeurs choisies par les autres agents, comme l'illustre la figure 3.2. Ces travaux peuvent également être classés comme des algorithmes de résolution de DCSP/DCOP vus précédemment, mais n'explicitent pas la fonction globale pour déterminer le comportement des agents.

Théorie des jeux et choix social. L'économie et les mathématiques ont également apporté un cadre de modélisation de problèmes multi-agents qui est la théorie des jeux [SHOHAM et al., 2009]. Ici,

L'idée est de concevoir des comportements optimaux d'agents, à partir d'observations, d'anticipations, et de modélisation des actions des autres agents. Cette théorie présente une typologie de jeux riche, correspondant à de nombreux problèmes réels (e.g. jeux coopératifs et jeux non coopératifs pour la formation de coalitions). Ici encore, afin d'évaluer la solution globale, une fonction d'agrégation peut-être utilisée, comme dans la théorie du choix social [MOULIN, 2003]. Cependant, une difficulté majeure avec une telle modélisation est de définir, de limiter ou de prendre en compte tous les facteurs et variables influençant les stratégies et les gains. Des approches, stochastiques, permettent de passer outre certaines de ces limitations [RAJNARAYAN et al., 2006].

Positionnement (Cadre de modélisation). Dans nos travaux, nous avons principalement adopté une *approche de résolution de problèmes sous contraintes* (sans forcément recourir à une fonction objectif globale, mais plutôt par émergence), et avons comparé nos résultats avec des algorithmes issus de la communauté DCSP/DCOP [GLIZE et PICARD, 2011 ; CLAIR et al., 2008b ; PICARD et al., 2007b ; PICARD et GLIZE, 2006]. Cependant, nous avons également attaqué des problèmes d'affectation sous l'angle du choix social, sans toutefois aborder la dimension dynamique de ces problèmes [EVERAERE et al., 2013 ; EVERAERE et al., 2012 ; PIETTE et al., 2013].

Choisir un cadre de modélisation du problème n'est cependant pas suffisant pour développer un système multi-agent pour résoudre ce problème. Il est ensuite nécessaire d'identifier les agents du système, puis les techniques et mécanismes de coordination à mettre en place pour garantir le comportement du système.

3.3 Stratégies d'agentification

Nous pouvons identifier trois familles d'approches multi-agents pour la résolution de problèmes, qui correspondent à trois stratégies d'*agentification* distinctes des problèmes, car, dans les faits, plusieurs choix de modélisation peuvent être faits pour développer un solveur multi-agent, ou pour résoudre un problème multi-agent.

Recherche par population d'agents ou division du travail. Chaque agent représente un point de l'espace de recherche (chaque agent fournit *une affectation complète*). Dans ce cas les agents explorent de manière distribuée le même espace de recherche mais avec des stratégies ou des points de vue différents, en se déplaçant de point en point. Ici, la solution au problème est l'état (la position, la valeur) d'un ou plusieurs agents en fin d'exécution du système multi-agent. L'optimisation par essaim particulière [KENNEDY et al., 1995] et l'optimisation par colonie de fourmis [DORIGO et al., 2004] appartiennent à cette famille. La figure 3.3 illustre cette notion de division du travail par un autre exemple qui est le partitionnement de l'espace de recherche.

x_1	1	2	1	1	0	0
x_2	0	1	0	0		
x_3	0	1	1	1	1	

FIGURE 3.2 – Un exemple de DCSP représenté sous forme d'environnement marqué. Les cellules représentent les valeurs possibles pour chacune des variables (en ligne). Les cercles représentent les positions (donc les valeurs) courantes des agents. Les nombres dans les cellules représentent le nombre de conflits sur une affectation donnée [LIU et al., 2002].

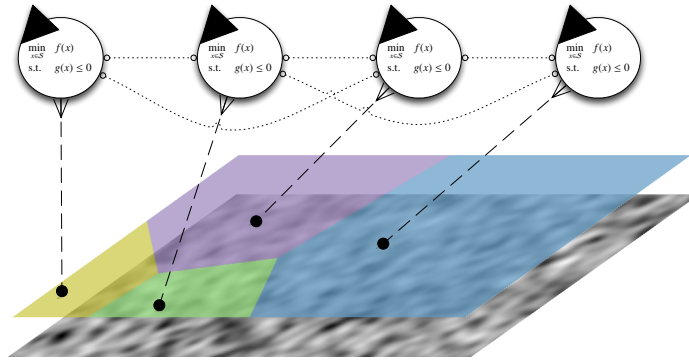


FIGURE 3.3 – Exemple de division du travail par partitionnement de l'espace de recherche en optimisation sous contrainte [PICARD, VILLANUEVA et al., 2013]. Les agents ont tous le même objectif qui correspond à l'objectif global limité à leur sous-région.

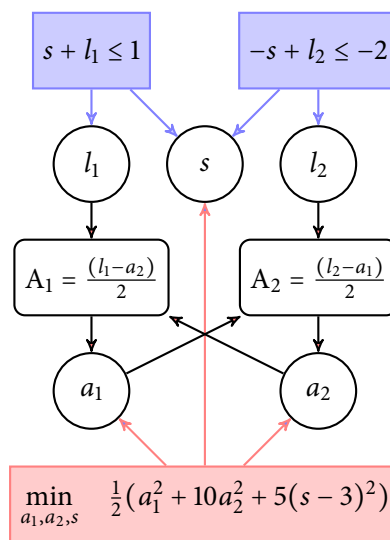


FIGURE 3.4 – Agentification d'un problème d'optimisation multi-disciplinaire (MDO) par AMAS [JORQUERA, 2013]. Les concepts de variables (cercles), de contraintes (rectangles bleus), d'objectifs (rectangle rouge) et de disciplines (rectangles arrondis) sont agentifiés.

Décomposition du problème par dimensions et critères. Dans ce cas le problème global est décomposé en plusieurs sous-problèmes affectés aux agents (chaque agent fournit une *affectation partielle*). Un problème peut classiquement être décomposé par dimensions — e.g. les variables comme dans [MODI et al., 2005 ; RAJNARAYAN et al., 2006] — ou par critères — e.g. dans les approches basées sur la théorie des jeux pour l'optimisation multicritère [SHOHAM et al., 2009 ; RAJNARAYAN et al., 2006]. Certaines approches poussent le concept plus loin en agentifiant à la fois les variables *et* les critères [JORQUERA, 2013], comme l'illustre la figure 3.4.

Décomposition par modélisation *ad-hoc* du domaine. Ici, les agents représentent des entités spécifiques au domaine du problème, et ne résolvent pas directement des concepts de résolution de problèmes (contraintes, critères, objectifs, etc.) [CLAIR et al., 2008b]. Par exemple, modéliser l'affectation de ressources en contrôle manufacturier peut être effectué par une agentification des machines, des conteneurs ou des opérateurs [BUSSMANN et al., 2004 ; MAVRÍK et al., 2007 ; CAPERA, BERNON et al., 2006]. Ces approches mêlent souvent les deux approches précédentes. Par exemple, un agent est responsable d'un ensemble de variables *et/ou* de régions de l'espace de recherche. D'autres approches font également coopérer plusieurs solveurs pour améliorer la convergence du processus de résolution [DANOY et al., 2010], ce qui les rendent très proches des approches de parallélisation de la recherche [TALBI, 2006].

Quelle que soit l'approche choisie, l'implémentation du système nécessite également de définir une politique de partage d'informations, de *coordination* (e.g. échanger des points et leur évaluation). Ceci va souvent se traduire par des règles de *cohérence* des états de différents agents, ou en des *protocoles* de communication, basés sur des topologies spécifiques (voisinages, arbres, pseudo-arbres, etc.).

Positionnement (Stratégie d'agentification). Lors de nos travaux, nous avons abordé des problèmes suivant *ces trois approches*. Par exemple, dans la résolution de problèmes de contrôle manufacturier nous avons développé une approche basée sur ABT (DCSP), donc basée sur une *décomposition par dimension*, mais également une *approche plus* ad-hoc [CLAIR et al., 2008b]. Nous avons également développé des optimiseurs dans le cadre de l'optimisation par métamodélisation basés sur des *populations* d'agents [RIVIÈRE et al., 2013 ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2013b].

3.4 Adaptation dans la résolution

L'adaptation, au sens où nous l'avons abordée dans le chapitre 2, peut intervenir pour deux raisons : (i) lorsque l'organisation est le résultat du processus de résolution (l'adaptation est le moyen d'explorer l'espace de recherche) ou (ii) lorsque l'organisation est un moyen de modéliser le problème et que ce dernier change au cours de l'exécution (l'adaptation est un moyen de répondre à la dynamique de l'environnement qui modifie le problème en tant que tel).

Adaptation pour explorer l'espace de recherche. Dans ce cas, l'organisation est modifiée ou adaptée par le processus de recherche d'une solution. L'organisation finale *est* un résultat au problème posé. Ceci est très commun dans les approches auto-organisées, comme les approches par stigmergie [DORIGO et al., 2004]. En effet, le processus d'exploration va créer des artefacts environnementaux, supports de l'organisation des agents. Le résultat de l'activité du système est alors une organisation *stable* représentant une solution adéquate au problème (e.g. le chemin le plus court pour aller du nid à une source de nourriture). On retrouve également ce genre de dynamique de résolution dans les approches de résolution par AMAS. Par exemple, dans un système de conception de mécanismes aéronautiques, les composants agentifiés du mécanisme vont s'auto-organiser spatialement pour obtenir une configuration optimale [CAPERA, M. GLEIZES et al., 2004]. D'autres méthodes consistent à partitionner l'espace de recherche de manière dynamique — l'organisation étant synonyme de partitionnement — afin de diviser l'espace de travail au fur et à mesure des découvertes des agents [CALVEZ et al., 2008 ; PICARD, VILLANUEVA et al., 2013]. L'organisation spatiale (le partitionnement, la topologie, les interactions possibles entre les agents) va résulter de ce processus, sans guidage par un objectif explicite global. Ces approches se positionnent en général parmi les approches ACPV, dans lesquelles l'organisation est implicite et non manipulée directement par les agents (cf. section 2.2).

Adaptation pour répondre à la dynamique de l'environnement. Dans certains cas d'application, le système doit être capable de fournir des solutions, alors que l'environnement et/ou l'organisation interne du système sont en train de changer. Par exemple, dans le cas du contrôle manufacturier, le système doit être capable de replanifier une affectation de ressources suite à l'arrivée d'une commande prioritaire, ou bien à la suite d'une panne machine. On parle alors d'adaptation à la dynamique. Ici, donc, suite à une phase de *monitoring*, le système détecte son inadaptation, et effectue une phase de réparation (cf. section 2.3.3). Ces phases peuvent bien sûr être effectuées par les agents eux-mêmes, ou par une entité externe. Dans le cadre des DCOP, plusieurs algorithmes ont été proposés, et reposent principalement sur l'ajout de nouveaux types de messages, et des traitements associés, pour prendre en compte l'ajout de variables ou de contraintes (ouverture du système), afin de ne pas reprendre l'exploration de l'espace de recherche à partir de zéro [MAILLER et V. LESSER, 2004 ; HATTORI et al., 2006]. Ce sont plutôt des approches dans lesquelles

l'organisation est explicite (cf. section 2.2.2), et dans lesquelles les phases de réparation sont centralisées. D'autres approches, basées sur des approches plus locales (comme l'auto-organisation) de propagation de valeurs de voisinage en voisinage sont, par essence, plus adaptées pour répondre à la dynamique de l'environnement (e.g. arrivée d'une nouvelle commande dans l'usine), tout en diminuant l'impact de la réparation sur le reste du système. On retrouve notamment tous les algorithmes de recherche locale distribuée [HIRAYAMA et al., 2005 ; ZHANG et al., 2005] ou les approches par auto-organisation [PICARD et al., 2007b ; CLAIR et al., 2008b], dans lesquelles l'organisation est explicite.

Positionnement (Adaptation dans la résolution). Nous avons principalement proposé des techniques de résolution de problèmes par *auto-organisation pour répondre à la dynamique* de l'environnement, notamment dans le domaine du contrôle manufacturier [CLAIR et al., 2008b] ou la gestion d'emplois du temps [PICARD, BERNON et M.-P. GLEIZES, 2005a], mais également dans le cadre plus générique des DCSP [GLIZE et PICARD, 2011 ; PICARD et GLIZE, 2006]. Nous avons également proposé des méthodes de partitionnement adaptatif de l'espace de recherche pour l'optimisation par métamodélisation, dans le cadre du projet ID4CS [PICARD, VILLANUEVA et al., 2013 ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2013b], ou de détection de multiples optima locaux [RIVIÈRE et al., 2013]. Ces deux approches reposent sur des populations d'agents se *divisant l'espace de recherche en cours de résolution* pour minimiser le nombre d'appel à des fonctions coûteuses.

Nous allons présenter dans les deux chapitres suivants nos contributions dans le domaine de la résolution de problèmes. Le chapitre 4 présente des méthodes dans lesquelles l'adaptation permet d'explorer l'espace de recherche, alors que le chapitre 5 aborde des méthodes de résolution par auto-organisation.

4 Adaptation dans l'exploration par population d'agents

Ce chapitre présente nos travaux dans le domaine de la résolution de problèmes — plus précisément en optimisation continue. Ici le système multi-agent consiste en une *population* d'agents explorant l'espace de recherche, par division du travail (voir section 3.3). Le but d'une telle division est double : (i) répartir l'effort calculatoire de la recherche (ces algorithmes sont parallélisables, bien que synchrones) et (ii) améliorer la compréhension du problème, en fournissant, par exemple, un partitionnement de l'espace associé à des métamodèles spécifiques à chaque sous-région. Ici l'adaptation permet d'explorer l'espace de recherche de telle sorte que l'organisation finale du système est la solution (voir section 3.4).

La section 4.1 présente brièvement le cadre de l'optimisation par métamodélisation, qui est le contexte technique des travaux présentés dans ce chapitre. La section 4.2 présente un travail préliminaire dans lequel plusieurs agents-métamodèles vont concourir pour trouver la meilleure représentation de l'espace de recherche à des fins d'optimisation sous incertitudes. La section 4.3 présente nos travaux sur le partitionnement adaptatif de l'espace de recherche dans le cadre de l'optimisation par métamodélisation. Ces travaux ont donné suite à une approche alternative de métamodélisation adaptative, sans partitionnement, mais toujours dans le cadre de l'optimisation par appel à des simulateurs coûteux, présentés en section 4.4.

4.1 Optimisation et métamodélisation

En ingénierie (et notamment en mécanique), les problèmes d'optimisation complexes, comme la conception d'avion, requièrent de nombreuses simulations coûteuses en calcul, comme la résolution des équations de la dynamique des fluides autour de l'avion. Ces simulations consistent en des appels à des logiciels qui, à un choix des variables de conception (un « point » dans l'espace de ces variables), associent des valeurs de performance de l'avion (e.g. la masse de l'avion simulé, son rayon d'action, son temps d'atteinte d'altitude de croisière, ...). De nombreux chercheurs ont contribué au développement de méthodes d'optimisation adaptées à ces simulations coûteuses. Les idées qui sous-tendent ces méthodes sont principalement (i) l'utilisation de *métamodèles* (ou approximations, ou *surrogates*) qui remplacent certains appels aux simulateurs coûteux par des appels à des fonctions rapides *appries* sur ces simulations, (ii) la décomposition du problème en des problèmes de plus petite taille, (iii) l'approximation du problème par des problèmes plus simples à résoudre et (iv) la parallélisation des calculs.

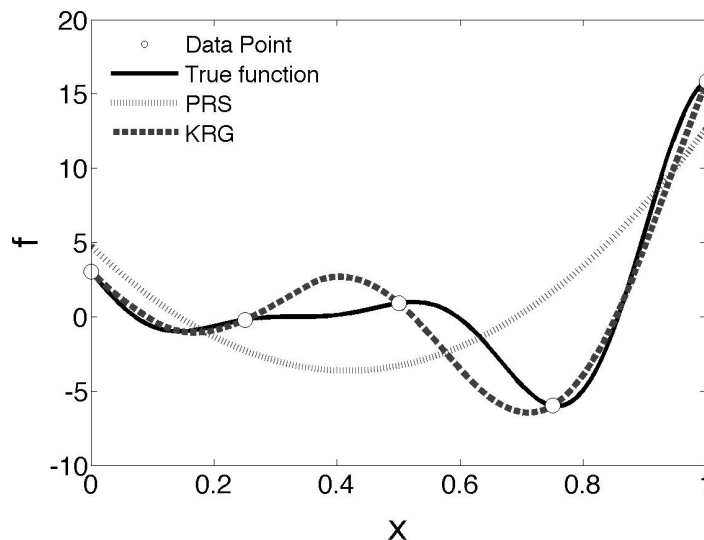


FIGURE 4.1 – Deux exemples de métamodèles pour une fonction à une dimension (ligne pleine) et pour un plan d'expérience de cinq points (cercles blancs) : surface de réponse quadratique (PRS) et krigeage (KRG).

4.1.1 Métamodèles et apprentissage

Un métamodèle est une fonction mathématique (i) approximant les réponses du modèle étudié (par exemple, le modèle représentant la masse d'un avion en fonction de ses différentes dimensions), construite le plus souvent à partir d'un ensemble de taille restreinte d'entrées-sorties de ce modèle (ii) de coût de calcul négligeable et (iii) ayant pour objectif de prédire de nouvelles réponses dans une partie de l'espace des entrées (éventuellement tout cet espace) [KLEIJEN, 2008].

L'ensemble des points initiaux est appelé plan d'expérience (ou *design of experiments, DOE*). Des exemples connus de métamodèles sont les surfaces de réponse polynomiales, les splines, les réseaux de neurones ou le krigeage. La préoccupation au sujet du coût est due au fait que, dans de nombreux domaines, le modèle étudié est un simulateur coûteux, pouvant nécessiter plusieurs heures (voire jours) de calcul pour produire un résultat. Un plan d'expérience et/ou la production de nouveaux points peuvent également être le résultat de mesures physiques réelles sur des prototypes ou des objets physiques existants. La figure 4.1 présente des exemples de métamodèles pour une fonction à une dimension et un plan d'expérience réduit. On voit ici que, pour un modèle et un sous-espace donnés, différents métamodèles peuvent avoir différents degrés de précision. L'erreur d'un métamodèle peut être mesurée; par exemple, en utilisant la somme des carrés des erreurs de prédiction.

L'utilisation de métamodèles pour remplacer des simulations et expérimentations coûteuses en optimisation a été largement étudiée [VILLANUEVA, PICARD et al., 2012]. De nombreux travaux ont proposé des stratégies d'utilisation de multiples métamodèles (*multi-surrogate*) pour l'optimisation, dont nous nous sommes inspirés sous le prisme du paradigme multi-agent. La métamodélisation est également une technique utilisée en *apprentissage automatique statistique* — les machines à vecteur de support et le krigeage en sont de bons exemples.

4.1.2 Optimisation sur métamodèles

Considérons la formulation générique d'un problème d'optimisation sous contraintes présentée dans l'équation (4.1). f et g sont des fonctions et x un vecteur de variables.

$$\begin{aligned} & \underset{x \in S}{\text{minimiser}} && f(x) \\ & \text{avec} && g(x) \leq 0 \end{aligned} \tag{4.1}$$

Algorithme 1 : Optimisation par métamodélisation

```

1  $t \leftarrow 1$  (état initial)
2 tant que  $t \leq t^{max}$  faire
3   Construire  $\hat{f}$  et  $\hat{g}$  à partir de  $(\mathbb{X}^t, \mathbb{F}^t, \mathbb{G}^t)$ 
   // Résoudre le problème (4.2) avec un algorithme d'optimisation interne:
4    $\hat{x}^* \leftarrow \underset{x \in \mathcal{S}}{\operatorname{argmin}} \hat{f}(x)$ , avec  $\hat{g}(x) \leq 0$ 
   // Appel au simulateur coûteux :
5   Calculer  $f(\hat{x}^*)$  et  $g(\hat{x}^*)$ 
   // Mettre à jour la base :
6    $(\mathbb{X}^{t+1}, \mathbb{F}^{t+1}, \mathbb{G}^{t+1}) \leftarrow (\mathbb{X}^t, \mathbb{F}^t, \mathbb{G}^t) \cup (\hat{x}^*, f(\hat{x}^*), g(\hat{x}^*))$ 
7    $t \leftarrow t + 1$ 

```

Par exemple, dans le cas de la conception d'un avion, x désigne la géométrie et les composants de l'avion, g des performances minimales indispensables à réaliser telles qu'une altitude de croisière, une masse embarquée, un rayon d'action, et f pourrait être une quantité de carburant consommée. À un x donné, un ou plusieurs programmes de simulation coûteux numériquement doivent être appelés. En optimisation par métamodèle, un métamodèle est construit à partir d'un plan d'expérience, noté \mathbb{X} , qui consiste en un ensemble de points pour le vecteur de variables x . Pour le plan d'expérience, nous disposons des valeurs calculées de la fonction objectif f et des contraintes g pour ces points, notées \mathbb{F} et \mathbb{G} , respectivement. Nous nous référons à \mathbb{X} et ses valeurs associées \mathbb{F} et \mathbb{G} comme une base de données (ou de croyances, lorsque intégrée à un agent). Cette base est utilisée pour construire une approximation par métamodèle de la fonction objectif, notée \hat{f} , et \hat{g} pour les contraintes. Nous pouvons approcher le problème de l'équation (4.1) en utilisant les métamodèles comme suit :

$$\begin{aligned} & \underset{x \in \mathcal{S}}{\operatorname{minimiser}} \quad \hat{f}(x) \\ & \text{avec} \quad \hat{g}(x) \leq 0 \end{aligned} \tag{4.2}$$

On note \hat{x}^* la solution de ce problème approché.

L'optimisation par métamodèle classique s'approche itérativement de la solution de (4.1). Elle est par conséquent dépendante du nombre d'itérations t^{max} , qui correspond au budget d'appels à la fonction coûteuse. Une fois l'optimum, \hat{x}^* , du problème de l'équation (4.2) trouvé, les vraies valeurs de f et g associées sont calculées et insérées dans la base pour la prochaine itération. Ensuite, le métamodèle est mis à jour et l'optimisation (4.2) est recommencée. Ainsi, au temps t , nous notons \mathbb{X}^t la base et l'ensemble des valeurs de fonction objectif et de contraintes associées \mathbb{F}^t et \mathbb{G}^t , respectivement. La procédure d'optimisation par métamodèle peut ainsi être résumée par l'algorithme 1.

4.2 Diversification de métamodèles

Afin d'exploiter les capacités de distribution et d'interaction des agents, nous proposons de concevoir un système multi-agent faisant évoluer plusieurs points de vue sur tout l'espace d'état. Les agents se construisent une vision personnelle de l'espace d'état — un métamodèle — et interagissent afin de déterminer le meilleur modèle, toujours dans l'objectif d'effectuer une optimisation sur cet espace métamodélisé. Ceci consiste en une *diversification* des métamodèles.

4.2.1 Principe

Une première intuition est de concevoir des agents exécutant chacun l'algorithme 1 de manière parallèle, avec des métamodèles distincts, et partageant des points afin d'affiner leur propre métamo-

dèle, afin de trouver quel métamodèle représente le mieux l'espace de recherche, au lieu de faire appel à une connaissance experte *a priori*. Ceci permet à chaque agent de se construire sa propre vision du problème, grâce à une base de croyance (des points) interne et au type de métamodèle utilisé pour approcher \hat{f} . Par exemple, dans un système à trois agents, l'agent a_1 peut utiliser une surface de réponse quadratique, l'agent a_2 du krigeage, et l'agent a_3 une régression à vecteurs de support. Les agents peuvent interagir en partageant, en diffusant et en recevant des résultats pour mettre à jour leur base de croyance interne.

4.2.2 Agents-métamodèles

Ce comportement basique est présenté dans l'algorithme 2, qui est basé sur l'algorithme 1 en ajoutant les lignes 6 et 7.

Algorithme 2 : Comportement d'un agent i , par diversification

```

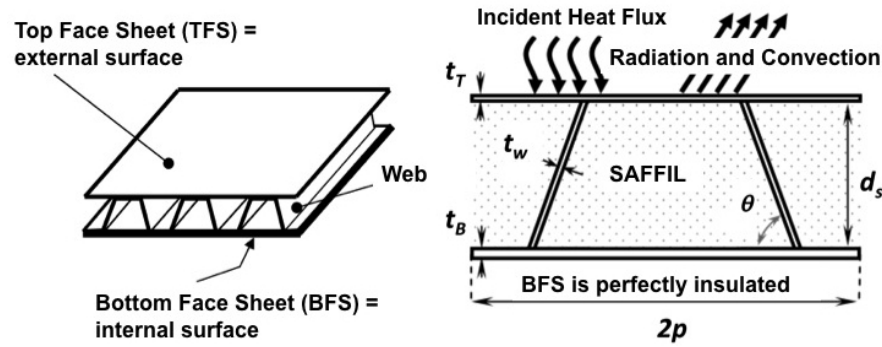
1  $t \leftarrow 1$  (état initial)
2 tant que  $t \leq t^{max}$  faire
3   Construire  $\hat{f}$  et  $\hat{g}$  à partir de  $(\mathbb{X}^t, \mathbb{F}^t, \mathbb{G}^t)$ 
   // Résoudre le problème (4.2) avec un algorithme d'optimisation interne:
4    $\hat{x}^* \leftarrow \underset{x \in \mathcal{S}}{\operatorname{argmin}} \hat{f}(x)$ , avec  $\hat{g}(x) \leq 0$ 
   // Appel au simulateur coûteux :
5   Calculer  $f(\hat{x}^*)$  et  $g(\hat{x}^*)$ 
6   Diffuser les résultats
7   Recevoir les résultats
   // Mettre à jour la base :
8    $(\mathbb{X}^{t+1}, \mathbb{F}^{t+1}, \mathbb{G}^{t+1}) \leftarrow (\mathbb{X}^t, \mathbb{F}^t, \mathbb{G}^t) \cup (\hat{x}^*, f(\hat{x}^*), g(\hat{x}^*))$ 
9    $t \leftarrow t + 1$ 

```

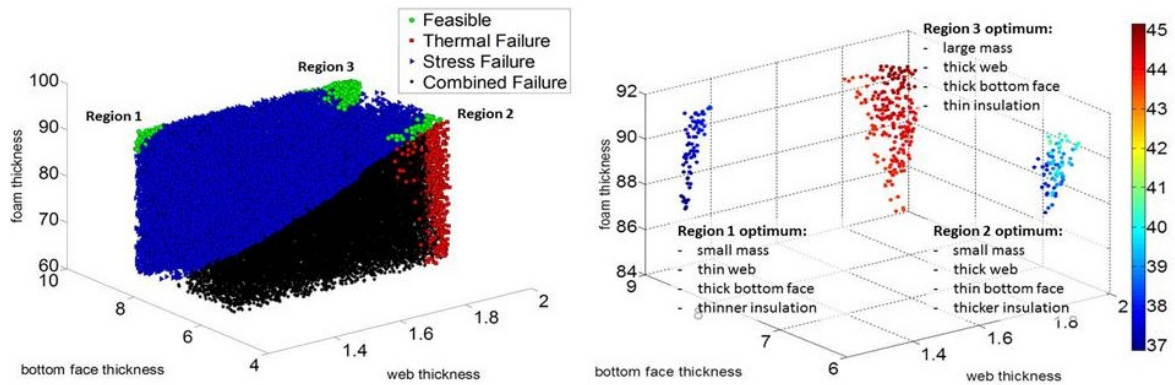
L'organisation des agents consiste en un réseaux connecté en fonction des politiques de partage (aucun partage, broadcast, envoi sélectif par type) des agents. Nous appelons cette organisation d'agents une *diversification* car les agents tentent de résoudre le même problème avec diverses méthodes et représentations (bases de croyance et métamodèles).

Applications et résultats. Cette approche a été appliquée à l'optimisation sous incertitudes (donc par appel à une simulation coûteuse Monte Carlo) d'un bouclier thermique pour véhicule spatial (voir figure 4.2) [VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2012b]. Différentes politiques de diffusion ont été envisagées. Ces dernières ont donné lieu à des résultats montrant qu'il était plus intéressant de partager des points provenant de certains métamodèles plus que d'autres, et que l'erreur moyenne d'un résultat obtenu par le système multi-agent était inférieure à celle obtenue par optimisation séparée sur chaque métamodèle, pour un budget équivalent d'appels à la fonction coûteuse.

Néanmoins, cette approche intuitive fixe *a priori* la politique de partage des agents (et donc l'organisation), ce qui demande une adaptation *ad-hoc* pour chaque problème. Les travaux présentés dans la suite de ce chapitre répondent à cette limitation en proposant des mécanismes d'adaptation de l'organisation opportuniste, en cours d'optimisation.



(a) L'ITPS fournit à la fois l'isolation et la capacité de charge, et par conséquent, peut conduire à des optima alternatifs pour une masse égale, avec différentes manières d'aborder les exigences thermiques et structurales.



(b) Les régions admissibles 1-3 et les régions non admissibles (à gauche), et la fonction objectif (masse) représentée par la couleur (à droite). Les deux optima concurrentiels dans les régions 1 et 2 s'appuient sur des concepts différents. Dans la région 1, la fonction thermique est satisfaite par une feuille épaisse de face inférieure servant de dissipateur de chaleur, tandis que dans la région 2, l'isolation est obtenue par une épaisse couche de mousse.

FIGURE 4.2 – Un exemple de problème de conception, le système de protection thermique intégré (ITPS), posédant plusieurs optima (compromis).

4.3 Partitionnement auto-adaptatif

Afin de répondre aux limitations de l'approche multi-métamodèles présentée dans la section précédente, nous avons proposé une approche *centrée agent* par partitionnement adaptatif de l'espace de conception [VILLANUEVA, PICARD et al., 2012 ; PICARD, VILLANUEVA et al., 2013 ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2013b].

4.3.1 Principe général

Chaque agent est responsable de l'exploration d'une sous-région. Cette exploration consiste en une optimisation globale dans cette sous-région en utilisant le métamodèle le plus précis (en fonction des erreurs de métamodèle calculées tout au long du processus d'optimisation). Un agent possède une bibliothèque de métamodèles qu'il va pouvoir utiliser (e.g. surface de réponse polynomiale ou krigeage). Un agent va optimiser la fonction métamodèle notée \hat{f} sous les contraintes \hat{g} dans son sous-espace, au lieu d'optimiser la fonction réelle f sous les contraintes réelles g dans tout l'espace. La tâche d'optimisation d'un agent i au temps t de la fonction \hat{f} sous les contraintes \hat{g} dans sa sous-région \mathcal{P}_i en fonction de sa base de croyances \mathbb{X}^t est notée $\text{optim}(\hat{f}, \hat{g}, \mathbb{X}^t, \mathcal{P}_i)$, ce qui correspond au problème d'optimisation suivant :

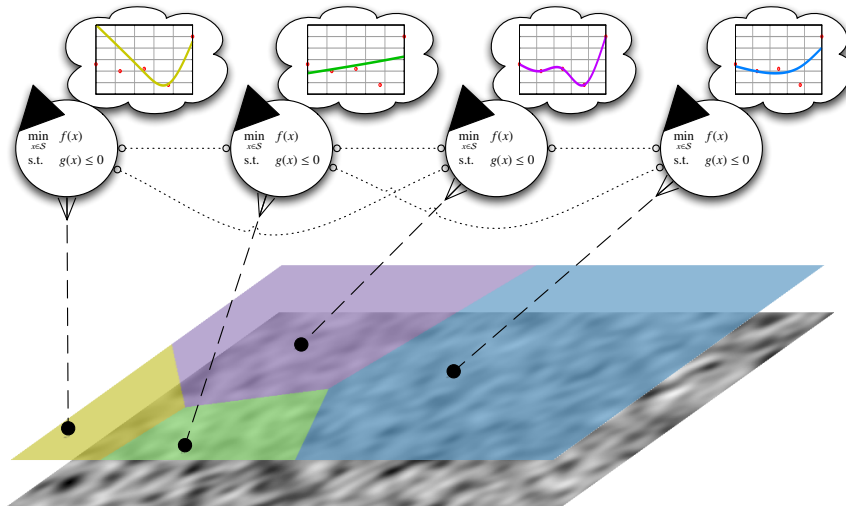


FIGURE 4.3 – Les agents-métamodèles optimisent une fonction commune sur une sous-région de l'espace de recherche (ici en 2D) suivant un métamodèle personnel, et communiquent des points à leurs voisins directs.

$$\begin{aligned} & \underset{x \in \mathcal{P}_i}{\text{minimiser}} && \hat{f}(x) \\ & \text{avec} && \hat{g}(x) \leq 0 \end{aligned} \quad (4.3)$$

Les sous-régions au temps t sont simplement des cellules de Voronoï [AURENHAMMER, 1991] dont les centres sont le meilleur point de chaque agent au temps t (l'optimum de la sous-région i , noté c_i). Une cellule contient plusieurs points déjà calculés et constitue la base de croyance de l'agent responsable de la sous-région. La base de croyance de l'agent i au temps t est notée $(\mathbb{X}^t, \mathbb{F}^t, \mathbb{G}^t)_i$. Deux caractéristiques vont changer pour chaque agent : le centre (donc les frontières des sous-régions) et le métamodèle utilisé pour approcher la fonction coûteuse. Les cellules et les métamodèles s'adaptent donc à l'espace de conception. Nous parlons alors d'optimisation par partitionnement adaptatif.

La figure 4.3 illustre cette approche (cf. section 3.3). Comme les frontières des cellules changent avec le temps, les agents se communiquent les points calculés afin de mettre à jour leurs métamodèles respectifs, mais uniquement sur leurs sous-régions respectives.

4.3.2 Agents-métamodèles

Le comportement suivi par chaque agent peut être résumé par l'algorithme 3 qui est une extension de l'algorithme 2. Ici encore, notre méthode d'optimisation est une méthode pas-à-pas, dans laquelle les agents agissent en parallèle puis attendent avant de passer au pas suivant, ce qui permet, entre autres, de paralléliser les appels aux fonctions coûteuses. La terminaison est paramétrée par un nombre de pas maximum, dépendant du budget d'appels au simulateur coûteux.

En supposant que les sous-régions sont déjà définies, chaque agent choisit le métamodèle le plus précis pour sa sous-région et sa base de croyance (lignes 5-9). Il est possible que l'agent demande des points à ses voisins s'il n'en a pas assez pour ajuster les métamodèles (lignes 5-7). La précision d'un métamodèle est déterminée par une mesure d'erreur par validation croisée, ici la somme des carrés des erreurs de prédiction VILLANUEVA, PICARD et al. [2012]. Une fois le métamodèle le plus précis choisi (ligne 9), l'agent cherche l'optimum dans sa sous-région en utilisant ce métamodèle (ligne 10). En cas d'obtention d'un point proche d'un point déjà connu ou d'un point non admissible, l'agent va explorer la sous-région. Ceci se traduit par le fait que l'agent ajoute un point dans la base de croyance qui maximise la distance minimum aux points déjà présents dans la base, et dans ce cas ce point remplace \hat{x}^* (lignes 11-12). L'appel au simulateur coûteux est alors effectué pour \hat{x}^* (ligne 13) et un nouveau point $(\hat{x}^*, f(\hat{x}^*), g(\hat{x}^*))$ est ajouté à la base de croyances (ligne 14). Ensuite, l'agent va repositionner

Algorithme 3 : Comportement d'un agent i , par partitionnement

```

1 t = 1 (état initial)
2 tant que t ≤ tmax faire
3   Mettre à jour sa partition :  $\mathcal{P}_i \leftarrow \{x \in \mathcal{S} \text{ s.t. } \|x - c_i\|^2 \leq \|x - c_j\|^2, j \neq i\}$ 
4   Mettre à jour sa base de croyances
5   si Pas suffisamment de points dans la base de croyance pour construire les métamodèles alors
6     Demander des points aux agents voisins
7     Ajouter les points à la base de croyances
8   Construire les métamodèles  $\hat{f}$  et  $\hat{g}$  à partir de  $(\mathbb{X}^t, \mathbb{F}^t, \mathbb{G}^t)_i$ 
9   Choisir le métamodèle le plus précis
10  Optimiser dans sa sous-région :  $\hat{x}^* \leftarrow \text{optim}(\hat{f}, \hat{g}, \mathbb{X}^t, \mathcal{P}_i)$ 
11  si  $\hat{x}^*$  est proche d'un point existant ou  $\hat{x}^*$  non admissible alors
12    Explorer
13  Calculer  $f(\hat{x}^*)$  et  $g(\hat{x}^*)$ 
14   $(\mathbb{X}^{t+1}, \mathbb{F}^{t+1}, \mathbb{G}^{t+1})_i \leftarrow (\mathbb{X}^t, \mathbb{F}^t, \mathbb{G}^t)_i \cup (\hat{x}^*, f(\hat{x}^*), g(\hat{x}^*))$ 
15  Mettre à jour le centre  $c_i$ 
16  Vérifier si fusion, séparation ou création
17  t = t + 1

```

son centre (ligne 15) et éventuellement réorganiser l'espace en fusionnant, en se séparant ou en créant un nouvel agent (ligne 16), avant de passer au pas suivant (ligne 17).

4.3.3 Partitionnement auto-organisé

Le partitionnement dépend fortement de la topologie de l'espace de recherche. Nous proposons donc un mécanisme auto-organisé pour partitionner l'espace en sous-régions qui s'adaptent à la topologie de l'espace. Par auto-organisé nous entendons que les agents (et ainsi les sous-régions) sont créés et supprimés au travers d'un processus coopératif d'optimisation par les agents eux-mêmes. Les agents se divisent lorsque les points de leurs sous-régions forment des clusters distincts (*création*), et fusionnent lorsque des centres d'agents (des optima locaux) convergent (*suppression*).

Les centres des agents (donc les agents) se déplacent dans l'espace. Le centre d'une sous-région est le meilleur point de cette sous-région. Le centre est déplacé à l'optimum de l'itération précédente s'il améliore la valeur de la fonction objectif, et si à la fois x^{*t-1} et c^{t-1} satisfont la contrainte g . Le centre est déplacé lorsque le dernier optimum satisfait la contrainte mais pas le centre précédent. Le centre est également déplacé s'il s'approche au mieux des contraintes alors que celles-ci ne sont respectées ni par le nouveau point, ni par le centre actuel. Ceci correspond à la ligne 15 de l'algorithme 3.

Une fois qu'un agent a ajouté un nouveau point dans sa base de croyances (ligne 14) et a déplacé son centre sur le meilleur point (ligne 15), il va vérifier s'il doit créer ou supprimer un agent ou fusionner avec un autre agent (ligne 16). Fusionner des agents (donc des sous-régions) évite que des agents ne se regroupent inutilement dans une même zone. Séparer un agent (donc créer un agent) est un moyen d'explorer plus efficacement l'espace par un affinement du partitionnement. La séparation et la fusion surviennent à la fin de chaque itération (ligne 16) : les agents sont d'abord fusionnés (si nécessaire), les points appartenant aux agents fusionnés sont partagés entre les agents restants en fonction du partitionnement, et chaque agent restant vérifie s'il se sépare ou non. Cette technique d'auto-organisation des agents et des sous-régions nous permet ainsi de ne pas fixer arbitrairement un nombre initial de sous-régions, et de lancer la procédure d'optimisation avec un unique agent qui se séparera si nécessaire.

Des agents sont fusionnés (et donc un agent est supprimé) si les centres de leurs sous-régions res-

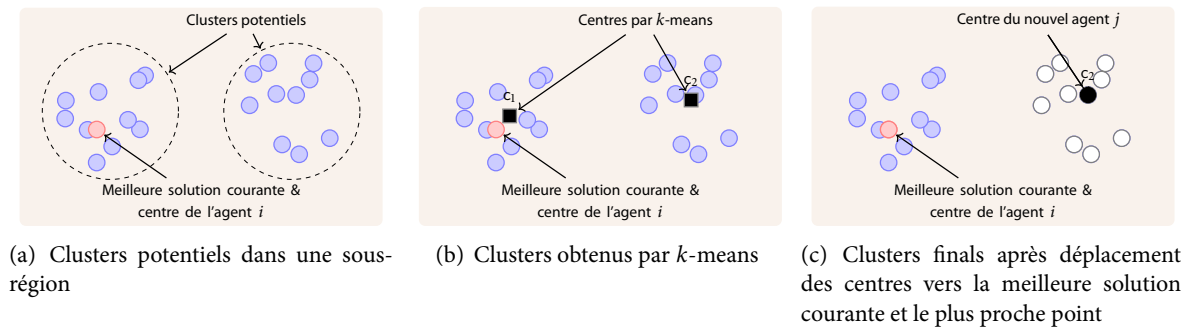


FIGURE 4.4 – Illustration du processus de création d'un agent j étant donnés des points de la sous-région de l'agent i .

pectives deviennent trop proches (en distance Euclidienne). Lorsque l'on examine les agents, l'agent avec la plus faible performance (le plus mauvais optimum) est supprimé. Avant la suppression, l'agent supprimé distribue ses points aux agents les plus proches de ces points.

Il est intéressant de créer un agent lorsqu'il se trouve que des points sont regroupés dans deux zones distinctes d'une même sous-région, comme illustré dans la Figure 4.4(a). Une telle situation peut apparaître lorsqu'il y a plusieurs optima locaux dans une même sous-région.

Les agents sont créés en utilisant une méthode de k -means [HARTIGAN et al., 1979] pour deux clusters ($k = 2$) étant donnés les points dans une sous-région, où la position initiale des centres est la meilleure solution courante de la sous-région (le centre actuel) et la moyenne de l'ensemble des points. Comme le clustering par k -means fournit des centres qui ne sont pas forcément des points existants comme illustré en Figure 4.4(b), nous déplaçons le nouveau centre vers le point disponible le plus proche dans la sous-région pour éviter un appel supplémentaire à la fonction coûteuse. Ceci est effectué en mesurant tout d'abord la distance des centres fournis par k -means au meilleur point actuel, puis en déplaçant le centre le plus proche vers le meilleur point, afin de conserver ce bon point. Pour l'autre centre, nous mesurons la distance de ce centre vers les autres points, et déplaçons le centre vers le plus proche. Le clustering final est illustré dans la Figure 4.4(c). Le résultat est donc un nouvel agent avec un centre positionné sur un point existant, où l'agent créateur reste sur son centre, à savoir le meilleur point. Le clustering final est ensuite validé en utilisant la *valeur de silhouette moyenne* des points de la sous-région ROUSSEEUW [1987].

Les agents peuvent atteindre un point pour lequel aucune amélioration n'est obtenue en plusieurs itérations. Par exemple, cela peut se produire lorsque chaque agent a localisé le meilleur point de sa sous-région. La zone autour du meilleur point est peuplée de points, et chaque agent est amené à explorer durant plusieurs itérations consécutives, et aucun autre optimum local potentiel n'est localisé. Ceci peut également se produire lors d'itérations en début de résolution lorsque les métamodèles sont mal ajustés dans la sous-région. Afin d'améliorer l'exploration, un nouvel agent est alors créé dans l'espace de recherche lorsque il n'y a pas d'amélioration depuis n itérations de résolution. Pour créer un nouvel agent, un nouveau centre est créé sur un point existant qui maximise la distance minimum aux centres existants. L'espace est alors partitionné à nouveau.

Applications et résultats. Cette approche a été appliquée à l'optimisation dans le cadre du projet ID4CS sur des cas d'étude théoriques en 2D et 6D [PICARD, VILLANUEVA et al., 2013 ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2013b] et pour la conception de boucliers thermiques [VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2013a], et le solveur développé en Scilab a été intégré à la plateforme ID4CS pour permettre à des agents de plus haut niveau, comme décrits dans [JORQUERA, 2013], d'effectuer des optimisations par appels à des fonctions coûteuses (disciplines). Il a été montré empiriquement qu'un agent seul (e.g. un optimiseur par métamodélisation classique) a moins de succès pour trouver tous les optima que le système multi-agent, pour un nombre équivalent d'appels au simulateur coûteux. De plus, dans certains cas notre approche multi-agent bénéficie d'un budget de calcul réduit dédié à la recherche aléatoire : à nombre constant d'évaluations, les agents avec un nombre initial de points élevé sont moins efficaces pour trouver les solutions admissibles que les agents débutant leur optimisation avec un faible nombre de points initiaux.

Dans cette approche, le SMA est bien plus efficace qu'un solveur standard pour (i) localiser l'optimum global et pour (ii) localiser tous les optima (à faible distance et faible valeurs g). Ceci démontre que notre comportement d'agent et de partitionnement auto-organisé entraîne que :

1. l'optimisation est moins coûteuse, car les agents s'exécutent en parallèle, pas-à-pas ;
2. le processus d'optimisation n'est pas seulement global mais peut également se stabiliser sur des optima locaux ;
3. le partitionnement final fournit une bonne compréhension du problème d'optimisation, comme chaque agent utilise le métamodèle le plus adapté à sa sous-région.

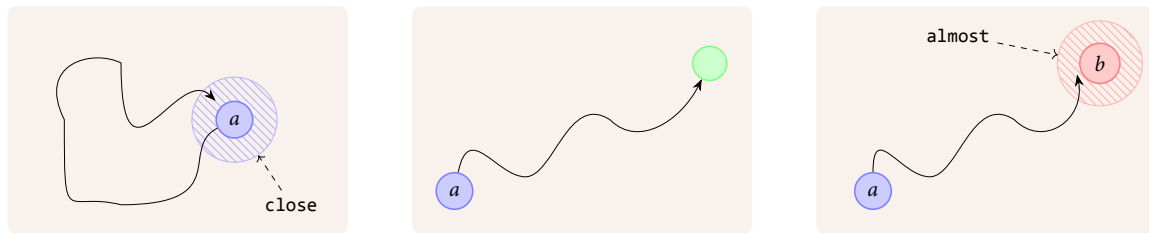
4.4 Affinement de métamodèles

Les conclusions de nos travaux sur le partitionnement auto-adaptatif, présentés dans la section précédente, nous ont encouragés à conduire des investigations plus poussées sur la localisation des optima locaux dans l'optimisation par métamodélisation. Ceci a conduit à l'algorithme LOOM (*Local Optima through Metamodels*), dans lequel les optima locaux sont identifiés en sortie du processus d'optimisation. Ici, la recherche distribue les ressources de calcul sur les bassins d'attraction [RIVIÈRE et al., 2013]. Alors que les bassins sont décrits comme des cellules de Voronoï dans les travaux précédents, LOOM utilise un métamodèle plus flexible pour les identifier et les décrire.

4.4.1 Principe général

La priorité est mise sur la recherche locale (*intensification*) et l'exploration s'effectue avec un budget limité d'appels à la fonction objectif. LOOM diffère donc des algorithmes heuristiques de recherche d'optima locaux (algorithmes évolutionnaires, essais particuliers) sur deux points : (i) LOOM peut être appliqué à l'optimisation coûteuse, et (ii) LOOM liste *explicitement* en sortie les optima locaux trouvés.

Le comportement global de LOOM diffère également de celui des deux précédentes approches, car les agents travaillent sur le même métamodèle, qui correspond à leur environnement commun, et qu'il *affine* au cours de l'exécution. Ainsi, les agents construisent collectivement un métamodèle commun. À l'initialisation de LOOM, un plan d'expérience initial, et un métamodèle relatif sont construits. Ensuite, un unique agent est créé dans l'espace du métamodèle, positionné aléatoirement ou sur un optimum global trouvé sur le métamodèle initial. À partir de là, la recherche commence, et va résulter dans la création et la destruction d'agents (et donc l'identification d'optima locaux) jusqu'à épuisement du budget d'appels. En fin de résolution, les agents sont positionnés sur tous les optima locaux (si le budget d'appels le permet), comme le présente la figure 4.7.



(a) Si l'optimisation locale retombe sur le même agent (dans la sphère de rayon `close`), alors réduction du rayon `close` et recherche pour un nouveau bassin

(b) Si l'optimisation locale trouve un nouveau point n'appartenant à aucun agent, alors mise à jour du métamodèle, déplacement et restauration de `close`

(c) Si l'optimisation locale trouve un nouveau point appartenant à autre agent (dans sa sphère `almost`), alors suppression du pire agent

FIGURE 4.5 – Exploitation dans LOOM

4.4.2 Agents LOOM

Les agents correspondent à des optima locaux potentiels, et effectuent chacun une optimisation locale dans leur bassin d'attraction (i.e. *exploitation*). Ils peuvent ajouter des points (par appel à la fonction objectif) — et ainsi changer le métamodèle — et ajouter ou supprimer d'autres agents — dans le même esprit que l'approche précédente. Afin de déterminer si l'agent améliore ou non sa position, et afin de garantir que l'approche est valide pour toute taille de bassin, les agents utilisent une sphère de rayon `close` qui détermine si l'agent est proche ou non du point (voir figure 4.6). En fonction de l'appartenance ou non du nouveau point à cette sphère, l'agent va pouvoir exploiter à nouveau, créer un nouvel agent ou explorer. Ils utilisent également une sphère de rayon `almost` pour déterminer si un agent doit être supprimé (voir figure 4.5(c)).

Le comportement de chaque agent peut-être résumé comme suit (voir figures 4.5 et 4.6) :

Exploitation : l'agent effectue une optimisation locale à partir de sa position actuelle sur le métamodèle actuel $f^{(t)}$ (e.g. Nelder-Mead avec une taille de simplexe réduite [NELDER et al., 1965]) afin de trouver un meilleur point que leur position actuelle (qui est la meilleure connue dans le bassin).

Création : si le résultat de l'optimisation locale est *proche* (voir Figure 4.6(a)) de la position de l'agent, l'agent réduit son paramètre de proximité `close` (voir figure 4.6) et cherche un nouveau bassin d'attraction sur le métamodèle :

- si un nouveau bassin inoccupé est trouvé, alors l'agent appelle f , met à jour le métamodèle et crée un nouvel agent dans le bassin;
- si aucun bassin n'est trouvé, ou si le bassin est déjà identifié (un agent est présent dans une sphère de rayon `close`), alors l'agent *explore*, i.e. cherche le point le plus loin de ses voisins, calcule f en ce point et met à jour le métamodèle.

Exploration : si le résultat de l'optimisation locale n'est pas proche de la position courante de l'agent, alors :

- si un autre agent est presque à la position du résultat de l'optimisation locale (voir figure 4.5(c)), il supprime le pire des deux agents;
- sinon, il appelle f sur l'optimum local, s'y déplace s'il est plus intéressant et met à jour le métamodèle.

Afin de trouver un nouveau bassin, un agent voyage à travers l'espace de recherche en suivant un ligne droite infinie partant de sa position actuelle. Tout au long de cette ligne, en fonction de paramètres de vitesse et d'accélération, l'agent effectue des optimisations locales (sur le métamodèle). Une telle optimisation résultant sur un point différent de sa position initiale signifie qu'un nouveau bassin a été découvert.

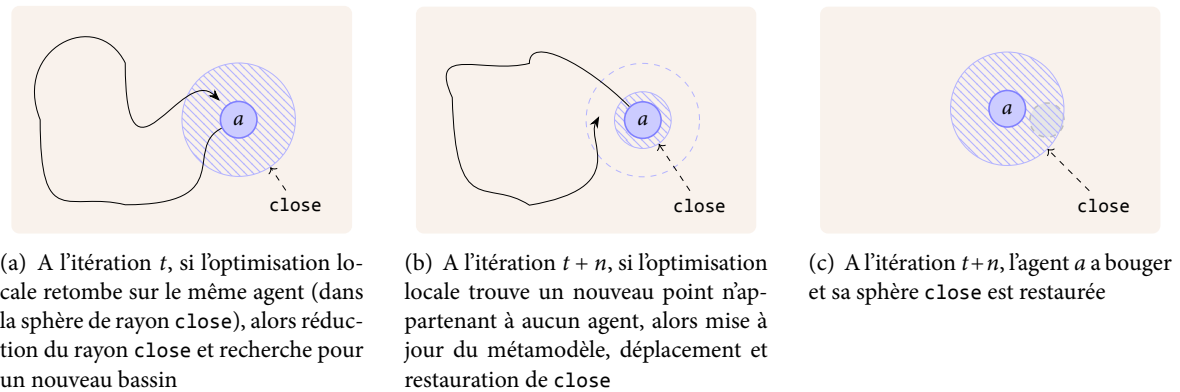
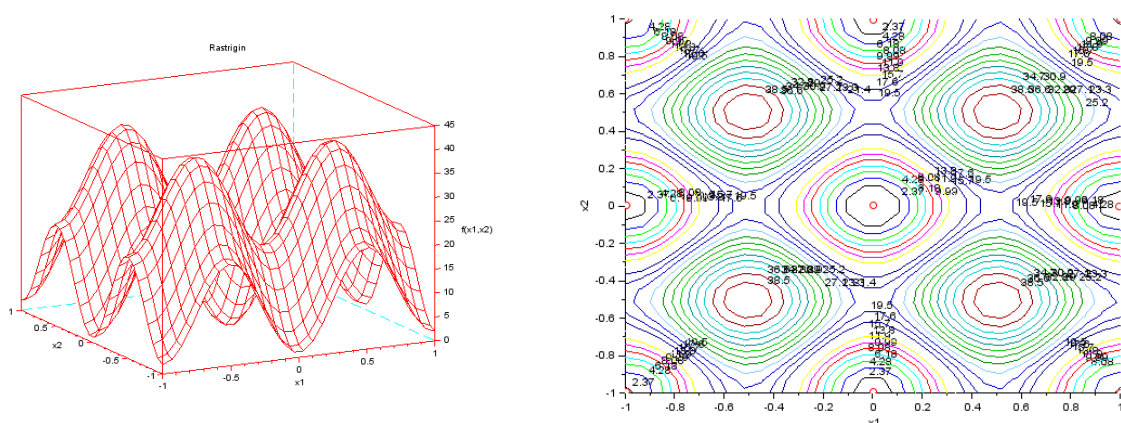
FIGURE 4.6 – Evolution de la valeur c_{close} LOOM

FIGURE 4.7 – Exemple de résultat de l'exécution de LOOM sur la fonction Rastrigin 2D (à gauche). Le nombre d'agents a convergé vers 9, et les agents sont positionnés sur les optima locaux de la fonction (cercles rouges, à droite).

Applications et résultats. LOOM a été principalement éprouvé sur des cas tests académiques piégeurs de 2D à 6D, sur une métamodélisation par krigeage et régression à vecteurs de support. Il a été montré expérimentalement que LOOM trouve les optima locaux à une précision inférieure à 1% et converge vers le nombre théorique d'optima sur des cas tests pouvant afficher plusieurs centaines d'optima locaux. Cependant, comme LOOM limite le nombre d'appels à la fonction à optimiser, ce budget doit être suffisant (donc supérieur au nombre d'optima) pour converger.

Ainsi, LOOM repose sur une coordination par auto-organisation basée sur deux paramètres de proximité (c_{close} et a_{almost}). Par l'évolution de c_{close} , les agents sont capables d'exploiter des bassins étroits ou d'explorer si nécessaire, garantissant un bon compromis exploitation-exploration.

4.5 Synthèse et analyse

Nous avons présenté dans ce chapitre des approches multi-agents qui ont consisté à fournir des optimiseurs dont la procédure de recherche est effectuée par des agents autonomes se partageant la tâche et se coordonnant pour améliorer l'exploration. L'organisation de ces agents est parfois *explicite* (e.g. partitionnement en cellule de Voronoï) ou *implicite* (e.g. dans LOOM), et les agents ne raisonnent que sur des informations locales, ou sur leur environnement direct. Les capacités d'**auto-organisation** des agents permettent au système de s'adapter à l'espace de recherche, de **apprendre**. Ici l'organisation est modifiée indirectement par le déplacement des agents suite à des processus d'optimisation locaux :

les agents cherchent à améliorer leur performance, et ainsi modifient l'organisation globale. Ces travaux ont donc produit de nouvelles techniques de métamodélisation, donc d'apprentissage. Nous pouvons placer ces travaux dans les cadrans (a) et (b) — pour ce qui concerne l'approche par partitionnement — de la figure 2.1 : ce sont des **approches centrées agent avec conscience limitée de l'organisation** (par exemple, les agents connaissent les cellules voisines et leurs frontières). Ces approches se prêtent parfaitement à la problématique de la *compréhension des topologies locales et de la métamodélisation d'espaces de recherche multimodaux*.

Ces travaux ont été menés dans le cadre du projet ID4CS, et notamment au cours de la thèse de Diane Villanueva [VILLANUEVA, 2013] (en cotutelle avec l'Université de Floride) et du post-doctorat de Jérémy Rivière. Ils sont également le fruit d'une collaboration pluridisciplinaire entre mécaniciens, informaticiens et mathématiciens. Il est intéressant de noter l'apport mutuel de cette collaboration. En effet, la perspective multi-agent a permis d'envisager les problèmes d'optimisation en mécanique d'une manière originale et finalement efficace dans la recherche des optima et le nombre d'appels aux fonctions coûteuses. Les résultats obtenus par nos optimiseurs multi-agents sont compétitifs par rapport aux standards utilisés classiquement par les académiques et les industriels, avec des propriétés intéressantes comme la *distributivité* et l'*identification des optima locaux*. Cette dernière est notamment primordiale pour des problèmes aux configurations inconnues, comme par exemple la conception d'avion à motorisation hybride, pour laquelle les experts n'ont aucune intuition sur la position des optima dans l'espace de conception.

Certains de ces travaux ont également été intégrés dans une plateforme d'optimisation multi-disciplinaire issue du projet ID4CS. Cette intégration consiste en l'encapsulation des solveurs multi-agents dans d'autres agents (de plus haut niveau) ayant besoin de faire appel à des optimiseurs peu gourmand en appel, pour calculer des corrélations entre entrées et sorties de modèles.

Les publications et communications suivantes sont accessibles pour plus de détails sur tous ces travaux : [VILLANUEVA, HAFTKA et al., 2013 ; PICARD, VILLANUEVA et al., 2013 ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2013b ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2013a ; VILLANUEVA, PICARD et al., 2012 ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2012a ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2012b ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2011 ; VILLANUEVA, LE RICHE, PICARD, HAFTKA et SANKAR, 2011 ; RIVIÈRE et al., 2013].

5 Auto-adaptation dans la résolution de problèmes

Les travaux présentés dans ce chapitre concernent la conception d'algorithmes distribués et de protocoles pour la résolution de problèmes combinatoires (bien que les approches présentées ont également été appliquées au domaine du continu dans des travaux antérieurs) dans lesquels une capacité d'adaptation à la dynamique est mise en place (voir section 3.4). Dans ce chapitre nous allons adopter des stratégies d'agentification par *décomposition du problème par dimensions et critères* et par *décomposition par modélisation ad-hoc du domaine* (voir section 3.3). Ici, donc, le système multi-agent consistera en un ensemble d'agents ayant des objectifs distincts (sous-objectifs de l'objectif global) interagissant au travers d'un environnement, ou tout simplement dialoguant directement, suivant une topologie dépendant du problème.

Avant de présenter les contributions, la section 5.1 discute de la notion d'auto-adaptation. La section 5.2 présente des travaux dans lesquels la coordination entre les agents est mise en œuvre via un environnement partagé de travail (voir section 3.2), appliqués notamment à la gestion d'emplois du temps ou le contrôle manufacturier. La section 5.3 présente des approches par négociation, dans lesquelles les agents dialoguent afin de converger vers une solution acceptable, appliquées notamment à l'affectation de fréquences, ou aux appariements stables.

5.1 Processus d'auto-adaptation

Comme nous l'avons vu dans la section 2.3, l'adaptation d'un système multi-agent, de notre point de vue, passe par une modification de son organisation. Il est donc nécessaire de déterminer qui sont les acteurs de ce processus d'adaptation lorsque l'on conçoit un tel système. Les approches présentées ici correspondent à des stratégies de changement émergent de l'organisation (voir section 2.3.3), dans lesquelles les agents n'auront qu'une vision limitée — voire aucune — sur l'état de l'organisation. Les agents autonomes sont ici les moteurs de cette adaptation : c'est pour cela que nous parlons d'*auto-adaptation*, à opposer à une adaptation guidée par une tierce partie, pouvant agir sur l'organisation, en changeant, par exemple, des attributions de rôles.

Dans le cadre de la résolution de problèmes, ce processus d'auto-adaptation est d'autant plus intéressant qu'il s'agit d'un processus continu, qui ne demande pas de redémarrer la résolution d'un problème suite à une modification de ses spécifications. Ainsi, dans l'exemple du contrôle manufacturier, il est dommageable de redémarrer tout un processus de résolution suite à l'arrivée d'une perturbation mineure. L'auto-adaptation est donc un moyen de *réparer* une solution en cours de construction. Elle sera

donc presque toujours synonyme de méthode de résolution par recherche locale décentralisée, dans laquelle le système fournit toujours une solution (son état courant) même si celle-ci n'est pas optimale. C'est au cours du processus d'adaptation que le système va améliorer la solution, par propagation de pressions locales, pour converger vers un optimum.

Une autre propriété intéressante de cette approche est l'*encapsulation* naturelle des données dans les agents composant le système. Il n'y a pas d'entité tierce résolvant les problèmes globaux et collectant les données des autres agents pour obtenir une solution. Ceci a deux avantages et un inconvénient. Tout d'abord, de tels systèmes peuvent garantir une certaine *privacité* des données, et donc être utilisés pour des applications dans lesquelles l'ouverture des données peut-être problématique. De plus, comme nous l'avons souligné plus haut, étant donné que le système répare son état au lieu de repartir à zéro après chaque perturbation, les solutions proposées *minimisent le nombre de changements* par rapport à une solution dégradée. Par contre, ces approches ne fournissent *aucune garantie de convergence* vers l'optimum.

Applications et résultats. Nous avons eu l'occasion d'étudier les propriétés des systèmes auto-organiseurs et l'auto-adaptation dans le cadre de la résolution de problèmes (illustrée par le contrôle manufacturier) dans le groupe de travail COLLINE^a (Collectif, Interaction, Émergence) et dans le groupe de travail européen TFG SO^b (*Self-Organisation in MAS*) initié dans le réseau d'excellence AgentLink. Nous avons également étudié comment évaluer ce type de systèmes, et proposé de nouvelles métriques afin de caractériser au mieux ces systèmes [GEORGÉ, M.-P. GLEIZES et al., 2010 ; KADDOUM, M.-P. GLEIZES, GEORGÉ et PICARD, 2009 ; KADDOUM, M.-P. GLEIZES, GEORGÉ, GLIZE et al., 2009].

a. www.irit.fr/COLLINE

b. Self-OrganisationinMAS

5.2 Approche par coordination via l'environnement

Cette section présente les approches où l'environnement est une abstraction de premier niveau — car utilisé pour représenter les contraintes ou pour servir de médium de coordination —, comme dans les approches par stigmergie, e.g. les algorithmes par colonies de fourmis [DORIGO et al., 2004 ; BONABEAU et al., 1999]. Cette approche a été une source d'inspiration pour des techniques multi-agents de résolution et de coordination [LIU et al., 2002 ; ARMETTA et al., 2007 ; REITBAUER et al., 2004 ; S. BRUECKNER, 2000 ; PARUNAK, S. BRUECKNER et al., 2002 ; S. BRUECKNER et PARUNAK, 2004]. Les agents se déplacent dans l'espace de recherche comme dans un environnement physique, et peuvent le marquer afin de laisser des informations ou de les altérer — ce qui peut s'avérer utile pour résoudre des conflits dans le cadre de la résolution de problèmes.

5.2.1 Du problème à l'environnement

Résoudre un problème, modélisé comme un DCSP (voir section 3.2), consiste à assigner des valeurs à des variables (des objectifs). Lorsque ce problème est partagé ou distribué, plusieurs acteurs doivent coopérer pour trouver une solution satisfaisante. Un exemple de tel problème est l'établissement d'emplois du temps universitaires, dans lequel des agents cherchent à trouver des créneaux communs, tout en exprimant des impossibilités ou des préférences, comme nous l'avons abordé dans ETTO (*Emergent Timetabling Organisation*) [PICARD, BERNON et M.-P. GLEIZES, 2005a]. Un moyen direct de permettre cette coopération est de concevoir un environnement commun de travail, sur lequel les agents vont déposer des informations pour leur coordination (e.g. des réservations de créneaux), comme l'illustre la figure 5.1.

L'environnement représente le produit cartésien des domaines de valeurs possibles des objectifs

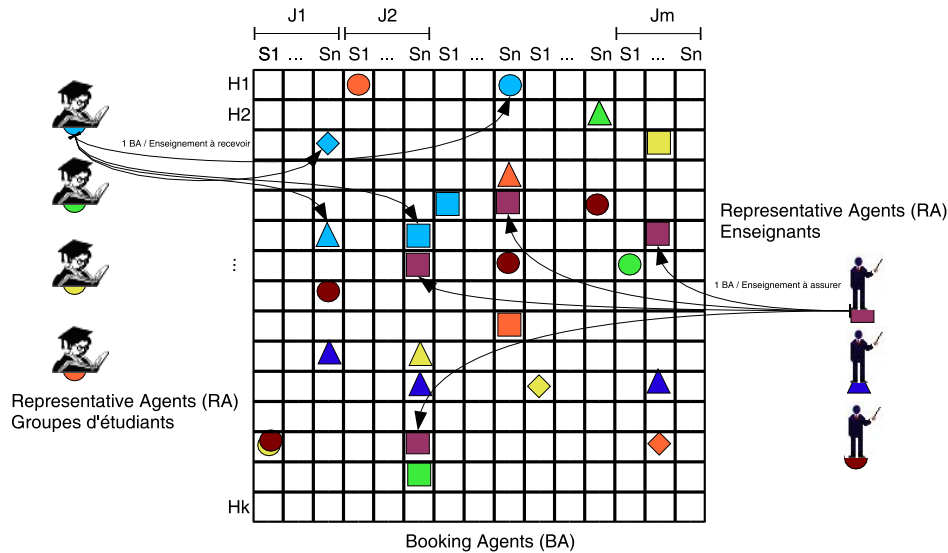


FIGURE 5.1 – Exemple de l'environnement pour la résolution de problèmes d'emplois du temps universitaires [PICARD, BERNON et M.-P. GLEIZES, 2005b]. Des agents représentant les enseignants et les étudiants établissent leur emploi du temps en réservant des créneaux horaires et des salles.

des agents. Par exemple, dans la figure 5.1, l'environnement est une grille à deux dimensions : créneaux temporels, salles. Contrairement à des approches par dépôt de phéromones, cette approche ne nécessite pas de doter l'environnement d'une dynamique propre (e.g. évaporation des marqueurs) ; seuls les agents modifient cet environnement. Les informations déposées dans l'environnement peuvent être de diverses natures, en fonction du problème, et plus ou moins riches de sens. Par exemple, dans ETTO, les agents déposent des « post-its » virtuels contenant leur adresse (afin d'établir des partenariats) et un *timestamp* afin de résoudre des conflits. L'adaptation d'ETTO pour le contrôle manufacturier, ETTO4MC (*ETTO for manufacturing control*), a été effectuée de manière directe, en modifiant uniquement les dimensions de l'espace. Le concept de partenariat reste le même : des agents de réservation cherchent des partenariats pour des stations de travail, et des créneaux afin de compléter une liste de commandes — qui évolue avec le temps [CLAIR et al., 2008b ; CLAIR, 2008]. Dans un travail parallèle, DAMASCOP, nous avons également proposé une agentification *domaine*, dans lesquelles les agents représentent les stations, les opérateurs et les conteneurs [CLAIR et al., 2008b ; KADDOUM, 2008], sans faire appel à un environnement.

En poussant la simplification des informations déposées, il est possible de ne considérer que les conflits. Ainsi nous transformons le problème de recherche de valeurs, en un problème de minimisation des conflits entre agents. Cette approche s'inspire de l'heuristique *min-conflict*, ayant fait ses preuves sur des CSP de grande taille [MINTON et al., 1994]. Ce genre d'heuristique est notamment utilisé dans ERA [LIU et al., 2002], comme nous l'avons vu dans la section 3.2 (figure 3.2). Cet exemple peut être complété par la figure 5.2(a) représentant deux environnements pour deux problèmes différents (dont celui des n -reines classiquement résolu avec *min-conflict*).

5.2.2 Le comportement coopératif des agents

Une fois que l'environnement a été défini, il est nécessaire de déterminer quels sont les comportements des agents, et notamment concernant l'altération de l'environnement. Les agents ont bien sûr leurs objectifs individuels, et ne vont se déplacer et marquer l'environnement que dans les cellules admissibles, c'est-à-dire leur domaine de valeur.

Une approche « simpliste » est de faire se déplacer les agents dans les cases admissibles, tout en minimisant leurs propres conflits. Ceci mène souvent vers des optima locaux, dont il faut se dégager. Les approches classiques vont consister à ajouter du bruit et de l'aléatoire dans les décisions. Par exemple,

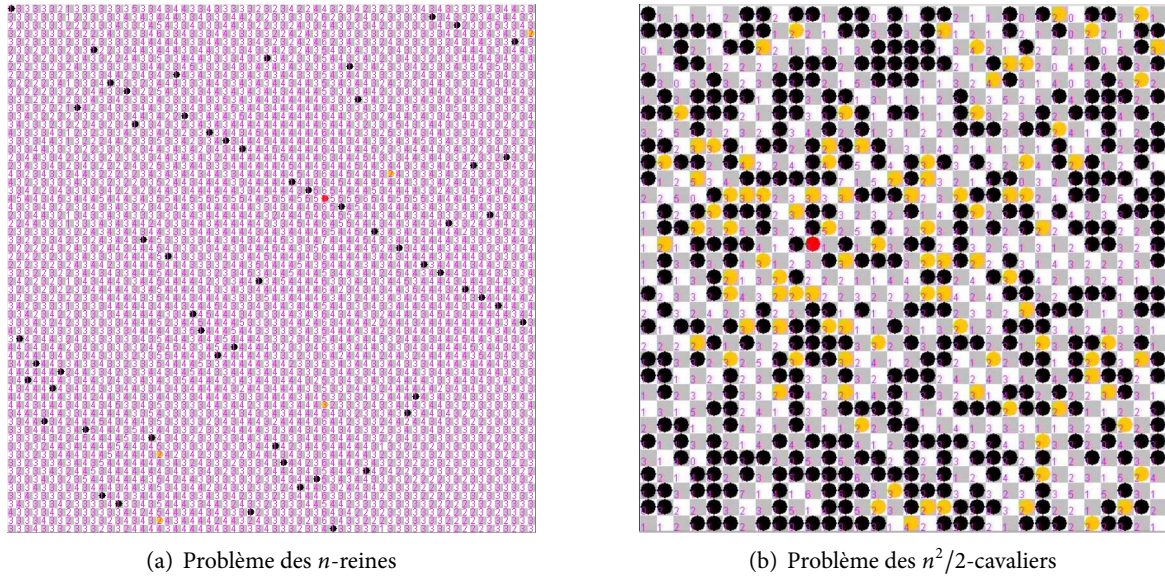


FIGURE 5.2 – Exemple d'environnement pour deux problèmes généralisables à d'autres CSP [GLIZE et PICARD, 2011]. Les agents (disques colorés) marquent l'environnement en incrémentant le nombre de conflits potentiels pour chaque case du damier (nombre en violet).

dans ERA, les agents ne vont pas nécessairement vers la case la moins conflictuelle, mais peuvent dans certains cas dégrader la qualité de leur solution locale [LIU et al., 2002] — ce qui est fortement inspiré par les modèles probabilistes de comportements d'insectes sociaux [BONABEAU et al., 1999].

Notre approche se base sur une approche coopérative des agents : les AMAS [GLIZE, 2001 ; M.-P. GLEIZES, 2004], illustrée dans l'algorithme 4. Ici, les agents ne choisissent pas les cellules sur lesquelles se déplacer ou à réserver, uniquement en tenant compte de leur objectif propre, mais également en tenant compte de l'état d'avancement des autres agents et de leur difficulté à trouver une solution.

Algorithme 4 : Un algorithme générique issu des AMAS, basé sur une valeur de criticité (κ), avec $v(x_i)$ désignant le voisinage d'un agent.

```

foreach agent  $i$  do
    set an initial assignment to  $x_i$ 
     $x_{i,\kappa} \leftarrow 0$ 
while termination conditions not met do concurrently
    order the possible solutions according to their  $\kappa$  value
     $x_{worst} \leftarrow \operatorname{argmax} \{x_{j,\kappa} \mid x_j \in v(x_i)\}$ 
    assign cooperative value to  $x_i$  such as  $x_{worst,\kappa}$  decreases
    compute  $x_{i,\kappa}$ 
    send  $x_{i,\kappa}$  to neighbours in  $v(x_i)$ 
    
```

Dans d'ETTO, nous avons par exemple utilisé un *timestamp* pour favoriser les agents en recherche de créneaux depuis longtemps, et ainsi déterminer la criticité. En conséquence, les agents les plus en difficulté sont prioritaires, ce qui permet de sortir de certains optima locaux. Ceci ne suffit pas forcément, et une marche aléatoire est souvent ajoutée, mais rarement déclenchée [PICARD, BERNON et M.-P. GLEIZES, 2005a].

Il est également possible d'équiper les agents d'une liste tabou, afin qu'ils ne reviennent pas sans arrêt vers les mêmes optima locaux [PICARD et GLIZE, 2006]. Ainsi, les agents se déplacent dans l'environnement et le marque de telle sorte que la solution (le positionnement des agents) minimise les conflits individuels, comme le montre l'algorithme 5 qui est une spécialisation de l'algorithme 4. La

Algorithme 5 : Un comportement basique pour un agent (q_i). Ici au lieu d'envoyer directement les valeurs aux autres agents, les agents marquent leur environnement.

```

while alive do
  updatePerceptions() // removing previous marks and adding new ones
  if (cost( $q_i$ ) == 0) then not occupying a conflictual cell
    rest()
  else
    moveTo(minConflictCell( $q_i$ ))
    mark(pCells( $q_i$ ))

```

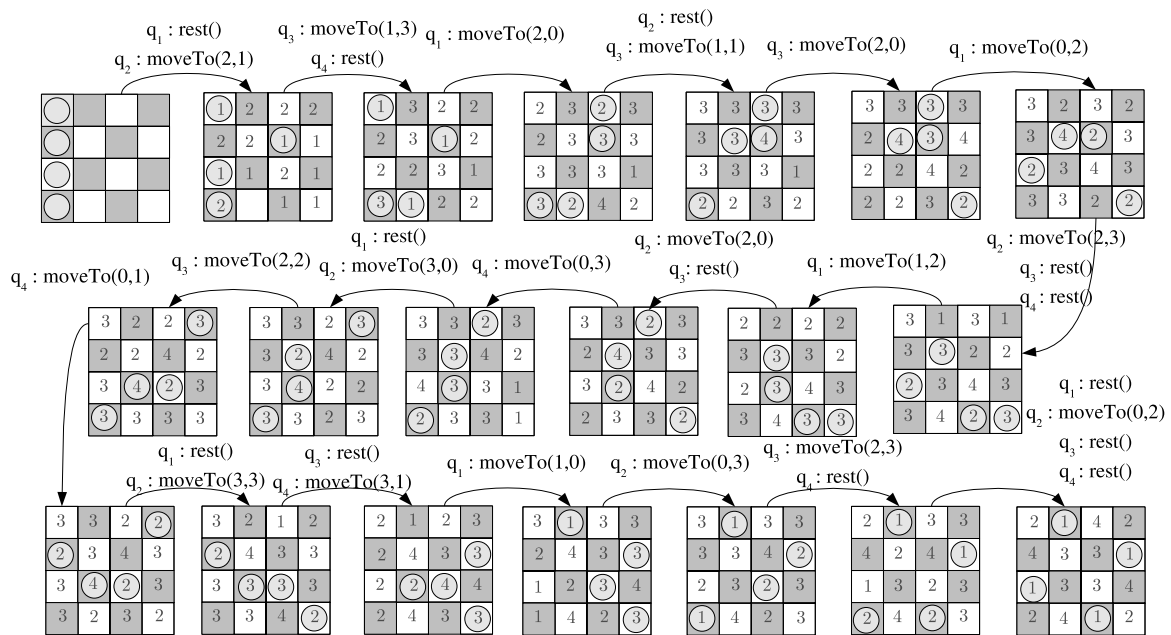


FIGURE 5.3 – Exemple de l'évolution de l'environnement de résolution du problème des 4-reines [GLIZE et PICARD, 2011].

figure 5.3 illustre la trace de la résolution d'un problème 4-reines par coopération. Ici les reines perçoivent et peuvent se déplacer sur toutes les cases qu'elles peuvent attaquer. Notons que contrairement aux approches classiques de résolution de ce problème qui limitent le déplacement d'une reine à une seule ligne — injectant une connaissance métier *a priori* —, les agents peuvent se déplacer sur tout l'échiquier¹. Ils se déplacent vers les cases les moins conflictuelles et impactant moins les autres agents percevant cette case. Les agents procèdent à une évaluation de la *criticité* des cases à choisir calculée à partir du nombre de conflits (les marques sur l'environnement) et de *max* du nombre de conflits sur la case actuelle de chaque agent percevant la case à évaluer.

5.2.3 Adaptation du système

L'avantage de reposer sur un état représentant une affectation complète, mais pas forcément optimale, et sur un comportement d'agent basé sur la coopération et des perceptions/actions locales, est de permettre au système de se réparer en cas de perturbation. En effet, une perturbation n'aura d'impact que sur les agents la percevant et sera réparée directement par l'agent ou par propagation, depuis la source de la perturbation jusqu'à sa solution. Ce profil adaptatif peut s'illustrer par la figure 5.4 présen-

1. Il est cependant remarquable que finalement les agents se positionnent un par ligne dans les premiers pas de la résolution, juste en se laissant guider par les règles de coopération, comme le montre la figure 5.2(a)

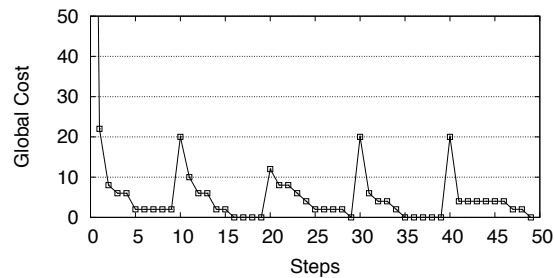


FIGURE 5.4 – Coût de la solution (nombre total de conflits) en fonction des pas de résolution pour plus de 500 reines coopératives subissant des perturbations aléatoires (tous les 10 pas, 10% des agents sont déplacés aléatoirement sur l'échiquier).

tant la qualité de la solution globale au cours du temps. Suite à des perturbations régulières (déplacement aléatoire d'agents), le système s'auto-organise pour revenir à un état acceptable.

Ces approches sont aisément généralisables à d'autres problèmes combinatoires. Cependant l'approche ETTO suivant une agentification plutôt guidée par le domaine (voir section 3.3), avec des agents représentant les utilisateurs et des agents représentant les réservations, elle est plus adéquate pour des problèmes d'affectation de ressources, comme le contrôle manufacturier avec ETTO4MC.

Applications et résultats. Les travaux présentés dans cette section ont été d'abord éprouvés sur des cas d'école (e.g. n -reines) [PICARD et GLIZE, 2006]. ETTO, initialement conçu pour les emplois du temps universitaires [PICARD, BERNON et M.-P. GLEIZES, 2005a], a également été appliqué au contrôle manufacturier et a montré de meilleures performances que des algorithmes classiques de résolution de DCSP [CLAIR et al., 2008b ; CLAIR, 2008], et a été implanté dans le cadre du groupe de travail COLLINE, dans la plateforme MASC. Nous avons auparavant utilisé des techniques équivalentes à ETTO dans la coordination de collectifs de robots transporteurs [PICARD et M.-P. GLEIZES, 2005b].

Les approches de résolution coopérative de problèmes que nous venons de voir s'appuient ainsi fortement sur l'environnement comme médium de coordination sans lequel les agents ne pourraient sortir des conflits (dans le cas des CSP) ou trouver des partenaires (dans le cas d'ETTO ou ETTO4MC). Nous allons voir maintenant des approches purement dialogiques, dans lesquelles les agents résolvent les conflits par échange de messages.

5.3 Approche par négociation

Résoudre un problème via d'un environnement partagé est efficace, comme nous avons pu le voir à la section précédente. Cependant nous pouvons identifier deux limitations à ce type d'approches : (i) l'environnement partagé peut se trouver être un frein à un réel déploiement distribué et (ii) les informations partagées laissent des traces, lisibles par d'autres agents, et donc représentent un frein à la *privacé*. Pour outrepasser ces limitations, des protocoles de négociation peuvent être envisagés. Ici les agents cherchent à affecter une ou plusieurs valeurs internes en respectant des contraintes inter-agents, en échangeant directement des messages représentant des propositions, des accords ou des refus — sans forcément dévoiler à tout le monde toutes les informations personnelles. Les approches classiques de résolution de DCOP (voir section 3.2) opèrent de cette manière, et ont été une grande source d'inspiration pour les travaux présentés dans cette section. Nos travaux seront illustrés par les cas d'étude de l'affectation de fréquences (FAPP) [PICARD et al., 2007b] et des appariements stables (CASANOVA) [EVERAERE et al., 2012].

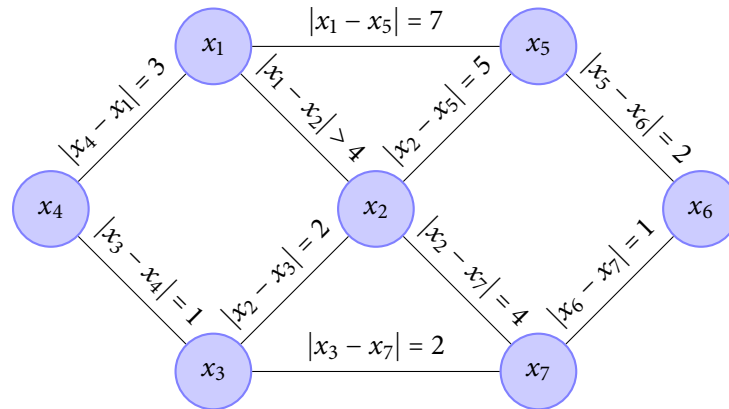


FIGURE 5.5 – Un exemple de FAPP représenté sous forme de graphe variables-contraintes.

5.3.1 Organisation et topologie

Dans la plupart des méthodes et algorithmes de résolution de DCOP, les problèmes sont représentés sous forme de graphe, comme illustré dans la figure 3.1. Cependant, en fonction de l’algorithme et de ses propriétés, les agents ne vont pas forcément suivre cette topologie pour envoyer des messages.

Un exemple représentatif est celui d’ABT qui organise les agents en hiérarchie totale (idéalement un chemin couvrant du graphe), dans laquelle les messages de propositions sont envoyés de bas en haut, alors que les refus ou les notifications d’impossibilités (pour déclencher un retour en arrière, *backtrack*) sont envoyés de bas en haut. Cette organisation, rigide, assure la complétude de l’algorithme ABT, mais nuit à ses performances (i.e. exploration complète de l’espace) et à sa robustesse en cas de disparition d’agents (e.g. suite à une panne de réseau). Des approches plus flexibles comme celle de AWCS rendent cette ordre total dynamique, les agents changeant de place dans la hiérarchie en cours de résolution, guidés par une heuristique de type *min-conflict* [YOKOO, 2001].

D’autres approches, comme DBA, APO ou OptAPO [YOKOO, 2001 ; MAILLER et V. R. LESSER, 2006], reposent sur le graphe de contraintes, et les agents limitent leur envois de messages aux agents avec lesquels ils partagent des contraintes. Ces approches sont pertinentes alors pour prendre en charge la dynamique (ouverture du systèmes, ou bruit dans les communications). Les approches que nous avons développées sont proches de ces dernières. Cependant les approches citées possèdent certaines limitations, comme par exemple la possibilité dans APO qu’un agent se retrouve à collecter toutes les valeurs des autres agents et à résoudre le problème seul.

Le problème FAPP correspond à un CSP classique : la coloration de graphe. Ici les agents représentent des antennes pouvant émettre à différentes fréquences, mais devant choisir leur fréquence en fonction des fréquences des antennes voisines afin de limiter les interférences. Le graphe de contraintes correspond alors à un maillage de l’espace (graphe plan), les variables étant les fréquences des antennes, et les contraintes des distances à respecter entre ces fréquences, entre antennes voisines, comme l’illustre la figure 5.5. Les agents n’échangeront des messages qu’entre voisins directs. Dans CASANOVA, par contre, compte tenu de la nature du problème d’appariement (graphe complet de contraintes), les agents peuvent potentiellement agir directement avec tous les agents, mais dans les faits, ils n’envoient des propositions qu’à destination des agents préférés (dans le haut de leur liste de partenaires potentiels), ou ne répondent qu’à des propositions.

Lors de la résolution, les agents peuvent jouer différents rôles, explicites ou non. Ces rôles impliquent que les agents utiliseront des protocoles et des performatives différents, ou se verront affectés des priorités différentes. Par exemple, dans les travaux sur FAPP, des agents sont *élus* (d’où le caractère auto-organisateur du processus) pour gérer les conflits dans un voisinage donné. Ou bien, dans nos travaux sur les problèmes classiques de mariages stables [GUSFIELD et al., 1989], nous avons proposé l’algorithme CASANOVA, dans lequel les agents jouent à la fois les rôles de proposant et de disposant

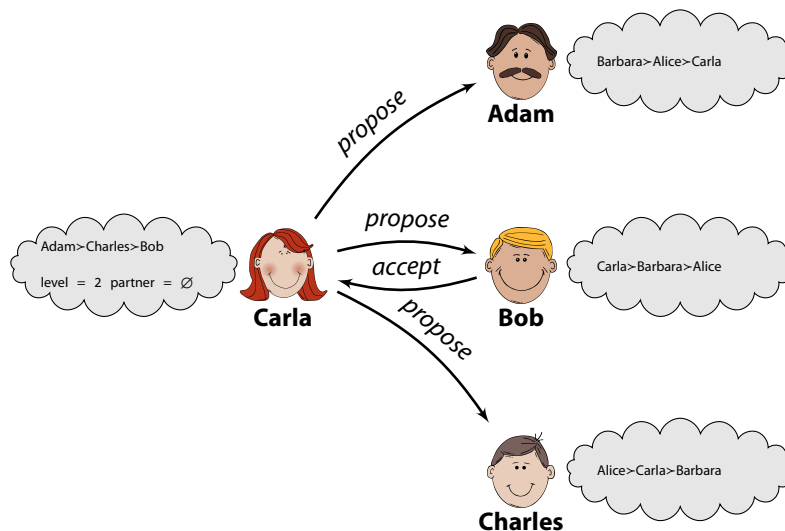


FIGURE 5.6 – Un exemple d’interactions directes dans CASANOVA [EVERAERE et al., 2012]. Carla propose un appariement à ses partenaires préférés mais ne dévoilent pas ses préférences.

[EVERAERE et al., 2012], comme l’illustre la figure 5.6.

Dans nos travaux, donc, les agents sont organisés topologiquement comme dans le graphe de contraintes, mais peuvent s’auto-organiser dans le sens où ils pourront changer de rôles en cours de résolution et pourront modifier les propriétés des liens entre-eux (e.g. priorité ou criticité).

5.3.2 Comportement des agents

Une fois l’organisation définie, il faut déterminer quels sont les messages envoyés et comment les agents réagissent aux messages, ainsi que les changements éventuels de rôles. Que ce soit dans CASANOVA ou dans FAPP, nous avons conçu des protocoles de négociation asynchrone. Les agents n’agissent pas avec un tour de parole prédéterminé, mais sont encapsulés dans des fils d’exécution distincts. Nous présentons ici deux stratégies de résolution : la coopération dans FAPP et la concession minimale dans CASANOVA.

Comportement coopératif

L’approche que nous avons adoptée pour FAPP est assez générique, puisque le problème est un CSP dont le réseau de contraintes correspond à un maillage (pour un ensemble homogène d’émetteurs-récepteurs). L’algorithme proposé dans [PICARD et al., 2007b] repose sur l’approche AMAS (voir section 5.2.2). Ici les agents vont déterminer qui est l’agent le plus critique du voisinage, en utilisant une somme pondérée des métriques suivantes :

- la *meilleure amélioration* possible pour cet agent par rapport à sa situation courante, au regard du nombre de contraintes insatisfaites;
- le *nombre d’affectations* possibles pour l’agent, capables de vérifier cette contrainte;
- le nombre de contraintes insatisfaites dans la situation courante (comme pour les reines, section 5.2.2);
- la *durée d’insatisfaction* mesurée en nombre de sessions d’affectation depuis la dernière satisfaction (comme dans ETTO, section 5.2.2).

Les agents suivent alors un processus classique d’envoi de propositions, d’attentes de réponse, et ensuite mettent en place une élection pour l’agent ayant le droit de modifier sa valeur, pour un voisinage

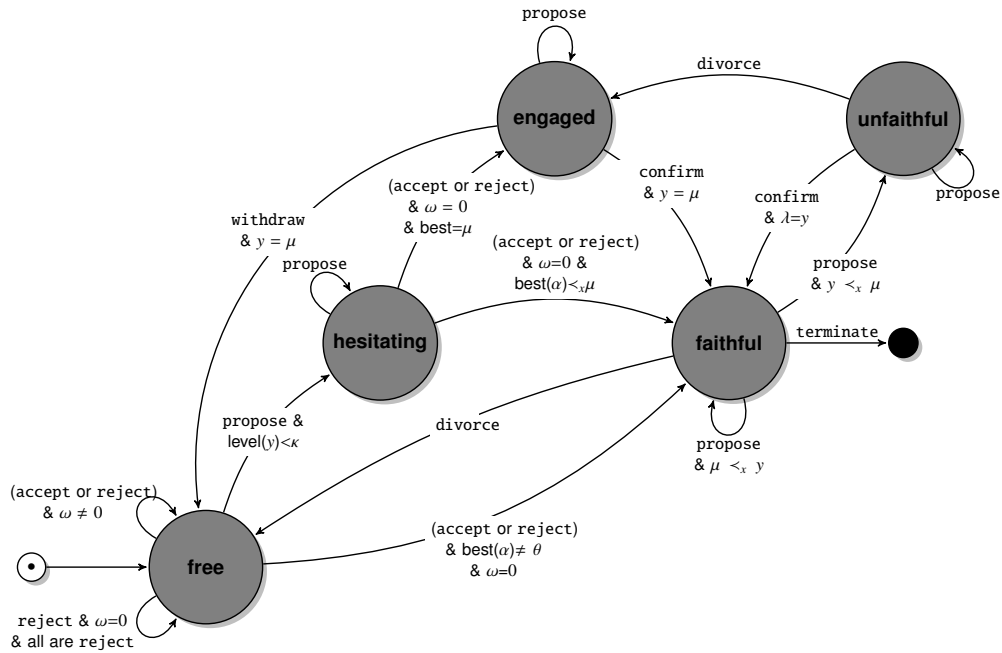


FIGURE 5.7 – Comportement des agents CASANOVA représenté comme un automate fini déterministe [EVERAERE et al., 2012].

donné. Cet agent élu représente alors un point de synchronisation afin de garantir la cohérence des données. Par contre, tous les voisins d'un agent connaissent la valeur et la criticité de cet agent — cette approche n'est donc pas forcément adéquate en terme de *privacit *. Ce protocole repr sente une recherche locale d centralis e, et ne garantit donc pas la convergence vers une solution. Par contre, compte tenu de son fonctionnement par propagation et affectation par voisinage, cette approche peut r pondre naturellement   la dynamique (e.g. modification de contrainte, changement de fr quences admissibles pour un agent) comme dans ETTO.

Nous avons utilis  la m me approche dans le d veloppement d'un algorithme par AMAS pour le contr le manufacturier : DAMASCOP [CLAIR et al., 2008b ; KADDOUM, 2008]. Dans cette approche, par agentification du domaine, des agents repr sant les stations, les op rateurs et les conteneurs n gocient pour trouver une affectation respectant un planning de commandes,  voluant dans le temps.

Comportement de concession minimale

Dans CASANOVA, pour r soudre des probl mes d'appariements stables (mariages stables plus pr cis ment), nous n'avons pas suivi une approche coop rative, mais une strat gie de *concession minimale* ROSENSCHEIN et al. [1994], bas e sur le regret des agents, pour leur permettre de faire  merger un appariement stable et  quitable. Selon cette strat gie, les agents se proposent   leurs partenaires pr f r s. En cas d' chec, un agent conc de, ce qui se traduit par l'abandon de ses exigences vis- -vis des partenaires potentiels. Ainsi, au d but, un agent envoie une proposition   son unique partenaire pr f r , et si ce dernier refuse l'agent conc de et donc envoie une nouvelle proposition aux deux partenaires qu'il pr f re, etc.

Nous pouvons repr senter le comportement des agents gr ce   un automate fini d terministe compos  de cinq  tats : libre (free), h sitant (hesitating), fianc  (engaged), mari  fid le (faithful), et mari  infid le (unfaithful), comme illustr  dans la figure 5.7. Un agent peut envoyer diff rents messages (propose, accept, reject, confirm, withdraw et divorce) et y r agira diff remment en fonction de son  tat interne (c libataire ou non, niveau de concession, etc.). Les agents vont ainsi alterner des phases de mariage et de c libat jusqu'  atteindre un appariement stable.

CASANOVA permet d'atteindre des solutions plus équitables (entre communautés) et optimales (au sens du bien-être utilitaire) que les algorithmes centralisés classiques, ne respectant pas la confidentialité des préférences [EVERAERE et al., 2012]. Par contre, dans certains cas dans lesquels les listes de préférences présentent des cycles, CASANOVA boucle. Nous avons étudié ce problème et fourni des solutions — distribuables mais non décentralisées — dans [PIETTE et al., 2013].

Applications et résultats. Nos travaux dans le domaine de protocoles de résolution de problèmes par auto-organisation ont été appliqués au problème d'affectation de fréquences et testés sur le benchmark FAPP [PICARD et al., 2007b]. Une approche similaire par coopération, DAMASCO, a également été développée pour le contrôle manufacturier, et a exhibé les meilleures performances en temps, en qualité et en adaptativité, par rapport aux approches DCOP classiques [CLAIR et al., 2008b ; KADDOUM, 2008]. L'approche CASANOVA quant à elle a été initialement conçue pour la recherche de mariages stables [EVERAERE et al., 2012], qui représente un point de départ vers la résolution de problèmes comme le covoiturage ou le *homeswapping*. Des investigations sur d'autres algorithmes de résolution de tels problèmes (de manière synchrone et non auto-organisatrice) ont été poursuivies dans la continuité, notamment pour résoudre des problèmes de blocages dans CASANOVA [PIETTE et al., 2013 ; EVERAERE et al., 2013]. Nous avons également des investigations en cours pour l'application de telles techniques dans le domaine des *smart grids*, dans le cadre du projet ITEA SEAS^a.

a. <https://itea3.org/project/seas.html>

5.4 Synthèse et analyse

Nous avons présenté dans ce chapitre des approches multi-agents de résolution de problèmes distribués et/ou dynamiques dans lesquelles les agents se coordonnent soit via un environnement partagé, soit grâce à l'utilisation de messages directs et de sessions. L'organisation de ces agents est *explicite* (e.g. réseau de contraintes couplé aux priorités des agents, position dans l'espace couplé aux marqueurs de conflits) mais jamais perceptible dans sa totalité par les agents — les agents ne raisonnent que sur des *informations locales*, ou sur leur *environnement direct* (par exemple, les agents ne connaissent que les valeurs avec des agents partageant des contraintes avec eux). Les agents modifient également l'organisation en *changeant de position* dans leur domaine de recherche et en *changeant leur priorité* explicitement ou par le biais de la notion de *criticité*. Dans le cas de la négociation dans l'affectation de fréquences, cette (auto-)organisation se traduit en l'*élection* de l'agent pouvant changer sa valeur. Nous pouvons donc placer ces travaux à la frontière des cadrans (a) et (b) de la figure 2.1 : ce sont des **approches centrées agent avec vision locale de l'état de l'organisation**. Les capacités d'*auto-organisation* des agents permettent ici au système de s'adapter aux changements de la spécification du problème en cours de résolution (e.g. ajout de contraintes) et aux changements de valeurs des agents (e.g. un utilisateur change ses préférences pour l'emploi du temps, en cours de résolution). Cette résolution *anytime* est mise en œuvre par un *processus continu de réparation et d'amélioration* de l'état courant de l'organisation. Suivant une agentification *ad-hoc* ou par décomposition du problème, l'approche par systèmes multi-agents adaptatifs par auto-organisation a ainsi permis de mettre en place des capacités de *robustesse à la dynamique*, de *distributivité* et éventuellement de *confidentialité*.

Ces travaux ont été menés lors d'encadrement de masters recherche : Florian Cornet [CORNET, 2005], Gaël Clair [CLAIR, 2008], Elsy Kaddoum [KADDOUM, 2008], Santiago Villarreal [VILLARREAL, 2010], et Lucas Cerqueira-Martins [CERQUEIRA-MARTINS, 2012]. Ils ont également été le fruit de collaborations avec l'IRIT (Jean-Pierre Georgé, Marie-Pierre Gleizes et Pierre Glize) et le LIFL (Maxime Morge et Patricia Everaere). Certaines de ces recherches ont été initiées dans des groupes de travail français (COLLINE) ou européen (TFG SO), et ont été l'occasion de mettre en place une plateforme

de simulation pour le contrôle manufacturier (MASC²).

Les publications et communications suivantes sont accessibles pour plus de détails sur tous ces travaux : [PIETTE et al., 2013 ; EVERAERE et al., 2013 ; EVERAERE et al., 2012 ; MORGE et al., 2011 ; PICARD et MORGE, 2011 ; GEORGÉ, M.-P. GLEIZES et al., 2010 ; KADDOUM, M.-P. GLEIZES, GEORGÉ et PICARD, 2009 ; KADDOUM, M.-P. GLEIZES, GEORGÉ, GLIZE et al., 2009 ; PICARD et al., 2009b ; CLAIR et al., 2008b ; CLAIR et al., 2008a ; PICARD et al., 2007b ; PICARD et al., 2007a ; PICARD et GLIZE, 2006 ; PICARD et M.-P. GLEIZES, 2006 ; PICARD et GLIZE, 2005b ; PICARD et M.-P. GLEIZES, 2005b ; PICARD, BERNON et M.-P. GLEIZES, 2005a ; PICARD et GLIZE, 2005c ; PICARD, 2005b ; PICARD et GLIZE, 2005a ; PICARD, BERNON et M.-P. GLEIZES, 2005b].

2. <http://www.irit.fr/COLLINE/>

Troisième partie

Ingénierie des systèmes multi-agents adaptatifs

Ingénierie orientée multi-agent : un positionnement

Ce chapitre présente un positionnement de nos travaux dans le domaine de l'ingénierie orientée multi-agent et de l'adaptation. Les sections 6.1 et 6.2 présentent respectivement des modèles d'agents et de systèmes multi-agents, des points de vue architecture et programmation. La section 6.3 aborde la question méthodologique, au travers des méthodes associées à certains modèles présentés dans la section précédente. Enfin, la section 6.4 discute de la place de l'adaptation dans les différents modèles et architectures présentés dans ce chapitre.

6.1 Modèles et architectures d'agents

La définition du concept d'agent n'étant pas consensuelle, de nombreux modèles et architectures d'agents ont vu le jour depuis les débuts de ce champ d'étude. Cependant, le comportement interne d'un agent est souvent représenté comme un cycle de vie perception-décision-action, comme illustré par le modèle de la figure 6.1. Lors de la phase de perception, l'agent met à jour ses connexions avec l'environnement (senseurs, capteurs ou boîte aux lettres). Lors de la phase de décision, les perceptions permettent à l'agent de mettre à jour ses connaissances et croyances sur le monde, puis de décider d'une action à exécuter pendant la phase d'action. Les architectures d'agent sont donc principalement composées de trois modules — un pour chaque phase de ce cycle de vie. L'implantation de ce cycle va mener à plusieurs familles de modèles d'agents, dont les deux extrêmes sont les agents *réactifs* et les agents *délibératifs* [WOOLDRIDGE et al., 1995]. Une synthèse récente sur ces modèles et architectures se trouve dans l'ouvrage de WEISS [2013].

Architectures d'agents réactifs. De nombreux modèles réactifs ont été proposés, souvent inspirés par la biologie ou l'éthologie. On trouve des modèles probabilistes de comportement, dans lesquels chaque perception va influencer sur la distribution de probabilités des actions à effectuer [BONABEAU et al., 1999], comme dans les travaux menés au LORIA [CHEVRIER, 2002 ; CHEVRIER et al., 2011]. L'architecture ECO issue du paradigme de l'eco-résolution permet également de développer des agents réactifs guidés par une mesure de satisfaction et des capacités de pression sur les autres agents (agression) [FERBER, 1995]. L'approche AMAS fournit également une architecture d'agent coopératif, qui fait une distinction entre le module de décision nominal (fonctionnement classique) et le module de décision coopérative (pour réparer ou anticiper des situations de non coopération) [PICARD, 2004]. Enfin, IODA propose un cadre de conception orienté interaction pour les systèmes multi-agents, couplé à une architecture d'agent réactif [KUBERA et

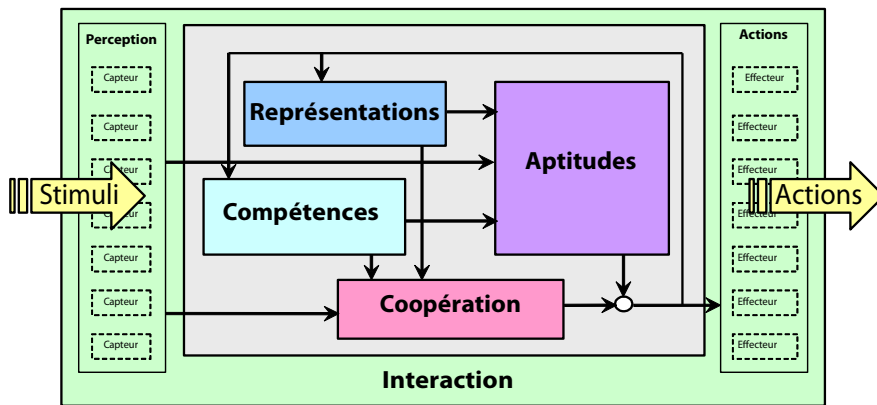


FIGURE 6.1 – Architecture d'un agent coopératif AMAS [PICARD, 2004] utilisé dans certains travaux de la partie II. La phase de décision de cet agent plutôt réactif repose sur des représentations du monde, des compétences (e.g. base de faits d'expertise métier), des aptitudes (des capacités d'inférence sur les compétences) et un module de coopération qui devient prioritaire lorsque l'agent se trouve en situation non coopérative.

al., 2008]. Tous ces modèles n'ont pas fait forcément l'objet d'effort d'ingénierie logicielle et seules quelques plateformes ou bibliothèques implantent de tels modèles : MadKit¹, IODA² ou MAY³ pour ne citer qu'elles. D'autres plateformes, plus orientées simulation multi-agent, reposent sur des architectures d'agents peu contraignantes, comme NetLogo⁴ ou Repast⁵.

Architectures d'agents délibératifs. Dans le livre de WEISS [2013], plusieurs modèles d'agents sont proposés, et contribuent chacun, plus ou moins, à fournir des capacités délibératives (de haut niveau) aux agents. Par exemple, des modèles BDI ou par décision markovienne sont conseillés dans des problèmes nécessitant des capacités de planification. Un exemple d'environnement de programmation proposant une implantation de BDI est Jason⁶. Il est également possible d'équiper les agents de modules d'apprentissage par renforcement ou d'apprentissage statistique, afin de permettre aux agents de modifier leur comportement par observation de l'évolution de leur environnement. Enfin, il est possible de munir les agents de capacités d'interaction et de communication par messages, suivant des spécifications comme KQML [FININ et al., 1994] ou FIPA-ACL [POSLAD, 2007]. Certains peuvent même avoir des raisonnements sur l'organisation.

Architectures hybrides. Il est bien sûr possible de concevoir des agents qui possèdent des caractéristiques propres au réactif *et* au délibératif. Par exemple, il est possible de munir des agents réactifs de capacité de communications élaborées, comme dans l'exemple FAPP du chapitre 5. On peut également imaginer des agents munis de capacité de planification, capable également de réagir à des imprévus, comme en robotique, avec l'architecture hybride proposée par ARKIN [1998]. Le modèle d'agent AMAS présenté dans [PICARD, 2004] peut être assimilé à une architecture hybride par subsomption [BROOKS, 1986], dans laquelle un comportement coopératif prend le dessus sur un autre comportement nominal (réactif ou délibératif) pour installer les capacités de coopération nécessaires aux agents.

1. <http://www.madkit.org>
 2. <http://www.lifl.fr/SMAC/projects/ioda/>
 3. <http://www.irit.fr/MAY>
 4. <https://ccl.northwestern.edu/netlogo/>
 5. <http://repast.sourceforge.net>
 6. <http://jason.sourceforge.net>

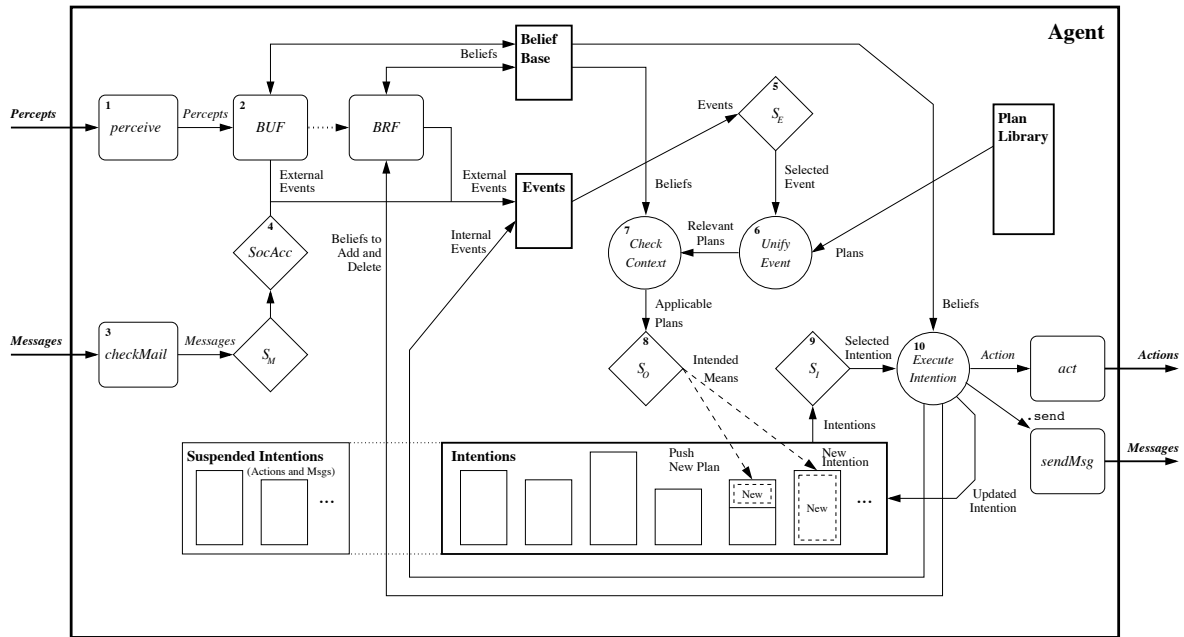


FIGURE 6.2 – L'architecture d'agent BDI [RAO et al., 1995] telle qu'implantée dans le langage de programmation agent Jason que nous avons utilisé dans certains de nos travaux [YAICH et al., 2013 ; EVERAERE et al., 2012 ; PERSSON et al., 2012b ; SORICI, PICARD, BOISSIER, SANTI et al., 2012 ; SORICI, BOISSIER, PICARD et SANTI, 2011].

Positionnement (Modèles agents). Dans nos travaux sur la résolution de problèmes (voir chapitres 4 et 5), nous avons utilisé des modèles d'agents *réactifs*, soit avec le modèle d'agent AMAS issu de mes travaux de thèse PICARD [2004], soit de manière *ad-hoc* — par exemple, dans le cadre du projet ID4CS (thèse de Diane Villanueva et post-doc de Jérémy Rivière), compte tenu du contexte applicatif et technologique, nous avons utilisé des environnements de développement comme SciLab ou MatLab, non dotés de bibliothèques ou d'architectures d'agents. Le modèle AMAS a également été l'objet d'un travail sur la transformation de modèles (approche dirigée par les modèles) pour la génération automatique de code [OTTENS et al., 2006], poursuivi plus tard par d'autres travaux comme la thèse de Sylvain Rougemaille [ROUGEMAILLE, 2008] ou dans le cadre de la plateforme MAY. Nous avons également utilisé des agents communicants avec JADE, ou des plateformes reposant sur l'intégriciel de JADE, pour certains développements comme pour CASANOVA [EVERAERE et al., 2012 ; CERQUEIRA-MARTINS, 2012 ; VILLARREAL, 2010]. Cependant, nous verrons dans la suite de cette partie que pour le développement d'agents intégrés dans des organisations explicites, spécifiées en MOISE, nous avons fait le choix de modèles d'agents plus *délibératifs* BDI, programmés en Jason qui est intégré dans la plateforme de programmation multi-agent JaCaMo^a.

a. <http://jacamo.sourceforge.net>

6.2 Programmation et architectures de multi-agents

Les systèmes multi-agents ne se résument par forcément à un ensemble d'agents interagissant et nombres de concepts complémentaires peuvent être utiles pour développer un système multi-agent. Les agents peuvent être déployés dans un environnement (physique ou virtuel) ou une organisation dans laquelle ils jouent des rôles particuliers, par exemple. Généralement les plateformes multi-agents fournissent des métamodèles — à instancier pour pouvoir développer une application particulière — regroupant les concepts clés définissant l'architecture du système (agent, groupe, organisation, rôle,

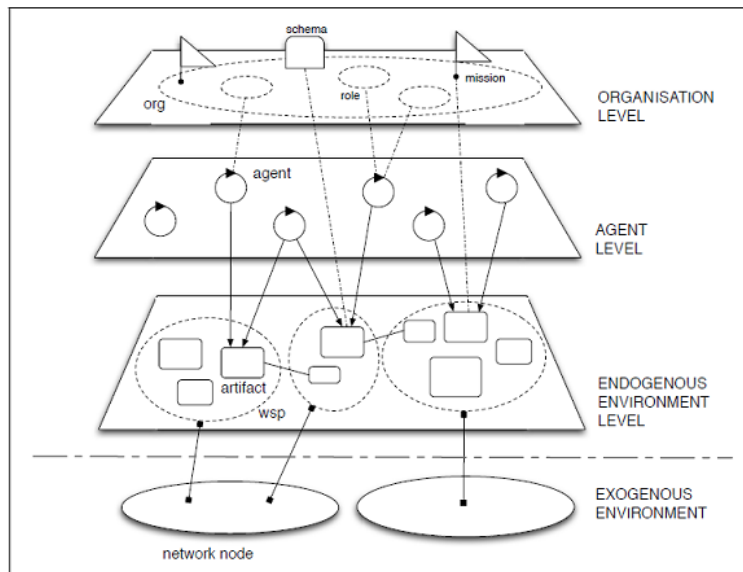


FIGURE 6.3 – L'architecture multi-agent en trois couches (organisation, agents, environnement) de JaCaMo [BOISSIER, BORDINI et al., 2013], utilisée dans nos travaux [YAICH, 2013 ; PERSSON et al., 2012b ; SORICI, BOISSIER, PICARD et SANTI, 2011 ; SORICI, PICARD, BOISSIER, SANTI et al., 2012].

etc.). Comme nous l'avons vu dans la section 2.2, la notion d'organisation pouvant être explicite, elle doit pouvoir être une entité manipulable par les agents, et nécessite alors d'être modélisée et instanciée ; de même pour le concept d'environnement, comme l'illustre la figure 6.3.

Dimension organisation. Lorsque l'on ajoute explicitement la notion d'organisation dans le système, l'architecture s'en trouve fortement impactée, compte tenu de l'aspect structurant et fonctionnel d'une organisation. Comme nous l'avons vu dans la section 2.2.3, de nombreuses contributions dans le domaine proposent des environnements de spécification de modèles organisationnels et éventuellement d'exécution de ces modèles : AGR [FERBER et GUTKNECHT, 1998], TAEMS [V. LESSER et al., 2004], STEAM [TAMBE, 1997], *MOISE*⁺ [HÜBNER et al., 2002], ISLANDER [ESTEVA et al., 2001] ou le métamodèle CRIO dans la plateforme JANUS⁷ [COSSENTINO, GAUD, HILAIRE et al., 2010]. Dans *MOISE*⁺, par exemple, les agents ont accès à l'organisation pour faire état de leur souscription à des rôles, l'avancement de leurs plans, ou l'atteinte de leurs objectifs. Il est également possible d'avoir accès à l'état des normes régissant le système. On peut regrouper des travaux dans la famille des approches de programmation orientée organisation, car elles fournissent les outils de spécification et les primitives de manipulation des organisations [PYNADATH et al., 1999].

Dimension environnement. L'environnement est à la fois un support à l'exécution et à l'interaction. Il peut donc être considéré comme l'une des parties essentielles de l'architecture d'un système multi-agent. Si la définition de l'environnement ne pose pas de restriction, de nombreuses mises en œuvre existent selon une dimension choisie pour l'environnement : l'environnement physique [WEYNS et al., 2007 ; ODELL et al., 2002 ; GALLAND et GAUD, 2014 ; BADEIG et al., 2012], l'environnement social [SAUNIER et al., 2014], et l'environnement d'exécution [RODRIGUEZ et al., 2014]. L'idée est de placer, ou de déplacer certaines fonctionnalités attendues du système dans l'environnement de ces derniers. Ici, donc, l'environnement n'est pas juste une donnée, mais bien un artefact (ou un ensemble d'artefacts) à spécifier, concevoir et déployer. Cette entité de premier ordre doit alors être programmée — on peut alors parler de programmation orientée environnement [RICCI et al., 2011], adoptée dans la plateforme CARTAGO⁸.

7. <http://www.janus-project.org>

8. <http://cartago.sourceforge.net>

Dimension interaction. Concevoir les protocoles de communication, ainsi que les intergiciels mettant en œuvre ces communications, consiste à fournir l'infrastructure d'interaction du système. La plateforme JADE⁹ propose un intergiciel et un modèle d'agent pour mettre en place de telles capacités, mais ne fournit que la syntaxe des performatives ACL, et non la sémantique. Au-delà du simple médium de communication, la couche interactionnelle peut être envisagée comme une entité de premier ordre, comme dans IODA, dans laquelle la conception et l'architecture du système sont centrées sur une matrice d'interaction capturant toutes les dynamiques physiques et sociales pilotant le comportement du système multi-agent [KUBERA et al., 2008]. Cette approche peut être vue comme une programmation orientée interaction [HUHNS, 2001]. Dans une certaine mesure, l'approche AMAS peut-être vue comme une approche orientée interaction car le cœur de la conception d'un tel système multi-agent passe par la définition de règles de coopération et de gestion de situations non coopératives entre les agents eux-mêmes et les agents et leur environnement [GLIZE, 2001].

Toutes ces dimensions peuvent également être regroupées dans une approche de programmation orientée multi-agent (*multi-agent oriented programming*, retrouvée dans la plateforme JaCaMo [BOISSIER, BORDINI et al., 2013], illustrée en figure 6.3.

Positionnement (Modèles multi-agents). Les travaux présentés dans la partie précédente, comme nous avons pu le voir, ont abordé la résolution de problèmes par auto-organisation, et donc ont fait appel à des *architectures de systèmes émergents*. L'environnement était parfois purement social (FAPP ou CASANOVA, voir section 5.3) ou physique, i.e. un médium de coordination ou de communication indirecte (ETTO, voir section 5.2). Dans les travaux postérieurs, notamment les travaux de thèses de YAICH [2013] et de PERSSON [2014], nous avons par contre utilisé des systèmes multi-agents organisationnels, basés sur MOISE, et implantés en JaCaMo. Cette approche nous permet en effet de spécifier l'organisation et de la rendre manipulable par les agents à des fins d'adaptation, comme nous le présenterons dans les chapitres à venir. Les architectures résultantes sont essentiellement composées de trois couches : organisation, agent, environnement (artefacts). Ces travaux ont porté sur des problématiques de communautés virtuelles, d'intelligence ambiante et de réseaux *machine-to-machine* dans le cadre du projet FUI SensCity.

6.3 Les méthodes orientées agents et multi-agents

Passer d'un cahier des charges, à des modèles d'agents et de systèmes multi-agents, puis à la réelle implantation en vue de déploiement, peut être fastidieux pour des ingénieurs non spécialistes du domaine des SMA. Pour cette raison, depuis la fin des années 1990, de nombreuses méthodes de développement de systèmes multi-agents on vu le jour BERNON, M.-P. GLEIZES et PICARD [2009], HENDERSON-SELLERS et al. [2005] et WEISS et al. [2004], inspirées par de multiples domaines de l'ingénierie logicielle, comme l'illustre la figure 6.4. Nous n'allons pas présenter de panorama de ces approches, mais redirigeons plutôt le lecteur vers des synthèses déjà établies dans nos travaux [M.-P. GLEIZES et al., 2008 ; BERNON, M.-P. GLEIZES et PICARD, 2009], qui comparent ces méthodes suivant leurs processus, leurs métamodèles, leurs notations et leurs outils de support. Cependant, aucune méthode n'a réellement émergé du lot. Nous pouvons attribuer ceci à (i) la spécialisation de la plupart des méthodes pour un modèle d'agent et/ou un modèle multi-agent (e.g. ADELFE pour les AMAS, ASPECT pour CRIO) ; (ii) le manque d'outils pour passer des modèles à la programmation.

9. <http://jade.tilab.com>

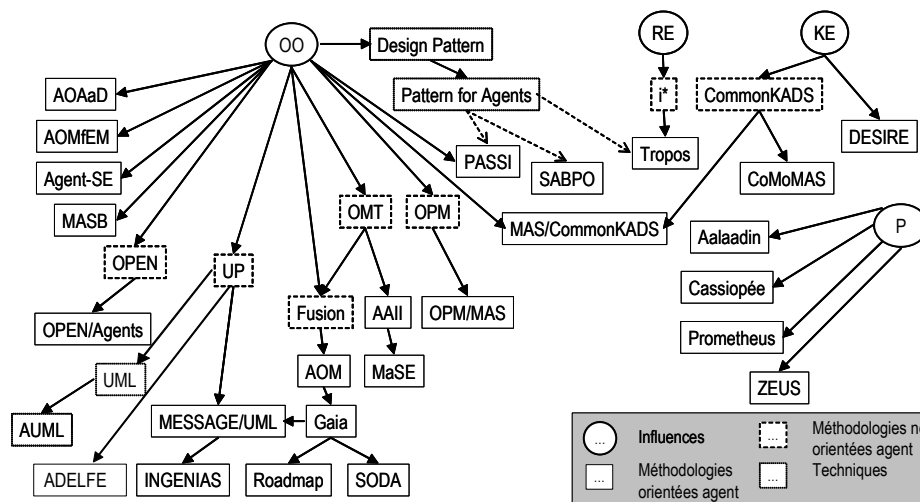


FIGURE 6.4 – De la richesse des méthodes orientées agents, en 2004 [PICARD, 2004], s’inspirant de l’orienté objet (O), de l’ingénierie des besoins (RE), de l’ingénierie des connaissances (KE) et des plateformes de simulation (P).

Positionnement (Méthodes de développement). Mes travaux de thèse ont contribué au mouvement des méthodes orientées multi-agent en proposant la méthode ADELFE dédiée aux systèmes multi-agent adaptatifs par auto-organisation coopérative [PICARD, 2004 ; BERNON et al., 2005 ; PICARD et M.-P. GLEIZES, 2004] appliquée à la robotique collective, à la gestion d’emplois du temps ou à la conception mécanique [PICARD, BERNON, CAMPS et al., 2003 ; PICARD et M.-P. GLEIZES, 2003 ; CAPERA et al., 2004]. Les travaux présentés dans ce mémoire n’ont pas contribué à ce mouvement, mais ont plutôt utilisé les résultats, notamment dans le développement d’applications en suivant la méthode ADELFE. Concernant les travaux de conception de systèmes multi-agents organisationnels [YAICH, 2013 ; PERSSON et al., 2012b], nous avons suivi une approche de décomposition du systèmes en dimensions à la *Voyelles* [DEMAZEAU, 1995], illustrée en figure 6.5. Elle se prête idéalement au MAOP, qui repose sur les quatre dimensions : agent, environnement, organisation et l’interaction. Cependant, nous n’avons pas suivi de processus de développement systématique ni utilisé d’environnement de conception, car aucun n’existe, pour l’instant, pour une approche MAOP.

6.4 L’adaptation dans l’ingénierie multi-agent

Dans le but d’installer des capacités d’adaptation dans le systèmes multi-agent à concevoir, il est nécessaire de se pencher sur la définition des concepts implantant ces capacités. Comme nous l’avons vu dans le chapitre 2, nous considérons que l’adaptation du système passe par l’adaptation de son organisation, qu’elle soit explicite ou non. Cependant, cette capacité n’est pas forcément localisée ou à mettre sous la responsabilité de la couche organisationnelle du système — d’autant plus lorsque celle-ci n’est pas explicitement présente. Par exemple, dans les AMAS, que nous pourrions construire avec seulement trois dimensions (agent, interaction et éventuellement environnement), l’adaptation vient se loger principalement dans le comportement coopératif (agent), dans la capacité à se coordonner pour prendre en compte la criticité (interaction), en faisant éventuellement appel à l’environnement pour cette coordination. Dans l’approche OR4MAS [J. HÜBNER, BOISSIER et al., 2010], suivant une modélisation organisationnelle *MOISE*, il est également possible d’instrumenter l’environnement, au moyen d’artefacts, pour manipuler et donc adapter l’organisation — e.g. les agents peuvent entrer dans une organisation, changer de rôle, redémarrer un plan, etc. De tels artefacts, directement manipulables par les agents sont illustrés dans la figure 6.6. L’environnement peut également être le vecteur de l’adapta-

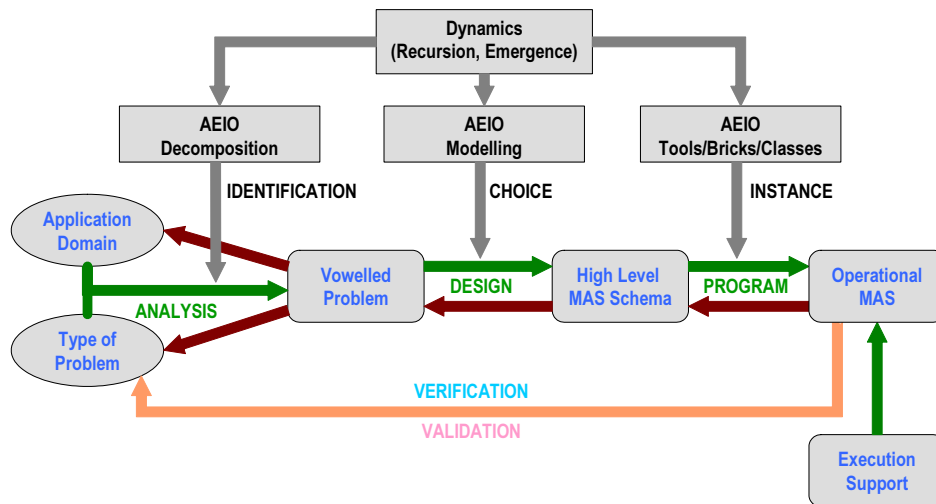


FIGURE 6.5 – Le processus de la méthode Voyelles [DEMAZEAU, 1995], qui consiste principalement en la décomposition du problème en quatre dimensions (Agent, Environnement, Interaction et Organisation), puis l’instanciation de ces dimensions dans des technologies cibles choisies — e.g. JaCaMo.

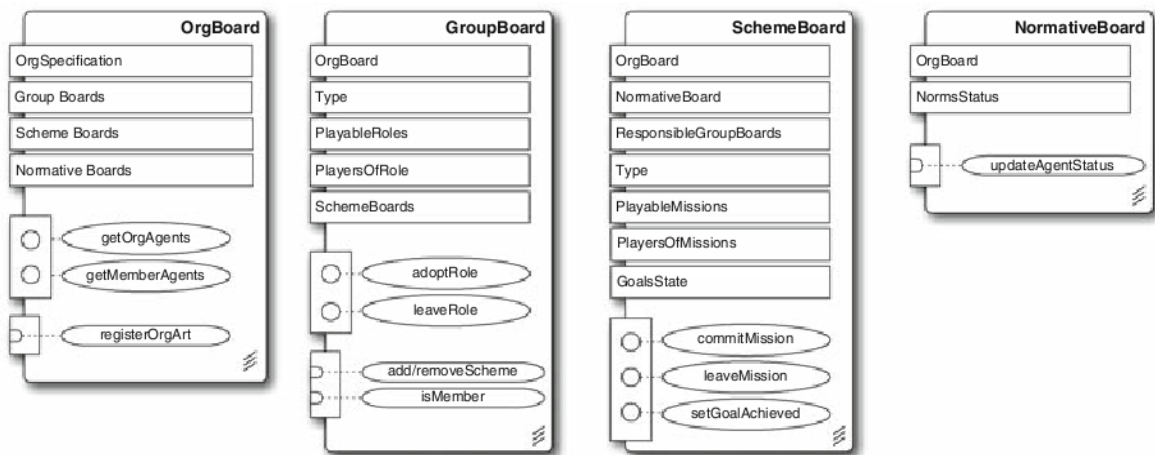


FIGURE 6.6 – Les artefacts d’ORA4MAS mettant en place dans l’environnement des moyens de manipuler les trois dimensions de l’organisation [J. HÜBNER, BOISSIER et al., 2010] : structurelle (GroupBoard), fonctionnelle (SchemeBoard) et normative (NormativeBoard).

tion, dans des approches agnostiques de l’organisation, de par sa dynamique propre. Par exemple, dans des approches auto-organisatrices par dépôt de phéromones, c’est bien la capacité donnée à l’environnement de faire évaporer des informations qui donne la capacité au système multi-agent à s’adapter à des perturbations, comme la disparition de routes [BONABEAU et al., 1999].

Positionnement (Ingénierie de l'adaptation). La partie précédente a illustré la manière de développer des solveurs adaptatifs reposant sur des comportements d'agents coopératif et/ou un environnement de coordination (voir partie II). Cependant, nous n'avons pas adopté dans ces travaux une approche *ingénierie de l'environnement*, mais plutôt une *conception orientée agent et interaction*. Dans les travaux de thèse de YAICH [2013], les agents forment des communautés (groupes dans une organisation) et sont capables créer et de modifier les règles de confiance les régissant. Ceci place clairement ces travaux dans une approche d'*adaptation organisationnelle*. Dans les travaux de PERSSON et al. [2012b], les agents sont capables de se réorganiser, de modifier certaines propriétés de l'organisation (e.g. des cardinalités de rôles), ou de créer de nouvelles organisations correspondant à de nouvelles exigences applicatives de la plateforme M2M, en suivant l'approche ORA4MAS — nous sommes ici dans une *adaptation organisationnelle coordonnée*.

Nous allons présenter dans les deux chapitres suivants des travaux d'ingénierie *architecturale* de systèmes multi-agents s'adaptant par réorganisation et auto-réorganisation. Plus que les aspects algorithmiques que nous avons pu voir dans la partie précédente, nous allons nous intéresser aux modèles organisationnels mis en place pour concevoir des applications pour la gouvernance de réseaux M2M [PERSSON et al., 2012b] ou la gestion de la confiance dans des communautés virtuelles [YAICH et al., 2013].

Approche organisationnelle de la réorganisation

Alors que dans les travaux des chapitres précédents, les agents ne disposaient pas de modèle explicite de l'organisation — ou sinon de manière très limitée avec la notion de *voisinage* —, les travaux de ce chapitre reposent sur une spécification explicite de l'organisation sur laquelle les agents peuvent raisonner, et qu'ils peuvent manipuler afin de l'adapter. Dans ce chapitre nous allons donc adopter un *point de vue orienté organisation*. L'agentification dépendra du domaine d'application, et consistera ici principalement à fournir à des agents génériques le choix d'adopter des rôles correspondant à des tâches et objectifs spécifiques.

Nous posons le cadre théorique de l'approche organisationnelle adoptée dans ces travaux, à savoir le modèle *MOISE*, qui s'appuie sur trois facettes complémentaires pour spécifier les organisations : structurelle, fonctionnelle et normative. La section 7.1 va brièvement conceptualiser la notion de *réorganisation* dans nos travaux, au regard de la définition 1 présentée en section 2.4. La section 7.2 présente nos travaux sur la définition et la mise en œuvre d'une organisation de la réorganisation, appliquée à l'intelligence ambiante.

7.1 Approche *MOISE* de l'adaptation

Dans ce contexte, nos travaux ont reposé sur une modélisation organisationnelle exprimée en *MOISE* [HÜBNER et al., 2002]. Les trois facettes modulaires (structurelle, fonctionnelle et normative) sont autant de leviers pour le changement de l'organisation.

La dimension structurelle décrit l'organisation en terme de *groupes composés de rôles* impliqués dans le fonctionnement du systèmes multi-agent. Pour chaque rôle, une cardinalité d'agents engagés dans le rôle est spécifiée. La dimension structurelle définit également les relations d'accointances, de communication, de compatibilité et de hiérarchie entre les rôles.

La dimension fonctionnelle définit les objectifs du système multi-agent comme un ensemble de *schémas sociaux*. Un schéma est une décomposition arborescente d'un but social en sous-buts qui sont sous la responsabilité d'un groupe. Les ensembles consistants de buts sont regroupés en missions. Buts et missions peuvent être qualifiés avec une cardinalité d'agents (nombre d'agents nécessaires pour la mise en œuvre) et en temps limite de complétion.

La dimension normative sert d'interface entre structurel et fonctionnel en définissant des normes qui sont utilisées pour affecter des missions aux rôles. Une norme est composée (i) d'une condition contextuelle de déclenchement de la norme; (ii) d'une relation déontique (i.e. permission ou

obligation) d'engagement; (iii) d'une mission; (iv) d'un rôle. Il est également possible de spécifier un temps limite de prise en charge de la mission, avant que la norme soit considérée comme violée.

Ces trois dimensions du modèle *MOISE* permettent une définition flexible, et adaptable, de l'organisation. Il est possible de considérer l'adaptation à deux niveaux dans ce cadre : au niveau des instances, et au niveau du modèle. Au *niveau instance* (ou entité, noté OE), l'adaptation signifie que les agents vont rester dans le cadre spécifié par le modèle, mais vont toutefois changer la façon dont le modèle est mis en œuvre, en changeant par exemple de rôle, de mission, etc. En effet, la même mission peut être affectée à différents rôles dans différents contextes. Au *niveau spécification* (noté OS), il est possible de considérer la spécification organisationnelle comme un mutable que les agents vont manipuler et adapter. Il est ainsi possible de redéfinir la structure afin de renforcer ou de relâcher des contraintes sociales. Dans le même ordre d'idée, les schémas fonctionnels peuvent être étendus ou restreints pour changer les degrés de liberté des agents — car les agents peuvent raisonner sur les contraintes organisationnelles et décider de les respecter ou non. Enfin, il est également possible de considérer l'adaptation d'un système par *l'ajout de nouvelles organisations* (donc de modifier les trois dimensions en même temps), avec chacune leur propre spécification. Nous illustrerons cette approche *multi-organisation* dans le chapitre suivant.

Exemple (Réorganisation niveau instance par allocation de rôles). Une des premières idées pour adapter un système dans lequel l'organisation est explicite est de modifier l'affectation des rôles, afin de la rendre plus adéquate au contexte actuel. Nous avons adopté ce point de vue afin de permettre à des agents AMAS (raisonnant sur la criticité, voir section 5.2.2) de changer leur rôle et de gérer leur affectation en utilisant le principe de coopération dans le cadre de la gestion de ressources [PICARD, MELLOULI et al., 2005]. A chaque fois qu'un agent fait face à une situation non coopérative due à des conflits de ressources, il déclenche une réorganisation par changement de rôle.

Cette approche peut être pleinement implantée en utilisant une approche organisationnelle à la *MOISE*, en fournissant le moyen de coordonner l'affectation des rôles; en utilisant des artefacts dédiés, par exemple.

Algorithme 6 : Un algorithme centralisé d'affectation de rôle en fonction de la disponibilité de ressources et de la compatibilité des agents [PICARD, MELLOULI et al., 2005].

R un ensemble de rôles à attribuer;

pour chaque $r_i \in R$ **faire**

 vérifier les ressources et actions interdites à r (normes);
 éliminer tous les agents ayant accès aux ressources interdites;
 éliminer tous les agents qui n'ont pas accès aux actions nécessaires;

si aucun agent n'est disponible alors

 Déterminer les agents auquel le rôle r_i peut être affecté;
 pour chaque agent non disponible faire évaluer le coût de sa réactivation;
 réactiver l'agent de coût minimal

sinon

 choisir l'agent qui jouera le rôle r_i ;
 affecter le rôle à l'agent choisi

Pour la mise en œuvre et le développement de systèmes suivant l'approche *MOISE*, le langage OML (*Organisation Modelling Language*) permet de spécifier les trois dimensions organisationnelles. Un langage de définition de normes est également fourni, NOPL (*Normative Organisation Programming Language*), en complément d'OML [J. F. HÜBNER et al., 2010].

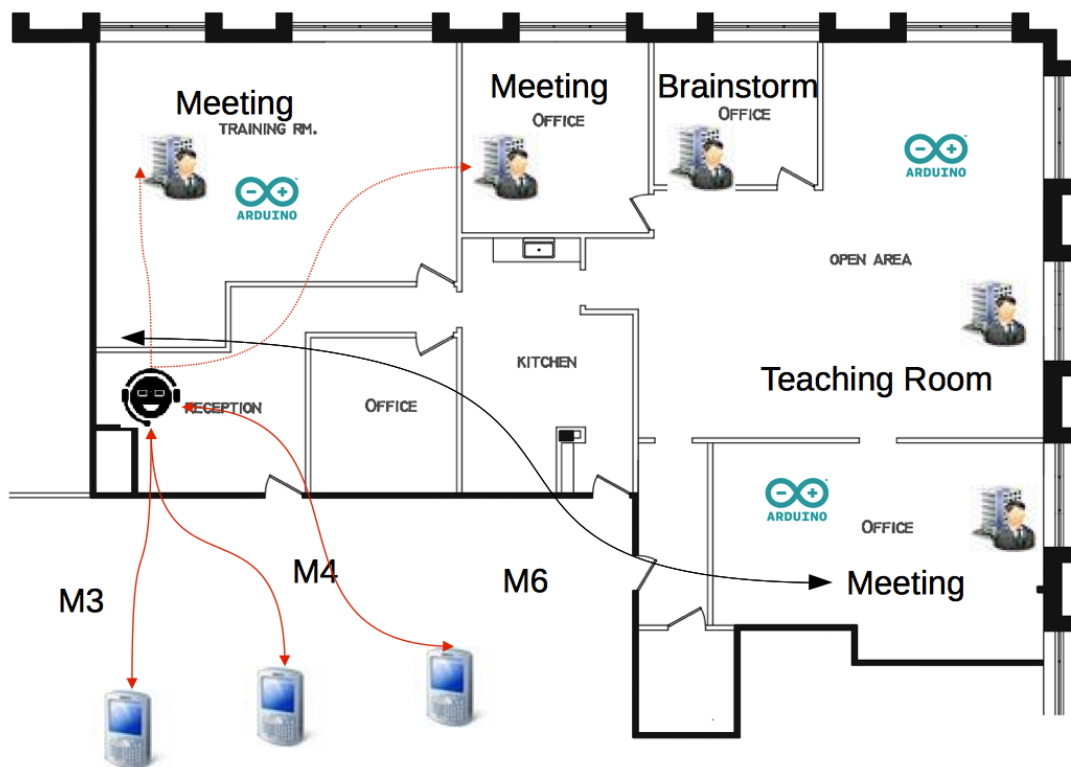


FIGURE 7.1 – Le scénario des salles de réunion intelligentes : en fonction des besoins et de leur occupation (mesurée par des capteurs Arduino), la réservation des salles ainsi que leur rôle (réunion, brainstorm ou enseignement) changent. Les utilisateurs peuvent émettre des préférences, et sont notifiés directement sur leur agenda personnel des changements de salles.

Applications et résultats. Nous avons adopté les deux approches (niveau instance et niveau spécification) dans nos travaux sur l'adaptation d'organisations explicites. Au niveau instance, dans les travaux de thèse de Camille Persson [PERSOON et al., 2012b] ou de master d'Alexandru Sorici, les agents changent l'adoption de rôles en cours de déroulement des schémas fonctionnels, afin d'effectuer un équilibre de charge (d'occupation des salles ou de débit de données provenant de capteurs). Nous avons suivi une telle approche également dans une étude autour d'ETTO, où les types d'agents sont reconsidérés comme des rôles jouables par différents agents « génériques » [PICARD, MELLOULI et al., 2005]. Au niveau modèle, les travaux de Persson et Sorici ont également proposé la modification de la structure par les cardinalités des rôles afin d'adapter une spécification a priori trop contraignante, dans le cadre de l'intelligence ambiante [SORICI, BOISSIER, PICARD et SANTI, 2011 ; SORICI, PICARD, BOISSIER, SANTI et al., 2012] (voir figure 7.1). Enfin, dans la thèse de Reda Yaich sur la gestion de la confiance dans les communautés virtuelles, les agents sont capables de créer de nouvelles communautés (des groupes), de créer et de modifier les normes régissant chaque communauté, par une participation active de leurs membres [YAICH, 2013].

7.2 Organisation de la réorganisation

Nous allons présenter dans cette section l'approche organisationnelle de la réorganisation, étendant les travaux de J. HÜBNER, J. SICHMAN et al. [2004]. En plus de spécifier une organisation de la réorganisation, nous avons également instrumenté ce processus en définissant un environnement composé

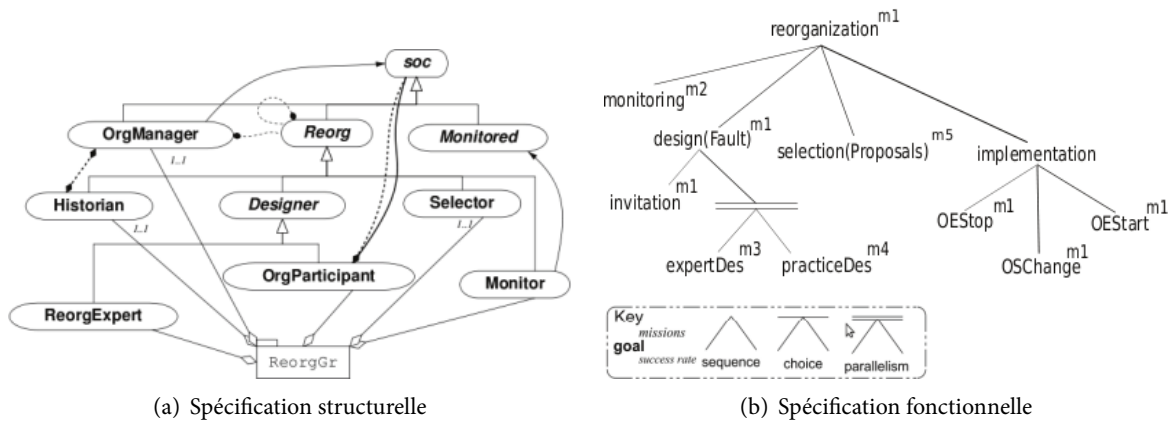


FIGURE 7.2 – Spécification de l’organisation de la réorganisation dans \mathcal{MOISE}^+ [J. HÜBNER, J. SICHMAN et al., 2004].

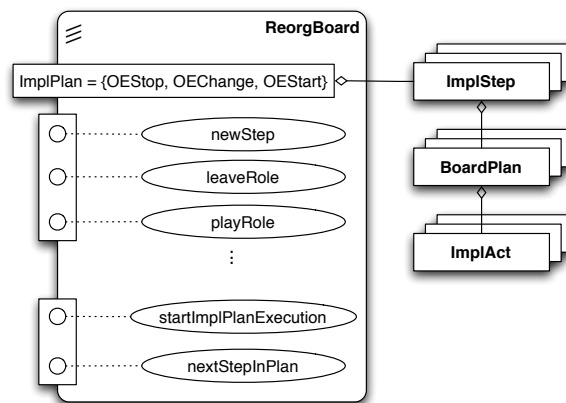


FIGURE 7.3 – L’artefact ReorgBoard permettant aux agents de mettre en œuvre la réorganisation du système [SORICI, PICARD, BOISSIER, SANTI et al., 2012].

d’artefacts de réorganisation, comme dans l’approche ORA4MAS [J. HÜBNER, BOISSIER et al., 2010]. Du point de vue ingénierie, une approche par transformation de modèles permet de passer d’une spécification organisationnelle en OML et NOPL à un ensemble d’artefacts de manipulation de cette organisation [J. F. HÜBNER et al., 2010].

Nous considérons ici l’adaptation comme étant une fonctionnalité explicite du système; il est alors possible, de manière *réflexive*, de spécifier dans l’organisation, cette fonctionnalité, en fournissant une structure, des normes et des schémas fonctionnels pour le mettre en œuvre. Ceci correspond à des travaux initiés dans \mathcal{MOISE}^+ , par exemple, dans lequel les phases de réorganisation telles que nous les avons abordées dans la section 2.3.3 sont spécifiées dans une organisation, comme illustré dans la figure 7.2(b). Ici, les agents peuvent jouer des rôles dans la réorganisation du système (voir figure 7.2(a)).

Nous nous sommes principalement intéressés aux phases de *conception* et de *mise en œuvre*. Dans le cadre de JaCaMo (voir section 6.2), nous avons instrumenté l’environnement des agents avec des artefacts dédiés à la réorganisation, permettant aux agents d’agir sur l’organisation et d’implanter les phases de réorganisation. Nous avons défini un nouveau type d’artefact, appelé le ReorgBoard (illustré dans la figure 7.3). Cet artefact est créé par les agents jouant le rôle de OrgManager (voir figure 7.2(a)), à chaque fois que la phase de *monitoring* est terminée, déclenchant la phase de *design* pour réparer l’échec ou la faute de fonctionnement (voir figure 7.2(b)).

Atteindre le but de *conception* consiste à définir un ensemble coordonné de modifications de l’organisation au niveau de la *spécification* ou des *instances*, pour chaque sous-but du but d’*implémentation*. Cet ensemble constitue un plan d’*implémentation*, noté ImplPlan, principalement découpé en

trois phases : *OEStop* (actions à mettre en œuvre pour stopper l'instantiation courante de l'organisation), *OSChange* (actions à mettre en œuvre pour modifier la spécification de l'organisation), *OESTart* (actions à mettre en œuvre pour remettre l'organisation en marche). L'*ImpPlan* est stocké dans le *ReorgBoard* et est ainsi accessible aux agents via un ensemble de propriétés observables de cet artefact. De cette manière, les agents en charge de mettre en place la nouvelle organisation auront le plan de mise en œuvre comme une croyance et l'exécuteront.

En plus de ces propriétés observables, l'interface d'usage du *ReorgBoard* propose deux ensembles distincts d'opérations pour aider les agents dans la réorganisation. Les actions de *design* permettent de modifier l'organisation au niveau spécification (e.g. *addRole*, *removeGroup*) ou au niveau instance (e.g. *leaveMission*, *playRole*), comme synthétisé dans le tableau 7.1. Les opérations d'*implementation* consistent à démarrer un *ImpPlan* (*startImpPlanExecution(p)*) ou à passer à la prochaine étape du plan (*nextStepInPlan(p)*).

OS Level Operations	Description
<i>addRole</i>	add a role to a group specification
<i>removeRole</i>	remove a role from a group specification
<i>addMission</i>	add a mission to a scheme specification
<i>removeMission</i>	remove a mission from a scheme specification
<i>addRoleObligation</i>	add a normative assignment of a mission to a role
<i>removeAllRoleObligations</i>	clear a role of all its normative obligations
<i>changeRoleCardinality</i>	change the min..max interval of agents playing this role
<i>changeSubgroupCardinality</i>	change the min and max number for the instances of a subgroup
<i>removeGroup</i>	remove the given group specification from the organisation
<i>removeScheme</i>	remove the given scheme specification from the organisation
OE Level Operations	Description
<i>leaveRole</i>	target agent must leave the specified role
<i>playRole</i>	target agent must start playing the specified role
<i>leaveMission</i>	target agent must discontinue pursuing the goals in given mission
<i>commitMission</i>	target agent must commit to solving goals in given mission

TABLE 7.1 – Opérations de réorganisation, lors de la phase de *design*, au niveau instance (OE) et au niveau spécification (OS), fournies par l'artefact *ReorgBoard*.

Nous décrivons les différentes interactions prenant place entre le *ReorgBoard*, les agents impliqués dans la réorganisation et les instances de *GroupBoard* ou de *SchemeBoard* dans la figure 7.4. Ici, deux agents jouent respectivement les rôles d'*OrgManager* et de *Designer*. Tout d'abord l'agent jouant le rôle de *Designer* construit le plan de réorganisation pas à pas (étapes 1 et 2) en utilisant les actions élémentaires fournies par l'artefact *ReorgBoard*. Ensuite, l'agent jouant le rôle d'*OrgManager* démarre (étape 3) puis contrôle pas à pas (étapes 4 et 5) l'exécution du plan de réorganisation. Lors de l'exécution du plan, les agents impliqués par la réorganisation vont recevoir des notifications via leurs artefacts *GroupBoard* et *SchemeBoard*, afin d'éventuellement changer de rôle ou de mission (étapes 4.x et 5.x).

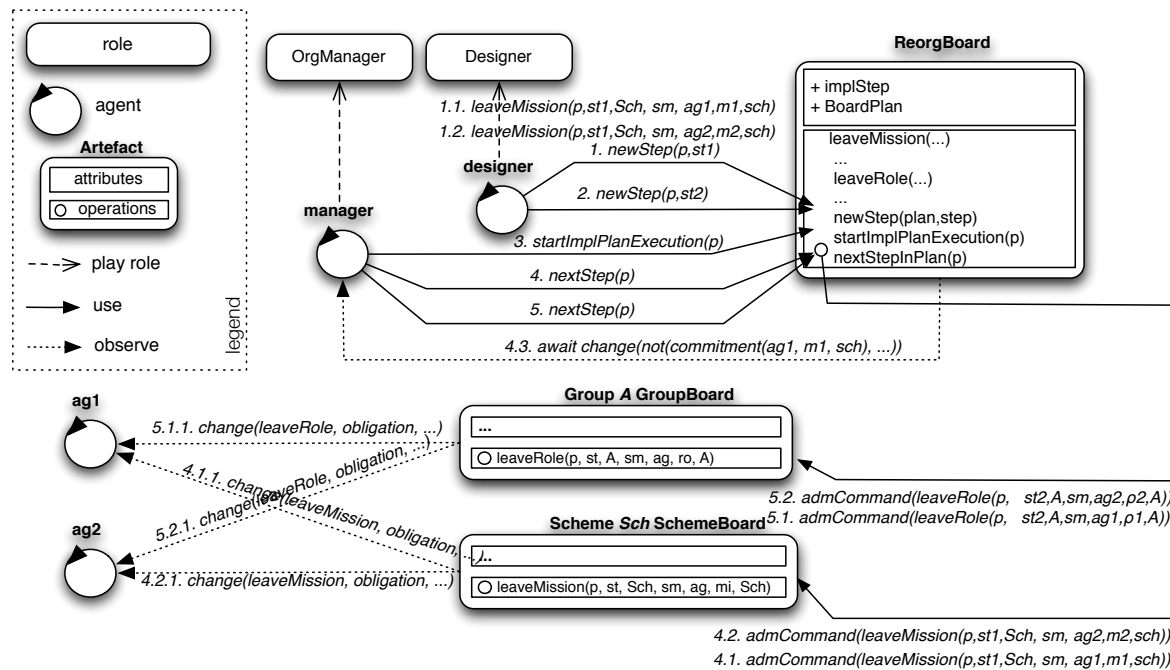


FIGURE 7.4 – Diagramme de collaboration du processus de réorganisation [SORICI, PICARD, BOISSIER, SANTI et al., 2012].

Applications et résultats. Ce travail résulte du master d’Alexandru Sorici [SORICI, 2011] et de collaborations lors du projet région MAOP, dans lequel nous avons étendu l’organisation pour la réorganisation proposée dans [J. HÜBNER, J. SICHMAN et al., 2004]. Il a été implanté en JaCaMo et appliqué dans le cadre de l’intelligence ambiante — plus précisément pour la gestion dynamique et l’affectation de salles dans le contexte d’espaces de co-travail intelligents [SORICI, BOISSIER, PICARD et SANTI, 2011], dont l’architecture est illustrée dans la figure 7.5. Le code est disponible sur le site officiel de JaCaMo ^a, comme exemple d’implantation. Nous avons uniquement appliqué la *réorganisation au niveau structurel* (changement de rôles), et avons montré la capacité pour le système de répartir la charge des salles en fonction de la demande. Ces travaux ont servi de point de départ pour d’autres travaux dans le domaine des réseaux M2M [PERSSON et al., 2012b] que nous présenterons dans le chapitre suivant et ayant également contribué à l’évolution de JaCaMo.

a. http://jacamo.sourceforge.net/?page_id=87

7.3 Synthèse et analyse

Nous avons présenté une approche multi-agent de l’adaptation d’organisations par modification (i) au niveau des instances (affectation de rôles, par exemple), ou (ii) au niveau de la spécification organisationnelle (ajout de rôles, par exemple), à l’initiative des agents qui détectent des dysfonctionnements (e.g. des réservations impossibles dues à un manque d’agents jouant un rôle particulier). Suivant une approche organisationnelle (OCPV), l’organisation de ces agents est *explicite* et perceptible partiellement par les agents — les agents raisonnent sur la spécification de l’organisation et des groupes auxquels ils appartiennent. Ils construisent (mais nous laissons toutefois cette partie à la discrétion du développeur des agents) les étapes de la réorganisation, et peuvent suivre le processus de réorganisation pas à pas. Pour ce faire, *l’environnement est instrumenté* pour implanter cette perception et les actions des agents sur l’organisation. Nous pouvons donc placer ces travaux dans le cadran (d) de la figure 2.1 : ce sont des **approches centrées organisation dont les agents sont les acteurs de l’adaptation**. L’approche par

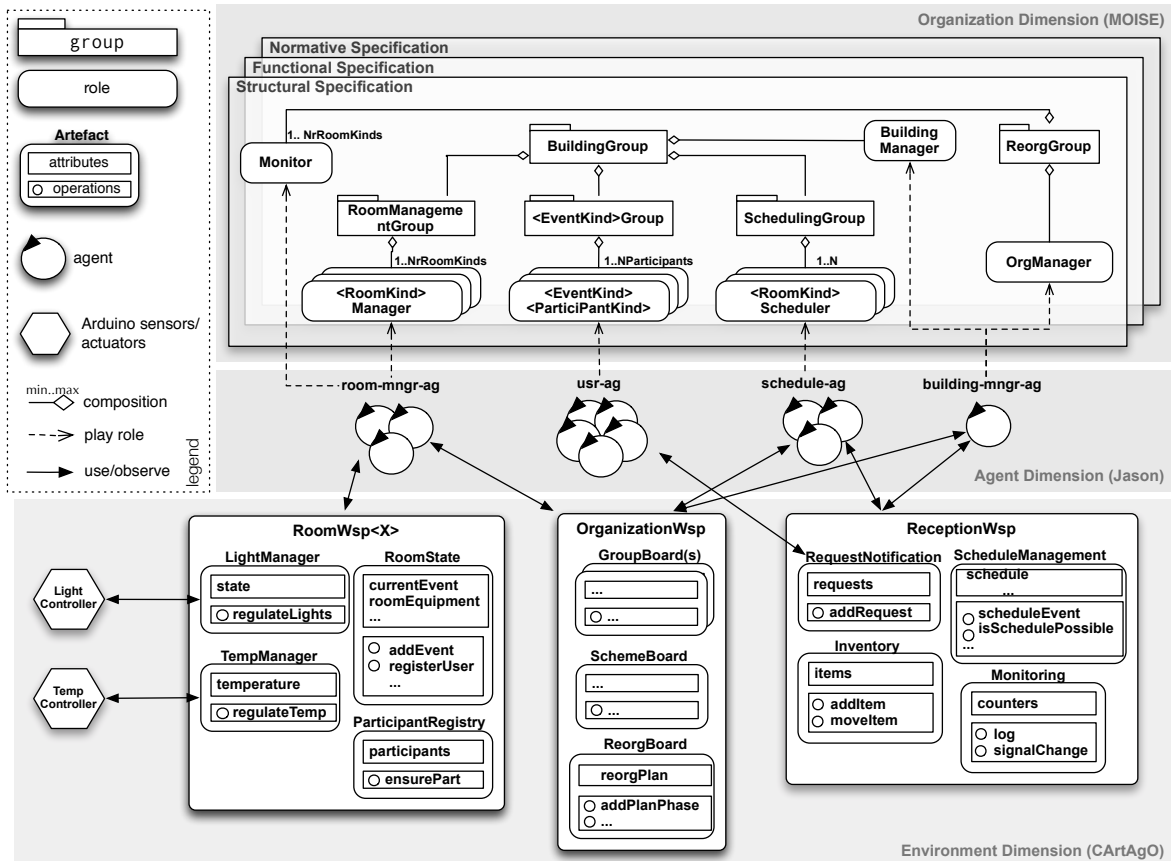


FIGURE 7.5 – Architecture JaCaMo de l’application de gestion d’espaces de co-travail [SORICI, 2011]. Le système est connecté à un réseaux réels de capteurs et d’effecteurs Arduino, interfacés grâce à des artefacts CArtAgO.

programmation multi-agent a permis de mettre en place des capacités de *réorganisation décentralisée* et donc d’adaptation aux perturbations de l’environnement et aux dysfonctionnements de l’organisation qui est représentée explicitement dans l’environnement via des artefacts dédiés.

Ces travaux ont été menés lors d’encadrements d’étudiants en master recherche d’Alexandru Sorici [SORICI, 2011] et Mustapha Bilal [BILAL et al., 2012], ainsi que lors la thèse de Camille Persson [PERSSON, 2014]. Ils ont également été le fruit de collaborations avec l’Université Polytechnique de Bucharest (Roumanie), l’Université de Bologne (Italie), et l’Université Fédérale de Santa Catarina (Brésil), dans le cadre du projet région MAOP. Certaines de ces recherches ont également été initiées dans le cadre des workshops ESAW [PICARD, MELLOULI et al., 2005].

Les publications et communications suivantes sont accessibles pour plus de détails sur tous ces travaux : [BILAL et al., 2012 ; PERSSON et al., 2012b ; SORICI, PICARD, BOISSIER, SANTI et al., 2012 ; PERSSON et al., 2011a ; PERSSON, PICARD et RAMPARANY, 2011 ; SORICI, BOISSIER, PICARD et SANTI, 2011 ; PERSSON et al., 2011b ; PICARD, MELLOULI et al., 2005].

Cependant, nous n’avons présenté qu’une approche centrée organisation de la réorganisation, sans spécifier le comportement des agents évoluant dans ces organisations. Ces travaux ne décrivent pas le contexte ni les moyens donnés aux agents pour mettre en œuvre la réorganisation — Quel plan mettre en œuvre pour réparer l’organisation ? Comment attribuer les rôles dans ces organisations ? Nous présentons donc, dans le chapitre suivant, deux systèmes — de gouvernance de systèmes socio-techniques et de gestion de communautés virtuelles — dans lesquelles les agents sont équipés de moyens de décision pour la réorganisation.

Architectures pour l'adaptation par auto-réorganisation

Dans le chapitre précédent nous avons vu comment mettre en place une organisation de la réorganisation. Nous allons voir dans ce chapitre deux architectures inspirées de cette approche pour rendre adaptatives des architectures de systèmes, et voir quels comportements les agents adoptent pour assurer la réorganisation. Dans ce chapitre nous allons donc suivre un *point de vue orienté organisation*, bien que les comportements des agents ne soient pas forcément centrés sur l'organisation. L'agentification consistera à fournir à des *agents typés* le choix d'adopter des *rôles* correspondant à des tâches et objectifs spécifiques. Les travaux présentés dans ce chapitre illustrent également comment les dynamiques descendantes (des contraintes de l'organisation vers les comportements des agents) et ascendantes (des comportements des agents vers les changements de l'organisation) peuvent coexister et permettre au système de s'adapter à des dynamiques endogènes et exogènes. Nous parlons alors d'*auto-réorganisation*.

Les travaux présentés dans ce chapitre reposent sur une approche par programmation multi-agent avec l'architecture JaCaMo, présentée et illustrée dans les deux chapitres précédents (voir section 6.2 et 7.2). La section 8.1 présente une architecture générique de gouvernance de systèmes ambiants à large échelle, déployés dans le cadre du projet FUI SensCity pour la gouvernance de réseaux M2M, lors de la thèse CIFRE de Camille Persson [PERSSON, 2014]. Ici, la gouvernance et l'adaptation du système est installée de manière ascendante et descendante, et propose une vision multi-organisation. Enfin, la section 8.2 expose l'approche que nous avons adoptée pour la gestion de la confiance dans des communautés virtuelles, lors de la thèse de Reda Yaich [YAICH, 2013]. Dans ce travail, les agents suivent des politiques de confiance préconisées par leurs communautés, et peuvent également les faire évoluer de manière ascendante lorsque qu'elles ne sont plus adaptées à leurs usages.

8.1 Architecture de gouvernance adaptative de systèmes

Dans cette section nous présentons une extension de l'adaptation par réorganisation dans le contexte de la gouvernance de systèmes complexes [PERSSON, 2014]. Ici, la mise en place d'une connexion à un système déjà déployé, qu'il faut munir de capacités d'adaptation et de passage à l'échelle, est au cœur de nos préoccupations. Nous reposons sur l'interaction entre multiples organisations (horizontales et verticales), et une adaptation définie suivant différents niveaux opérationnels. La figure 8.1 illustre un tel système, connecté à un environnement intergiciel et physique existant. Nous suivons également l'approche programmation orientée multi-agent (MAOP) avec JaCaMo comme architecture.

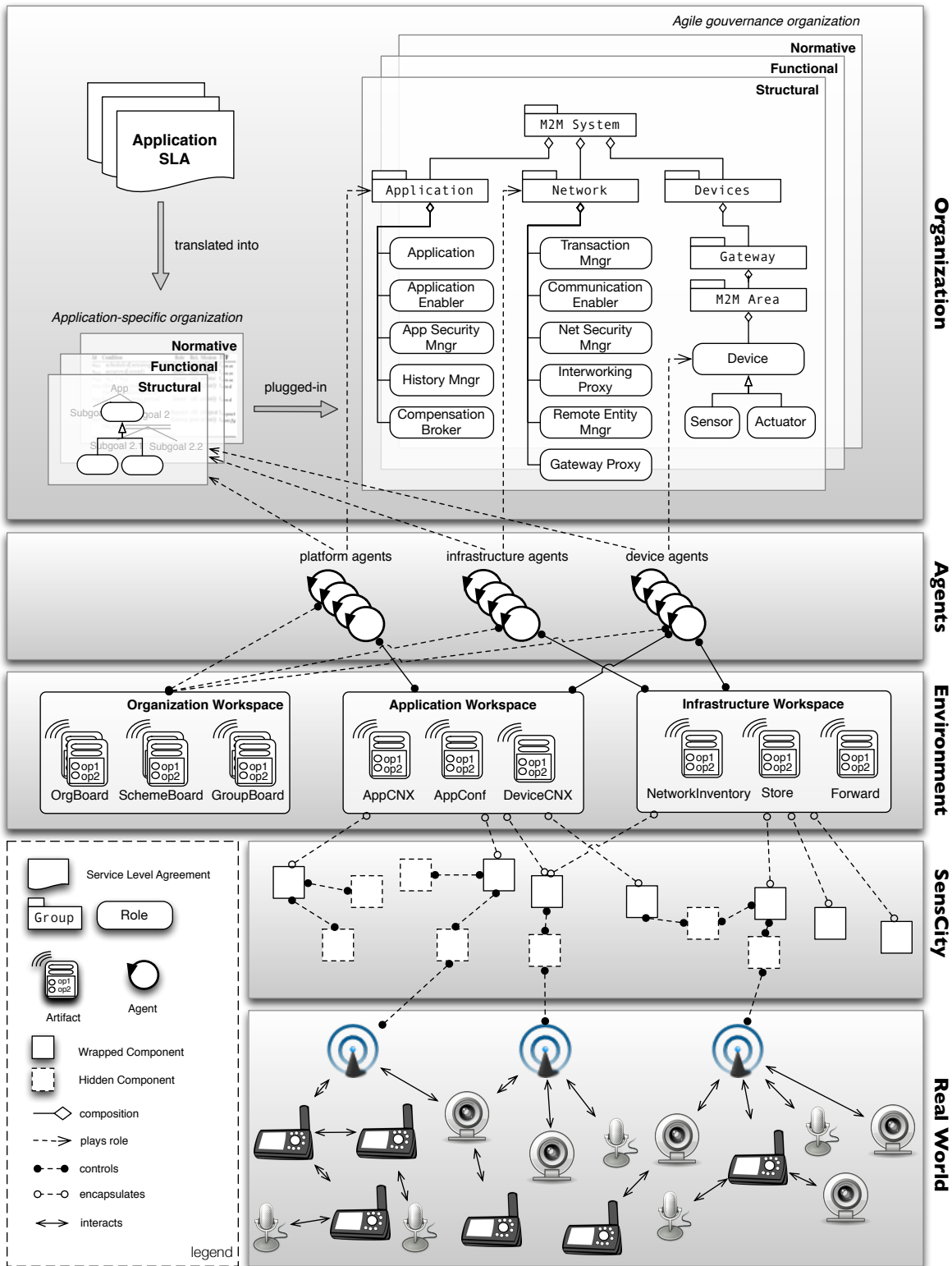


FIGURE 8.1 – Un exemple de système multi-agent connecté à une infrastructure réelle : la plateforme AGAME-MON de gestion de réseaux M2M et des applications associées [PERSSON et al., 2012b].

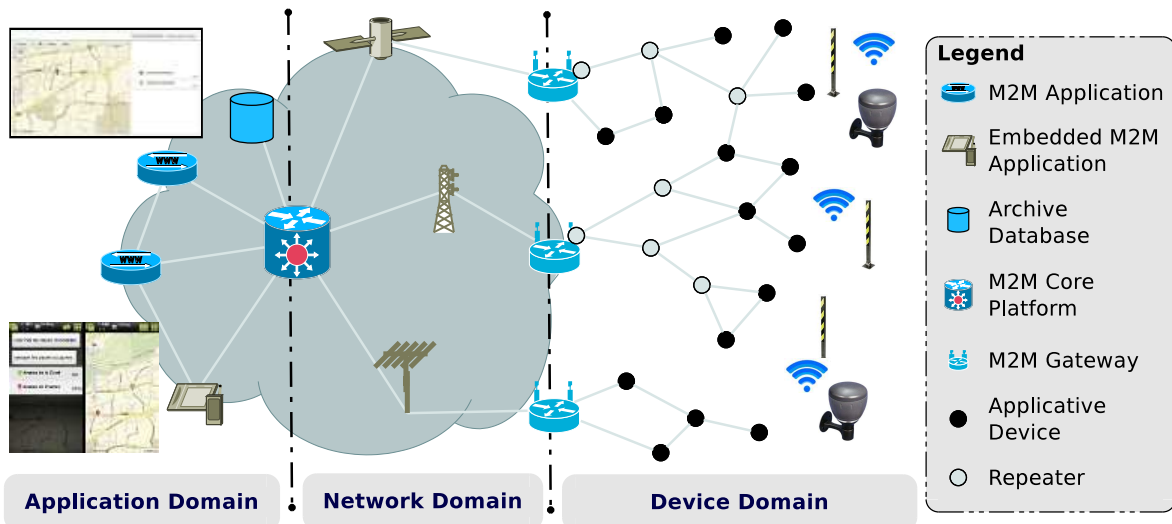


FIGURE 8.2 – L'architecture pour la gestion des réseaux M2M structurée en trois domaines (selon l'ETSI, *European Telecommunications Standards Institute*) : équipement, réseau et application. AGAMEMON se situe dans la *Core Platform*, entre application et réseau.

8.1.1 Gouvernance du système

La gouvernance d'un tel système peut être découpée en cinq couches : (i) la couche *métier* composée des utilisateurs impliqués dans l'infrastructure (compagnie, particulier, etc.); (ii) la couche *stratégique* spécifiant les objectifs haut niveau et les contrats de fourniture de données ou d'utilisation de la plateforme; (iii) la couche *tactique* spécifiant les buts intermédiaires pour bien mener la stratégie; (iv) la couche *politique* spécifiant les moyens d'atteindre les buts intermédiaires; (v) la couche *gouvernée* représentant l'infrastructure gouvernée et les composants métiers. Nous nous sommes focalisés sur les trois couches intermédiaires — les deux extrêmes étant déjà déployées par ailleurs.

Dans ce cadre, nous concevons un système multi-agent pour piloter un système déjà existant, voire déjà déployé. Par exemple, dans le cadre du projet SensCity, Orange Labs, ayant développé un système de gestion de réseaux M2M, a soulevé le besoin de doter le système en cours de production de capacités d'adaptation pour *passer à l'échelle*. Pas question donc de modifier les interfaces de communications avec les utilisateurs, avec l'intergiciel ou avec les capteurs. L'idée est ici d'encapsuler les composants de l'intergiciel, au cœur de cette plateforme, dans des artefacts CArAgO, afin de les rendre manipulables par des agents. Au-delà du fonctionnement nominal du système (qui fonctionne dans une configuration prédéterminée), le système multi-agent doit permettre des réorganisations internes de la plateforme, en permettant, par exemple, de répondre à un nombre croissant d'applications demandant des données provenant de capteurs, ou inversement à un nombre croissant de capteurs fournissant de plus en plus de données à traiter.

Ainsi, les différentes dimensions de la gouvernance sont mises en œuvre par différentes dimensions du système multi-agent : la stratégie est définie comme une organisation, les tactiques sont détenues par les agents et les politiques sont encapsulées dans les artefacts composant l'environnement.

8.1.2 Dynamique environnementale

Le système déployé peut faire face à plusieurs perturbations pouvant mettre en cause la gouvernance. Tout d'abord de nouveaux utilisateurs peuvent souscrire à la plateforme pour accéder à ses services (e.g. récupération de données provenant de capteurs, ou inscription d'un message sur un panneau d'affichage). Chaque souscription passe par un SLA (*Service Level Agreement*) spécifiant les services requis et leur qualité attendue (e.g. récupérer les images d'une caméra toutes les minutes avec une qualité HD). Techniquement parlant, chaque SLA est traduit en une organisation à laquelle certains agents

vont participer pour rendre le service requis. Le système est alors *multi-organisation*, avec une organisation *horizontale* assurant la gouvernance du système, et une *organisation verticale* par SLA assurant la qualité de service.

Pour un ensemble stable d'applications ayant souscrit aux services du système, d'autres dynamiques environnementales peuvent entrer en jeu : l'augmentation du nombre de capteurs, et donc du nombre de données, des pannes de capteurs, etc. Ces informations passent de l'environnement physique (e.g. les capteurs) aux agents via des artefacts encapsulant les équipements.

8.1.3 Adaptation de la gouvernance

L'adaptation du système va se traduire par un *raffinement de la gouvernance* du système. C'est un processus transversal consistant en un cycle d'évaluation-amélioration de la performance de la gouvernance — donc de l'organisation, des agents et de l'environnement. En fonction de cette performance (e.g. délai de réception de données provenant de capteurs), le système multi-agent adapte la gouvernance afin d'en améliorer le fonctionnement.

Nous définissons trois types d'améliorations : (i) la *mise en conformité* est une adaptation descendante de la gouvernance, déclenchant une nouvelle adaptation de gouvernance de plus bas niveau pour mettre en application les objectifs de haut-niveau (SLA) — elle consiste principalement à modifier l'allocation de rôles dans le système (voir chapitre précédent); (ii) la *réorganisation* qui est une adaptation ascendante de la gouvernance, conséquence de l'inadéquation d'objectifs de haut-niveau (SLA) avec des contraintes de bas niveau (capacité des équipements) — elle consiste à modifier les spécifications organisationnelles des applications ayant souscrit aux services (modification des termes du SLA); et (iii) l'*auto-organisation* qui consiste à adapter un niveau de gouvernance pour améliorer les processus sous-jacents en fonction de l'évolution de l'infrastructure — ceci se traduit en la création de nouveaux agents, voire de nouvelles machines virtuelles à déployer sur des serveurs pour assurer le passage à l'échelle.

L'amélioration de la gouvernance s'appuie sur la dimension *interaction* du système multi-agent. La mise en conformité est mise en œuvre par des interactions entre agents et organisation (comme dans la section précédente) et entre agents et artefacts encapsulant des composants de l'intergiciel ou des équipements physiques. La réorganisation est mise en œuvre par des interactions entre ressources physiques et artefacts, entre artefacts et agents et entre agents et organisations. Enfin, l'auto-organisation est mise en œuvre par la création, la suppression et la coordination au sein d'une dimension du système multi-agent.

8.1.4 Comportement des agents

Ce sont les agents qui pilotent l'adaptation de la gouvernance en appliquant une procédure standard de réorganisation — telle que nous avons pu la décrire dans le chapitre précédent. Les agents surveillent le comportement du système en suivant les métriques associées aux SLA contractés avec les applications tierces, et le comportement général du système. Par exemple, si une application tierce a demandé la fourniture de données capteurs avec un délai donné, l'agent sera notifié si le délai réel s'éloigne de celui défini dans le SLA.

Nous avons défini trois types d'agents BDI pour prendre en charge la gouvernance du système et l'adapter : les agents Application, les agents Device et les agents Infrastructure. Les agents Application cherchent à maximiser l'utilité des services auxquels il souscrit tout en minimisant le coût induit par les souscriptions et la violation des SLA. Un agent Device cherche à maximiser le nombre d'applications clientes qu'il fournit tout en maximisant la durée de vie des équipements. Enfin, un agent Infrastructure cherche à assurer les transmissions de messages des équipements/services et des applications tout en minimisant le coût de l'infrastructure (e.g. le nombre de machines virtuelles déployées). Chaque type d'agent pourra jouer différents rôles dans la plateforme. Ces rôles correspondent aux fonctionnalités

identifiées du systèmes (e.g. gérer l'historique des messages, gérer le routage des messages, etc.).

Les comportements d'adaptation de la gouvernance sont codés comme des *plans (tactiques)* suivis par les agents pour réguler le système. Chaque type d'agent possède des plans propres. Une fois un problème détecté, les agents vont principalement effectuer un équilibrage de charge en ce qui concerne l'affectation des rôles (les agents les moins impliqués sont prioritairement affectés à des rôles vacants), la création/destruction d'agents, ou le paramétrage d'équipements, par appel à des *politiques de régulation implantées dans les artefacts* de gouvernance. Ces politiques vont impacter *l'environnement* (paramètres des équipements) ou *l'organisation* (changement de rôles, modification des normes correspondant aux SLA, etc.).

Cette approche transverse (plusieurs applications peuvent coexister dans la même plateforme) et multi-niveau (du physique à l'applicatif et *vice versa*) de l'adaptation de systèmes gouvernés, au travers des comportements d'agents individuels distincts, permet d'obtenir un bon niveau de performance et de qualité des données collectées. Ceci se traduit par un profil comportemental comme celui illustré dans la figure 8.3, montrant une adaptation de la précision des données en fonction des heures de la journée. Ce système permet alors d'obtenir une bonne qualité des données (objectif défini dans un SLA) tout en garantissant une durée de vie des capteurs supérieure à 10 ans (contrainte de haut-niveau).

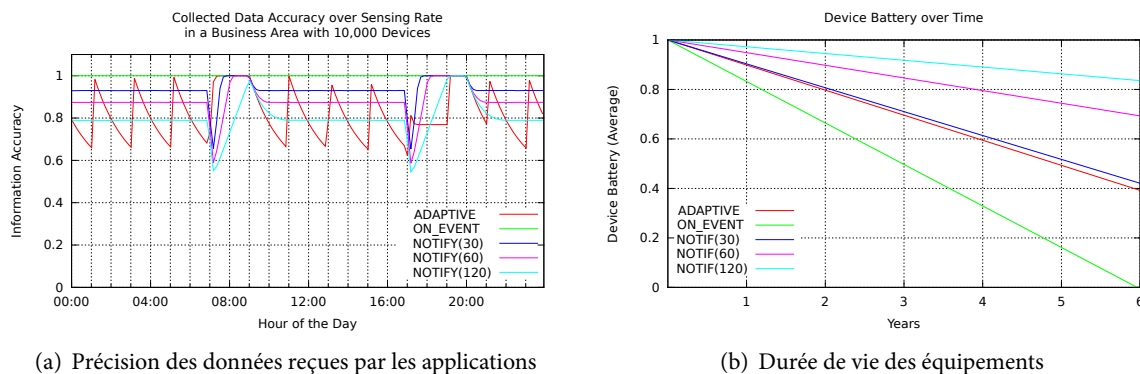


FIGURE 8.3 – Comparaison entre différents profils d'application [PERSSON, 2014]. Nous comparons ici une plateforme non adaptative qui envoie les données à la demande — ON_EVENT — ou à fréquences de collecte fixe — NOTIFY(f) —, à un système adaptatif.

Applications et résultats. Ces travaux ont été appliqués dans le cadre du projet FUI SensCity, dans la thèse de Camille Persson effectuée avec Orange Labs [PERSSON, 2014]. Ce projet a eu pour contexte applicatif la *gestion de réseaux M2M* à l'échelle de la ville, illustré dans la figure 8.2. L'approche résultante, AGAMEMON, implantée en JaCaMo et connectée à l'intergiciel de SensCity, peut aisément être généralisée à d'autres contextes d'*intelligence ambiante* et a été appliquée à la gestion intelligente de places de parking BILAL et al. [2012].

8.2 Architecture de gestion de la confiance dans les communautés virtuelles

Dans cette section nous présentons une extension de l'adaptation par réorganisation dans le contexte de la gestion de la confiance dans des communautés virtuelles [YAICH, 2013]. Nous nous intéressons ici à la conception d'un système multi-agent permettant de soutenir les dynamiques de préférences ascendantes (des préférences des agents vers les préférences de groupes) et descendantes (des groupes vers les agents) et permettant d'analyser l'apport de ces dynamiques dans le contexte des communautés d'innovation ouverte.

$$\pi_{bob}^{(read, file)} = \{ \langle \text{identity}, \geq, \text{marginal}, 2, m \rangle, \langle \text{age}, >, 18, 2, m \rangle, \langle \text{age}, <, 30, 2, o \rangle, \langle \text{reputation}, \geq, 50\%, 3, o \rangle, \langle \text{recommendation}, >, 2, 1, o \rangle \}$$

FIGURE 8.4 – Un exemple de politique de confiance dans ASC-TMS [YAICH et al., 2013]. Pour accéder en lecture à un fichier détenu par *bob*, il est préférable de présenter une identité au moins marginale, d'avoir plus de 18 ans mais il est obligatoire d'avoir moins de 30 ans, d'avoir une réputation supérieure à 50% et d'avoir 2 recommandations.

8.2.1 Gestion multi-agent de communautés

Ici, le but du système est de permettre aux utilisateurs, via des agents assistants, de gérer leur participation dans des communautés virtuelles (modélisées ici en tant que groupes *MOISE*). La participation consiste à mettre à disposition des ressources (e.g. le code source d'une application en cours de développement) pour que celles-ci soient consultées ou modifiées. Ces ressources peuvent être individuelles (appartenant à un agent) ou communautaires (appartenant à un groupe). Chaque utilisateur définit les politiques de confiance auxquelles doivent se conformer les autres utilisateurs souhaitant interagir avec lui ou utiliser ses ressources. De plus, les communautés peuvent également définir des politiques que leurs membres doivent respecter. Enfin, les agents doivent être capables de modifier les politiques de confiance de leurs communautés lorsque celles-ci sont trop contraignantes.

Chaque utilisateur est doté d'un *agent assistant*, lui-même couplé à un *agent de gestion de la confiance* qui gère l'accès aux ressources de l'utilisateur, en fonction de politiques de confiances spécifiées par l'utilisateur, mais pouvant évoluer en cours de fonctionnement. L'assistant s'occupe de mettre en œuvre le comportement « métier » des utilisateurs (e.g. modifier un code source, créer une ressource, etc.), alors que les agents de gestion de la confiance gèrent de manière décentralisée les politiques de confiances des utilisateurs (e.g. mettre à disposition un code source à une personne, uniquement si elle a de bonnes qualifications). Nous nous intéresserons ici principalement aux agents de gestion, moteurs des dynamiques ascendantes et descendantes.

La figure 8.5 présente l'architecture JaCaMo de ce système, nommé ASC-TMS (*Adaptive and Socially Compliant Trust Management System*). La couche organisationnelle correspond aux communautés, représentées par des groupes, dont les politiques sont des normes. La couche agent comporte les agents-assistants, couplés aux agents de gestion de la confiance. Enfin, l'environnement consiste en l'ensemble des ressources créées et modifiées par les agents. Cette architecture peut être généralisée à d'autres problématiques que la gestion de la confiance, dans lesquelles les agents doivent gérer tout type de préférences exprimées sur des individus ou des communautés, grâce à un langage de définition des politiques. Dans le cadre d'ASC-TMS, nous avons développé un langage de politiques de confiance, ainsi qu'une ontologie des termes apparaissant dans ces politiques [YAICH, 2013]. Pour résumer, ces *politiques sont des conjonctions de critères pondérés* à respecter pour être considéré de confiance (e.g. avoir une certification Java **et** avoir une carte d'identité valide **et** avoir une réputation supérieure à 80%). Le poids affecté à chaque critère permet d'évaluer la confiance de manière non binaire. Chaque critère peut être considéré comme obligatoire (négociable) ou non. Un exemple de politique est présenté dans la figure 8.4.

Le but de ce système est de permettre aux communautés de se structurer pour répondre au mieux à leurs intérêts. Par exemple, dans le cas de l'innovation ouverte, les agents se regroupent pour répondre à des défis de développement d'applications informatiques. Ainsi, les agents et les communautés vont s'adapter aux défis lancés par des clients. Concrètement, les agents vont recevoir des propositions de problèmes à résoudre dans un délai fixé, ce qui déclenchera un plan de *résolution de problème*. La plupart des problèmes étant non solvables par un individu seul, dans le délai imparti, les agents vont chercher à trouver des partenaires « compatibles » pour venir à bout du défi. Les agents vont ensuite

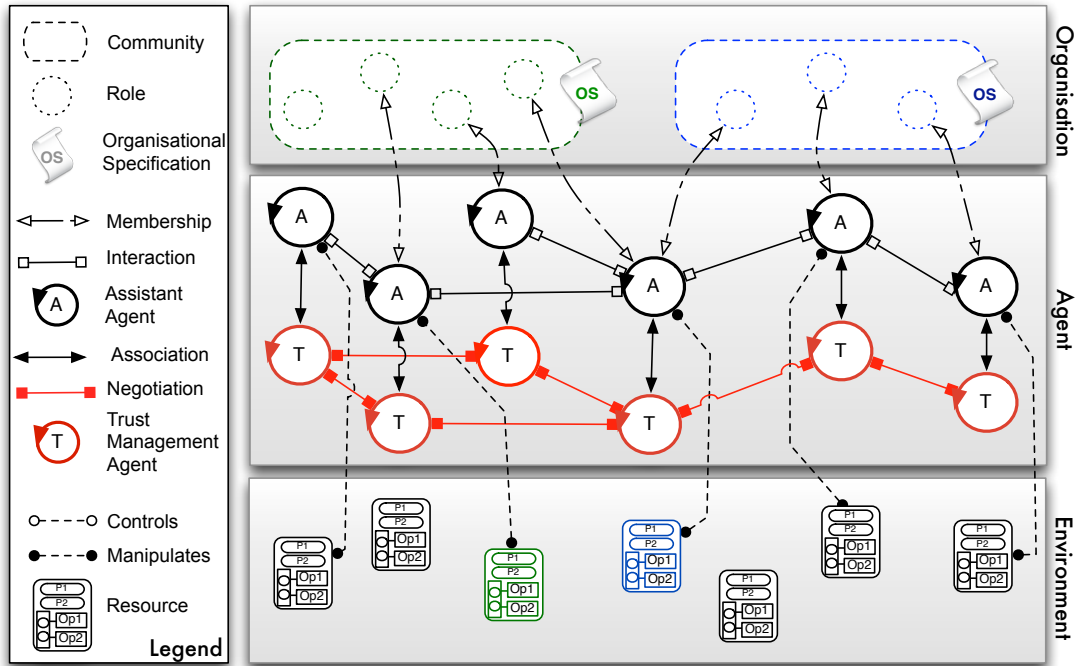


FIGURE 8.5 – Architecture JaCaMo de la plateforme ASC-TMS [YAICH, 2013].

créer, se partager et améliorer des ressources permettant de résoudre le problème. Ce partage pose bien sûr un problème de confiance, car donner l'accès à une ressource à un agent malicieux peut être dommageable pour la réussite du défi. Les agents vont ainsi se baser sur leurs politiques de confiance pour interagir avec les autres agents et les communautés. Les communautés sont ainsi un moyen de regrouper des individus partageant les mêmes intérêts et des politiques de confiances compatibles afin de les rendre plus efficaces pour la résolution de problèmes.

8.2.2 Adaptation des politiques

Au cours de leur vie, les agents vont devoir adapter leurs politiques de confiance devenues obsolètes. Cette adaptation est mise en œuvre grâce à des *méta-politiques* décrivant quand modifier des politiques. Ces méta-politiques sont des règles *event-condition-action* faisant appel à des opérateurs de modification des politiques (e.g. si une ressource a été vandalisée, augmenter le niveau de réputation nécessaire pour accéder à ce type de ressource). Les opérateurs d'adaptation des politiques sont : *ajout/suppression d'un critère* de la politique (e.g. ajouter un critère sur la réputation), *augmenter/diminuer le niveau du seuil* d'acceptabilité (e.g. passer d'une réputation requise de 70% à 80%), *augmenter/diminuer le poids* du critère dans la décision de confiance (e.g. considérer la réputation en priorité par rapport aux autres critères). Il est également possible d'intégrer des politiques de confiance entre elles suivant différentes heuristiques (la moins restrictive, la plus restrictive, etc.). Ainsi, l'utilisateur est capable d'exprimer les moyens de l'évolution et de l'adaptation de ses politiques de confiance de manière explicite et paramétrique.

8.2.3 Adaptation et conformité sociale

Nous avons identifié quatre situations nécessitant une part d'adaptation ou de mise en conformité des politiques, que ce soit au niveau individuel ou collectif, comme l'illustre la figure 8.8.

Instantiation : les agents vont tout d'abord adapter leur politiques de confiance au contexte d'exécution, donc à l'état des ressources, et des autres agents (e.g. plus une ressource prend de la valeur,

$$\text{Instantiate}(\pi_{\text{Bob}}^{\langle _, file \rangle}, R) : R.\text{value}^t > R.\text{value}^{t-1} \ \& \ R.\text{sensitivity}^t > R.\text{sensitivity}^{t-1} \ \leftarrow \\ \text{RestrictCriterion}(\pi_{\text{Bob}}^{\langle _, file \rangle}, \text{reputation}), \\ \text{RestrictCriterion}(\pi_{\text{Bob}}^{\langle _, file \rangle}, \text{recommendation})$$

FIGURE 8.6 – Méta-politique d'adaptation à la valeur de la ressource (instanciation). Si la valeur et la sensibilité de la ressource augmentent, devenir plus restrictif concernant la réputation et les recommandations.

$$\pi_{\text{bob}}^{\langle read, file \rangle} = \{ \langle \text{identity}, \geq, \text{marginal}, 2, m \rangle, \\ \langle \text{age}, >, 18, 2, m \rangle, \\ \langle \text{age}, <, 30, 2, o \rangle, \\ \langle \text{reputation}, \geq, 60\%, 3, o \rangle, \\ \langle \text{recommendation}, >, 3, 1, o \rangle \}$$

FIGURE 8.7 – La politique de confiance de la figure 8.4 une fois adaptée par instanciation.

plus les politiques gérant cette ressource doivent devenir contraignante). Ceci est illustré dans les figures 8.6 et 8.7, présentant une méta-politique définie par l'utilisateur et automatiquement déclenchée pour adapter la politique d'accès à une ressource en fonction du contexte (ici l'augmentation de la valeur et de la sensibilité de la ressource).

Intégration : les agents vont intégrer les politiques de leur communauté dans leur décision, en suivant par exemple la plus restrictive, afin d'assurer leur conformité sociale (respect des normes de leur groupe). Les politiques (celle de l'agent et celle de la communauté) sont analysées critère par critère, et pour chacun d'entre eux le plus restrictif est conservé.

Négociation : les agents vont modifier leur politique en fonction des demandes des autres agents (e.g. un agent possédant une ressource unique dont on a besoin ne passe pas le niveau de confiance, il faut alors changer notre politique ou faire changer la sienne par négociation). Ceci marche comme l'instanciation, mais au lieu de réagir au contexte, les méta-politiques sont déclenchées par des requêtes d'autres agents.

Combinaison : un agent qui n'arrive pas à respecter la politique de sa communauté (car trop restrictive), peut demander à la modifier, ou plus simplement deux agents peuvent s'entendre pour créer une nouvelle communauté combinant leurs propres politiques (changer ou créer des normes de groupe). Dans ce cas, un agent désirant modifier une politique de communauté va faire appel à un vote en faveur d'une proposition d'adaptation de la politique. En cas de majorité absolue, la politique de groupe est modifiée. Nous pouvons parler alors d'*émergence de normes*.

Dans le cas de la création et de la modification de communautés, les agents doivent être capables d'accéder à la spécification organisationnelle des communautés pour les manipuler, comme nous l'avons décrit dans le chapitre précédent. Grâce aux mécanismes d'adaptation des politiques présentés dans la section précédente, les agents et les communautés sont ainsi capables de s'adapter à ces différentes situations, de manière auto-organisée (appel au vote) et réorganisée (modification des politiques individuelles et collectives).

Nous avons validé cette approche par auto-réorganisation en simulant des communautés d'innovation ouverte. Les agents étaient soumis à des défis et recevaient des récompenses en fonction de leurs partages, de leurs accès et de la résolution des défis. Il a été empiriquement démontré que : (i) sans comportement d'intégration de la part des agents, les communautés avaient tendances à disparaître, et donc à laisser des défis sans solutions; (ii) la dynamique des communautés est fortement affectée par la conformité des agents (moins les agents se conforment à la politique de la communauté, moins la communauté et donc les agents ne résolvent les défis); et (iii) la capacité donnée aux agents de faire évoluer leur communauté, même en présence d'agents non conformes (jusqu'à 20%), permet à une communauté de perdurer et de répondre positivement à des défis [YAICH, 2013]. Ainsi les capacités de mise en conformité (processus descendant) et d'émergence de normes (processus ascendant) assurent la

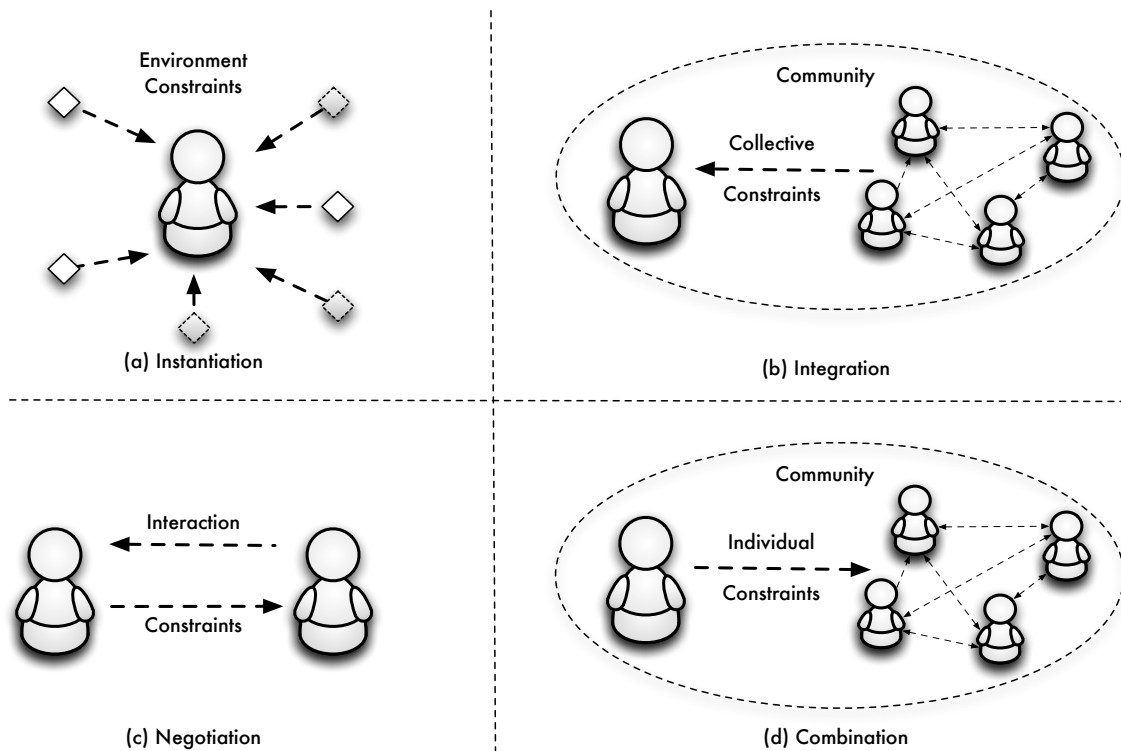


FIGURE 8.8 – Les causes de l’adaptation des agents et du système [YAICH, 2013] : *l’instantiation* (adaptation des politiques individuelles à l’environnement physique de l’agent), *l’intégration* (adaptation des politiques collectives dans les politiques individuelles), la *négociation* (adaptation des politiques individuelles suite aux négociations pair-à-pair) et la *combinaison* (adaptation des politiques collectives aux demandes individuelles).

robustesse et le compétitivité des communautés.

Applications et résultats. Le système ASC-TMS a été appliqué au domaine de communautés d’innovation ouverte, notamment dans le domaine du *problem solving* par appel à des communautés de développeurs [YAICH et al., 2013 ; YAICH, 2013]. Outre la plateforme multi-agent développée en JaCaMo, ces travaux ont résulté sur un modèle de confiance associé à un langage de spécification de politiques de confiance, mêlant facteurs de confiance durs (e.g. clés, preuves d’identité) et souples (e.g. réputation, recommandation). Un client Android de l’agent de gestion de la confiance, ainsi qu’un simulateur Repast de systèmes dans le cadre de l’innovation ouverte ont été développés. Ces travaux ont été le point de départ d’une collaboration avec la société Sereale, dans le domaine des services à la personne rémunérés entre particuliers.

8.3 Synthèse et analyse

Nous avons présenté une approche multi-agent de l’adaptation de systèmes par auto-réorganisation, conjuguant point de vue orienté agent et point de vue orienté organisation (ACPV et OCPV). Ici, les agents évoluant dans une organisation sont dotés de comportements (plans) pour réguler le système dans lequel ils évoluent, soit au niveau de leur environnement, soit au niveau de l’organisation. En dernier recours, lorsque le système s’avère non adapté, ils peuvent mettre en place une procédure de modification de l’organisation, soit au niveau instance (changement de rôles) soit au niveau spécification (changement des normes). L’organisation de ces agents est donc forcément *explicite* et perceptible partiellement par les agents — les agents raisonnent sur la spécification de l’organisation et des groupes auxquels ils appartiennent. L’environnement est instrumenté pour implanter cette perception et les ac-

tions des agents sur l'organisation et l'environnement physique. Les capacités d'*auto-organisation* des agents permettent au système de s'adapter aux perturbations endogènes (e.g. pannes de l'infrastructure M2M, agents malicieux) et exogènes (e.g. nouveaux SLA, nouveaux défis), par la *création de nouveaux agents ou de nouveaux groupes*. L'auto-organisation est également assurée par des *mécanismes de coordination locaux* soit via l'environnement (e.g. changement de paramètres de capteurs ou redirection de données dans le cas du M2M) soit par négociation/vote (e.g. vote à la majorité pour modifier une politique collective), impactant l'ensemble de la dynamique du système. Ainsi, nous pouvons placer ces travaux à la frontière des cadrans (b) et (d) de la figure 2.1 : ce sont des **approches centrées organisation dans lesquelles les agents s'auto-organisent**. La différence avec le chapitre précédent est qu'ici les agents sont dotés de capacités de coordination extra-organisation (e.g. le vote ou la répartition de charge via l'environnement) qui ont un impact sur indirect l'organisation. Ces systèmes couplent donc une approche centrée organisation et une approche centrée agent, d'où la qualification d'**auto-réorganisation** — permettant de coupler contraintes organisationnelles et adaptation. Ainsi, *l'approche organisationnelle serait un moyen de définir un espace de recherche des comportements (organisations) du système, l'auto-organisation un moyen d'explorer les états possibles de cet espace, et l'auto-réorganisation un moyen de modifier l'espace de recherche si ce dernier a été incorrectement ou incomplètement défini, ou que les exigences ont changé depuis le déploiement du système.*

Ces travaux ont été menés lors des thèses de Camille Persson [PERSSON, 2014] et de Reda Yaich [YAICH, 2013]. La thèse CIFRE de Camille Persson a été effectuée en collaboration avec Orange Labs, dans le cadre du projet FUI SensCity. Ces travaux ont été appliqués à la gestion de places de parking lors du stage de master de Mustapha Bilal [BILAL, 2011]. Les travaux *pluridisciplinaires* de Reda Yaich, à la frontière de la sécurité et des systèmes multi-agents ont permis de démontrer l'apport de la programmation multi-agent au domaine de la confiance dans les réseaux sociaux. Le modèle de confiance ainsi que le langage de spécification des politiques de confiance ont donné lieu à une collaboration avec la startup grenobloise Sereale¹. Les concepts d'adaptation et de conformité sociale sont également sources d'inspiration pour nos travaux dans le projet ANR ETHICAA² sur la mise en place de règles éthiques de groupes.

Les publications et communications suivantes sont accessibles pour plus de détails sur tous ces travaux : [PERSSON et al., 2014 ; YAICH et al., 2013 ; PERSSON et al., 2012a ; YAICH et al., 2012a ; YAICH et al., 2012b ; BILAL et al., 2012 ; PERSSON et al., 2012b ; PERSSON et al., 2011a ; PERSSON, PICARD et RAMPARANY, 2011 ; SORICI, BOISSIER, PICARD et SANTI, 2011 ; YAICH, JAILLON, BOISSIER et al., 2011 ; PERSSON et al., 2011b ; YAICH et al., 2011b ; YAICH et al., 2011a].

1. <https://www.sereale.fr>

2. <https://ethicaa.greyc.fr>

Quatrième partie

Conclusion et perspectives

Synthèse des travaux de recherche

Ce manuscrit a dressé un panorama de mes travaux de recherche menés depuis 2004. Ce chapitre synthétise les contributions majeures dans le domaine de l'adaptation par auto-organisation dans les systèmes multi-agents, les projets dans lesquels ces recherches ont été menées et ainsi que les applications de ces recherches.

9.1 Contributions

La présentation de mes travaux a été structurée suivant deux axes de recherche — (i) l'adaptation des systèmes multi-agents dans la résolution de problèmes et (ii) l'ingénierie des systèmes multi-agents adaptatifs. Ces travaux ont principalement contribué aux domaines des systèmes multi-agents en général, mais également en sécurité et en génie mécanique.

9.1.1 Contributions à l'adaptation des systèmes multi-agents et dans résolution de problèmes

Le premier axe de recherche a résulté sur plusieurs algorithmes multi-agents, solveurs et optimiseurs inédits de part leur capacité d'adaptation — soit pour la métamodélisation de l'espace de recherche, soit pour le processus de recherche en tant que tel. Nous avons également abordé différents types de problèmes (combinatoires, continus, sous contraintes, temporels, etc.) suivant deux types d'approches (par population ou par décomposition des dimensions du problème) mais toujours suivant des principes de coordination par auto-organisation.

Adaptation pour la métamodélisation (chapitre 4). Nous avons défini et implanté plusieurs algorithmes par métamodélisation d'espaces continus, pour l'optimisation par appel à des simulateurs coûteux. Ces algorithmes reposent sur des populations d'agents qui s'auto-organisent pour explorer de manière coordonnée l'espace de recherche, dans toutes ses dimensions (*chaque agent cherche une solution au problème global*). La métamodélisation adaptative est obtenue soit par partitionnement de l'espace de recherche (en cellules de Voronoï) couplé à l'utilisation de plusieurs métamodèles dans chaque sous-région [VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2013b ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2012a], soit par raffinement d'un métamodèle unique pour identifier tous les optima locaux, dans LOOM [RIVIÈRE et al., 2013]. Ces algorithmes ont montré de très bonnes performances (qualité des solutions à budget d'appels

limité) sur des problèmes académiques et industriels (e.g. conception de boucliers thermiques) en comparaison d'optimiseurs par métamodélisation classiques.

Adaptation pour la résolution (chapitre 5). Nous avons conçu des algorithmes et protocoles multi-agents pour la résolution de problèmes combinatoires, dans le cadre des DCOP. Ces algorithmes consistent en la mise en place de techniques de coordination pour permettre aux agents de trouver une solution commune à un problème global, étant données des contraintes inter-agents et des objectifs individuels. Cette coordination a été mise en place soit via un environnement de travail commun dans ETTO [PICARD, BERNON et M.-P. GLEIZES, 2005a] et ETTO4MC [CLAIR et al., 2008b], soit par négociation pair-à-pair et sessions d'affectation dans FAPP [PICARD et al., 2007b] ou DAMASCOP [PICARD et al., 2007b]. Ces algorithmes reposent sur une réparation continue de l'état du système par auto-organisation, et apportent des capacités de robustesse à la dynamique environnementale. Nous avons également abordé les problèmes de *privac  * dans la r  solution de probl  mes d'appariement stable, avec le protocole CASANOVA [EVERAERE et al., 2012].

Notons que *ces deux approches ont pu   tre coupl  es* dans le cadre du projet ANR ID4CS ¹, dans lequel des agents, obtenus par d  composition par dimensions et crit  res, utilisent en interne l'algorithme LOOM (donc un autre syst  me multi-agent de plus bas niveau) pour effectuer des optimisations par appel    des simulateurs co  teux de conception a  ronautique, afin d'effectuer des r  solutions de probl  mes inverses pour identifier les corr  lations entre leurs entr  es et leurs sorties.

9.1.2 Contributions    l'ing  nierie des syst  mes multi-agents adaptatifs

Le deuxi  me axe de recherche a produit des mod  les et des architectures pour l'adaptation par auto-r  organisation. Nous avons utilis   et   tendu des mod  les organisationnels de la r  organisation, et propos   des architectures de syst  mes de gouvernance ou de gestion de communaut  s bas  es sur ces mod  les.

Organisation de la r  organisation (chapitre 7). Nous avons   tendu l'approche d'organisation de la r  organisation en instrumentant l'environnement par le biais d'*artefacts de r  organisation*, permettant aux agents de (i) d  finir un plan de r  organisation pas    pas, gr  ce    des actions   l  mentaires, et de (ii) contr  ler et piloter les   tapes de r  organisation — impactant sur l'adoption des r  les des agents, ou sur les normes gouvernant le syst  me, par exemple. Ce travail a permis de tracer les directions de recherche suivies dans les th  ses de Reda Yaich et Camille Persson [YAICH, 2013 ; PERSSON, 2014]. Ceci constitue mes premiers travaux sur des organisations explicites, en collaboration avec l'Universit   de Bologne (Italie), l'Universit   F  d  rale de Santa Catarina (Br  sil) et l'Universit   Polytechnique de Bucharest (Roumanie), dans le cadre du projet MAOP.

Architectures pour l'adaptation par auto-r  organisation (chapitre 8). Les travaux pr  c  dents ont permis de d  finir des architectures de syst  mes multi-agents, suivant une approche programmation orient  e multi-agent, adaptatifs par auto-r  organisation. Ici l'organisation permet de d  finir les objectifs m  tiers du syst  me, que les agents peuvent modifier en cas d'obsolescence ou d'impossibilit   physique de les atteindre, dans le cas des syst  mes de gestion de r  seaux M2M [PERSSON, 2014]. Nous avons   galement trait   l'ouverture du syst  me    des souscriptions d'applications tierces par l'ajout de nouvelles organisations orthogonales    l'organisation centrale du syst  me. Les agents,    la crois  e de ces organisations, sont les moteurs de l'adaptation et du passage    l'  chelle. Dans le domaine des communaut  s virtuelles, nous avons mis en place des m  canismes ascendants et descendants de mise en conformit   sociale des agents, et d'  volution des communaut  s, en fonction des besoins du syst  me. Ainsi, nous avons install   dans les deux cas et de mani  re concomitante, des *processus d'adaptation des agents    l'organisation, et de l'organisation aux agents*.

1. <http://www.irit.fr/id4cs>

Transformation de modèles dans l'ingénierie orientée multi-agent. De manière plus anecdotique et au-delà des travaux rentrant dans les deux sous-axes précédents, juste après ma thèse sur la méthode ADELFE, nous avons développé une approche — inédite à l'époque — de transformation de modèles. Nous avons adopté une approche dirigée par les modèles (MDA) dans le cadre de la méthode ADELFE, pour la génération automatique de code d'agents coopératifs AMAS [OTTENS et al., 2006]. Ces travaux ont été poursuivis par l'équipe SMAC, notamment dans la thèse de Sylvain Rougemaille [ROUGEMAILLE, 2008].

9.1.3 Contributions à d'autres domaines

Génie mécanique. La thèse de Diane Villanueva, dans le cadre du projet interdisciplinaire ID4CS, ayant fait l'objet d'une cotutelle avec le département de génie mécanique de l'Université de Floride, a été le point de départ de travaux appliqués à la conception d'objets mécaniques complexes. La métamodélisation, très utilisée dans ce contexte, a été la technique à la base de l'algorithme de partitionnement d'espace de recherche et de LOOM [VILLANUEVA, 2013 ; RIVIÈRE et al., 2013]. Ces algorithmes ont été et sont encore utilisés pour s'attaquer à des problèmes venant de la mécanique et de la conception aéronautique. Ces travaux ont naturellement été *valorisés dans la communauté de génie mécanique et aérospatial* [VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2013a ; VILLANUEVA, HAFTKA et al., 2013 ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2012a ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2012b ; VILLANUEVA, LE RICHE, PICARD, HAFTKA et SANKAR, 2011].

Sécurité. La thèse de Reda Yaich, au croisement des systèmes multi-agents et de la sécurité informatique, a permis de définir un système décentralisé de gestion de la confiance basée sur une approche hybride inédite de la confiance, à des fins de contrôle d'accès. Ces travaux ont été valorisés devant une audience de ce domaine [YAICH et al., 2012a].

9.2 Domaines d'application et projets

Comme nous l'avons vu tout au long de ce document, nous avons pris soin de valoriser, de diffuser et d'appliquer nos résultats de recherche à des domaines variés, notamment dans le cadre de projets collaboratifs. Voici une synthèse de ces domaines d'application.

Aéronautique. Dans le cadre du projet ID4CS et de la thèse de Diane Villanueva, nous avons appliqué nos techniques d'optimisation par métamodélisation multi-agent à des problèmes issus de l'aéronautique, notamment la conception préliminaire avion et la conception de boucliers thermiques.

Intelligence ambiante. Les travaux de master, puis de thèse d'Alexandru Sorici, se sont intéressés à l'intelligence ambiante, dans le cadre du projet MAOP et de montage du laboratoire AmI Lab à l'Université Polytechnique de Bucharest. Les scénarios développés consistent en la mise en place d'espaces de co-travail intelligents (détection de réunions *ad-hoc*, réservation automatique de salles de réunions, etc.).

Ville intelligente. Le projet FUI SensCity, dans lequel s'est déroulée la thèse CIFRE de Camille Persson, a mis en place un intergiciel de gestion de réseaux M2M pour la ville intelligente. Nous avons développé une couche de gestion adaptative illustrée sur des scénarios comme la réservation de places de parking, et la collecte adaptative de déchets ménagers. Dans une autre mesure, nos travaux sur l'affectation adaptative de fréquences s'attaquent à une problématique apparaissant dans la gestion de réseaux de capteurs.

Communautés virtuelles. Les travaux de Reda Yaich, ainsi que les travaux sur les appariements stables, sont autant d'applications de techniques multi-agents pour la gestion de la confiance et de la confidentialité dans les communautés virtuelles. Reda Yaich a illustré ses travaux dans le domaine de

l'innovation ouverte, ce qui a donné lieu à une collaboration avec la startup Sereale. Dans un autre registre, CASANOVA est appliqué au domaine des mariages stables par approche multi-agent. Ceci constitue un premier pas vers la conception de protocoles décentralisés de rencontre en ligne, de consommation collaborative ou de covoiturage.

Contrôle manufacturier. Dans le cadre du groupe de travail français COLLINE et du groupe de travail européen TFG SO, nous avons développé une plateforme de *benchmarking* pour les systèmes multi-agents dans le domaine du contrôle manufacturier, et avons produit plusieurs algorithmes de résolution adaptatifs à la dynamique des commandes, des pannes machines et des indisponibilités des opérateurs.

10

Perspectives et projet(s) de recherche

Compte tenu de mes travaux antérieurs, présentés dans ce document, l'objectif général de mon projet de recherche futur reste l'adaptation par auto-organisation dans les systèmes multi-agents, pour concevoir des systèmes capables de répondre à des dynamiques endogènes et exogènes, tout en garantissant un comportement cohérent. Cet objectif a été décliné suivant deux axes de recherche, qui feront encore partie de mon projet, mais enrichis des questionnements apparus en cours d'investigation, comme les questions de confidentialité ou d'émergence de normes.

10.1 Résolution multi-agent de problèmes

La résolution de problèmes par systèmes multi-agents adaptatifs propose une approche centrée individu de la résolution. Ceci permet de fournir au système des capacités de robustesse à la dynamique du problème et de compréhension des phénomènes localisés dans l'espace de recherche. Cependant, cette approche conserve actuellement des limites concernant son applicabilité dans un certains nombres de domaines — les garanties, la généralité des modèles, et la confidentialité.

10.1.1 Vers un cadre générique de résolution par coopération

Les approches développées dans nos recherches ont été dirigées par les problèmes (emploi du temps, affectation de fréquences, contrôle manufacturier), et bien qu'un certain effort de généralisation des approches ait été fourni (dans le cas d'ETTO), il semble primordial de fournir une méthode, ou une famille de méthodes, plus générique, sous la forme d'optimiseurs ou solveurs basés sur la résolution par systèmes multi-agents adaptatif et la coopération, et directement utilisables par les ingénieurs.

La première méthode, basée sur ETTO ou sur la résolution dirigée par les conflits, permettrait de s'attaquer à des problèmes dynamiques d'optimisation combinatoire (DCOP dynamiques), pour lesquels un déploiement avec un environnement commun de coordination est envisageable. Le passage du problème au modèle environnemental est généralisable : des domaines de valeurs à l'environnement discrétisé. Par contre, les comportements coopératifs n'ont pas forcément été conçus pour être génériques, et donc l'effort devra être mis en priorité sur cet aspect.

Lorsque un déploiement physique ou des exigences de confidentialité des données rendent difficile ou impossible le recours à un environnement partagé, les approches par négociation pair-à-pair deviennent pertinentes. Nos travaux sur l'affectation de fréquences ou les appariements, sont deux

exemples d'applications de ce type d'approches, comparables aux algorithmes classiques de résolution de DCOP — plus précisément des techniques de *propagation de croyances*, provenant du domaine des *modèles graphiques probabilistes* pour la décision [KOLLER et al., 2009], qui ont inspiré de nouveaux algorithmes distribués et efficaces, comme max-sum [FARINELLI et al., 2008]. Une perspective à court terme est donc de généraliser voire d'intégrer l'approche de priorisation par coopération dans ces techniques qui possèdent encore des limitations en terme de convergence à l'optimum en cas de présence de cycles dans le réseau de contraintes. Habituellement, des heuristiques *ad-hoc* sont proposées pour y répondre (ajout de bruit, amortissement, ordonnancement intelligent des messages, etc.). La notion de coopération, qui a montré ses preuves dans de nombreux problèmes, pourrait ainsi être une solution générique (métaheuristique) à ce type de décisions distribuées, qui nous semble être un cadre théorique compatible avec la résolution de problèmes par auto-organisation coopérative. Inversement, il est envisageable de définir la notion de coopération au regard de ces techniques éprouvées.

10.1.2 Vers des garanties de convergence et de terminaison

Les algorithmes proposés jusqu'à présent dans nos travaux reposent sur une approche par recherche locale décentralisée, et ne fournissent aucune garantie de convergence ou de terminaison (sauf dans le cas de budget de calcul limité). Nous pourrions tenter de répondre à ces limitations, en nous plaçant dans un cadre formel ou technique ayant permis d'obtenir, de garantir ou de caractériser ses propriétés. Encore une fois, rapprocher les techniques d'optimisation par auto-organisation coopérative et les techniques de propagation de croyances semble une piste prometteuse à moyen terme. En effet, dans le domaine des modèles graphiques, plusieurs propriétés de convergence et de qualité des solutions peuvent être déduites de la topologie du réseau et des domaines de valeurs; ce dont nous pourrions aisément nous inspirer une fois le modèle de résolution par auto-organisation exprimé comme un modèle de propagation.

Nous n'écartons pas non plus le rapprochement avec d'autres cadres formels comme la mécanique statistique qui s'intéresse à l'analyse des invariants dans les comportements collectifs large échelle [JAYNES, 1957]. Déjà appliqué de manière anecdotique aux systèmes multi-agents [GLINTON et al., 2010], ce cadre théorique pourrait être un moyen d'analyser le comportement du système de manière macroscopique et ainsi garantir, en moyenne, la convergence de nos systèmes.

10.1.3 Vers une résolution de problèmes respectueuse de la privacité

La *privacités* devient une préoccupation majeure, remplaçant l'individu au centre des services. La résolution de problèmes doit s'inscrire dans cette dynamique, car forte de solutions pour résoudre des problèmes rencontrés dans les communautés virtuelles, dans les applications mobiles, ou dans des contextes d'informatique ambiante. Cependant, garantir la privacité peut aller à l'encontre du fonctionnement optimal d'un système. Nous proposons donc d'explorer un espace de compromis suivant deux dimensions allant (i) de la délégation complète des données et des traitements à un tiers de confiance jusqu'à la fermeture totale des données, et (ii) de la performance optimale du service jusqu'au bien-être de chaque participant. Fournir une telle connaissance et les protocoles associés représenterait une contribution importante pour la compréhension et la gestion de la vie numérique des usagers.

Nous proposons d'étudier ces compromis dans le cadre des DCOP, de l'auto-organisation, et de la négociation. Proche du *mechanism design* [KEARNS et al., 2012], cet enjeu sociétal, soulevant des questions théoriques et techniques difficiles, devra se nourrir de travaux en sécurité, en confiance, théorie du choix social et les systèmes multi-agents en général.

10.2 Ingénierie et théorie de l'auto-réorganisation

Mes travaux dans l'ingénierie des systèmes multi-agents ont porté sur les méthodes et outils de développement (i) de systèmes multi-agents adaptatifs par coopération (lors de ma thèse et dans sa continuité), puis (ii) de systèmes multi-agents adaptatifs par réorganisation (à partir de mon intégration dans l'équipe de l'École des Mines de Saint-Étienne). Au dernier chapitre de la partie III nous avons ouvert la porte à l'auto-réorganisation, au croisement des approches ACPV par auto-organisation et des approches OCPV par réorganisation. Nos travaux dans cette direction n'en sont qu'à leur début, et de nombreuses perspectives sont ouvertes.

10.2.1 Vers une approche auto-réorganisationnelle de l'adaptation

Nos travaux ont proposé des architectures permettant de mettre en place les mécanismes d'auto-réorganisation. Cependant, ils ont été menés en parallèle, et même si les résultats sont complémentaires, une prise de recul sur la formalisation et la généralisation des résultats est nécessaire. Le point de départ pour la définition et l'articulation de ces dynamiques ascendantes et descendantes peut se résumer par la conclusion du chapitre 8 : *l'approche organisationnelle serait un moyen de définir un espace de recherche des comportements (organisations) du système, l'auto-organisation un moyen d'explorer les états possibles de cet espace, et l'auto-réorganisation un moyen de modifier l'espace de recherche si ce dernier a été incorrectement ou incomplètement défini, ou que les exigences ont changé depuis le déploiement du système*. Ceci représente notre première étape d'investigation, notamment si nous nous plaçons dans un formalisme qui nous est familier : la résolution de problèmes de satisfaction de contraintes.

Pourquoi ne pas envisager les approches de résolution par auto-organisation comme un moyen d'explorer l'espace de recherche sous contraintes organisationnelles, et éventuellement modifier cet espace par auto-réorganisation si l'environnement change ou que les besoins évoluent, et donc changer explicitement le modèle du problème sous contraintes organisationnelles? Nous pourrions alors nous appuyer sur des résultats obtenus dans le premier axe pour améliorer notre théorie de l'auto-organisation et de l'adaptation — et ainsi boucler la boucle entre résolution de problèmes et ingénierie...

10.2.2 Vers des outils intégrés de conception de systèmes multi-agents adaptatifs

L'approche organisationnelle par programmation multi-agent nous a été précieuse pour la conceptualisation et le développement de nos architectures multi-agents. Bien qu'instrumentée par des langages de spécification, des éditeurs, ou des environnements d'exécution, elle n'est pas encore intégrée dans un processus de développement spécifique. La méthode AEIO n'est qu'un cadre de modélisation permettant d'attacher les composantes du système à des dimensions données (agent, environnement, interaction et organisation), par contre rien ne guide le concepteur sur l'identification ou la conception des agents — seuls les rôles sont clairement structurés dans l'organisation. La typologie des agents et leurs compétences restent assez difficiles à déterminer dans ce cadre.

Un effort méthodologique semble donc nécessaire pour fournir des modèles et des outils intégrés à un processus rigoureux de développement de systèmes multi-agents adaptatifs par auto-réorganisation, comme nous avons pu le faire dans ADELFE¹ pour les AMAS.

10.2.3 Vers une approche *privacy-by-design-at-runtime*

Nous avons souligné dans l'axe précédent le souci de mettre en place des mécanismes de garantie et d'évaluation de la privacité dans la résolution multi-agent. En nous plaçant dans une perspective d'auto-réorganisation, il est alors tout à fait possible de considérer la privacité au moment de la conception

1. <http://www.irit.fr/ADELFE/>

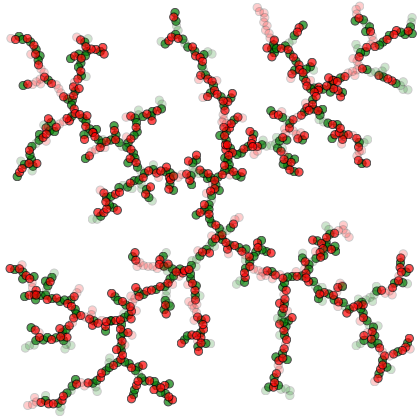


FIGURE 10.1 – Réseau de distribution d'énergie dans lequel des prosommateurs (usagers à la fois producteurs et consommateurs) échangent de l'énergie pour répartir la charge énergétique.

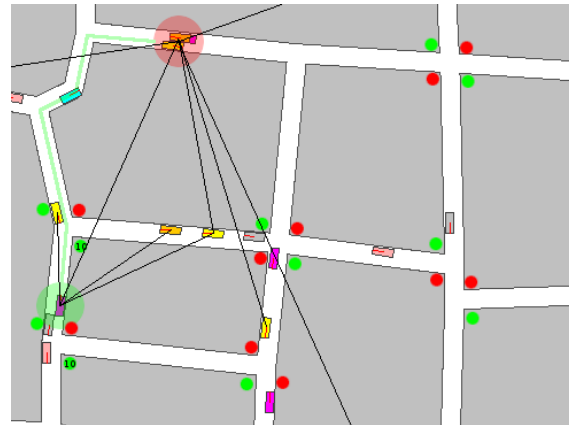


FIGURE 10.2 – Les systèmes multi-agents pour le contrôle de réseaux routiers et la coordination de véhicules intelligents, en cas d'accident sur la voie.

(*privacy-by-design*), notamment au travers de la définition d'organisations dédiées, par exemple, ou lors de l'exécution du système (*privacy-at-runtime*), par des mécanismes auto-organisés de régulation sociale basés sur la confiance et la réputation, par exemple.

Cette approche pourra faire l'objet d'investigations dans le cadre du projet ANR ETHICAA², qui propose d'étudier la définition et la mise en place de principes éthiques au sein de systèmes multi-agents.

10.3 Applications

Nos recherches ont toujours été motivées par un souci d'applicabilité des résultats à des problèmes réels, industriels ou sociétaux. Outre les domaines que nous avons pu aborder dans les chapitres précédents, d'autres champs applicatifs ont émergé et nous paraissent autant d'opportunités de théorisation, d'expérimentation, de développement de preuves de concept et bien sûr de transfert de connaissances.

10.3.1 Énergie et *smart grids*

Les infrastructures de production, de transport et de distribution, ainsi que les systèmes de contrôles des réseaux électriques ont fortement évolué ces dernières années, ouvrant un large potentiel d'investigations et d'innovations dans le domaine de l'informatique. Compte tenu de la distribution naturelle et de l'implication de plus en plus croissante de l'utilisateur dans ces réseaux — notion de *prosommateur* (voir figure 10.1) — les systèmes multi-agents se positionnent comme un paradigme de premier choix pour proposer de nouvelles méthodes de modélisation, de gestion ou de développement de telles infrastructures.

Le projet européen ITEA SEAS³ (*Smart Energy Aware Systems*), auquel nous participons, a pour but de permettre l'interopérabilité des systèmes produisant de l'énergie, des TIC et des systèmes d'automatisation des sites de consommations, et d'introduire des solutions fondées sur des technologies innovantes pour contrôler et suivre l'estimation de la consommation d'énergie. Dans cette optique, nous allons notamment contribuer à la représentation des connaissances et à mise en place de protocoles de négociation de partenariats production-consommation auto-organisés. Les données permettront de valider la thèse de S. Gillani en co-encadrement avec F. Laforest (Télécom Saint-Etienne). Ces travaux soulèvent des problèmes de coordination massive et seront un moyen d'éprouver, de développer

2. <https://ethicaa.greyc.fr>

3. <https://itea3.org/project/seas.html>

les techniques de résolution auto-organisationnelles et décentralisées et de formaliser les mécanismes d'auto-organisation. Dans de tels systèmes, compte tenu de l'infrastructure énergétique actuelle, il est également nécessaire de combiner des approches ascendantes à des approches descendantes (reflétant l'organisation du réseau énergétique en place), sans faire table rase du passé du marché.

Sur cette problématique, nous avons également initié une collaboration avec le laboratoire IIIA⁴ de Barcelone, pour la conception d'algorithmes d'optimisation décentralisée pour maximiser le profit et assurer l'équilibre de charge dans des réseaux de distribution peuplés de prosummateurs. Nous envisageons d'étendre les algorithmes classiques — notamment max-sum — par des mécanismes de coopération, pour permettre une application à des réseaux cycliques (voir les remarques de la section 10.1). Cette collaboration a donné lieu au dépôt d'un projet CNRS PICS, DOT4eT (*Distributed Optimization Techniques to enable Energy Trading in Prosumer Communities*).

10.3.2 Transport et véhicules intelligents

Le domaine des transport est également en train de subir une mutation importante. Fondés sur l'utilisation des nouvelles technologies de la communication, les systèmes de transport intelligents connaissent un essor important car ils permettent une meilleure utilisation des ressources disponibles. Pour la gestion du trafic urbain routier, l'utilisation de ces systèmes a déjà permis d'améliorer la fluidité du trafic par une augmentation du niveau d'information disponible pour l'établissement des plans de feux.

Des progrès concernant la technologie des capteurs ont permis le déploiement d'algorithmes adaptatifs des plans de feux exploitant les informations de plus en plus précises sur la position et la vitesse des véhicules. Les modèles coopératifs qui reposent sur l'utilisation de la communication entre véhicules ou avec l'infrastructure contribuent à cette évolution par une intégration du conducteur au sein du système de régulation du trafic urbain. Cependant, cette information caractérise l'état courant du trafic et ne permet donc qu'une régulation à court terme.

Une approche multi-agent nous semble pertinente pour dépasser cette limite en proposant d'intégrer des informations liées aux conducteurs et permettant une planification sur un plus long horizon et donc une meilleure robustesse et efficacité de la régulation. Nous avons très récemment initié une collaboration avec l'Université de Belfort dans cette direction, par la définition d'un environnement et une infrastructure de coordination multi-agent, appliquée au domaine routier [GALLAND, GAUD et al., 2014] (voir figure 10.2), et qui ouvre de nombreuses perspectives.

10.3.3 Aéronautique

Le monde des constructeurs aéronautiques — et du transport en général — est également en train de changer, dans le contexte de la pénurie des sources d'énergie fossile. Ceci force les bureaux d'études de ces constructeurs à proposer de nouveaux concepts d'aéronefs propulsés par des moteurs alimentés par des énergies alternatives. Au regard des processus de conception préliminaire adoptés chez ces constructeurs, basés sur la réutilisation de modèles déjà produits, le passage à des conceptions en rupture avec les modèles et expériences acquises ouvre la porte à de nouvelles techniques d'optimisation, et de détection des optima locaux.

Dans le cas des avions à propulsion hybride (voir figure 10.3), les ingénieurs n'ont aucune intuition de la position des optima (donc des concepts potentiels) dans l'espace de recherche complètement différent de celui des avions à propulsion thermodynamique — l'utilisation de turbines électriques, la présence de batteries qui ont une masse constante, à l'opposé des carburants liquides ou gazeux, etc. Airbus nous a donc contacté pour développer, sur la base de LOOM, un optimiseur pour les aider à comprendre l'espace de conception. Ceci va faire le sujet d'une étude et de projets à venir.

4. <http://www.iiia.csic.es>



FIGURE 10.3 – L'avion de ligne hybride est au cœur des enjeux de conception en aéronautique moderne, en rupture avec les conceptions connues.

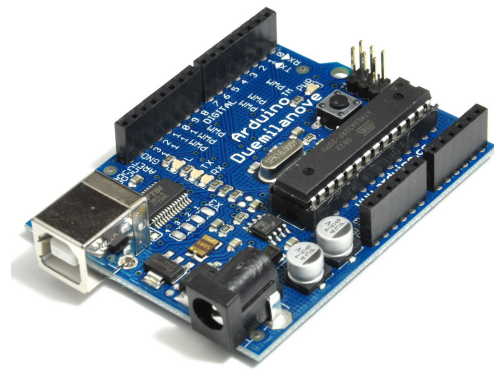


FIGURE 10.4 – La popularisation des contrôleurs et des capteurs bon marché contribue à l'essor de l'intelligence ambiante fondée sur le Web des objets.

10.3.4 Informatique ambiante et Web des objets

L'informatique ambiante est aujourd'hui une grande source d'innovation (ville intelligente, transport intelligent, domotique, grilles intelligentes, etc.). Ce contexte applicatif, couplé à l'arrivée sur le marché de technologies de capteurs et effecteurs connectés à bas prix (voir figure 10.4), est l'environnement de déploiement idéal pour le Web des objets. Ce paradigme propose à la fois autonomie et connectivité aux objets déployés, et se présente comme un terrain d'application direct de l'approche multi-agent.

Une approche par auto-réorganisation pourra notamment proposer une solution à l'imbrication grandissante des mondes physiques, sociaux et numériques dans notre société, par l'extension du Web des objets avec deux dimensions décisives : la sociabilité (s'étendant des réseaux sociaux au Web) et la sensibilité au contexte (reconnaissance continue et d'adaptation à des situations qui se produisent à l'intérieur et à l'extérieur). Nous pourrions ainsi concevoir une infrastructure logicielle unificatrice d'un Web social sensible au contexte, capable de mettre en œuvre les aspects très exigeants de l'ouverture, de l'hétérogénéité et l'évolutivité des applications et des services omniprésents dans le monde réel. Ce travail est abordé dans l'équipe ISCOD de l'École des Mines de Saint-Étienne en collaboration avec l'Université Polytechnique de Bucharest, au travers de deux thèses, dont une que j'encadre sur la mise en place de l'infrastructure multi-agent de gestion du contexte (Alexandru Sorici) [SORICI, PICARD et BOISSIER, 2014 ; SORICI, BOISSIER, PICARD et ZIMMERMANN, 2013].

Cinquième partie

Références

11 Publications

Pour respecter les critères d'évaluation des publications définis par l'AERES pour le domaine des STIC¹, le classement de mes publications distingue les articles publiés en revue internationale, les contributions à des ouvrages qui sont des publications issues d'une sélection faisant suite à une communication à une conférence ou un workshop, et les publications en conférence en classant celles reconnues comme importantes par leur rang d'après le classement CORE², EIC³, MRS⁴ ou CNU 27⁵. Sauf indication contraire, toutes les publications à des conférences ou des workshops listées ici ont subi un **processus de sélection par comité de lecture sur soumission d'un article long** (et non par *abstract*).

Quand ils sont connus les taux de sélection sont indiqués par l'annotation [AR=XX%]. D'autres annotations signalent le type de publications (*pre-* ou *post-proceedings*) et le nombre de tours de relecture (*review phase*) de la conférence ou du journal. Enfin, les DOI (*Digital Object Identifier*) sont fournis pour les articles publiés chez des éditeurs reconnus (IEEE, ACM, Springer, IOSPress, etc.).

11.1 Chapitres de livres

- GLIZE, P. et PICARD, G. (2011). « Self-Organisation in Constraint Problem Solving ». In : *Self-organizing Software : From Natural to Artificial Adaptation*. Sous la dir. de G. SERUGENDO, M.-P. GLEIZES et A. KARAGEORGOS. Natural Computing Series. Springer. Chap. 14, p. 347–377. ISBN : 978-3-642-17348-6. DOI : 10.1007/978-3-642-17348-6_14. URL : <http://www.springer.com/computer/ai/book/978-3-642-17347-9> (cité pp. 22, 24, 27, 44, 45). [Chapter on invitation – 1 review phase]
- BERNON, C., GLEIZES, M.-P. et PICARD, G. (2009). « Méthodes orientées agent et multi-agent ». In : *Technologies des systèmes multi-agents et applications industrielles*. Sous la dir. d'A. EL FALLAH-SEGHRUCHNI et J.-P. BRIOT. Collection IC2. Hermès. Chap. 2, p. 45–76. URL : <http://www.lavoisier.fr/livre/notice.asp?ouvrage=2138883> (cité p. 59). [Chapter on invitation – 1 review phase]
- BERNON, C., CAMPS, V., GLEIZES, M.-P. et PICARD, G. (2005). « Engineering Self-Adaptive Multi-Agent Systems : the ADELFE Methodology ». In : *Agent-Oriented Methodologies*. Sous la dir. de B. HENDERSON-SELLERS
-
1. http://obspm.flext.net/wp-content/uploads/2009/02/aeres_criteres_publi.pdf
 2. <http://www.core.edu.au/rankings/tier-conf-final2007.pdf>
 3. <http://perso.crans.org/~genest/conf.html>
 4. <http://academic.research.microsoft.com> (catégorie *Artificial Intelligence*)
 5. <http://cnu27.lri.fr/Publis-IA.html> (catégorie *intelligence artificielle*)

et P. GIORGINI. Idea Group Publishing. Chap. 7, p. 172–202. DOI : [10.4018/978-1-59140-581-8](https://doi.org/10.4018/978-1-59140-581-8). URL : <http://www.igi-global.com/book/agent-oriented-methodologies/62> (cité p. 60). [Chapter on invitation – 2 review phases]

PICARD, G. et GLEIZES, M.-P. (2004). « The ADELFE Methodology – Designing Adaptive Cooperative Multi-Agent Systems ». In : *Methodologies and Software Engineering for Agent Systems*. Sous la dir. de G. WEISS, F. BERGENTI, M.-P. GLEIZES et F. ZAMBONELLI. T. 11. Multiagent Systems, Artificial Societies, And Simulated Organizations. Kluwer Publishing. Chap. 8, p. 157–176. ISBN : 1-4020-8057-3. DOI : [10.1007/1-4020-8058-1_11](https://doi.org/10.1007/1-4020-8058-1_11). URL : <http://www.springerlink.com/content/ku3714781x30q625/> (cité p. 60). [Chapter on invitation – 2 review phases]

11.2 Éditions de livre

ALDEWERELD, H., DIGNUM, V. et PICARD, G., édés. (2009). *Engineering Societies in the Agents World X - 10th International Workshop, ESAW 2009, Utrecht, The Netherlands, November 18-20, 2009*. T. 5881. Lecture Notes in Artificial Intelligence (LNAI). Springer, p. 258. ISBN : 978-3-642-10202-8. DOI : [10.1007/978-3-642-10203-5](https://doi.org/10.1007/978-3-642-10203-5). URL : <http://www.springer.com/computer/ai/book/978-3-642-10202-8>.

ARTIKIS, A., PICARD, G. et VERCOUTER, L., édés. (2008). *Engineering Societies in the Agents World IX - 9th International Workshop, ESAW 2008, Saint-Etienne, France, September 24-26, 2008, Revised Selected Papers*. T. 5485. Lecture Notes in Artificial Intelligence (LNAI). Springer, p. 281. ISBN : 978-3-642-02561-7. DOI : [10.1007/978-3-642-02562-4](https://doi.org/10.1007/978-3-642-02562-4). URL : <http://www.springer.com/computer/ai/book/978-3-642-02561-7>.

11.3 Journaux et revues

SORICI, A., PICARD, G., BOISSIER, O., ZIMMERMANN, A. et FLOREA, A. (2014). « CONSERT : Applying Semantic Web Technologies to Context Modeling in Ambient Intelligence ». In : *Computers and Electrical Engineering - An International Journal*. [in submission]

YAICH, R., BOISSIER, O., PICARD, G. et JAILLON, P. (2013). « Adaptiveness and Social-Compliance in Trust Management within Virtual Communities ». In : *Web Intelligence and Agent Systems (WIAS)* 11.4, p. 315–338. DOI : [10.3233/WIA-130278](https://doi.org/10.3233/WIA-130278). URL : <http://iospress.metapress.com/content/q2659685221703r7/?issue=4&genre=article&page=315&issn=1570-1263&volume=11> (cité pp. 10, 57, 62, 76, 79, 80, 125).

EVERAERE, P., MORGE, M. et PICARD, G. (2012). « Casanova : un comportement d'agent pour l'équité des mariages préservant la confidentialité ». In : *Revue d'Intelligence Artificielle* 26.5, p. 471–494. DOI : [10.3166/ria.26.471-494](https://doi.org/10.3166/ria.26.471-494). URL : <http://ria.revuesonline.com/article.jsp?articleId=17808> (cité pp. 24, 46, 48–51, 57, 84). [CNU27]

GLEIZES, M.-P., BERNON, C., MIGEON, F. et PICARD, G. (2008). « Méthodes de développement de systèmes multi-agents ». In : *Génie Logiciel, GL & IS* 86, p. 2–7 (cité p. 59).

OTTENS, K., PICARD, G. et CAMPS, V. (2006). « Transformation de modèles d'agents dans la méthode ADELFE : Des stéréotypes de conception à l'implémentation ». In : *Revue Technique et Science Informatique – L'objet* 12.4, p. 43–72. DOI : [doi : 10.3166/objet.12.4.43-72](https://doi.org/10.3166/objet.12.4.43-72). URL : <http://objet.e-revues.com/article.jsp?articleId=9174> (cité pp. 57, 85).

PICARD, G. et GLIZE, P. (2006). « Model and Analysis of Local Decision Based on Cooperative Self-Organization for Problem Solving ». In : *Multiagent and Grid Systems – An International Journal (MAGS)* 2.3, p. 253–265. URL : <http://iospress.metapress.com/content/2wgykx6vbfje5wt4/>

(cité pp. 10, 16, 22–24, 27, 44, 46, 51). [Cet article de revue correspond à mes premiers travaux d'après thèse sur la résolution de problèmes par systèmes multi-agents.]

PICARD, G., BERNON, C., CAMPS, V. et GLEIZES, M.-P. (2003). « ADELFE : Atelier de développement de logiciels à fonctionnalité émergente ». In : *Revue Technique et Science Informatique* 22.4. Sous la dir. de J. BRIOT et K. GHEDIRA, p. 387–391. URL : <http://tsi.revuesonline.com/article.jsp?articleId=4789> (cité p. 60). [CNU27]

PICARD, G. et GLEIZES, M.-P. (2003). « Outils pour la réalisation de systèmes multi-agents adaptatifs dans le cadre de la méthode ADELFE ». In : *Revue Technique et Science Informatique* 22.4. Sous la dir. de J. BRIOT et K. GHEDIRA, p. 249–253. URL : <http://tsi.revuesonline.com/article.jsp?articleId=4777> (cité p. 60). [CNU27]

11.4 Conférences internationales (avec comité de lecture)

CERQUIDES, J., PICARD, G. et RODRÍGUEZ-AGUILAR, J. (2015). « Designing a marketplace for the trading and distribution of energy in the smart grid ». In : *14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. International Foundation for Autonomous Agents et Multiagent Systems. [in submission]

GALLAND, S., BALBO, F., GAUD, N., RODRIGUEZ, S., PICARD, G. et BOISSIER, O. (2015). « Contextualize Agent Interactions by Combining Social and Physical Dimensions in the Environment ». In : *14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. International Foundation for Autonomous Agents et Multiagent Systems. [in submission]

BELLONI, A., BERGER, A., BOISSIER, O., BONNET, G., BOURGNE, G., CHARDEL, P., COTTON, J., EVREUX, N., GANASCIA, J., JAILLON, P., MERMET, B., PICARD, G., REVER, B., SIMON, G., SWARTE, T. de, TESSIER, C., VEXLER, F., VOYER, R. et ZIMMERMANN, A. (2014). « Towards A Framework To Deal With Ethical Conflicts In Autonomous Agents And Multi-Agent Systems ». In : *12th International Conference on Computer Ethics and Philosophical Enquiry (CEPE'14)*, p. 1–10.

EVERAERE, P., MORGE, M. et PICARD, G. (2013). « Minimal Concession Strategy for Reaching Fair, Optimal and Stable Marriages ». In : *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'13)*. Sous la dir. de T. ITO, C. JONKER, M. GINI et O. SHEHORY. International Foundation for Autonomous Agents et Multiagent Systems, p. 1319–1320. URL : <http://dl.acm.org/citation.cfm?id=2484920.2485203> (cité pp. 24, 50, 51). [AR=44.28%] [Core A+ – Pre-proceedings – 1 review phase]

VILLANUEVA, D., LE RICHE, R., PICARD, G. et HAFTKA, R. (2013a). « Dynamic Design Space Partitioning for Optimization of an Integrated Thermal Protection System ». In : *9th AIAA Multidisciplinary Design Optimization Specialist Conference co-located with the 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (SDM'13)*. AIAA. DOI : [10.2514/6.2013-1534](https://doi.org/10.2514/6.2013-1534). URL : <http://arc.aiaa.org/doi/abs/10.2514/6.2013-1534> (cité pp. 37, 40, 85, 126). [Pre-proceedings – 1 review phase]

PERSSON, C., PICARD, G., RAMPARANY, F. et BOISSIER, O. (2012a). « A JaCaMo-Based Governance of Machine-to-Machine Systems ». In : *Advances on Practical Applications of Agents and Multiagent Systems, Proc. of the 10th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 12)*. Sous la dir. d'Y. DEMAZEAU, J. P. MÜLLER, J. M. C. RODRÍGUEZ et J. B. PÉREZ. T. 155. Advances in Soft Computing Series. Springer, p. 161–168. DOI : [10.1007/978-3-642-28786-2_18](https://doi.org/10.1007/978-3-642-28786-2_18). URL : http://dx.doi.org/10.1007/978-3-642-28786-2_18 (cité pp. 80, 125). [AR=30.6%] [EIC 52/701 in Artificial Intelligence – Pre-proceedings – 1 review phase]

VILLANUEVA, D., LE RICHE, R., PICARD, G. et HAFTKA, R. (2012a). « Dynamic Partitioning for Balancing Exploitation and Exploration in Constrained Optimization : A Multi-Agent Approach ». In :

- 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (MAO)*. AIAA. DOI : [10.2514/6.2012-5440](https://doi.org/10.2514/6.2012-5440). URL : <http://arc.aiaa.org/doi/abs/10.2514/6.2012-5440> (cité pp. 40, 83, 85, 126). [Pre-proceedings – 1 review phase]
- VILLANUEVA, D., LE RICHE, R., PICARD, G. et HAFTKA, R. (2012b). « Surrogate-Based Agents for Constrained Optimization ». In : *14th AIAA Non-Deterministic Approaches Conference, Honolulu, HI*. AIAA. DOI : [10.2514/6.2012-1935](https://doi.org/10.2514/6.2012-1935). URL : <http://arc.aiaa.org/doi/abs/10.2514/6.2012-1935> (cité pp. 32, 40, 85, 126). [Pre-proceedings – 1 review phase]
- YAICH, R., BOISSIER, O., JAILLON, P. et PICARD, G. (2012a). « An Adaptive and Socially-Compliant Trust Management System for Virtual Communities ». In : *The 27th ACM Symposium On Applied Computing (SAC 2012)*. ACM Press, p. 2022–2028. DOI : [10.1145/2245276.2232112](https://doi.org/10.1145/2245276.2232112) (cité pp. 80, 85, 125). [CORE B – Pre-proceedings – 1 review phase]
- (2012b). « An Agent Based Trust Management System for Multi-Agent Based Virtual Communities ». In : *Advances on Practical Applications of Agents and Multiagent Systems, Proc. of the 10th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 12)*. Sous la dir. d'Y. DEMAZEAU, J. P. MÜLLER, J. M. C. RODRÍGUEZ et J. B. PÉREZ. T. 155. Advances in Soft Computing Series. Springer, p. 217–223. DOI : [10.1007/978-3-642-28786-2_24](https://doi.org/10.1007/978-3-642-28786-2_24). URL : http://dx.doi.org/10.1007/978-3-642-28786-2_24 (cité pp. 80, 125). [AR=30.6%] [EIC 52/701 in Artificial Intelligence – Pre-proceedings – 1 review phase]
- MORGE, M. et PICARD, G. (2011). « Privacy-Preserving Strategy for Negotiating Stable, Equitable and Optimal Matchings ». In : *Advances on Practical Applications of Agents and Multiagent Systems, Proc. of the 9th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 11)*. Advances in Intelligent and Soft-Computing. Springer, p. 97–102. DOI : [10.1007/978-3-642-19875-5](https://doi.org/10.1007/978-3-642-19875-5). URL : <http://www.springerlink.com/content/978-3-642-19874-8#section=867062&page=1&locus=0> (cité p. 51). [AR=48.1%] [EIC 52/701 in Artificial Intelligence – Pre-proceedings – 1 review phase]
- PERSSON, C., PICARD, G. et RAMPARANY, F. (2011). « A Multi-Agent Organization for the Governance of Machine-To-Machine Systems ». In : *IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'11)*. IEEE Computer Society, p. 421–424. DOI : [10.1109/WI-IAT.2011.161](https://doi.org/10.1109/WI-IAT.2011.161) (cité pp. 69, 80, 125). [AR=21%] [CORE C – Pre-proceedings – 1 review phase]
- VILLANUEVA, D., LE RICHE, R., PICARD, G., HAFTKA, R. et SANKAR, B. (2011). « Decomposition of System Level Reliability-Based Design Optimization to Reduce the Number of Simulations ». In : *ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Washington, DC, USA*. ASME, p. 117–126. DOI : [10.1115/DETC2011-47815](https://doi.org/10.1115/DETC2011-47815) (cité pp. 40, 85, 126). [Pre-proceedings – 1 review phase]
- KADDOUM, E., GLEIZES, M.-P., GEORGÉ, J.-P. et PICARD, G. (2009). « Characterizing and Evaluating Problem Solving Self-* Systems ». In : *International Conference on Adaptive and Self-adaptive Systems and Applications (ADAPTIVE 2009), 2009 Computation World : Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, Athens, Greece, November 15-November 20*. IEEE Computer Society, p. 137–145. ISBN : 978-0-7695-3862-4. DOI : [10.1109/ComputationWorld.2009.100](https://doi.org/10.1109/ComputationWorld.2009.100). URL : <http://www.computer.org/portal/web/csdl/doi/10.1109/ComputationWorld.2009.100> (cité pp. 42, 51, 126). [Pre-proceedings – 1 review phase]
- CLAIR, G., GLEIZES, M.-P., KADDOUM, E. et PICARD, G. (2008b). « Self-Regulation in Self-Organising Multi-Agent Systems for Adaptive and Intelligent Manufacturing Control ». In : *Second IEEE International Conference on Self-Adaption and Self-Organization (SASO 2008), Venice, Italy, 20-24 October 2008*. IEEE Computer Society, p. 107–116. DOI : [10.1109/SASO.2008.19](https://doi.org/10.1109/SASO.2008.19). URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4663415 (cité pp. 9, 22, 24–27, 43, 46, 49–51, 84, 126). [AR=27.1%] [Pre-proceedings – 1 review phase]

- PICARD, G., GLEIZES, M.-P. et GLIZE, P. (2007b). « Distributed Frequency Assignment Using Cooperative Self-Organization ». In : *First IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'07), Boston, Mass., USA, July 9-11, 2007*. IEEE Computer Society, p. 183–192. DOI : [10.1109/SASO.2007.18](https://doi.org/10.1109/SASO.2007.18). URL : <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4274902> (cité pp. 9, 13, 16, 22, 24, 27, 46, 48, 50, 51, 84). [AR=20.8%] [Pre-proceedings – 1 review phase]
- PICARD, G., BERNON, C. et GLEIZES, M.-P. (2005a). « Emergent Timetabling Organization ». In : *Multi-Agent Systems and Applications IV - 4th International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS'05), 15-17 September 2005, Budapest, Hungary*. T. 3690. Lecture Notes in Artificial Intelligence (LNAI). Springer-Verlag, p. 440–449. DOI : [10.1007/11559221_44](https://doi.org/10.1007/11559221_44). URL : <http://www.springerlink.com/content/bcvm1jjad712erpv/> (cité pp. 23, 27, 42, 44, 46, 51, 84). [AR=40.7%] [Pre-proceedings – 1 evaluation round]
- PICARD, G. et GLEIZES, M.-P. (2005b). « Cooperative Self-Organization to Design Robust and Adaptive Collectives ». In : *Second International Conference on Informatics in Control, Automation and Robotics (ICINCO'05), 14-17 September 2005, Barcelona, Spain, Volume I*. INSTICC Press, p. 236–241 (cité pp. 46, 51). [AR=45.07%] [Pre-proceedings – 1 evaluation round]
- PICARD, G. et GLIZE, P. (2005b). « Model and Experiments of Local Decision Based on Cooperative Self-Organization ». In : *Second International Indian Conference on Artificial Intelligence (IICAI'05), 20-22 December 2005, Pune, India*. Sous la dir. de PRASAD, B., 3009–3024. ISBN : 0-9727412-1-6 (cité p. 51). [AR=35%] [Pre-proceedings – 1 evaluation round]
- CAPERA, D., PICARD, G., GLEIZES, M.-P. et GLIZE, P. (2004). « Applying ADELFE Methodology to a Mechanism Design Problem ». In : *Third Joint Conference on Multi-Agent System (AAMAS'04)*. New York, USA : IEEE Computer Society, p. 1508–1509. ISBN : 1-58113-864-4. DOI : [10.1109/AAMAS.2004.66](https://doi.org/10.1109/AAMAS.2004.66). URL : <http://portal.acm.org/citation.cfm?id=1019006> (cité p. 60). [AR=49.7%] [CORE A+ – Pre-proceedings – 1 review phase]
- PICARD, G., BERNON, C. et GLEIZES, M.-P. (2004). « Cooperative Agent Model within ADELFE Framework : An Application to a Timetabling Problem ». In : *Third Joint Conference on Multi-Agent System (AAMAS'04)*. New York, USA : IEEE Computer Society, p. 1506–1507. ISBN : 1-58113-864-4. DOI : [10.1109/AAMAS.2004.93](https://doi.org/10.1109/AAMAS.2004.93). URL : <http://portal.acm.org/citation.cfm?id=1019005>. [AR=49.7%] [CORE A+ – Pre-proceedings – 1 review phase]
- PICARD, G. et GLEIZES, M.-P. (2002). « An Agent Architecture to Design Self-Organizing Collectives : Principles and Application ». In : *AISB'02 Symposium on Adaptive Multi-Agent Systems (AAMASII)*. Sous la dir. de D. KAZAKOV, D. KUDENKO et E. ALONSO. T. 2636. Lecture Notes in Artificial Intelligence (LNAI). University of London, UK : Springer-Verlag, p. 141–158. ISBN : 3-540-40068-0. DOI : [10.1007/3-540-44826-8_9](https://doi.org/10.1007/3-540-44826-8_9). URL : <http://www.springerlink.com/content/8gqx5072vmb132ta/>. [Post-proceedings – Selected and extended papers – 2 review phases]

11.5 Workshops internationaux (avec comité de lecture)

- BELLONI, A., BERGER, A., BOISSIER, O., BONNET, G., BOURGNE, G., CHARDEL, P., COTTON, J., EVREUX, N., GANASCIA, J., JAILLON, P., MERMET, B., PICARD, G., REVER, B., SIMON, G., SWARTE, T. de, TESSIER, C., VEXLER, F., VOYER, R. et ZIMMERMANN, A. (2015). « Dealing With Ethical Conflicts In Autonomous Agents And Multi-Agent Systems ». In : *Workshop on AI and Ethics at The Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15)*. [AR=40%]
- SORICI, A., PICARD, G., BOISSIER, O. et FLOREA, A. (2015). « Flexible Design and Deployment of Context-Aware Applications with CONSERT ». In : *11th International Workshop on Context and Activity Modeling and Recognition (CoMoRea'15)*. IEEE. [in submission]

- GILLANI, S., LAFOREST, F. et PICARD, G. (2014). « A Generic Ontology for Prosumer-Oriented Smart Grid ». In : *3rd Workshop on Energy Data Management at 17th International Conference on Extending Database Technology*. CEUR Workshop Proceedings. URL : <http://ceur-ws.org/Vol-1133/#paper-21> (cité pp. 10, 126).
- GILLANI, S., PICARD, G. et LAFOREST, F. (2014a). « IntelSCEP : Towards an Intelligent Semantic Complex Event Processing Framework for Prosumer-Oriented SmartGrid ». In : *International Workshop on Web Intelligence and Smart Sensing (IWWISS'14)*. Sous la dir. de P. MARET et S. HONDA. ACM Digital Library. DOI : [10.1145/2637064.2637110](https://doi.org/10.1145/2637064.2637110). URL : <http://dl.acm.org/citation.cfm?id=2637110&CFID=422236107&CFTOKEN=63603169> (cité p. 126). [Poster]
- (2014b). « Towards a Distributed Semantically Enriched Complex Event Processing and Pattern Matching ». In : *3rd International Workshop on Ordering and Reasoning (OrdRing'14)*. CEUR Workshop Proceedings (cité p. 126).
- PERSSON, C., PICARD, G., RAMPARANY, F. et BOISSIER, O. (2014). « A Multi-Agent based Governance of Machine-to-Machine Systems ». In : *International Workshop on Web Intelligence and Smart Sensing (IWWISS'14)*. Sous la dir. de P. MARET et S. HONDA. ACM Digital Library, p. 1–2. DOI : [10.1145/2637064.2637112](https://doi.org/10.1145/2637064.2637112). URL : <http://dl.acm.org/citation.cfm?id=2637112&CFID=422236107&CFTOKEN=63603169> (cité pp. 80, 125). [Poster]
- SORICI, A., PICARD, G. et BOISSIER, O. (2014). « Towards an Agent enabled Context Management Middleware ». In : *International Workshop on Web Intelligence and Smart Sensing (IWWISS'14)*. Sous la dir. de P. MARET et S. HONDA. ACM Digital Library. DOI : [10.1145/2637064.2637111](https://doi.org/10.1145/2637064.2637111). URL : <http://dl.acm.org/citation.cfm?id=2637111&CFID=422236107&CFTOKEN=63603169> (cité pp. 92, 126). [Poster]
- RIVIÈRE, J., LE RICHE, R. et PICARD, G. (2013). « LOOM, an algorithm for finding local optima of expensive functions ». In : *International France-China Workshop, NICST'2013 (New and smart Information Communication Science and Technology to support Sustainable Development), 18-20 September 2013, Clermont Ferrand, France*. LIMOS/UPB. ISBN : 978-2-9544948-0-7 (cité pp. 26, 27, 37, 40, 83, 85). [Pre-proceedings – 1 review phase]
- SORICI, A., BOISSIER, O., PICARD, G. et ZIMMERMANN, A. (2013). « Applying Semantic Web Technologies to Context Modeling in Ambient Intelligence ». In : *Evolving Ambient Intelligence : AmI 2013 Workshops, Dublin, Ireland, December 3-5, 2013. Revised Selected Papers*. Sous la dir. de M. O'GRADY, H. VAHDAT-NEJAD, K. WOLF, M. DRAGONE, J. YE, C. RÖCKER et G. O'HARE. Communications in Computer and Information Science 413. Springer, p. 217–229. DOI : [10.1007/978-3-319-04406-4_22](https://doi.org/10.1007/978-3-319-04406-4_22). URL : <http://www.percam.org> (cité pp. 10, 92, 126). [CORE C – Pre-proceedings – 1 review phase]
- VILLANUEVA, D., LE RICHE, R., PICARD, G. et HAFTKA, R. (2013b). « Self-organized Space Partitioning for Multi-Agent Optimization ». In : *6th International Workshop on Optimisation in Multi-Agent Systems (OPTMAS 2013, in conjunction with AAMAS 2013 6th-7th May 2013)* (cité pp. 26, 27, 33, 37, 40, 83, 126). [Pre-proceedings – 1 review phase]
- BILAL, M., PERSSON, C., RAMPARANY, F., PICARD, G. et BOISSIER, O. (2012). « Multi-Agent based governance model for Machine-to-Machine networks in a smart parking management system ». In : *Proceedings of IEEE International Conference on Communications, ICC 2012, Ottawa, ON, Canada, June 10-15, 2012, 3rd IEEE International Workshop on SmARt COmmunications in NEtwork Technologies ('ICC'12 WS - SaCoNet-III)*. IEEE Computer Society, p. 6468–6472. DOI : [10.1109/ICC.2012.6364789](https://doi.org/10.1109/ICC.2012.6364789) (cité pp. 69, 75, 80, 125, 126). [EIC 160/788 in Communications – Pre-proceedings – 1 review phase]
- PERSSON, C., PICARD, G., RAMPARANY, F. et BOISSIER, O. (2012b). « Multi-Agent Based Governance of Machine-to-Machine Systems ». In : *9th European Workshop (EUMAS 2011), Revised Selected Papers*. Sous la dir. de M. COSSENTINO, M. KAISERS, K. TUYLS et G. WEISS. T. 7541. Lecture Notes

- in Computer Science (LNCS). Springer, p. 205–220. DOI : [10.1007/978-3-642-34799-3_14](https://doi.org/10.1007/978-3-642-34799-3_14). URL : <http://www.springer.com/computer/ai/book/978-3-642-34798-6> (cité pp. 57, 58, 60, 62, 65, 68, 69, 72, 80, 125). [AR=26%] [Post-proceedings – Selected and extended papers – 2 review phases]
- SORICI, A., PICARD, G., BOISSIER, O., SANTI, A. et HÜBNER, J. (2012). « Multi-Agent Oriented Reorganisation within the JaCaMo infrastructure ». In : *The 3rd International Workshop on Infrastructures and Tools for Multiagent Systems (ITMAS 2012)* (cité pp. 57, 58, 65, 66, 68, 69, 126). [No proceedings – 1 review phase]
- PERSSON, C., PICARD, G., RAMPARANY, F. et BOISSIER, O. (2011a). « A Multi-Agent Organization for the Governance of Machine-to-Machine Systems ». In : *European Workshop on Multi-agent Systems (EUMAS'11)* (cité pp. 69, 80, 125). [CORE C – Pre-proceedings – 1 review phase]
- SORICI, A., BOISSIER, O., PICARD, G. et SANTI, A. (2011). « Exploiting the JaCaMo Framework for Realising an Adaptive Room Management Application ». In : *AGERE! (Actors and aGents REloaded) Programming Systems, Languages, and Applications based on Actors, Agents, and Decentralized Control workshop at ACM SPLASH 2011*. ACM Press, p. 239–242. DOI : [10.1145/2095050.2095088](https://doi.org/10.1145/2095050.2095088). URL : <http://doi.acm.org/10.1145/2095050.2095088> (cité pp. 57, 58, 65, 68, 69, 80, 126). [Pre-proceedings – 1 review phase]
- YAICH, R., BOISSIER, O., JAILLON, P. et PICARD, G. (2011a). « Social-Compliance in Trust Management within Virtual Communities ». In : *3rd International Workshop on Web Intelligence and Communities (WI&C'11) at the International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT 2011)*. IEEE Computer Society, p. 322–325. DOI : [10.1109/WI-IAT.2011.212](https://doi.org/10.1109/WI-IAT.2011.212) (cité pp. 80, 125). [CORE C – Pre-proceedings – 1 review phase]
- YAICH, R., BOISSIER, O., PICARD, G. et JAILLON, P. (2011b). « Social-Compliance in Trust Management within Virtual Communities ». In : *European Workshop on Multi-agent Systems (EUMAS'11)* (cité pp. 80, 125). [CORE C – Pre-proceedings – 1 review phase]
- GEORGÉ, J.-P., GLEIZES, M.-P., KADDOUM, E., MASCIARDI, L., PICARD, G. et RAIBULET, C. (2010). « Criteria for Self-* Systems Evaluation : a Unified Proposal ». In : *ICSE 2010 Workshop on Software Engineering for Adaptive and Self-managing Systems (SEAMS), Cape Town, South Africa*. ACM/IEEE, p. 29–38. DOI : [10.1145/1808984.1808988](https://doi.org/10.1145/1808984.1808988) (cité pp. 42, 51, 126). [Pre-proceedings – 1 review phase]
- PICARD, G., HÜBNER, J., BOISSIER, O. et GLEIZES, M.-P. (2009a). « Reorganisation and Self-organisation in Multi-Agent Systems ». In : *International Workshop on Organizational Modeling (OrgMod'09)*, p. 66–80. [AR=57.3%] [Pre-proceedings – 1 review phase]
- GARCIA RUIZ, J. E., GEORGÉ, J.-P. et PICARD, G. (2008). « The AmICriM Project : A Truly Body Area Network Application ». In : *First International Workshop on Sensor Networks (SN 2008), in conjunction with ICCCN 2008, August 4-7, Virgin Islands, USA*. [No proceedings – 1 review phase]
- HÜBNER, J., BORDINI, R. H. et PICARD, G. (2008a). « Jason and MOISE+ : Organisational programming in the Agent Contest 2008 ». In : *Dagstuhl Seminar on Programming Multi-Agent Systems*. Sous la dir. de R. BORDINI, M. DASTANI, J. DIX et A. EL FALLAH-SEGHROUCHNI. T. 08361.
- (2008b). « Using jason and MOISE+ to develop a team of cowboys ». In : *Proceedings of the Seventh international Workshop on Programming Multi-Agent Systems (ProMAS 08), Agent Contest, held with The Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2008)*. Sous la dir. de K. HINDRIKS, A. POKAHR et S. SARDINA, p. 238–242. DOI : [10.1007/978-3-642-03278-3](https://doi.org/10.1007/978-3-642-03278-3). URL : <http://www.springerlink.com/content/t402t436636r/#section=185314&page=1&locus=0>. [CORE B – Post-proceedings – Demo Contest selected paper – 1 review phase, 1 demonstration]
- BERNON, C., GLEIZES, M.-P. et PICARD, G. (2006). « Enhancing Self-Organising Emergent Systems Design with Simulation ». In : *Seventh International Workshop on Engineering Societies in the Agents World (ESAW'06), Dublin, Ireland from the 6th - 8th September, 2006*. Lecture Notes in Compu-

- ter Science (LNCS) 4457. Springer-Verlag, p. 284–299. DOI : [10.1007/978-3-540-75524-1](https://doi.org/10.1007/978-3-540-75524-1). URL : <http://www.springerlink.com/content/978-3-540-75522-7/#section=347810&page=1&locus=0>. [AR=58%] [MRS 79/361 – Post-proceedings – Selected and extended papers – 2 review phases]
- CAPERA, D., PICARD, G., GLEIZES, M.-P. et GLIZE, P. (2005). « A Sample Application of ADELFE Focusing on Analysis and Design : The Mechanism Design Problem ». In : *Fifth International Workshop on Engineering Societies in the Agents World (ESAW'04), 20-22 October 2004, Toulouse, France*. Sous la dir. de M.-P. GLEIZES, A. OMICINI et F. ZAMBONELLI. T. 3451. Lecture Notes in Artificial Intelligence (LNAI). Springer-Verlag, p. 231–244. DOI : [10.1007/11423355_17](https://doi.org/10.1007/11423355_17). URL : <http://www.springerlink.com/content/3kc056tmv1bf78bn/>. [AR=51.1%] [MRS 79/361 – Post-proceedings – Selected and extended papers – 2 review phases]
- PICARD, G. (2005a). « Cooperative Agent Model Instantiation to Collective Robotics in ADELFE ». In : *Fifth International Workshop on Engineering Societies in the Agents World (ESAW'04), 20-22 October 2004, Toulouse, France*. Sous la dir. de M.-P. GLEIZES, A. OMICINI et F. ZAMBONELLI. T. 3451. Lecture Notes in Artificial Intelligence (LNAI). Springer-Verlag, p. 209–221. DOI : [10.1007/11423355_15](https://doi.org/10.1007/11423355_15). URL : <http://www.springerlink.com/content/5ek2vg08a3qcyxuw/>. [AR=51.1%] [MRS 79/361 – Post-proceedings – Selected and extended papers – 2 review phases]
- PICARD, G., BERNON, C. et GLEIZES, M.-P. (2005b). « ETTO : Emergent Timetabling by Cooperative Self-Organization ». In : *Engineering Self-Organizing Applications – Third International Workshop (ESOA) at the Fourth International Joint Conference on Autonomous Agents and Multi-Agents Systems (AAMAS'05), July 2005, Utrecht, Netherlands*. T. 3910. Lecture Notes in Artificial Intelligence (LNAI). Springer-Verlag, p. 31–45. DOI : [10.1007/11734697_3](https://doi.org/10.1007/11734697_3). URL : <http://www.springerlink.com/content/v5q7611867rq3011/> (cité pp. 43, 51). [AR=47%] [Post-proceedings – Selected and extended papers – 2 review phases]
- PICARD, G. et GLEIZES, M.-P. (2005a). « Cooperative Self-Organization : Designing Robust and Adaptive Robotic Collectives ». In : *3rd European Workshop on Multi-Agent Systems (EUMAS'05), 7-8 December, Brussels, Belgium*. Koninklijke Vlaamse Academie van Belie voor Wetenschappen en Kunsten, p. 495–496. [CORE C – Pre-proceedings – 1 review phase]
- PICARD, G. et GLIZE, P. (2005a). « Cooperative Self-Organization : Modeling and Experiments of Local Decision to Solve Distributed Problems ». In : *3rd European Workshop on Multi-Agent Systems (EUMAS'05), 7-8 December, Brussels, Belgium*. Koninklijke Vlaamse Academie van Belie voor Wetenschappen en Kunsten, p. 497–498 (cité p. 51). [CORE C – Pre-proceedings – 1 review phase]
- PICARD, G., MELLOULI, S. et GLEIZES, M.-P. (2005). « Techniques for Multi-Agent System Reorganization ». In : *Sixth International Workshop on Engineering Societies in the Agents World (ESAW'05), 26-28 October 2005, Kuşadası, Aydın, Turkey*. Sous la dir. d'O. DIKENELLI, M.-P. GLEIZES et A. RICCI. T. 3963. Lecture Notes in Artificial Intelligence (LNAI). Springer-Verlag, p. 142–152. DOI : [10.1007/11759683_9](https://doi.org/10.1007/11759683_9). URL : <http://www.springerlink.com/content/u5t7k34040506374/> (cité pp. 64, 65, 69). [AR=44%] [MRS 79/361 – Post-proceedings – Selected and extended papers – 2 review phases]
- BERNON, C., CAMPS, V., GLEIZES, M.-P. et PICARD, G. (2003a). « Designing Agents' Behaviours within the Framework of ADELFE Methodology ». In : *Fourth International Workshop on Engineering Societies in the Agents World (ESAW'03)*. Sous la dir. d'A. OMICINI, P. PETTA et J. PITT. T. 3071. Lecture Notes in Artificial Intelligence (LNAI). Imperial College London, UK : Springer-Verlag, p. 311–327. DOI : [10.1007/978-3-540-25946-6_20](https://doi.org/10.1007/978-3-540-25946-6_20). URL : <http://www.springerlink.com/content/m4ya2cvcjx1grhcm/>. [AR=32.2%] [MRS 79/361 – Post-proceedings – Selected and extended papers – 2 review phases]
- (2003b). « Tools for Self-Organizing Applications Engineering ». In : *Engineering Self-Organizing Applications – First International Workshop (ESOA) at the Second International Joint Conference on Autonomous Agents and Multi-Agents Systems (AAMAS'03)*. Sous la dir. de G. DI MARZO SERUGENDO,

- A. KARAGEORGOS, O. F. RANA et F. ZAMBONELLI. T. 2977. *Lecture Notes in Artificial Intelligence (LNAI)*. Melbourne, Australia : Springer-Verlag, p. 283–298. ISBN : 3-540-21201-9. DOI : [10.1007/978-3-540-24701-2_19](https://doi.org/10.1007/978-3-540-24701-2_19). URL : <http://springerlink.metapress.com/content/0muewchfkxx1nym0/>. [Post-proceedings – Extended papers – 2 review phases]
- GEORGÉ, J.-P., PICARD, G., GLEIZES, M.-P. et GLIZE, P. (2003). « Living Design for Open Computational Systems ». In : *International Workshop on Theory And Practice of Open Computational Systems (TAPOCS) at 12th IEEE International Workshop on Enabling Technologies : Infrastructure for Collaborative Enterprises (WETICE'03)*. Sous la dir. de M. FREDRIKSSON, A. RICCI, R. GUSTAVSSON et A. OMICINI. Linz, Austria : IEEE Computer Society, p. 389–394. ISBN : 0-7695-1963-6. DOI : [10.1109/ENABL.2003.1231442](https://doi.org/10.1109/ENABL.2003.1231442). URL : <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1231442>. [AR=76.7%] [CORE B – Pre-proceedings – 1 review phase – Workshop best paper]
- PICARD, G. (2003b). « UML Stereotypes Definition and AUML Notations for ADELFE Methodology with OpenTool ». In : *The First European Workshop on Multi-Agent Systems (EUMAS'03), St Catherine College, Oxford, 18th and 19th December 2003*. [CORE C – 1 review phase]
- BERNON, C., GLEIZES, M.-P., PEYRUQUEOU, S. et PICARD, G. (2002). « ADELFE : a Methodology for Adaptive Multi-Agent Systems Engineering ». In : *Third International Workshop on Engineering Societies in the Agents World (ESAW'02)*. Sous la dir. de P. PETTA, R. TOLKSDORF et F. ZAMBONELLI. T. 2577. *Lecture Notes in Computer Science (LNCS)*. Madrid, Spain : Springer-Verlag, p. 156–169. ISBN : 3-540-14009-3. DOI : [10.1007/3-540-39173-8_12](https://doi.org/10.1007/3-540-39173-8_12). URL : <http://www.springerlink.com/content/rheud1raydumk5g1/>. [AR=57.1%] [MRS 79/361 – Post-proceedings – Selected and extended papers – 2 review phases]
- BERNON, C., GLEIZES, M.-P., PICARD, G. et GLIZE, P. (2002). « The ADELFE Methodology For an Intranet System Design ». In : *Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002)*. Sous la dir. de P. GIORGINI, Y. LESPÉRANCE, G. WAGNER et E. YU. T. 57. CAiSE'02. Toronto, Canada : CEUR Workshop Proceedings. URL : <http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-57/>. [CORE B – Online pre-proceedings – 1 review phase]

11.6 Conférences nationales (avec comité de lecture)

- GALLAND, S., GAUD, N., RODRIGUEZ, S., BALBO, F., PICARD, G. et BOISSIER, O. (2014). « Contextualiser l'interaction entre agents en combinant dimensions sociale et physique au sein de l'environnement ». In : *22es Journées Francophones sur les Systèmes Multi-Agents (JFSMA'14)*. Cépaduès (cité p. 91). [AR=28%] [CNU27 – Pre-proceedings – 1 review phase]
- PICARD, G., VILLANUEVA, D., LE RICHE, R. et HAFTKA, R. (2013). « Méthode multi-agent d'optimisation par partitionnement auto-organisé ». In : *21es Journées francophones des systèmes multi-agents (JFSMA'13)*. Cépaduès (cité pp. 25–27, 33, 37, 40, 126). [AR=33%] [CNU27 – Pre-proceedings – 1 review phase]
- PIETTE, E., MORGE, M. et PICARD, G. (2013). « Swing++ : méthode multi-agents pour la résolution du problème des mariages stables ». In : *Septièmes journées francophones Modèles Formels de l'Interaction (MFI'13)* (cité pp. 24, 50, 51).
- VILLANUEVA, D., PICARD, G., LE RICHE, R. et HAFTKA, R. T. (2012). « Optimisation multi-agent par partitionnement adaptatif de l'espace de conception ». In : *20es Journées francophones des systèmes multi-agents (JFSMA'12)*. Cépaduès, p. 149–158 (cité pp. 30, 33, 34, 40, 126). [AR=50%] [CNU27 – Pre-proceedings – 1 review phase]
- EVERAERE, P., MORGE, M. et PICARD, G. (2011). « Casanova : un comportement d'agent pour l'équité des mariages préservant la confidentialité ». In : *19es Journées francophones des systèmes multi-agents*

- (JFSMA'11). Cépaduès, p. 203–212. [AR=32.5%] [CNU27 – Pre-proceedings – 1 review phase – Best Paper Award]
- PERSSON, C., PICARD, G., RAMPARANY, F. et BOISSIER, O. (2011b). « Organisation multi-agent pour la gouvernance de systèmes Machine-to-Machine ». In : *19es Journées francophones des systèmes multi-agents (JFSMA'11)*. Cépaduès, p. 11–20 (cité pp. 69, 80, 125). [AR=32.5%] [CNU27 – Pre-proceedings – 1 review phase]
- YAICH, R., JAILLON, P., BOISSIER, O. et PICARD, G. (2011). « Gestion de la confiance et intégration des exigences sociales au sein de communautés virtuelles ». In : *19es Journées francophones des systèmes multi-agents (JFSMA'11)*. Cépaduès, p. 213–222 (cité pp. 80, 125). [AR=48.8%] [CNU27 – Pre-proceedings – 1 review phase]
- KADDOUM, E., GLEIZES, M.-P., GEORGÉ, J.-P., GLIZE, P. et PICARD, G. (2009). « Analyse des critères d'évaluation de systèmes multi-agents adaptatifs ». In : *Journées Francophones sur les Systèmes Multi-Agents (JFSMA'09)*, Lyon, France, October 19-21. Sous la dir. de Z. GUESSOUM et S. HASSAS. Cépaduès, p. 123–124 (cité pp. 42, 51, 126). [AR=23.5%] [CNU27 – Pre-proceedings – 1 review phase]
- PICARD, G., HÜBNER, J., BOISSIER, O. et GLEIZES, M.-P. (2009b). « Réorganisation et auto-organisation dans les systèmes multi-agents ». In : *Journées Francophones sur les Systèmes Multi-Agents (JFSMA'09)*, Lyon, France, October 19-21. Sous la dir. de Z. GUESSOUM et S. HASSAS. Cépaduès, p. 89–98 (cité pp. 10, 11, 51). [AR=39.2%] [CNU27 – Pre-proceedings – 1 review phase]
- CLAIR, G., GLEIZES, M.-P., KADDOUM, E. et PICARD, G. (2008a). « Approches multi-agents auto-organisatrices pour un contrôle manufacturier intelligent et adaptatif ». In : *Journées Francophones sur les Systèmes Multi-Agents (JFSMA'08)*, Brest, France, Octobre 15-17. Cépaduès, p. 191–200 (cité pp. 51, 126). [AR=40.7%] [CNU27 – Pre-proceedings – 1 review phase]
- PICARD, G., GLEIZES, M.-P. et GLIZE, P. (2007a). « Affectation distribuée de fréquences par auto-organisation coopérative ». In : *Journées Francophones sur les Systèmes Multi-Agents (JFSMA'07)*, Carcassonne, France, Octobre 17-19. Cépaduès, p. 33–42 (cité p. 51). [AR=22.2%] [CNU27 – Pre-proceedings – 1 review phase]
- PICARD, G. et GLEIZES, M.-P. (2006). « Auto-organisation coopérative pour la conception de collectifs adaptatifs et robustes ». In : *7ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'06)*, 6, 7 et 8 Février 2006, Lille, France. Presses Universitaires de Valenciennes, p. 385–400 (cité p. 51). [AR=58%] [Pre-proceedings – 1 review phase]
- PICARD, G. (2005b). « Résolution d'emploi du temps dynamique et distribuée par auto-organisation coopérative ». In : *7^{es} Rencontres des Jeunes Chercheurs en Intelligence Artificielle (RJCIA'05)*, Plateforme AFIA, Nice. Presses Universitaires de Grenoble (PUG), p. 127–140. ISBN : 2-7061-1285-9 (cité p. 51). [AR=55%] [Pre-proceedings – 1 review phase]
- PICARD, G. et GLIZE, P. (2005c). « Modélisation et expérimentations d'une décision locale basée sur l'auto-organisation coopérative ». In : *Journées Francophones sur les Systèmes Multi-Agents (JFSMA'05)*, à Calais du 23 au 25 novembre 2005. Hermès-Lavoisier, p. 161–174 (cité p. 51). [CNU27 – Pre-proceedings – 1 review phase]

11.7 Autres communications (avec comité de lecture)

- VILLANUEVA, D., HAFKA, R., LE RICHE, R. et PICARD, G. (2013). « Locating Multiple Designs with Dynamic Partitioning and Surrogates ». In : *10th World Congress on Structural and Multidisciplinary Optimization (WCSMO'13)*. ISSMO (cité pp. 40, 85, 126). [No proceedings – 1 review phase (abstract)]
- PICARD, G. et MORGE, M. (2011). « Stratégie multi-agent pour la négociation d'appariements stables, équitables et optimaux ». In : *12e Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'11)*, Saint-Etienne (cité p. 51). [No proceedings – 1 review phase (abstract)]

- VILLANUEVA, D., LE RICHE, R., PICARD, G. et HAFTKA, R. (2011). « A Multi-Agent System Approach To Reliability Based Design Optimization Including Future Tests ». In : *12e Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Decision (ROADEF'11)*, Saint-Etienne (cité pp. 40, 126). [No proceedings – 1 review phase (abstract)]
- YAICH, R., JAILLON, P., PICARD, G. et BOISSIER, O. (2010). « Toward an adaptive trust policy model for open and decentralized virtual communities ». In : *Workshop on Trust and Reputation*. Interdisciplines. URL : <http://www.interdisciplines.org/paper.php?paperID=110> (cité p. 125). [Online pre-proceedings – 1 review phase (abstract)]

11.8 Thèses et mémoires

- PICARD, G. (2004). « Méthodologie de développement de systèmes multi-agents adaptatifs et conception de logiciels à fonctionnalité émergente ». Thèse de doct. Université Paul Sabatier Toulouse III (cité pp. 55–57, 60).
- (2001). « Etude de l'émergence comportementale d'un collectif de robots par auto-organisation coopérative ». Rapport de DEA. Université Paul Sabatier Toulouse III.

Bibliographie

- ALBEROLA, J. M., JULIAN, V. et GARCIA-FORNES, A. (2012). « Cost-Aware Reorganization Service for Multiagent Systems ». In : *Proceedings of the 10th International Conference on Advanced Agent Technology*. AAMAS'11. Berlin, Heidelberg : Springer-Verlag, p. 442–456. DOI : [10.1007/978-3-642-27216-5_34](https://doi.org/10.1007/978-3-642-27216-5_34). URL : http://dx.doi.org/10.1007/978-3-642-27216-5_34 (cité p. 14).
- ARKIN, R. (1998). *Behavior-Based Robotics*. MIT Press (cité p. 56).
- ARMETTA, F., HASSAS, S., PIMONT, S. et LEFEVRE, O. (2007). « Vers le contrôle de l'émergence par la coordination d'agents pour le partage de ressources critiques ». In : *Revue d'Intelligence Artificielle* 21.5-6, p. 673–702 (cité p. 42).
- AURENHAMMER, F. (1991). « Voronoi diagrams : a survey of a fundamental geometric data structure ». In : *ACM Computing Surveys (CSUR)* 23.3, p. 345–405 (cité p. 34).
- BADEIG, F. et BALBO, F. (2012). « Définition d'un cadre de conception et d'exécution pour la simulation multi-agent ». In : *Revue d'Intelligence Artificielle* 26.3, p. 255–280 (cité p. 58).
- BERNOUX, P. (1985). *La sociologie des organisations*. 3ème. Seuil.
- BILAL, M. (2011). « Multi-agent governance model for M2M networks : Application to a smart parking management system ». Rapport de master. Université de technologie de Troyes (cité p. 80).
- BOISSIER, O., BORDINI, R., HÜBNER, J., RICCI, A. et SANTI, A. (2013). « Multi-agent oriented programming with JaCaMo ». In : *Sci. Comput. Program.* 78.6, p. 747–761 (cité pp. 58, 59).
- BOISSIER, O., HÜBNER, J. et SICHTMAN, J. (2007). « Organization Oriented Programming From Closed to Open Organizations ». In : *Engineering Societies in the Agents World VII (ESAW 06)*. T. 4457. LNCS. Springer-Verlag, p. 86–105. DOI : [10.1007/978-3-540-75524-1_5](https://doi.org/10.1007/978-3-540-75524-1_5) (cité p. 10).
- BONABEAU, E., DORIGO, M. et THERAULAZ, G. (1999). *"Swarm Intelligence : From Natural to Artificial Systems"*. Oxford University Press (cité pp. 42, 44, 55, 61).
- BROOKS, R. (1986). « A robust layered control system for a mobile robot ». In : *Robotics and Automation, IEEE Journal of* 2.1, p. 14–23. DOI : [10.1109/JRA.1986.1087032](https://doi.org/10.1109/JRA.1986.1087032) (cité p. 56).
- BRUECKNER, S. (2000). « Return from the Ant : Synthetic Ecosystems for Manufacturing Control ». Thèse de doct. Department of Computer Science, Humboldt University Berlin (cité p. 42).
- BRUECKNER, S. et PARUNAK, H. V. D. (2004). « Self-Organizing MANET Management ». In : *Engineering Self-Organising Systems, Nature-Inspired Approaches to Software Engineering [revised and extended papers presented at the Engineering Self-Organising Applications Workshop, ESOA 2003, held at AAMAS 2003 in Melbourne, Australia, in July 2003 and selected invited papers from leading researchers in self-organisation]*. T. 2977. Lecture Notes in Computer Science (LNCS). Springer, p. 20–35 (cité p. 42).
- BUSSMANN, S., JENNINGS, N. R. et WOOLDRIDGE, M. (2004). *Multiagent Systems for Manufacturing Control*. Springer (cité p. 25).

- CALVEZ, B. et HUTZLER, G. (2008). « Optimisation Dichotomique Adaptative : une nouvelle méthode pour le calibrage de modèles à base d'agents (présentation courte) ». In : *16es journées francophones sur les systèmes multi-agents (JFSMA)*, p. 181–190 (cité p. 26).
- CAPERA, D., BERNON, C. et GLIZE, P. (2006). « Étude d'un processus d'allocation coopératif de ressources entre agents pour la gestion de production ». In : *7^e Congrès de la ROADEF*. PUV, p. 369–383 (cité p. 25).
- CAPERA, D., GLEIZES, M. et GLIZE, P. (2004). « Mechanism Type Synthesis based on Self-Assembling Agents ». In : *Journal of Applied Artificial Intelligence* 18.9-10, p. 921–936 (cité pp. 13, 14, 26).
- CARRON, T. et BOISSIER, O. (2001). « Towards a Temporal Organizational Structure Language for Dynamic Multi-Agent Systems ». In : *Pre-Proceeding of the 10th European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW'2001)* (cité p. 15).
- CASTELFRANCHI, C. (2000). « Engineering Social Order ». In : *ESAW '00 : Proceedings of the First International Workshop on Engineering Societies in the Agent World*. Springer-Verlag, p. 1–18 (cité p. 13).
- CERQUEIRA-MARTINS, L. (2012). « Decentralized stable matching in mixed communities ». Rapport de master. Université Jean Monnet - Ecole des Mines de Saint-Etienne (cité pp. 50, 57).
- CHEVALEYRE, Y., DUNNE, P. E., ENDRISS, U., LANG, J., MAUDET, N. et RODRÍGUEZ-AGUILAR, J. A. (2005). « Multiagent Resource Allocation ». In : *Knowledge Engineering Review* 20.2, p. 143–149. DOI : [10.1017/S0269888905000470](https://doi.org/10.1017/S0269888905000470). URL : <http://dx.doi.org/10.1017/S0269888905000470> (cité p. 22).
- CHEVRIER, V. (2002). « Contributions au domaine des systèmes multi-agents. » Thèse de doct. Université Henri Poincaré (cité p. 55).
- CHEVRIER, V., BOURJOT, C. et THOMAS, V. (2011). « Region Detection in Images ». In : *Self-organising Software*. Sous la dir. de G. DI MARZO SERUGENDO, M.-P. GLEIZES et A. KARAGEORGOS. Natural Computing Series. Springer Berlin Heidelberg, p. 425–445. DOI : [10.1007/978-3-642-17348-6_17](https://doi.org/10.1007/978-3-642-17348-6_17). URL : http://dx.doi.org/10.1007/978-3-642-17348-6_17 (cité p. 55).
- CLAIR, G. (2008). « Self-organisation for manufacturing control based on multi-agent systems ». Rapport de master. Université Jean Monnet - Ecole des Mines de Saint-Etienne (cité pp. 43, 46, 50).
- CORKILL, D. et LESSER, V. (1983). « The Use of Meta-Level Control for Coordination in Distributed Problem Solving Network ». In : *Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI'83)*, p. 748–756 (cité p. 13).
- CORNET, F. (2005). « Study of a frequency assignment problem using adaptive multi-agent systems ». Rapport de master. Université Paul Sabatier - Toulouse III (cité p. 50).
- COSENTINO, M., GAUD, N., GALLAND, S., HILAIRE, V. et KOUKAM, A. (2007). « A Holonic Metamodel for Agent-Oriented Analysis and Design ». In : *Holonic and Multi-Agent Systems for Manufacturing" (HoloMAS'07)*. T. LNAI 4659. Springer, p. 237–246 (cité p. 12).
- COSENTINO, M., GAUD, N., HILAIRE, V., GALLAND, S. et KOUKAM, A. (2010). « ASPECS : an agent-oriented software process for engineering complex systems ». In : *Autonomous Agents and Multi-Agent Systems* 20.2, p. 260–304. DOI : [10.1007/s10458-009-9099-4](https://doi.org/10.1007/s10458-009-9099-4). URL : <http://dx.doi.org/10.1007/s10458-009-9099-4> (cité p. 58).
- COUTINHO, L., SICHMAN, J. et BOISSIER, O. (2007). « Modeling Dimensions for Multi-Agent Systems Organizations ». In : *Agent Organizations : Models and Simulations (AOMS), Workshop held at IJCAI 07*.
- DANOY, G., BOUVRY, P. et BOISSIER, O. (2010). « A Multi-Agent Organizational Framework for Coevolutionary Optimization ». In : *Transactions on Petri Nets and Other Models of Concurrency IV*. Sous la dir. de K. JENSEN, S. DONATELLI et M. KOUTNY. T. 6550. Lecture Notes in Computer Science.

- Springer Berlin Heidelberg, p. 199–224. DOI : [10.1007/978-3-642-18222-8_9](https://doi.org/10.1007/978-3-642-18222-8_9). URL : http://dx.doi.org/10.1007/978-3-642-18222-8_9 (cité p. 25).
- DE WEERDT, M. et CLEMENT, B. (2009). « Introduction to Planning in Multiagent Systems ». In : *Multiagent and Grid Systems – An International Journal (MAGS)* 5.4, p. 345–355. URL : <http://dl.acm.org/citation.cfm?id=1735317.1735318> (cité pp. 13, 22).
- DECHTER, R. (2003). *Constraint Processing*. Morgan Kaufmann (cité p. 22).
- DELOACH, S. A. (2004). « The MaSE Methodology ». In : *Methodologies and Software Engineering for Agent Systems*. Kluwer (cité p. 12).
- DEMAZEAU, Y. (1995). « From Interactions to Collective Behaviour in Agent-Based Systems ». In : *Proceedings of the First European Conference on Cognitive Science (ECCS'95), Saint Malo, France*, p. 117–132 (cité pp. 60, 61).
- DI MARZO SERUGENDO, G., GLEIZES, M. P. et KARAGEORGOS, A. (2006). « Self-Organisation and Emergence in Multi-Agent Systems : An Overview ». In : *Informatica* 30.1, p. 45–54 (cité p. 13).
- DIGNUM, V., DIGNUM, F. et SONENBERG, L. (2004). « Towards dynamic organization of agent societies ». In : *Workshop on Coordination in Emergent Agent Societies*. Sous la dir. de G. VOUROS, p. 70–78 (cité p. 14).
- DORIGO, M. et STÜTZLE, T. (2004). *Ant Colony Optimization*. MIT Press (cité pp. 23, 24, 26, 42).
- DROGOUL, A., CORBARA, B. et LALANDE, S. (1995). « MANTA : New experimental results on the emergence of (artificial) ant societies ». In : *Artificial Societies : the Computer Simulation of Social Life*. UCL Press, p. 119–221 (cité pp. 10, 13, 16).
- ESTEVA, M., RODRIGUEZ-AGUIAR, J. A., SIERRA, C., GARCIA, P. et ARCOS, J. L. (2001). « On the formal specification of Electronic Institutions ». In : *Proceedings of the Agent-mediated Electronic Commerce*. LNAI 1191. Berlin : Springer, p. 126–147 (cité pp. 12, 58).
- FARINELLI, A., ROGERS, A., PETCU, A. et JENNINGS, N. R. (2008). « Decentralised Coordination of Low-power Embedded Devices Using the Max-sum Algorithm ». In : *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2. AAMAS'08*. Richland, SC : International Foundation for Autonomous Agents et Multiagent Systems, p. 639–646. URL : <http://dl.acm.org/citation.cfm?id=1402298.1402313> (cité p. 88).
- FERBER, J. (1995). *Les systèmes multi-agents*. Dunod (cité pp. 9, 10, 55).
- FERBER, J. et GUTKNECHT, O. (1998). « A Meta-Model for the Analysis and Design of Organizations in Multi-Agents Systems ». In : *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS'98)*. IEEE Press, p. 128–135 (cité pp. 12, 58).
- FININ, T., FRITZSON, R., MCKAY, D. et McENTIRE, R. (1994). « KQML As an Agent Communication Language ». In : *Proceedings of the Third International Conference on Information and Knowledge Management. CIKM '94*. New York, NY, USA : ACM, p. 456–463. DOI : [10.1145/191246.191322](https://doi.org/10.1145/191246.191322). URL : <http://doi.acm.org/10.1145/191246.191322> (cité p. 56).
- GALLAND, S. et GAUD, N. (2014). « Holonic Model of a Virtual 3D Indoor Environment for Crowd Simulation ». In : *International Workshop on Environments for Multiagent Systems (E4MAS14)*. Springer (cité p. 58).
- GASSER, L. et ISHIDA, T. (1991). « A Dynamic Organization Architecture for Adaptive Problem Solving ». In : *Proceedings Ninth National Conference on Artificial Intelligence (AAAI'91)*. The MIT Press & AAAI Press, p. 185–90 (cité p. 13).
- GENESERETH, M. R. et NILSSON, N. J. (1987). *Logical Foundations of Artificial Intelligence*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc. (cité p. 9).

- GEORGÉ, J.-P., EDMONDS, B. et GLIZE, P. (2004). « Making Self-Organizing Adaptive Multi-Agent Systems Work - Towards the engineering of emergent multi-agent systems (chapter 8) ». In : *Methodologies and Software Engineering for Agent Systems*. Kluwer, p. 319–338 (cité pp. 12, 14).
- GLASER, N. et MORIGNOT, P. (1997). « The Reorganization of Societies of Autonomous Agents ». In : *Multi-Agent Rationality*. LNAI 1237. Springer, p. 98–111 (cité p. 13).
- GLEIZES, M.-P. (2004). « Vers la résolution de problèmes par émergence ». Mémoire d'habilitation à diriger des recherches. Université Paul Sabatier Toulouse III (cité pp. 21, 23, 44).
- GLINTON, R., SYCARA, K., SCERRI, D. et SCERRI, P. (2010). « The Statistical Mechanics of Belief Sharing in Multi-agent Systems ». In : *Inf. Fusion* 11.3, p. 256–266. DOI : [10.1016/j.inffus.2009.09.003](https://doi.org/10.1016/j.inffus.2009.09.003). URL : <http://dx.doi.org/10.1016/j.inffus.2009.09.003> (cité p. 88).
- GLIZE, P. (2001). « L'adaptation des systèmes à fonctionnalité émergente par auto-organisation coopérative ». Mémoire d'habilitation à diriger des recherches. Université Paul Sabatier - Toulouse III (cité pp. 44, 59).
- GUSFIELD, D. et IRVING, R. W. (1989). *The Stable Marriage Problem : Structure and Algorithms*. MIT Press (cité p. 47).
- HARTIGAN, J. et WONG, M. (1979). « Algorithm AS 136: A K-Means Clustering Algorithm ». In : *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28.1, p. 100–108 (cité p. 36).
- HASSAS, S., DI MARZO-SERUGENDO, G., KARAGEORGOS, A. et CASTELFRANCHI, C. (2006). « On Self-Organising Mechanisms from Social, Business and Economic Domains ». In : *Informatica* 30.1, p. 63–71 (cité p. 13).
- HATTORI, H. et ITO, T. (2006). « A Quick Adaptation Method for Constraint Satisfaction in a Real-time Environment ». In : *International Journal of Computer Science and Network Security* 6.78, p. 107–112 (cité p. 26).
- HENDERSON-SELLERS, B. et GIORGINI, P., éd. (2005). *Agent-Oriented Methodologies*. Idea Group Publishing. DOI : [10.4018/978-1-59140-581-8](https://doi.org/10.4018/978-1-59140-581-8) (cité p. 59).
- HIRAYAMA, K. et YOKOO, M. (2005). « The distributed breakout algorithms ». In : *Artificial Intelligence* 161.1-2, p. 89–115 (cité p. 27).
- HORLING, B., BENYO, B. et LESSER, V. (2001). « Using Self-Diagnosis to Adapt Organizational Structures ». In : *Proceedings of the 5th International Conference on Autonomous Agentes (Agents' 01)* (cité p. 16).
- HÜBNER, J. F., BOISSIER, O. et BORDINI, R. (2010). « A Normative Organisation Programming Language for Organisation Management Infrastructures ». In : *Coordination, Organizations, Institutions and Norms in Agent Systems V*. Sous la dir. de J. PADGET, A. ARTIKIS, W. VASCONCELOS, K. STATHIS, V. da SILVA, E. MATSON et A. POLLERES. T. 6069. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 114–129. DOI : [10.1007/978-3-642-14962-7_8](https://doi.org/10.1007/978-3-642-14962-7_8). URL : http://dx.doi.org/10.1007/978-3-642-14962-7_8 (cité pp. 64, 66).
- HÜBNER, J., SICHMAN, J. et BOISSIER, O. (2002). « A Model for the Structural, Functional, and Deontic Specification of Organizations in Multiagent Systems ». In : *Advances in Artificial Intelligence*. Sous la dir. de G. BITTENCOURT et G. RAMALHO. T. 2507. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 118–128. DOI : [10.1007/3-540-36127-8_12](https://doi.org/10.1007/3-540-36127-8_12). URL : http://dx.doi.org/10.1007/3-540-36127-8_12 (cité pp. 12, 58, 63).
- HÜBNER, J., SICHMAN, J. et BOISSIER, O. (2004). « Using the $\mathcal{M}OISE^+$ for a Cooperative Framework of MAS Reorganisation ». In : *Advances in Artificial Intelligence – SBIA 2004*. Sous la dir. d'A. L. BAZZAN et S. LABIDI. T. 3171. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 506–515. DOI : [10.1007/978-3-540-28645-5_51](https://doi.org/10.1007/978-3-540-28645-5_51). URL : http://dx.doi.org/10.1007/978-3-540-28645-5_51 (cité pp. 13, 16, 65, 66, 68).

- HÜBNER, J. F. (2003). « Um Modelo de Reorganização de Sistemas Multiagentes ». Thèse de doct. Universidade de São Paulo, Escola Politécnica. URL : <http://www.inf.furb.br/~jomi/pubs/2003/Hubner-tese.pdf> (cité pp. 10, 11).
- HÜBNER, J., BOISSIER, O., KITIO, R. et RICCI, A. (2010). « Instrumenting multi-agent organisations with organisational artifacts and agents ». In : *Autonomous Agents and Multi-Agent Systems* 20, p. 369–400 (cité pp. 60, 61, 66).
- HUHNS, M. N. (2001). « Interaction-oriented programming ». In : *First international workshop, AOSE 2000 on Agent-oriented software engineering*. Springer, p. 29–44 (cité p. 59).
- JAYNES, E. T. (1957). « Information Theory and Statistical Mechanics ». In : *Phys. Rev.* 106, p. 620–630. DOI : [10.1103/PhysRev.106.620](https://doi.org/10.1103/PhysRev.106.620). URL : <http://link.aps.org/doi/10.1103/PhysRev.106.620> (cité p. 88).
- JORQUERA, T. (2013). « An adaptive multi-agent system for self-organizing continuous optimization ». Thèse de doct. Université de Toulouse (cité pp. 25, 37).
- KADDOUM, E. (2008). « Auto-régulation du contrôle manufacturier par système multi-agent auto-organisateur ». Rapport de master. Université Paul Sabatier - Toulouse III (cité pp. 43, 49, 50).
- (2011). « Optimisation sous contraintes de problèmes distribués par auto-organisation coopérative ». Thèse de doct. Université de Toulouse.
- KAMBOJ, S. et DECKER, K. S. (2007). « Organizational self-design in semi-dynamic environments ». In : *AAMAS '07 : Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. ACM, p. 1–8. DOI : <http://doi.acm.org/10.1145/1329125.1329370> (cité p. 13).
- KEARNS, M., PAI, M. M., ROTH, A. et ULLMAN, J. (2012). « Mechanism Design in Large Games : Incentives and Privacy ». In : *CoRR abs/1207.4084* (cité p. 88).
- KENNEDY, J. et EBERHART, R. C. (1995). « Particle swarm optimization ». In : *Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ*. P. 1942–1948 (cité p. 24).
- KLEIJEN, J. (2008). *Design and analysis of simulation experiments*. Springer (cité p. 30).
- KOLLER, D. et FRIEDMAN, N. (2009). *Probabilistic Graphical Models*. MIT Press (cité p. 88).
- KOTA, R., GIBBINS, N. et JENNINGS, N. R. (2012). « Decentralized Approaches for Self-adaptation in Agent Organizations ». In : *ACM Trans. Auton. Adapt. Syst.* 7.1, 1 :1–1 :28. DOI : [10.1145/2168260.2168261](https://doi.org/10.1145/2168260.2168261). URL : <http://doi.acm.org/10.1145/2168260.2168261> (cité p. 13).
- KUBERA, Y., MATHIEU, P. et PICAULT, S. (2008). « Interaction-Oriented Agent Simulations : From Theory to Implementation ». In : *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI'08)*. IOS Press, p. 383–387 (cité pp. 55, 59).
- LEMAÎTRE, C. et EXCELENTE, C. B. (1998). « Multi-Agent Organization Approach ». In : *Proceedings of II Iberoamerican Workshop on DAI and MAS* (cité p. 10).
- LESSER, V., DECKER, K., WAGNER, T., CARVER, N., GARVEY, A., HORLING, B., NEIMAN, D., PODOROZHNY, R., NAGENDRAPRASAD, M., RAJA, A., VINCENT, R., XUAN, P. et ZHANG, X. (2004). « Evolution of the GPGP/TAEMS Domain-Independent Coordination Framework ». In : *Autonomous Agents and Multi-Agent Systems* 9.1, p. 87–143 (cité pp. 12, 58).
- LIU, J., JING, H. et TANG, Y. Y. (2002). « Multi-agent Oriented Constraint Satisfaction ». In : *Artificial Intelligence* 136.1, p. 101–144 (cité pp. 23, 24, 42–44).
- LONG, M. T., MURPHY, R. R. et HICINBOTHOM, J. (2007). « Social roles for taskability in robot teams ». In : *International Conference on Intelligent Robots and Systems (IROS'07)*. IEEE, p. 2338–2344 (cité pp. 14, 16).

- MAHESWARAN, R. T., PEARCE, J. P. et TAMBE, M. (2004). « Distributed Algorithms for DCOP : A Graphical-Game-Based Approach ». In : *Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems (PDCS)*, p. 432–439 (cité p. 23).
- MAILLER, R. et LESSER, V. (2004). « Solving Distributed Constraint Optimization Problems Using Cooperative Mediation ». In : *AAMAS'04*. IEEE Computer Society, p. 438–445 (cité pp. 13, 16, 26).
- MAILLER, R. et LESSER, V. R. (2006). « Asynchronous Partial Overlay : A New Algorithm for Solving Distributed Constraint Satisfaction Problems ». In : *Journal of Artificial Intelligence Research* 25, p. 529–576 (cité p. 47).
- MAVRÍK, V., VYATKIN, V. et COLOMBO, A. W., édés. (2007). *Holonic and Multi-Agent Systems for Manufacturing, 3rd International Conference on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS'07)*. T. 4659. Springer (cité p. 25).
- MICHEL, F. (2007). « The IRM4S model : the influence/reaction principle for multiagent based simulation ». In : *Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS07)*. ACM. DOI : [10.1145/1329125.1329289](https://doi.org/10.1145/1329125.1329289).
- MINTON, S., JOHNSTON, M., PHILIPS, A. et LAIRD, P. (1994). « Minimizing Conflicts : a Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems ». In : *Constraint-Based Reasoning* 58.1-3, p. 161–205 (cité p. 43).
- MODI, P. J., SHEN, W., TAMBE, M. et YOKOO, M. (2005). « ADOPT : Asynchronous Distributed Constraint Optimization with Quality Guarantees ». In : *Artificial Intelligence* 161.2, p. 149–180 (cité pp. 22, 23, 25).
- MOULIN, H. (2003). *Fair Division and Collective Welfare*. MIT Press (cité p. 24).
- MÜLLER, J. P. (1998). « Vers une méthodologie de conception de systèmes multi-agents de résolution de problème par émergence ». In : *JFIADSMA'98*. Hermès (cité pp. 10, 21).
- NELDER, J. et MEAD, R. (1965). « A simplex for function minimization ». In : *Computer J.* 7, p. 308–313 (cité p. 38).
- ODELL, J., PARUNAK, H., FLEISHER, M. et BRUECKNER, S. (2002). « Modeling Agents and their Environment ». In : *Agent-Oriented Software Engineering III*. Sous la dir. de F. GIUNCHIGLIA, J. ODELL et G. WEISS. T. 2585. Lecture Notes In Computer Science. Springer (cité p. 58).
- PARUNAK, H. V. D. et BRUECKNER, S. A. (2004). « Engineering Swarming Systems ». In : *Methodologies and Software Engineering for Agent Systems*. Kluwer, p. 341–376 (cité p. 12).
- PARUNAK, H. V. D., BRUECKNER, S. et SAUTER, J. (2002). « Digital Pheromone Mechanisms for Coordination of Unmanned Vehicles ». In : *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'02)*. ACM Press, p. 449–450 (cité p. 42).
- PAVÓN, J. et GÓMEZ-SANZ, J. J. (2003). « Agent Oriented Software Engineering with INGENIAS ». In : *CEEMAS*, p. 394–403 (cité p. 12).
- PERSSON, C. (2014). « A Decentralized and Distributed Adaptive Governance for Machine-to-Machine Systems : A Multi-Agent Oriented Programming Approach ». Thèse de doct. Ecole des Mines de Saint-Etienne (cité pp. 59, 69, 71, 75, 80, 84).
- PETCU, A. (2007). « A Class of Algorithms for Distributed Constraint Optimization ». Thèse de doct. Swiss Federal Institute of Technology (EPFL) (cité p. 23).
- POSLAD, S. (2007). « Specifying Protocols for Multi-agent Systems Interaction ». In : *ACM Trans. Auton. Adapt. Syst.* 2.4. DOI : [10.1145/1293731.1293735](https://doi.org/10.1145/1293731.1293735). URL : <http://doi.acm.org/10.1145/1293731.1293735> (cité p. 56).
- PYNADATH, D. V., TAMBE, M., CHAUVAT, N. et CAVEDON, L. (1999). « Toward Team-Oriented Programming ». In : *ATAL*, p. 233–247 (cité p. 58).

- RAJNARAYAN, D., WOLPERT, D. et KROO, I. (2006). « Optimization Under Uncertainty Using Probability Collectives ». In : *Proceedings of 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, VA, AIAA-2006-7033* (cité pp. 24, 25).
- RAO, A. S. et GEORGEFF, M. P. (1995). « BDI Agents : from theory to practice ». In : *Proceedings of the First International Conference on MultiAgent Systems (ICMAS'95)*. Sous la dir. de V. LESSER. AAAI Press, p. 312–319 (cité pp. 9, 57).
- REITBAUER, A., BATTINO, A., SAINT GERMAIN, B., KARAGEORGOS, A., MEHANDJIEV, N. et VALCKENAERS, P. (2004). « The Mabe Middleware : Extending Multi-Agent Systems to Enable Open Business Collaboration ». In : *6th IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services (BASYS)*. T. 159. Springer, p. 53–60 (cité p. 42).
- RICCI, A., PIUNTI, M. et VIROLI, M. (2011). « Environment programming in multi-agent systems : an artifact-based perspective ». In : *Autonomous Agents and Multi-Agent Systems 23*, p. 158–192 (cité p. 58).
- RODRIGUEZ, S., GAUD, N. et GALLAND, S. (2014). « SARL : a general-purpose agent-oriented programming language ». In : *The 2014 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*. Warsaw, Poland : IEEE Computer Society Press (cité p. 58).
- ROSENSCHEIN, J. et ZLOTKIN, G. (1994). *Rules of encounter : designing conventions for automated negotiation among Computers*. Cambridge, Massachusetts, USA : MIT Press (cité p. 49).
- ROUGEMAILLE, S. (2008). « Ingénierie des systèmes multi-agents adaptatifs dirigée par les modèles ». Thèse de doct. Université Paul Sabatier - Toulouse III (cité pp. 57, 85).
- ROUSSEEUW, P. J. (1987). « Silhouettes : a Graphical Aid to the Interpretation and Validation of Cluster Analysis ». In : *Computational and Applied Mathematics 20*, p. 53–65. DOI : [10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7) (cité p. 36).
- SAUNIER, J., BALBO, F. et PINSON, S. (2014). « A Formal Model of Communication and Context Awareness in Multiagent Systems ». In : *Journal of Logic, Language and Information*, p. 1–29. DOI : [10.1007/s10849-014-9198-8](https://doi.org/10.1007/s10849-014-9198-8) (cité p. 58).
- SHEHORY, O. et KRAUS, S. (1998). « Methods for Task Allocation via Agent Coalition Formation ». In : *Artificial Intelligence 101.1-2*, p. 165–200. DOI : [10.1016/S0004-3702\(98\)00045-9](https://doi.org/10.1016/S0004-3702(98)00045-9). URL : [http://dx.doi.org/10.1016/S0004-3702\(98\)00045-9](http://dx.doi.org/10.1016/S0004-3702(98)00045-9) (cité p. 22).
- SHOHAM, Y. et LEYTON-BROWN, K. (2009). *Multiagent Systems : Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press (cité pp. 21–23, 25).
- SICHMAN, J. S., CONTE, R., DEMAZEAU, Y. et CASTELFRANCHI, C. (1994). « A Social Reasoning Mechanism Based on Dependence Networks ». In : *Proceedings of the 11th European Conference on Artificial Intelligence*, p. 188–192 (cité pp. 10–12).
- SMITH, R. G. (1980). « The Contract Net Protocol : High-Level Communication and Control in a Distributed Problem Solver ». In : *IEEE Transaction on Computers 29.12*, p. 1104–1113 (cité p. 12).
- SO, Y. et DURFEE, E. H. (1993). « An Organizational Self-Design Model for Organizational Change ». In : *Proceedings of AAAI93 Workshop on AI and Theories of Groups and Organizations* (cité p. 14).
- SORICI, A. (2011). « Dynamic, reactive and pro-active context information aggregation in an AmI environment ». Rapport de master. Master Universitatea Politehnica Bucuresti, EURAMUS (cité pp. 68, 69).
- STONE, P. et VELOSO, M. (1999). « Task Decomposition and Dynamic Role Assignment for Real-Time Strategic Teamwork ». In : *ATAL'98. LNCS 1555*, p. 293–308 (cité p. 15).
- TALBI, E., éd. (2006). *Parallel Combinatorial Optimization*. Wiley (cité p. 25).

Bibliographie

- TAMBE, M. (1997). « Towards Flexible Teamwork ». In : *Journal of Artificial Intelligence Research* 7, p. 83–124 (cité pp. 12, 58).
- VILLANUEVA, D. (2013). « Reliability Based Design Including Future Tests and Multi-Agent Approaches ». Thèse de doct. University of Florida et Ecole des Mines de Saint-Etienne (cité pp. 40, 85).
- VILLARREAL, S. (2010). « Distributed constraint-based Optimisation and Social Choice ». Rapport de master. Université Jean Monnet - Ecole des Mines de Saint-Etienne (cité pp. 50, 57).
- WEISS, G., éd. (2013). *Multiagent Systems*. 2nd edition. MIT Press (cité pp. 55, 56).
- WEISS, G., BERGENTI, F., GLEIZES, M.-P. et ZAMBONELLI, F., éd. (2004). *Methodologies and Software Engineering for Agent Systems*. T. 11. Multiagent Systems, Artificial Societies, And Simulated Organizations. Kluwer Publishing. DOI : [10.1007/1-4020-8058-1_11](https://doi.org/10.1007/1-4020-8058-1_11) (cité p. 59).
- WEYNS, D., OMICINI, A. et ODELL, J. (2007). « Environment as a First-class Abstraction in Multi-Agent Systems ». In : *Autonomous Agents and Multi-Agent Systems* 14.1, p. 5–30 (cité p. 58).
- WOOLDRIDGE, M. et JENNINGS, N. R. (1995). « Intelligent agents : theory and practice ». In : *The Knowledge Engineering Review* 10, p. 115–152. DOI : [10.1017/S0269888900008122](https://doi.org/10.1017/S0269888900008122). URL : http://journals.cambridge.org/article_S0269888900008122 (cité p. 55).
- WOOLDRIDGE, M. (2002). *An Introduction to MultiAgent Systems*. John Wiley et Sons (cité p. 9).
- YAICH, R. (2013). « Adaptation et conformité sociale dans la gestion de la confiance - Une approche multi-agent ». Thèse de doct. Ecole des Mines de Saint-Etienne (cité pp. 58–60, 62, 65, 71, 75–80, 84).
- YOKOO, M. (2001). *Distributed Constraint Satisfaction : Foundations of Cooperation in Multi-Agent Systems*. Springer (cité pp. 21–23, 47).
- ZHANG, W., WANG, G., XING, Z. et WITTENBURG, L. (2005). « Distributed Stochastic Search and Distributed Breakout : Properties, Comparison and Applications to Constraint Optimization Problems in Sensor Networks ». In : *Artificial Intelligence* 161.1-2, p. 55–87. DOI : [10.1016/j.artint.2004.10.004](https://doi.org/10.1016/j.artint.2004.10.004). URL : <http://dx.doi.org/10.1016/j.artint.2004.10.004> (cité pp. 23, 27).

Sixième partie

Annexes

Activités d'enseignement

J'enseigne l'informatique depuis 2001 (à l'exception de l'année 2006-2007, étant ingénieur de recherche contractuel). J'ai donné des TP, des TD et des cours à toutes sortes de publics estudiantins du L1 au M2 dans des parcours d'informatique pure, pour des biologistes, des mathématiciens, et aujourd'hui principalement de futurs ingénieurs civils des Mines. J'ai également été (et le suis encore) responsable pédagogique et j'ai monté de nombreux cours (surtout depuis 2007), notamment sur Java, UML, Python, Django, JEE, Android, l'intelligence artificielle et les systèmes multi-agents (mon domaine de recherche). Quelle que soit la formation, les enseignements ont été des enseignements en informatique (programmation et modélisation essentiellement) et sur mon domaine de recherche.

J'ai également donné un tutoriel à l'École d'été EASSS 2010, et lors d'un déplacement ERASMUS à Polytechnica Bucharest en 2011.

A.1 Responsabilités pédagogiques

Depuis 2014 Intelligence artificielle

Ce nouveau cours de 80h est destiné aux élèves 2A de l'EMSE. Il est basé sur le livre de Russell & Norvig (*Artificial Intelligence, a Modern Approach*), et explore trois facettes de l'IA : la résolution de problèmes, le raisonnement et la représentation, et l'apprentissage. À l'issue de ce cours, les élèves doivent développer un agent disposant de ces trois capacités instanciées pour un cas d'étude particulier. Certaines séances sont effectuées en classe inversée.

<http://www.emse.fr/~picard/cours/ai/>

Depuis 2014 Introduction à logique formelle

Ce cours d'introduction pour les élèves 1A de l'EMSE présente les bases de la logique formelle (proposition et prédicats), en suivant une approche apprentissage par problème. L'objectif de ce cours est d'initier les élèves-ingénieurs à la modélisation logique et au raisonnement sur des systèmes logiques, en complément de leur capacité de modélisation mathématique de réalités physiques.

<http://www.emse.fr/~picard/cours/logic/>

Depuis 2014	Informatique ambiante Ce cours, mutualisé avec le Master Web Intelligence, présente les concepts en relation avec la notion d'informatique ambiante (e.g. machine-to-machine, web of thing, pervasive computing) et des technologies et plateformes mettant en œuvre ces approches : Arduino, Android, XBee. http://www.emse.fr/~picard/cours/ac/
2008-2014	Langages et concepts de programmation orientée objet L'objectif de ce cours est de présenter les bases de la programmation orientée objet à la promotion entière d'ingénieurs civils des Mines (environ 130 étudiants). Par souci pédagogique, ce cours consiste principalement en une série de TD et de TP en Java. Je suis en charge des cours magistraux, de la production du livret de cours, des sujets de TD, de TP, de projet et de la correction de tous les rendus (TP, projet, examen écrit). http://www.emse.fr/~picard/cours/1A/java/
2008-2014	Gestion de projets informatiques Ce cours présente les bonnes pratiques de la gestion de projets informatiques (planification, cahier des charges, processus de développement, versioning, etc.). Le module se conclut par le développement d'un projet sur tout le semestre avec des technologies web utilisées dans le monde des entreprises. http://www.emse.fr/~picard/cours/2A/svn-trac/ http://www.emse.fr/~picard/cours/2A/gp/
2010-2014	Développement de systèmes informatiques Afin d'initier les futurs ingénieurs au développement des systèmes d'information, et leur présenter une vision intégrée des applications d'aujourd'hui et de leur ingénierie, ce cours présente les outils de base pour le développement d'un projet en groupe. Initialement en JEE, j'ai reconstruit le cours autour de Django et Android, afin de proposer un développement web 2.0 rapide et l'usage des terminaux mobiles. http://www.emse.fr/~picard/cours/2A/devsi/
2011-2012	Introduction à l'intelligence artificielle Ce cours de 15h propose une introduction générale aux problématiques de l'IA, puis une mise en pratique avec Prolog et la programmation de réseaux de neurones. http://www.emse.fr/~picard/cours/1A/IA/

A.2 Interventions pédagogiques

Le statut de Maître Assistant des Ecoles des Mines n'impose pas de charge d'enseignement de 192 HETD, mais nous nous répartissons la charge entre collègues. Cependant, tous les ans nous avons un récapitulatif des cours auxquels nous avons participé qui comptabilise également les heures de préparation, de correction et d'encadrement de projets. Pour plus de clareté, **les heures présentées ici sont uniquement les heures en présentiel, puis rapportées en HETD** (excluant donc les heures de préparation et de correction). Depuis 2007, je donne *en moyenne 184 HETD par an*.

A.2.1 Synthèse des interventions par statuts et matières

Statut	Formation	Matière	HTP	HTD	HCM	HETD
Moniteur	DEUG SVT – S2	Informatique – Pascal	24	18	0	34,00
	DEUG MIAS – S2	Informatique – CAML	84	32	0	88,00
	IUP TMM – 2e année	Intelligence Artificielle (TP)	24	0	0	16,00
	Licence L1-S1	Introduction Informatique	12	0	0	8,00
	Licence L1-S2	Programmation - CAML	21	0	0	14,00
	Licence L3-S6	Parallélisme (UNIX)	36	10	0	34,00
ATER	Master Informatique M1	Intelligence Artificielle (TP)	96	0	0	64,00
	Master Informatique M1	Conception Orientée Objet (TP)	48	0	0	32,00
	Master Informatique M1	Java et Parallélisme (TD)	0	48	0	48,00
	Licence L2	Informatique - CAML (TP)	20	0	0	13,33
	IUP ISI (M1)	Robotique collective (CM)	0	0	3	4,50
	Licence L1	Informatique - CAML (TP)	48	0	0	32,00
MA	Master Web Intelligence (M2)	Systèmes Multi-Agents (CM)	4	24	51	149,83
	Mastère Génie Logiciel (M1)	Conception Orientée Objet	18	24	138	243,00
	Pôle Informatique (L3)	Projet Informatique en C	49	0	0	32,67
	Pôle Informatique (L3)	Programmation objet	168	54	15	229,00
	Pôle Informatique (L3)	Programmation et algorithmique	40	0	0	26,67
	Pôle Informatique (L3)	Bases de données	40	12	0	60,92
	Pôle XXI (L3)	Intelligence Artificielle	0	0	15	22,50
	Axe ISI (M1)	Développement SI	18	10	26	61,00
	Axe ISI (M1)	Analyse et Conception Objet	30	147	141	423,50
TOTAL			780	379	389	1 636,920000000

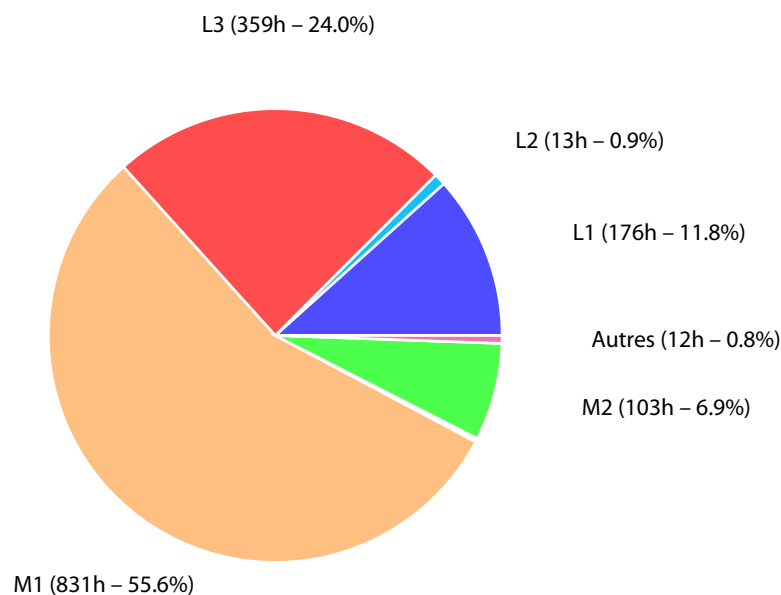


FIGURE A.1 – Répartition des enseignements par niveau de public

A.2.2 Synthèse des interventions par années

Année	2001	2002	2003	2004	2005	2007	2008	2009	2010	2011	2012	2013
HETD	63,33	66,67	64,00	89,83	104,00	161,25	237,75	226,33	157,50	137,67	202,67	168,92

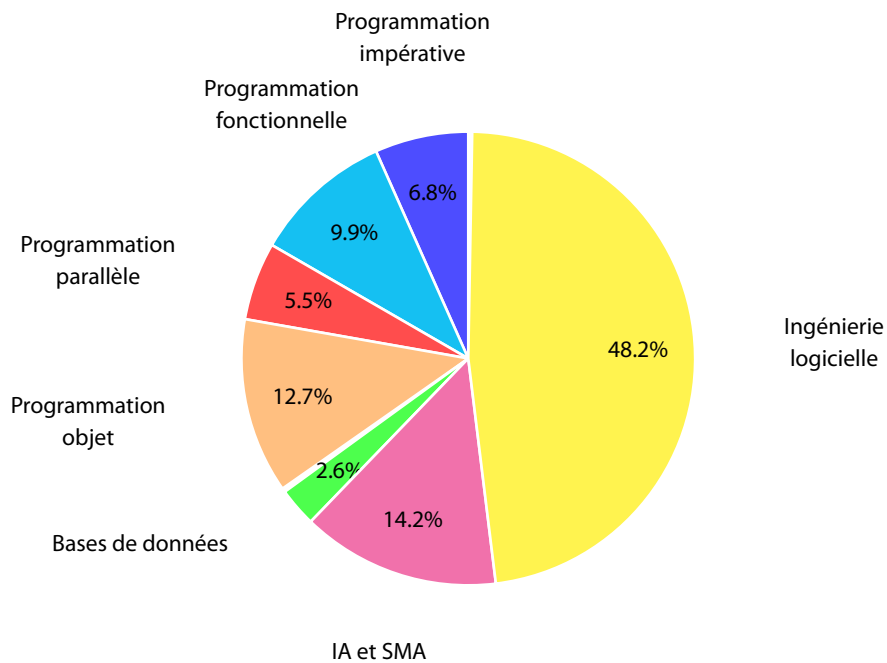


FIGURE A.2 – Répartition des enseignements par matière

A.3 Responsabilités administratives

Je suis responsable du module Coopération de l'option Informatique (module de 120 heures d'enseignements dans le cursus 3ème année d'Ingénieur Civil des Mines) depuis 2010. J'ai également été responsable des Unités pédagogiques suivantes :

- Langages et concepts de programmation orientée-objet (1A-L3, env. 130 étudiants) [2008-2014]
- Développement de Systèmes Informatiques (2A-M1, env. 25 étudiants) [2010-2014]
- Gestion de projets (2A-M1, env. 25 étudiants) [2008-2014]
- Intelligence artificielle (2A-M1, env. 20 étudiants) [2014+]

Je suis également élu au *Comité de l'enseignement* de l'EMSE depuis novembre 2013.

Activités de recherche

Ma vocation pour la recherche est née lors de mes premières années universitaires. Alors à Pau, je me suis inscrit en second cycle à Toulouse afin de pouvoir suivre un parcours recherche (DEA). J'ai été intégré dans des laboratoires depuis ma maîtrise (TER, travaux de études et de recherche), en 2000, lors de laquelle j'ai rejoint l'équipe dans laquelle j'allais plus tard effectuer ma thèse (équipe SMAC dirigée par Marie-Pierre Gleizes).

J'ai effectué tout d'abord mes recherches dans l'Institut de Recherche en Informatique de Toulouse (de 2000 à 2007, sous statut allocataire recherche, puis attaché temporaire de recherche, puis ingénieur CNRS en post-doc) puis à l'École des Mines (à partir de 2007, sous statut de maître assistant des Ecoles des Mines) au centre G2I puis l'Institut Henri Fayol, dans l'équipe ISCOD, dirigée par Olivier Boissier.

B.1 Parcours professionnel

depuis 2007	Maître-Assistant des Écoles des Mines à l'Institut Henri Fayol de l'École Nationale Supérieure des Mines de Saint-Etienne (ENSM-SE), France
2006-2007	Post-doc, responsable des projets européens à l'IRIT (Institut de Recherche en Informatique de Toulouse), France
2004-2006	Attaché temporaire d'enseignement et recherche (demi-ATER) à l'Université Paul Sabatier Toulouse III, France
2001-2004	Allocataire de Recherche à l'Université Paul Sabatier Toulouse III, France
2001-2004	Moniteur CIES à l'Université Paul Sabatier Toulouse III, France

B.2 Responsabilités scientifiques et de valorisation

B.2.1 Responsabilités

- Président de la session *Multi-Agent Systems* à l'*Indian International Conference on Artificial Intelligence (IICAI)* : 2007
- Membre du comité de pilotage de la série de workshops *Engineering Societies in the Agents World (ESAW)* : 2008, 2009
- Président du comité d'organisation d'*Engineering Societies in the Agents World (ESAW)* : 2008, 2009

- Président du comité de programme de *Engineering Societies in the Agents World (ESAW)* : 2008, 2009
- Président de la session sur l'*optimisation multi-disciplinaire* à la conférence de la *société française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF)* : 2011
- Président de la session *Demonstrations* à *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT)* : 2011
- Président du comité d'organisation d'IEEE International Conference on Self-Adaptive and Self-Organizing Systems (**SASO**) : 2012

B.2.2 Participations à des comités de programme

- Engineering Societies in the Agents World (**ESAW**) : 2004, 2008, 2009
- European Workshop on Multi-Agent Systems (**EUMAS**) : 2004, 2005
- Indian International Conference on Artificial Intelligence (**IICAI**) : 2007, 2009, 2011, 2013
- Rencontres des Jeunes Chercheurs en Intelligence Artificielle (**RJCIA**) : 2007
- Self-Adaptation for Robustness and Cooperation in Holonic MultiAgent Systems and in MultiAgent Systems (**SARC**) : 2008, 2009
- Workshop on Bio-Inspired and Self-* Algorithms for Distributed Systems (**BADS**) : 2009, 2010, 2011
- International Conference on Intelligent Agent & Multi-Agent Systems (**IAMA**) : 2009
- IEEE International Conferences on Self-Adaptive and Self-Organizing Systems (**SASO**) : 2009, 2010, 2011, 2012
- International Joint Conference on Autonomous Agents and Multi-Agent Systems (**AAMAS**) : 2010
- International Workshop on Agent-Oriented Software Engineering (**AOSE**) : 2010, 2011, 2012
- International Workshop on Web Intelligence and Virtual Enterprises (**WIVE**) : 2010
- International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (**IDETC**) : 2011
- International Conference on Practical Applications of Agents and Multi- Agent Systems (**PAAMS**) : 2011
- International Workshop on Multi-Agent Systems and Simulation (**MAS&S**) : 2011, 2012, 2013, 2014
- Journées Francophones sur les Systèmes Multi-Agents (**JFSMA**) : 2012, 2013
- International Joint Conference on Artificial Intelligence (**IJCAI**) : 2013
- International Conference on Robotics and Automation (**ICRA**) : 2013
- International Joint Workshop on Optimisation in Multi-Agent Systems and Distributed Constraint Reasoning (**OPTMAS-DCR**) : 2014

B.2.3 Participations en tant que relecteur

- Journal on Future Generation Computer Systems (**FGCS**)
- International Journal on Agent Oriented Software Engineering (**IJAOSE**)
- ACM Transaction on Adaptive and Autonomous Systems (**TAAS**)
- Revue d'Intelligence Artificielle (**RIA**)
- Journal on Simulation Modelling Practice and Theory (**SIMPAT**)
- Journal on Web Intelligence and Agent Systems (**WIAS**)
- International Journal of Production Research (**IJPR**)
- International Workshop on Coordination, Organisations, Institutions and Norms (COIN@AA-MAS) : 2008
- International Joint Conference on Autonomous Agents and Multi-Agent Systems (**AAMAS**) : 2005, 2008, 2013

- Agent-Oriented Software Engineering Methodologies and Systems (**AOMS@SAC**) : 2009
- Agent-Oriented Programming, Systems, Languages, and Applications (**APSLA@SAC**) : 2008
- Brazilian Symposium on Artificial Intelligence (**SBIA**) : 2008
- Congrès Reconnaissance des Formes et Intelligence Artificielle (**RFIA**) : 2008
- International Workshop on Agent-Oriented Software Engineering (**AOSE**) : 2009
- International Conference on Intelligent Systems and Agents (**ISA**) : 2009
- International Conference on Robotics and Automation (**ICRA**) : 2010
- IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (**WI-IAT**) : 2011
- AAAI Conference on Artificial Intelligence (**AAAI**) : 2012

B.2.4 Participations à des comités d'organisation

- Engineering Societies in the Agents World (**ESAW**) : 2004
- Journées Francophones sur les Systèmes Multi-Agent (**JFSMA**) : 2007
- European Agent Systems Summer School (**EASSS**) : 2010
- Multi-Agent Logics, Languages, and Organisations Federated Workshops (**MALLOW**) : 2010
- Ecole d'été Web Intelligence (**WI Summer School**) : 2011
- IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (**WI-IAT**) : 2011

B.3 Responsabilités administratives

En complément de mes activités opérationnelles d'enseignement et de recherche, j'ai également fait le choix de remplir des fonctions d'intérêt collectif et de m'impliquer dans les activités des équipes.

- Coordination des montages de projets européens du laboratoire IRIT (2006-2007).
- Responsabilité scientifique et technique du partenaire ARMINES-FAYOL dans le projet ID4CS (2009-2013).
- Responsabilité des montages de projets ANR et européens dans l'équipe ISCOD (actuellement).

Je suis également élu au *Comité de la recherche* depuis juin 2014.

Encadrements de thèses et de masters

Pour les co-encadrements, le pourcentage d'encadrement de chaque encadrant est indiquée entre crochets.

C.1 Thèses soutenues

- 2009-2014** | C. Persson (PhD ANRT CIFRE Orange Labs/EMSE) : « *A Decentralized and Distributed Adaptive Governance for Machine-to-Machine Systems : A Multi-Agent Oriented Programming Approach* »
Date de soutenance : 31 octobre 2014
Financement : ANRT CIFRE Orange Labs
Encadrement : Prof. O. Boissier [25%], G. Picard [45%], F. Ramparany [30%]
Publications : [PERSSON et al., 2014 ; PERSSON et al., 2012b ; BILAL et al., 2012 ; PERSSON et al., 2012a ; PERSSON et al., 2011a ; PERSSON et al., 2011b ; PERSSON, PICARD et RAMPARANY, 2011]
- 2009-2013** | R. Yaich : « *Adaptiveness and Social-Compliance in Trust Management - A Multi-Agent Based approach* »
Date de soutenance : 29 octobre 2013
Financement : Allocation Ecole des Mines
Encadrement : Prof. O. Boissier [25%], P. Jaillon [30%], G. Picard [45%]
Publications : [YAICH et al., 2013 ; YAICH et al., 2012b ; YAICH et al., 2012a ; YAICH et al., 2011b ; YAICH, JAILLON, BOISSIER et al., 2011 ; YAICH et al., 2011a ; YAICH, JAILLON, PICARD et al., 2010]
Poste actuel du doctorant : ATER
- 2010-2013** | D. Villanueva (co-tutelle EMSE/UF) : « *Reliability Based Design Including Future Tests and Multi-Agent Approaches* »
Date de soutenance : 13 mai 2013
Financement : Projet ANR ID4CS, Université de Floride
Encadrement : Prof. R. Le Riche [33%], Prof. R. Haftka (UF) [33%], G. Picard [33%]

Publications : [VILLANUEVA, HAFTKA et al., 2013 ; PICARD, VILLANUEVA et al., 2013 ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2013b ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2013a ; VILLANUEVA, PICARD et al., 2012 ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2012a ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2012b ; VILLANUEVA, LE RICHE, PICARD et HAFTKA, 2011 ; VILLANUEVA, LE RICHE, PICARD, HAFTKA et SANKAR, 2011]

Poste actuel du doctorant : post-doc aux US

C.2 Thèses en cours

- 2013-2016** | S. Gillani (TSE/EMSE) : « *Négociation distribuée et contextuelle dans un environnement consommateurs-producteurs énergétiques* »
Financement : Allocation EDSIS
Encadrement : Prof. F. Laforest [50%], G. Picard [50%]
Publications : [GILLANI, LAFOREST et al., 2014 ; GILLANI et al., 2014a ; GILLANI et al., 2014b]
- 2012-2016** | A. Sorici (co-tutelle EMSE/UPB) : « *Multi-Agent Context Management for Support of Ambient Computing Applications* »
Financement : Bourse MAEE
Encadrement : Prof. A. Florea (UPB) [25%], Prof. O. Boissier [25%], G. Picard [50%]
Publications : [SORICI, PICARD et BOISSIER, 2014 ; SORICI, BOISSIER, PICARD et ZIMMERMANN, 2013 ; SORICI, PICARD, BOISSIER, SANTI et al., 2012 ; SORICI, BOISSIER, PICARD et SANTI, 2011]

C.3 Masters recherche

- 2012** | L. Cerqueira Martins (Master EMSE/UJM) : « *Decentralized stable matching in mixed communities* »
- 2011** | A. Sorici (Master Universitatea Politehnica Bucuresti, ERAMUS) : « *Dynamic, reactive and pro-active context information aggregation in an AmI environment* »
Publications : [SORICI, BOISSIER, PICARD et SANTI, 2011]
- 2011** | M. Bilal (Master UTT, Orange Labs) : « *Multi-agent governance model for M2M networks : Application to a smart parking management system* »
Publications : [BILAL et al., 2012]
- 2010** | S. Villarreal (Master EMSE/UJM) : « *Distributed constraint-based Optimisation and Social Choice* »
- 2008** | G. Clair (Master EMSE/UJM) : « *Self-organisation for manufacturing control based on multi-agent systems* »
Publications : [CLAIR et al., 2008a ; CLAIR et al., 2008b]
- 2008** | E. Kaddoum (Master IRIT/UPS) : « *Self-regulation for manufacturing control using self-organising MAS* »
Publications : [GEORGÉ, M.-P. GLEIZES et al., 2010 ; KADDOUM, M.-P. GLEIZES, GEORGÉ et PICARD, 2009 ; KADDOUM, M.-P. GLEIZES, GEORGÉ, GLIZE et al., 2009 ; CLAIR et al., 2008b ; CLAIR et al., 2008a]
- 2006** | F. Cornet (Master IRIT/UPS) : « *Study of a frequency assignment problem using adaptive multi-agent systems* »

C.4 Jurys de thèses

- 2014** | C. Persson : "A Decentralized and Distributed Adaptive Governance for Machine-to-Machine Systems : A Multi-Agent Oriented Programming Approach" — Co-encadrant, soutenue le 31 octobre 2014
- 2014** | L. Pons : « Auto-ajustement de scénarios de jeux par systèmes multi-agents adaptatifs » — Examineur, soutenue le 7 juillet 2014
- 2013** | R. Yaich : « Adaptiveness and Social-Compliance in Trust Management - A Multi-Agent Based approach » — Co-encadrant, soutenue le 29 octobre 2013
- 2013** | T. Jorquera : « An Adaptive Multi-Agent System for Self-Organizing Continuous Optimization. » — Examineur, soutenue le 22 octobre 2013 à l'Université Paul Sabatier
- 2013** | D. Villanueva : « Reliability Based Design Including Future Tests and Multi-Agent Approaches » — Co-encadrant, soutenue le 13 mai 2013
- 2008** | S. Rougemaille : « Ingénierie des systèmes multi-agents adaptatifs dirigée par les modèles. » — Examineur, soutenue le 27 octobre 2008 à l'Université Paul Sabatier

D

Programmes nationaux et internationaux

Ce chapitre synthétise les différents projets nationaux et internationaux auxquels j'ai participé. Pour chacun d'entre eux, une courte description ainsi que quelques points marquants sont mentionnés. D'autres coopérations hors projet sont mentionnées en fin de chapitre.

D.1 Projets nationaux

2013-2017	<p>Projet ETHICAA [financement ANR]</p> <p>L'objectif du projet ETHICAA est de définir ce que devrait être une entité autonome pouvant gérer des conflits éthiques, aussi bien au niveau individuel qu'au niveau collectif. Les conflits éthiques que nous considérons sont susceptibles de survenir ; au sein d'un agent, entre un agent et les règles éthiques de son système d'appartenance, entre un agent et un utilisateur ou un opérateur humain, entre plusieurs agents incluant ou non des êtres humains (drones, agents virtuels).</p> <ul style="list-style-type: none">— <i>Montant de l'aide : 244 561 €</i>— <i>Participation au montage</i>— <i>Coopération avec le GREYC, l'Onera, le LIP6, Télécom Ecole de Management et Ardans</i>— <i>Modélisation et implémentation de mécanismes éthiques collectifs</i>
2009-2013	<p>Projet ID4CS [financement ANR]</p> <p>ID4CS est un projet ANR qui a pour ambition de proposer un environnement de modélisation et de simulation pour la conception de systèmes complexes comme des avions.</p> <ul style="list-style-type: none">— <i>Montant de l'aide : 221 293 €</i>— <i>Participation au montage</i>— <i>Co-encadrement d'une thèse en co-tutelle avec l'université de Floride</i>— <i>Coopération avec IRIT, Airbus, IMT, ICA, Upetec, SNECMA</i>— <i>Coordinateur scientifique pour le partenaire ARMINES-FAYOL</i>— <i>Coordinateur scientifique du lot « Modèle Agent »</i>— <i>http://www.irit.fr/id4cs</i>

2008-2012	Projet Web Intelligence [financement ISLE Cluster-RRA] L'objectif global du projet est de consolider et de structurer la communauté scientifique rhônealpine et de favoriser la synergie de coopération sur le sujet de la Web Intelligence. <ul style="list-style-type: none">— <i>Participation au lot « Future Web »</i>— <i>Participation à l'organisation et président de session démo à WI-IAT 2011</i>— http://www.web-intelligence-rhone-alpes.org/
2001-2004	Projet ADELFE [financement RNTL] Le but de l'atelier de développement ADELFE est de guider l'ingénieur lors du développement de systèmes multi-agents adaptatifs (AMAS). ADELFE est maintenant une méthode orientée multi-agent reconnue et a été publiée dans plusieurs ouvrages de référence sur l'ingénierie multi-agent. <ul style="list-style-type: none">— <i>ADELFE est une des méthodes multi-agents les plus citées</i>— <i>Doctorant associé au projet</i>— <i>Développement d'AdelfeToolkit pour aider le concepteur à suivre le processus ADELFE</i>— http://www.irit.fr/ADELFE/

D.2 Projets internationaux

2013-2015	Projet Smart Energy Aware Systems (SEAS) [financement ITEA2] L'objectif du projet SEAS est de permettre l'interopérabilité des systèmes produisant de l'énergie, des TIC et des systèmes d'automatisation des sites de consommations. IL vise également à introduire des solutions fondées sur des technologies dynamiques pour contrôler et suivre l'estimation de la consommation d'énergie. Un second but est d'explorer des modèles économiques et des solutions qui permettront aux acteurs du marché de l'énergie d'intégrer les environnements de réseaux de type micro-réseaux et les clients réactifs, en particulier des systèmes décentralisés intelligents (application à l'intelligence ambiante et la ville intelligente). <ul style="list-style-type: none">— <i>Montant de l'aide : 89 493 €</i>— <i>Participation au montage</i>— <i>Coopération avec 6 pays (Finland, France, Portugal, Romania, Spain, Turkey)</i>— <i>Ontologie pour les Smart Grids; privacité dans les Smart Grids; négociation automatique</i>— http://www.itea2.org/project/index/view?project=10156
------------------	--

D.3 Coopérations diverses

J'ai noué de multiples collaborations durables dans le cadre des projets de recherche auxquels j'ai participé (ANR ID4CS, CMIRA-RRA MAOP), d'encadrement de thèse (CIFRE Orange Labs, co-tutelle avec l'Université de Floride, et Télécom Saint-Etienne), de contacts lors de conférences (LIFL), ou à l'occasion de déplacement EURAMUS ("Politehnica" University of Bucharest). Suite à une invitation à l'Imperial College London, dans l'équipe de Jeremy Pitt, nous prévoyons également l'organisation d'un workshop international (SoRules 2014) sur les *self-organising rule-oriented systems*. Enfin, cette année, nous avons été invités, Olivier Boissier et moi-même, à effectuer des présentations au KAIST

(Korean Advanced Institute of Science and Technology) sur le thème de la robotique collective et les systèmes multi-agents. Ceci sera peut-être le point de départ d'une collaboration avec des collègues coréens (KAIST), américains (Purdue University) et australiens (Griffith University).

<i>Nationales</i>	Université de Toulouse (IRIT, ICA, IMT), Université de Lille (LIFL), Télécom Saint-Etienne (TSE), ONERA, Orange Labs, Upetec, Airbus, SNECMA
<i>Internationales</i>	University of Florida (US), Università di Bologna (IT), "Politehnica" University of Bucharest (RO), Federal University of Santa Catarina (BR), Imperial College London (UK)

