HAL
archives-ouvertes.fr

# Approches environnement-centrées pour la simulation de systèmes multi-agents

Fabien Michel

▶ **To cite this version:**

Fabien Michel. Approches environnement-centrées pour la simulation de systèmes multi-agents : Pour un déplacement de la complexité des agents vers l'environnement. Système multi-agents [cs.MA]. Montpellier II, 2015. <tel-01241307>

**HAL Id: tel-01241307**

**https://hal-lirmm.ccsd.cnrs.fr/tel-01241307**

Submitted on 10 Dec 2015

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

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

Académie de Montpellier

Université de Montpellier

Habilitation à Diriger des Recherches

Spécialité : Informatique
*École Doctorale* : *Information, Structures, Systèmes*

# Approches environnement-centrées pour la simulation de systèmes multi-agents

## Pour un déplacement de la complexité des agents vers l'environnement

par

Fabien Michel

Soutenue le 15 juin 2015 devant le jury composé de :

Marie-Pierre Gleizes, Professeur, Toulouse III, ..... Rapporteur
François Charpillet, Directeur de Recherche, LORIA, Nancy, ..... Rapporteur
Philippe Mathieu, Professeur, Lille I, ..... Rapporteur
Zahia Guessoum, Maître de conférences HDR, Université de Reims, ..... Examinateur
Salima Hassas, Professeur, Lyon I, ..... Examinateur
Jacques Ferber, Professeur, Université de Montpellier, ..... Examinateur
Danny Weyns, Professeur, Linnaeus University, Suède, ..... Examinateur

*À Isabelle, Vincent et Pauline.*

*It would be an unsound fancy and self-contradictory to expect that things which have never yet been done can be done except by means which have never yet been tried.*

Francis Bacon, *Novum organum*

# Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude aux membres du jury.

Mes plus vifs remerciements à Marie-Pierre Gleizes, François Charpillet et Philippe Mathieu qui m'ont fait l'honneur d'être rapporteurs. Leur disponibilité, malgré des contraintes temporelles fortes, m'a permis de finaliser sereinement ce document et je leur en suis infiniment reconnaissant. J'espère sincèrement que nos interactions futures seront nombreuses.

Soutenir une HDR, c'est bien sûr l'occasion de se tourner vers son passé et ce fut pour moi un moment singulier, d'une rare intensité, que de présenter mes travaux devant un jury composé des personnes qui ont profondément marqué mon parcours scientifique. Je dois à Salima Hassas de m'avoir initié aux SMA alors que j'étais en maîtrise à Lyon I. Son enthousiasme et sa vision des systèmes complexes ont été pour moi déterminants. Je tiens à remercier très chaleureusement Jacques Ferber, hier directeur de thèse, aujourd'hui collègue et ami. Je lui suis reconnaissant de m'avoir communiqué sa passion de la recherche et soutenu de manière indéfectible tout au long de ces années. J'ai débuté ma carrière de maître de conférences au côté de Zahia Guessoum et je la remercie de m'avoir fait confiance puis aiguillé au cours de mes premières années d'activité à Reims ; cela a été capital pour moi. Enfin, le contenu de cette habilitation serait sans doute bien différent si je n'avais pas eu le plaisir de collaborer avec Danny Weyns durant plusieurs années, autant de souvenirs, sur bien des plans.

Un document d'habilitation est aussi l'expression d'un travail de recherche collectif. Je remercie en particulier tous les (ex) doctorants impliqués dans celui-ci : Jason Mahdjoub, Samine Mazouzi, Nicolas Carlési, Ghulam Mahdi et Emmanuel Hermellin.

Je tiens également à remercier les collègues avec qui j'ai travaillé ou échangé ces dix dernières années, que ce soit à Reims au CReSTIC, à Montpellier au Lirmm, mais aussi dans le cadre des projets ou des conférences auxquels j'ai participé. Je salue en particulier Olivier Simonin, Grégory Beurier, Denis Payet, Van Parunak, Giuseppe Vizzari, Alessandro Ricci, Pierre Bommel, Paolo Petta, Abderrafiâa Koukam, Gildas Morvan, Sébastien Picault, Noura Faci, Julien Saunier, Fabien Badeig, Frédéric Gruau, Nicolas Marilleau, Pierre-Alain Laur, Olivier Gutknecht, Önder Gürcan, Raphaël Duboz, Jean-Pierre Müller, Stéphane Galland, Frédéric Armetta, Denis Phan, Sébastian Rodriguez, Paolo Petta, Vincent Chevrier, Alexander Helleboogh, Mahdi Zargayouna, Yoann Kubera, Jean-Christophe Soulié, Bruno Bonté, John Tranier, Nicolas Sébastien et Jean-Baptiste Soyez.

La soutenance d'une HDR étant aujourd'hui le fruit d'un long cheminement, je suis reconnaissant à Zohra Bellahsène, Christophe Dony et Olivier Strauss d'avoir été mes rapporteurs en vue de l'examen de mon dossier par l'école doctorale I2S. Leurs analyses et leurs conseils m'ont été précieux. Merci également à William Puech, alors responsable de la spécialité informatique pour l'école doctorale, pour toutes les réponses qu'il a apportées à mes nombreuses questions lors de ce processus.

Je tiens aussi à saluer ceux des personnels administratifs qui ont concrètement contribué à ce que tout se déroule au mieux : Louise Casteill, Caroline Ycre, Laurie Lavernhe et Nicolas Serrurier.

Mes pensées vont enfin aux personnes, collègues, proches, parfois les deux, qui m'ont aidé ou soutenu, de près ou de loin, dans la préparation de cet événement : Stefano Cerri, Vincent Emonet, Yannick Francillette, Abdelkader Gouaïch, Emmanuel Hermellin, Vincent Hinsinger, Clément Jonquet, Souhila Kaci, Philippe Lemoisson, Tibériu Stratulat et Chouki Tibermacine. Un merci tout particulier à Vincent H. pour ses relectures précises et ses remarques avisées. Un clin d'œil à Abdelkader, avec qui je partage bien plus qu'un bureau depuis 7 ans ; une communauté de pensée, depuis longtemps. Je le remercie ici pour sa disponibilité sans faille et ses précieux conseils.

Merci enfin à Isabelle, Danièle et mes parents pour leur aide et leur perpétuel soutien.

$\infty$

# Table des matières

**Remerciements** v

**Avant-propos** ix

**1 Introduction** 1
- 1.1 Domaine de recherche . . . 2
- 1.2 Problématiques et positionnement scientifiques . . . 6
- 1.3 Aspects méthodologiques . . . 11
- 1.4 Plan du document . . . 12

**I Travaux et coencadrements doctoraux passés** 13

**2 Le modèle IRM4S** 15
- 2.1 IRM4S : *Influence/Reaction Model for Simulation* . . . 15
- 2.2 Caractéristiques du modèle IRM4S . . . 19
- 2.3 Apports de IRM4S pour la simulation multi-agents . . . 21
- 2.4 Modéliser la dynamique de l'environnement . . . 23
- 2.5 Bilan . . . 25

**3 Sur le concept d'environnement dans les SMA** 27
- 3.1 Environments 4 Multi-Agent Systems . . . 27
- 3.2 Modèle 3-tiers de l'environnement dans les SMA . . . 29
- 3.3 Le modèle AGRE . . . 31
- 3.4 Intérêt d'une vision unifiée de l'environnement . . . 33
- 3.5 Leçon majeure : déplacer la complexité . . . 35

**4 Segmentation d'image par SMA** 37
- 4.1 Difficultés d'adaptation des méthodes TNI classiques . . . 37
- 4.2 L'image : un environnement pour SMA . . . 39
- 4.3 Détection et reconstruction de contours par SMA . . . 39
- 4.4 Segmentation d'images de profondeur par stigmergie . . . 43
- 4.5 Perspective E4MAS . . . 47

**5 Gestion par niveaux de détail des IA dans un jeu vidéo** **49**
5.1 Vers plus d'agents autonomes dans les jeux vidéo . . . 49
5.2 LOD pour la gestion adaptative des IA d'agents autonomes . . . 51
5.3 Utilisation d'un environnement social . . . 51
5.4 Délégation des moyens de perceptions et d'actions . . . 52

**6 Robotique mobile collective en milieu sous-marin** **55**
6.1 Vers des flottilles de robots sous-marins autonomes . . . 55
6.2 Approche organisationnelle réactive multi-agent . . . 56
6.3 Simulation avec TurtleKit 3 . . . 59
6.4 Combiner environnements physique et social . . . 60

**II Recherches en cours et perspectives** **61**

**7 Développements logiciels : MaDKit et TurtleKit** **63**
7.1 MaDKit : Multiagent Development Kit . . . 63
7.2 TurtleKit : simulation de systèmes complexes . . . 66
7.3 Morphogenèse et embryogénie artificielle avec TurtleKit . . . 69
7.4 Programmation situationnelle : *l'IA pour les nuls* . . . 70

**8 Simulation multi-agents et calcul haute performance sur GPU** **75**
8.1 GPGPU et simulation multi-agents : intérêts et limites . . . 75
8.2 Délégation GPU des perceptions agents . . . 78
8.3 GPGPU et SMA : un état des lieux . . . 82
8.4 Délégation GPU avec les *boids* de Reynolds . . . 83

**9 Perspectives et conclusion** **87**
9.1 Positionnement . . . 87
9.2 Vers une méthodologie pour la simulation sur GPU . . . 89
9.3 Vers plus de délégation dans les comportements agents . . . 90
9.4 Vers de nouveaux modèles et dynamiques multi-agents . . . 92
9.5 En conclusion . . . 96

**Bibliographie** **99**

**Curriculum Vitæ** **115**
Identité et parcours professionnel . . . 115
Publications . . . 116
Activités scientifiques . . . 123
Responsabilités scientifiques et pédagogiques . . . 126

**Résumé** **128**

# Avant-propos

Ce mémoire synthétise les recherches que nous avons menées depuis un peu plus d'une dizaine d'années. Celles-ci se situent majoritairement dans le domaine de la modélisation et de la simulation multi-agents. Thématique qui était déjà au cœur de la thèse que nous avons faite au Lirmm sous la direction de Jacques Ferber (soutenue en décembre 2004).

Après la thèse, nous avons tout d'abord passé quelques mois en tant qu'ingénieur de recherche, toujours au Lirmm, avant de rejoindre le laboratoire CReSTIC de l'université de Reims Champagne-Ardenne en septembre 2005, en tant que maître de conférences dans le groupe SIC (Signal, Image et Connaissance). Nous avons alors immédiatement travaillé avec Zahia Guessoum (dépôts de projets, coencadrements, etc.). En particulier, nous avons coencadré ensemble le master 2 de Jason Mahdjoub (2005-2006) qui a pu, grâce à un financement ministériel, poursuivre en thèse sur le thème de l'utilisation des SMA pour le traitement numérique de l'image. Toujours sur ce thème, nous avons aussi participé à l'encadrement de la thèse de Smaine Mazouzi lors de son séjour doctoral (2 ans) au sein du CReSTIC (il était alors maître assistant à l'université de Skikda en Algérie). par ailleurs, nous avons continué à cultiver les contacts que nous avions avec des chercheurs étrangers (notamment Van Parunak et Danny Weyns, puis Giuseppe Vizzari, avec qui nous avons coorganisé des événements scientifiques).

Suite à une mutation, nous sommes revenu au Lirmm en septembre 2008 et avons rejoint l'équipe SMILE de Jacques Ferber. Dès notre arrivée, nous avons intégré le coencadrement (avec Stefano Cerri et Abdelkader Gouaïch) de la thèse de Ghulam Mahdi qui débutait (IA et jeux vidéo). Dans le même temps nous avons participé à plusieurs collaborations industrielles sur le même thème. En 2010, nous avons intégré le projet ANR C-FLAM. Ce qui nous a donné l'occasion de collaborer avec l'équipe de robotique EXPLORE à travers le coencadrement de la thèse de Nicolas Carlési (flottilles de robots sous-marins sous la direction de Jacques Ferber et Bruno Jouvencel).

La première partie de ce mémoire discute des travaux et des coencadrements que nous avons effectués dans ces différents contextes.

Parallèlement, après un an d'intégration, fin 2009 nous avons reconsidéré notre projet de recherche et nous nous sommes orienté vers le calcul haute performance sur architectures massivement parallèles pour la simulation multi-agents. Nous avons proposé une première fois

un sujet de thèse sur ce thème en 2010 mais ce n'est qu'après nos premiers résultats, en 2012, que ce sujet a pu devenir prioritaire. Ce qui nous a permis de recruter Emmanuel Hermellin en doctorat en tant que moniteur (octobre 2013, codirigé avec Jacques Ferber). Par ailleurs, nous avons retravaillé sur les outils logiciels que nous utilisons (MaDKit et TurtleKit) afin de les faire évoluer pour qu'ils puissent servir efficacement nos objectifs. La deuxième partie de ce mémoire est entièrement consacrée à ces travaux et à leurs perspectives.

La mobilité qui caractérise notre parcours fut pour nous une chance immense.

Tout d'abord, cela nous a amené à utiliser la simulation multi-agents dans des domaines applicatifs divers, avec chaque fois des objectifs différents. Avec le recul, c'est un aspect important et, en cela, nous devons beaucoup aux travaux doctoraux que nous avons coencadrés. Aborder des thématiques aussi diverses que le traitement numérique de l'image, la robotique mobile collective, ou encore les jeux vidéo, nous a donné la possibilité d'expérimenter le paradigme multi-agents sous de multiples angles. Ce qui nous a permis d'en voir plus clairement les avantages et les limites. D'une manière plus générale, ces expériences ont forgé la vision que nous avons aujourd'hui des SMA. C'est pourquoi le lecteur trouvera dans la première partie de ce mémoire un chapitre pour chacun de ces thèmes.

Plus important sans doute est l'aspect humain : la diversité des contextes dans lesquels nous avons travaillé fut une expérience très enrichissante, notamment du point de vue de notre capacité d'adaptation et d'ouverture. Il existe de multiples façons de faire de la recherche et l'hétérogénéité des approches, des personnalités et des *interactions* constitue indéniablement une force motrice puissante. C'est en partie grâce à cela que nous pouvons aujourd'hui faire la synthèse qui est proposée dans ce mémoire.

# Chapitre 1

# Introduction

> "*Connaître la hauteur du grand mât, cela ne suffit pas pour calculer l'âge du capitaine. Quand vous aurez mesuré tous les morceaux de bois du navire, vous aurez beaucoup d'équations, mais vous ne connaîtrez pas mieux cet âge. Toutes vos mesures ayant porté sur vos morceaux de bois ne peuvent rien vous révéler que ce qui concerne ces morceaux de bois. De même vos expériences, quelque nombreuses qu'elles soient, n'ayant porté que sur les rapports des corps entre eux, ne nous révéleront rien sur les rapports mutuels des diverses parties de l'espace.*"
>
> Poincaré, 1902, *La science et l'hypothèse*, p.101

Les enjeux liés à la simulation informatique sont multiples. Tout d'abord, depuis ses tout débuts, elle constitue le troisième pilier des sciences modernes, au côté de la théorie et de l'expérience. En particulier, grâce à elle nous pouvons *animer* des modèles non linéaires dont la dynamique est trop complexe pour être résolue analytiquement [1]. C'est-à-dire des modèles dont on ne peut connaître l'état futur sans calculer de manière itérative toutes les étapes qui y conduisent. Il faut construire leur évolution, recréer *leur flèche du temps*. Ces modèles sont au cœur des problématiques étudiées dans le cadre des systèmes complexes (théorie du chaos, écologie, économie, biologie, etc.), ce qui fait de la simulation un outil transdisciplinaire.

La simulation est aussi incontournable pour la conception de nombreux systèmes opérationnels. Pour tous les systèmes ancrés dans le réel (e.g. robotique), elle permet de réaliser des expériences contrôlées afin de s'affranchir des coûts et du bruit dus aux expérimentations en conditions réelles. De plus, la complexité qu'on trouve dans le monde réel est une réalité pour les systèmes informatiques actuels. Autrement dit, la simulation est tout aussi nécessaire pour les concevoir, car ils sont eux-mêmes intrinsèquement complexes. Enfin, la simulation constitue le cœur même du fonctionnement de certains systèmes informatiques, ce qui est le cas pour la majorité des jeux vidéo par exemple (simulations interactives).

1. Ce qui a notamment rendu possibles les développements liés à la théorie du chaos, avec une leçon majeure pour les sciences : le déterminisme n'exclut pas le chaos. Prédire l'évolution à long terme d'un système non linéaire est tout simplement impossible dans la majorité des cas [Gleick, 1989 ; Motter et Campbell, 2013].

Les travaux de recherche que nous avons menés s'inscrivent majoritairement dans le domaine de la **modélisation et simulation de systèmes multi-agents** (SMA). À la différence des modèles mathématiques, la simulation multi-agents met en œuvre des modèles où les individus, leur environnement et leurs interactions sont directement représentés. Dans ces modèles, chaque individu –*agent autonome*– possède son propre comportement et produit ses actions en fonction d'une perception locale de son environnement.

Depuis nos travaux de thèse [Michel, 2004], nous nous intéressons à des problématiques liées à la **conception** et à l'**implémentation de simulations multi-agents** (modèles et outils). Nous nous plaçons dans un contexte générique et nos recherches ne sont pas liées à un domaine applicatif spécifique. Les questions que nous avons abordées, et qui restent largement ouvertes, peuvent être résumées ainsi :

— Quelles sont les abstractions les plus adéquates pour la modélisation et la simulation de SMA ?
— Quels sont les méthodes et les outils logiciels les mieux à même de les implémenter efficacement ?

Par ailleurs, notre parcours, en particulier à travers les différents travaux doctoraux que nous avons aidés à encadrer, nous a amené à utiliser la simulation multi-agents dans différents domaines tels que le traitement numérique de l'image, le jeu vidéo, la robotique mobile collective ou encore la vie artificielle.

Pour présenter le cadre de nos travaux, dans ce chapitre nous utilisons des éléments publiés dans [Michel et al., 2009] [2].

## 1.1 Domaine de recherche

### Les systèmes multi-agents pour la simulation

Dans le cadre de la modélisation des systèmes dynamiques, les modèles à base d'équations mathématiques (*Equation-Based Model* EBM) et les modèles multi-agents (*Agent-Based Model* ABM) sont complémentaires dans le sens où ils reposent sur des perspectives très différentes. Sans qu'ils s'excluent mutuellement, il existe ainsi deux différences majeures entre les EBM et les ABM [Parunak et al., 1998] :

— niveau d'abstraction : les EBM reposent sur des relations entre grandeurs macroscopiques observables (e.g. les tailles relatives des populations dans un modèle proies-prédateurs). Les ABM réifient les comportements individuels, l'environnement et les interactions du niveau microscopique (prédation, fuite, reproduction, etc.)
— niveau d'analyse : les EBM adoptent une posture analytique –*top-down*– et se focalisent sur l'évolution des grandeurs macroscopiques pour élaborer le modèle. Les ABM

2. Coécrit avec Jacques Ferber et Alexis Drogoul, il s'agit d'un article d'une cinquantaine de pages en forme d'introduction au domaine de la simulation multi-agents et à certaines de ses problématiques.

étudient des dynamiques construites de façon ascendante –*bottom-up*– qui émergent de la modélisation du niveau microscopique.

Les motivations qui ont conduit aux ABM sont anciennes, comme le suggère la citation d'Henri Poincaré que nous avons reprise et qui discute déjà les limites d'une approche analytique et des connaissances qu'elle apporte. Par exemple, dans l'étude des EBM de dynamique de populations proies-prédateurs dites de *Lotka-Volterra* [Lotka, 1920 ; Volterra, 1926], on voit déjà très bien dans [Gause, 1934] la difficulté d'intégrer dans le modèle mathématique des considérations liées au niveau microscopique (e.g. l'existence de refuges pour les proies). En sciences sociales, Guy Orcutt a mis très tôt en évidence le fait que les modèles macroéconomiques ne permettaient pas d'étudier l'influence des politiques gouvernementales sur les entités du niveau microscopique (ménages et sociétés). Il proposa alors le principe de *microsimulation* qui intègre explicitement des décisions individuelles [Orcutt, 1957]. La microsimulation est considérée comme l'un des ancêtres directs des ABM. La principale différence, qui tend à disparaître, est que les ABM réifient les actions et les interactions [Troitzsch, 2009]. Cette dernière référence contient une discussion sur les différents courants qu'on peut rapprocher des ABM, en particulier les automates cellulaires (*Cellular Automata* CA) qui sont depuis longtemps employés pour modéliser l'environnement dans certains modèles [Schelling, 1971]. D'où aussi une connexion évidente avec la vie artificielle (ALife) où les CA sont aussi couramment utilisés [Langton, 1986].

Issus de ces différents courants, les ABM sont donc des modèles individus-centrés où l'on modélise le comportement d'agents autonomes (un processus décisionnel basé sur une perception locale), situés dans un environnement (e.g. un CA) où ils interagissent (e.g. l'interaction de prédation sera concrétisée par l'élimination de la proie).

Les simulations basées sur des ABM (*Multi-Agent Based Simulation* MABS) ont réellement pris leur essor dans les années 80, grâce à l'augmentation des capacités de calcul. Il est alors possible de visualiser, en temps réel, le résultat de l'interaction d'agents (voir les *Boids* de Reynolds, 1987). Les MABS ont alors pu devenir de véritables *laboratoires virtuels* où il est possible de tester des hypothèses, dans des mondes artificiels, sur des systèmes complexes originaux ; voir les travaux de références : éthologie, ALife et IAD [Bonabeau et Theraulaz, 1994 ; Drogoul, 1993], apprentissage de la *pensée décentralisée* [Resnick, 1994], systèmes complexes adaptatifs (CAS) [Holland, 1995], sciences sociales et ALife [Epstein et Axtell, 1996 ; Gilbert et Conte, 1995], outil de simulation générique (Swarm) [Minar et al., 1996], théorie des jeux [Axelrod, 1997] [3].

Depuis, les MABS ont été utilisées pour simuler nombre de phénomènes complexes : trafic [Mandiau et al., 2008], foule [Galland, 2013], sciences sociales [Amblard et Phan, 2006], finance [Brandouy et al., 2013], réseaux sociaux [Zhao et al., 2014], diffusion d'information [Kiesling

3. Le premier workshop de la vivante série *MABS*, alors intitulé *Multi-Agent Systems and Agent-Based Simulation*, eu lieu en 1998 [Sichman et al., 1998]. Ce premier workshop avait alors pour but de réunir les chercheurs en sciences sociales, utilisateurs d'ABM, et les informaticiens spécialistes en SMA. Voir aussi *JASSS* : *Journal of Artificial Societies and Social Simulation*.

et al., 2012], gestion de ressources et simulation participative [Le Page et al., 2012], écologie et biologie [DeAngelis et Grimm, 2014], réseaux P2P [Gutiérrez, 2012], (voir aussi les exemples donnés dans [Treuil et al., 2008] : physique, biologie, géographie, etc.)…Comme le souligne très justement Mathieu et al., 2015, "*l'agent fait le bonheur*", et pas seulement celui de la recherche [4] ; jeux vidéo, cinéma, etc.

### La simulation pour les systèmes multi-agents

Les SMA, en tant que paradigme pour la modélisation et l'implémentation de systèmes informatiques, sont historiquement liés aux évolutions conjointes des langages de programmation [Hewitt, 1977] et de l'intelligence artificielle (IA). Dans le projet historique Hearsay II [Erman et al., 1980], il est question de reconnaissance vocale à l'aide d'un ensemble de micro systèmes experts (KS) aux fonctions spécialisées. On passe du système expert monolithique, difficile à concevoir et à maintenir, à un ensemble de programmes indépendants plus simples, dont la cohérence globale est plus facile à obtenir. De plus, dans les travaux fondateurs tels que le *Contract Net Protocol* [Smith, 1980] ou le *Distributed Monitoring Vehicle Testbed* (DVMT) [Lesser et Corkill, 1983], l'idée d'une décentralisation des traitements et des données se voit renforcée par le fait que les problèmes sont eux-mêmes de nature distribuée.

C'est dans ce contexte, celui de l'*IA Distribuée* (IAD), que les définitions usuelles du concept d'agent, en tant que programme, ont été formulées quelques années plus tard [Ferber, 1995 ; Wooldridge et Jennings, 1995]. Un agent est vu comme une entité logicielle *autonome* (capable de fonctionner sans interventions externes), *réactive* (qui agit en fonction des perceptions qu'il a de son environnement), *proactive* (capable de prendre l'initiative) et *sociale* (qui interagit avec d'autres agents).

Si les problématiques liées à l'implémentation de SMA peuvent être, dans une certaine mesure, décorrélées de celles inhérentes aux MABS, il est en revanche important de mesurer le rôle fondamental que la simulation joue en IAD, et ce depuis les premiers temps. Si le *Contract Net Protocol* a pu être élaboré, c'est grâce au *CNET simulator* que Smith a développé [Smith, 1980]. Dans DVMT, l'un des trois objectifs affichés était de concevoir un *simulation system* permettant de tester différents scénarios, un *testbed*. Lesser et Corkill [1983] expliquent :

> "*…this approach is the only viable way to gain extensive empirical experience with the important issues in the design of distributed problem solving systems. In short, distributed problem solving networks are highly complex. They are difficult to analyze formally and can be expensive to construct, to run, and to modify for*

4. Les applications industrielles données en référence dans cet article sont en effet très éloquentes. Le logiciel MASSIVE, de nombreuses fois utilisé pour le cinéma ou la publicité, permet de générer des simulations de foules dont la vraisemblance est très poussée. Le moteur de jeu GlassBox, utilisé dans SimCity 2013, a permis de donner des dynamiques beaucoup plus riches et nombreuses grâce à une approche où tous les éléments du jeu sont agentifiés (building, voitures, incendie, etc.). Ces dynamiques étaient impossibles à obtenir dans les approches supervisées habituellement utilisées dans les jeux vidéo.

> *empirical evaluation... Thus, it is difficult and expensive to gain these experiences by developing a "real" distributed problem solving application in all its detail.*"

Ici, la simulation joue son rôle d'aide à la conception en permettant des *expériences contrôlées* sur le fonctionnement du système[5]. Ainsi, si la simulation est incontournable pour élaborer des solutions multi-agents destinées à des applications ancrées dans le réel (e.g. robotique collective [Simonin et al., 2014]), les *testbeds* ont aussi toujours joué un rôle fondamental en IA(D) [Decker, 1996]. Derrière ce terme se cachent toujours des simulations. Et, si une fois la conception achevée, le rôle de la simulation passe au second plan, il ne faut pas oublier son importance pour le développement des solutions multi-agents. Par exemple, si la théorie des AMAS (*Adaptive Multi-Agent Systems*) a le succès qu'on lui connaît [Gleizes, 2004], c'est en partie grâce aux simulations qui ont permis sa conception (*Tileworld* [Camps et Gleizes, 1996], *ANTS* [Topin et al., 1999]). Cette démarche expérimentale reste pertinente même pour les SMA basés sur des modèles dont on peut extraire analytiquement des propriétés comme les *Decentralized Partially Observable Markov Decision Processes* (DEC-POMDPs) (e.g. [Szer et Charpillet, 2006]).

Pour réaliser à quel point la simulation est proche des SMA, dans [Michel et al., 2009] nous avions analysé les actes de la sixième conférence *Autonomous Agents and Multiagent Systems* (AAMAS 2007) et relevions 800 apparitions du terme *simulation*, réparties sur plus de 35% des 273 publications. Et pourtant, il n'y avait aucune session dédiée aux MABS et très peu (trop peu) de papiers traitant explicitement la question de leur implémentation[6].

Au-delà de ce constat, une question se pose : existe-t-il une différence entre la simulation multi-agents telle qu'elle est pratiquée dans les MABS et les *testbeds* utilisés pour concevoir des SMA ? À un haut niveau d'abstraction la réponse est clairement non. Dans les deux cas il s'agit de simuler des agents autonomes qui interagissent entre eux dans un environnement partagé. Du point de vue de la simulation de ces systèmes, les problématiques fondamentales restent identiques, ne serait-ce qu'au niveau des abstractions utilisées pour les implémenter (Agent, Environnement, Interaction, Organisation [Demazeau, 1995]). Le lien entre MABS et génie logiciel orienté agent (*Agent-Oriented Software Engineering* AOSE) est donc une évidence, encore récemment soulignée par Fortino et North [2013]. C'est la raison pour laquelle une partie de nos travaux se situe dans le cadre plus général de l'AOSE. C'est notamment dans ce contexte que nous avons initialement étudié l'intérêt des approches environnement-centrées.

5. C'est ce que le *ACM Special Interest Group on SImulation and Modeling SIGSIM*, fondé en 1969, appelle le *Software-in-the-Loop Modeling and Simulation* ; l'évaluation par simulation du fonctionnement d'un logiciel.

6. Cette tendance se confirme au fil des éditions. Les actes de AAMAS, considérée comme *LA conférence agent*, sont donc assez peu représentatifs des avancées liées aux MABS. Ce qui pose question sachant l'apport indéniable que celles-ci ont eu, et ont encore, pour l'ingénierie des SMA, sans même parler de leur nombre. Dans son HDR, Alexis Drogoul regrettait la spécialisation des conférences et l'absence grandissante de multidisciplinarité [Drogoul, 2000]. Quinze ans après, les JFSMA restent effectivement une exception de par la diversité et la richesse des thèmes abordés. On retrouve aussi une volonté d'ouverture et d'interdisciplinarité dans la conférence *Self-Adaptive and Self-Organizing Systems* SASO. La conférence PAAMS, dédiée aux applications concrètes des SMA, permet par ailleurs de mesurer le très large spectre des domaines qui peuvent bénéficier d'une approche multi-agents.

## 1.2 Problématiques et positionnement scientifiques

### Cadre théorique

Dans notre thèse, nous avions identifié quatre aspects explicitement ou implicitement présents dans tout modèle de simulation multi-agents (figure 1.1) :

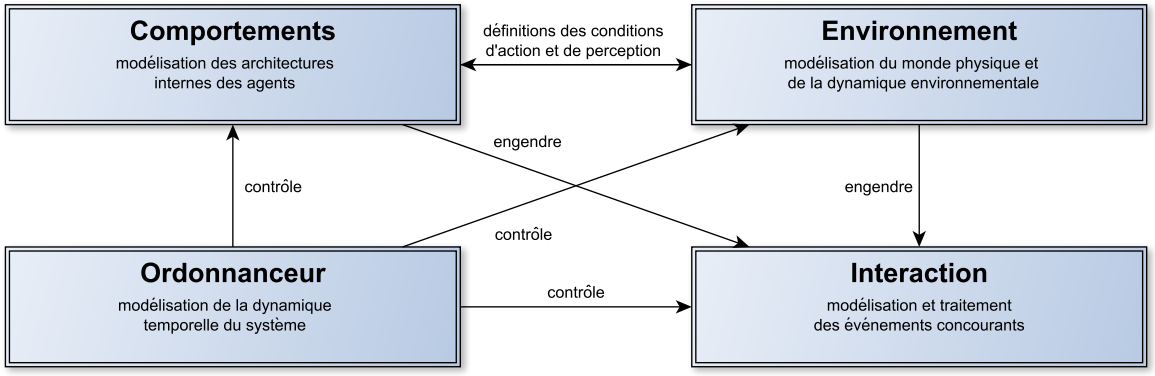



FIG. 1.1 – Les quatres aspects d'un modèle de simulation multi-agents

— ***comportements agent*** : modélisation des processus de délibération des agents (leur esprit). Il s'agit par exemple de modéliser la manière dont un prédateur décide de chasser une proie en fonction de ses perceptions.
— ***environnement*** : définition des différents objets physiques du monde (le corps des agents, les ressource naturelles, etc.), ainsi que la dynamique endogène de l'environnement (le fait que l'herbe pousse par exemple).
— ***ordonnancement*** (scheduling) : modélisation de l'écoulement du temps (granularité temporelle, ordre d'exécution entre agents, etc.).
— ***interaction*** : modélisation du résultat des actions et des interactions pour un instant $t$ (mouvement, prédation, etc.).

Dans ce cadre, nos motivations, ainsi que les questions de recherche que nous avons abordées, s'explicitent dans le contexte de la théorie de la modélisation et de la simulation (M&S) proposée par Zeigler et al. [2000]. C'est pour nous une référence depuis nos travaux de thèse. Bien que celle-ci ait été spécifiquement proposée pour fournir un socle formel aux simulations de systèmes dynamiques [7], on peut en extraire librement des éléments de réflexion transposables à tout contexte de simulation. En particulier, elle explicite très bien les *entités* clés de la M&S et caractérise les problématiques liées à leurs *relations*. Ces entités sont le *système source* (l'objet d'étude), le *cadre expérimental* (les conditions et les questions d'étude du système source), le *modèle* du *système source* et le *simulateur*.

7. Elle est généralement associée à l'utilisation du formalisme DEVS, qui est notamment très intéressant pour la multi-modélisation (EBM + ABM par exemple), cf. les thèses de Raphaël Duboz [Duboz, 2004], ainsi que l'habilitation de Jean-Christophe Soulié [Soulié, 2012].

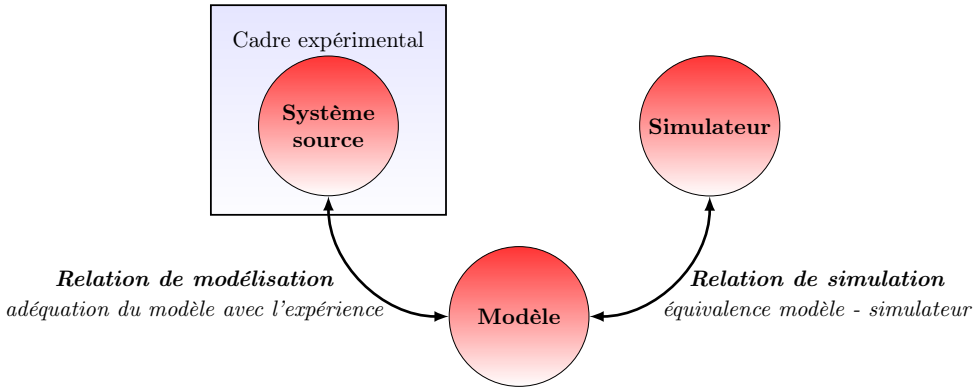



FIG. 1.2 – Cadre théorique pour la modélisation et la simulation [Zeigler et al., 2000]

L'entité *modèle* est évidemment au centre de la M&S. Si la sémantique des autres entités est assez intuitive, il est important de clarifier le terme modèle, car celui-ci possède bien sûr de multiples sens [8]. L'avantage du cadre théorique proposé dans [Zeigler et al., 2000] est de donner une définition très claire et non ambiguë de ce qu'est un *modèle* dans la M&S. Quelle que soit la manière de modéliser le *système source*, un *modèle* désigne l'ensemble des spécifications qui permettent de l'exécuter via un simulateur (Zeigler parle de *modèle de simulation*). En particulier, celui-ci est indépendant du langage utilisé pour l'implémenter, et ne doit souffrir aucune interprétation. Élaborer un *modèle* consiste donc à spécifier une fonction *Evolution* telle que, pour un état $\sigma(t) \in \Sigma$ du *modèle*, *Evolution* calcule l'état suivant $\sigma(t + dt)$ de manière déterministe :

$$\sigma(t + dt) = Evolution(\sigma(t)) \tag{1.1}$$

Par extension, un *simulateur* est un programme qui implémente fidèlement le *modèle*. Le point fondamental tient dans ce que le passage d'un état à un autre du système est calculé de manière déterministe : la reproductibilité est essentielle. Par exemple, en toute rigueur, l'algorithme éventuellement utilisé pour générer des nombres aléatoires fait partie du *modèle*. Un *modèle* est donc avant tout une description haut niveau d'un programme, il doit être *implémentable*.

La théorie de la M&S définit de façon générique les problématiques liées à une simulation à travers les deux relations qui lient un *modèle* aux autres entités (figure 1.2) :

- ***relation de modélisation*** : validité du *modèle* dans le contexte du *cadre expérimental* dans lequel est étudié le *système source*.
- ***relation de simulation*** : le modèle est implémentable (spécifications complètes) et correctement implémenté (simulateur sans biais et efficace).

Une partie de nos travaux s'apparente à des problématiques liées à la *relation de modélisation*. Dans ce document, celle-ci caractérise le fait d'être en capacité d'établir un *modèle* du

8. On y adjoint souvent des considérations épistémologiques en faisant référence à la définition de Marvin Minsky (qui traite spécifiquement du concept de modèle et non de simulation) : "*To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A*"[Minsky, 1965]. Voir par exemple les discussions développées dans l'habilitation de Guillaume Beslon [Beslon, 2008] ainsi que dans la thèse de Bruno Bonté sur l'usage de la *triade de Minsky* [Bonté, 2011].

système que l'on souhaite étudier (*système source*). En particulier, cela suppose de faire reposer la modélisation sur des *abstractions implémentables*. Pour les MABS, il est donc important de disposer des abstractions adéquates pour modéliser les agents, leur environnement et représenter leurs interactions dans le temps, afin d'être capable d'en construire un *modèle*.

Les problématiques inhérentes à la *relation de simulation* sont essentiellement liées à l'idée que le *simulateur* doit fidèlement reproduire le modèle. Le deuxième aspect que nous avons abordé est donc celui de l'ingénierie liée à l'implémentation des simulations multi-agents. Notamment, le simulateur ne doit pas introduire dans la simulation de biais dus à une mauvaise interprétation du *modèle*. De plus, l'efficacité d'exécution de la simulation est bien sûr un point important. Celle-ci dépend à la fois de la nature du *modèle* (complexité algorithmique) et de la manière dont le simulateur est conçu (implémentation efficace).

Les problématiques soulevées par chacune de ces relations sont différentes : modélisation d'un côté et implémentation de l'autre. Le point crucial est qu'elles sont fondamentalement reliées entre elles par l'entité *modèle*. En particulier, une modélisation ne reposant pas sur des abstractions implémentables posera problème lors de l'élaboration du modèle, et donc par conséquent pour son implémentation. De façon duale, le simulateur doit proposer des éléments permettant de simuler le modèle désiré.

### Sur la relation de modélisation

Les problématiques liées à la relation de modélisation sont très nombreuses et fortement intriquées [Bandini et al., 2009 ; Béhé et al., 2014 ; Michel et al., 2009]. Modéliser les comportements [9], le temps [10], l'environnement [11] ou les interactions [12] sont autant de questions difficiles et de nombreux points de vue sont possibles sur les relations qui les unissent.

Sur ces aspects, nos contributions reposent sur la proposition de modèles formels et conceptuels. Dans l'élaboration de ces modèles, nos principaux objectifs sont la **généricité** et l'**explicitation des concepts clés**.

Nous sommes intéressé en particulier par une problématique, identifiée de longue date, liée au flou entourant le passage du *SMA sur papier* (souvent désigné par *modèle conceptuel*) à son *modèle* concret, c'est-à-dire les spécifications du programme final (par extension le code de la simulation). Modéliser un SMA est une tâche non triviale et fortement liée au contexte. De fait, une question importante est de savoir si le *modèle de simulation* implémente effectivement ce qu'on souhaite modéliser (*système source*). Ce qui amène par exemple à poser la question suivante : *où sont les agents* dans le code final ? "*agents nowadays constitute a convenient*

9. La modélisation des comportements est bien sûr un élément central dans le cadre des SMA [Ferber, 1995]. On distingue classiquement les architectures réactives (e.g. [Drogoul, 1993 ; Simonin, 2001]) des architectures cognitives (e.g. *Believe Desire Intentions* (BDI) [Rao et Georgeff, 1992]).
10. [Payet et al., 2006a ; Sébastien et al., 2009].
11. [Badeig et Balbo, 2012 ; Balbo et al., 2013 ; Payet et al., 2006b ; Saunier et al., 2007 ; Soulié, 2001].
12. [Kubera et al., 2011 ; Michel et al., 2004 ; Sun, 2006].

*model for representing autonomous entities, but they are not themselves autonomous in the resulting implementation of these models*"[Drogoul et al., 2003].

C'est une question fondamentale. Il ne s'agit pas tant de valider ou de formaliser le processus de simulation de A à Z. Il s'agit de respecter les principes du paradigme multi-agents, notamment le concept d'autonomie décisionnelle (pas d'intervention extérieure) et le *principe de localité* : un agent *doit* réaliser ses perceptions et actions dans un contexte local, sur des données environnementales partielles. Cet aspect est au cœur du fonctionnement d'un SMA ; sans lui nous perdons une partie de l'intérêt du paradigme, ce qui est particulièrement bien exprimé dans [Parunak, 1997]. Par exemple, dans la méthodologie ADELFE, c'est un principe qui est au cœur de la conception d'un système adaptatif [Gleizes, 2004 ; Picard, 2004]. Son non-respect briserait le principe de résolution par émergence qui y est proposé. Comme cela a été souvent dit [Bonabeau, 2002 ; Resnick, 1994], les ABM reposent avant tout sur une approche conceptuelle plus que sur une technologie. Mais lorsque le code final n'a plus rien à voir avec des *agents autonomes*, les MABS perdent de leur potentiel à générer les *véritables* dynamiques que l'on souhaite étudier.

Il existe de nombreux travaux qui guident le processus de modélisation, mais nous partageons le constat fait par l'équipe SMAC de Philippe Mathieu et discuté dans l'habilitation de Sébastien Picault [Picault, 2013] : il existe d'une manière générale *une focalisation excessive sur l'individu* et, en proportion, la majorité des approches se focalisent sur la modélisation du processus délibératif des agents. Picault parle d'une potentielle perte complète de l'intelligibilité des comportements. Celle-ci tenant au fait que la modélisation commence par l'identification des agents puis s'attaque directement à leurs comportements, qui sont du coup modélisés/codés dans l'agent lui-même[13]. Dans ce cadre, SMAC propose de donner un rôle central au concept d'interaction avec la méthodologie IODA (*Interaction-Oriented Design of Agent simulations*) [Kubera et al., 2011]. À l'instar de [Badeig et Balbo, 2012] par exemple, nous verrons que notre parcours nous a pour notre part conduit à étudier plus spécifiquement le concept d'environnement. Au-delà de la représentation du monde physique, nous verrons que nos travaux illustrent l'intérêt de le considérer comme une entité de premier ordre dans la modélisation.

Sur les aspects modélisation, notre principale contribution repose sur un modèle formel générique pour la simulation multi-agents, IRM4S (*Influence Reaction Model for Simulation*, chapitre 2) [Michel, 2007a,b]. Celui-ci concrétise un certain nombre de concepts clés comme la simultanéité, l'autonomie ou encore le principe de localité. Ces travaux nous ont amené à collaborer avec d'autres chercheurs sur la modélisation de dynamiques environnementales dans les simulations [Helleboogh et al., 2007]. Par ailleurs, dans le cadre de l'AOSE, grâce à d'autres collaborations, nous avons étudié des modèles conceptuels qui s'intéressent directement au concept d'environnement (chapitre 3) [Ferber et al., 2005 ; Gouaïch et Michel, 2005 ; Weyns et al., 2005b]. Plus récemment, nous avons proposé un principe de conception qui repose sur

13. Ce qui induit aussi des problèmes de réutilisabilité, car le code des agents est du coup trop spécifique.

le déplacement d'une partie des calculs effectués par les agents dans l'environnement (chapitre 8).

### Sur la relation de simulation

À un haut niveau d'abstraction, définir un modèle de simulation multi-agents consiste à définir une fonction *Evolution* telle que les actions des agents à un instant $t$ $(A_1(t), A_2(t)...A_n(t))$ sont couplées avec l'évolution endogène de l'environnement $E_n(t)$, par un opérateur de composition $\uplus$, pour calculer l'état suivant du modèle :

$$\sigma(t + dt) = Evolution(\uplus(A_n(t), E_n(t)), \sigma(t)) \tag{1.2}$$

Étant donné la diversité des modèles et des approches existantes, il existe évidemment un nombre incalculable de façons de définir une telle fonction. Ce qui explique notamment le grand nombre de plates-formes disponibles. Kravari et Bassiliades [2015] recense et compare 24 plates-formes mais il en existe beaucoup plus. Les deux plates-formes les plus utilisées aujourd'hui sont RePast [M. North et al., 2013] et NetLogo [Sklar, 2007]. Au niveau francophone, on peut par exemple citer GAMA [Drogoul et al., 2013], JEDI (le moteur de simulation utilisé pour IODA) [Kubera et al., 2009] et GEAMAS-NG [Payet et al., 2006b] (évolution de GEAMAS [Marcenac et Giroux, 1998]).

Implémenter une simulation multi-agents (et a fortiori une plate-forme) est une tâche non triviale. On rencontre en particulier deux problèmes majeurs : (1) la concrétisation du modèle de simulation multi-agents dans une implémentation fiable et (2) les besoins importants en ressources de calcul ; simuler efficacement un modèle multi-agents est souvent difficile.

Sur le premier point, sans même parler des inévitables bugs, il est facile d'introduire dans le code de nombreux artéfacts, c'est-à-dire des biais dus aux choix réalisés pour implémenter certaines fonctions-clés (ordonnancement des agents, modélisation de l'interaction, etc.) [Bommel, 2009 ; Edmonds et Hales, 2003 ; Galán et al., 2009a ; Michel et al., 2004].

Face à ce problème, Edmonds et Hales [2003] ont par exemple défendu l'idée que les expériences (notamment l'implémentation du modèle conceptuel) doivent être répliquées le plus possible (idéalement par des équipes différentes) afin de détecter les artéfacts potentiels. Il est aussi de plus en plus question d'appliquer les méthodes de vérification et de validation classiques en M&S (V&V [Sargent, 2001]) dans le cadre des MABS [David, 2013], notamment grâce à une meilleure formalisation [Thiele, 2015].

Même si ces démarches ne peuvent pas nuire, on en revient souvent à la question précédemment posée : *où sont les agents* ? C'est là que se fait le lien entre la relation de modélisation et la relation de simulation, via le modèle. À l'instar de [Drogoul et al., 2003 ; Galán et al., 2009b ; Picault, 2013], nous pensons qu'une question centrale est de faire reposer la modélisation sur des abstractions qui se concrétisent dans l'implémentation, c'est-à-dire des *abstractions*

*implémentables*. La principale difficulté étant que ces dernières soient à la fois suffisamment simples et expressives pour permettre de modéliser le *système source* ciblé. Comme nous le verrons, IRM4S apporte des réponses sur cet aspect, car, en tant que modèle de simulation, il fait un lien direct entre certains concepts utilisés dans les SMA et leur implémentation.

Se pose ensuite directement la question des outils logiciels utilisés pour simuler les SMA. Ces derniers doivent permettre d'implémenter chacun des aspects déclinés précédemment (agents, environnement, interaction et ordonnancement, cf. figure 1.1), sans compromettre le modèle de simulation. Tout cela à l'heure où il est aussi de plus en plus question de simulations large échelle et/ou multi-niveaux [Morvan, 2012 ; Parry et Bithell, 2012 ; Picault et Mathieu, 2011], impliquant jusqu'à plusieurs millions d'agents.

Dans cette perspective, les outils logiciels que nous proposons ont deux objectifs majeurs :

— faciliter l'implémentation des simulations multi-agents grâce à des **librairies génériques** favorisant **accessibilité** et **réutilisabilité**.
— permettre la simulation de SMA **large échelle** grâce des moteurs de simulation efficaces et, plus récemment, grâce à l'utilisation du **calcul haute performance**.

Dans cette optique, nos contributions reposent essentiellement sur le développement des plates-formes génériques MadKit et TurtleKit (chapitre 7). Dans le chapitre 8, nous montrons comment une approche environnement-centrée nous a permis d'intégrer dans nos outils le calcul haute performance sur GPU (architectures massivement parallèles), tout en gardant à l'esprit nos premiers objectifs.

## 1.3 Aspects méthodologiques

Comme beaucoup de nos collègues nous mettons en avant le principe de parcimonie. C'était d'ailleurs le thème des JFSMA 2014. Pour nous, celui-ci joue un rôle à la fois au niveau des modèles et des outils que nous proposons. Il s'agit de faire appel au minimum de concepts nécessaires afin de rester le plus générique possible.

Un autre aspect de notre démarche est la place fondamentale que nous accordons à l'utilisateur final. Étant donné que nous nous plaçons dans une perspective où nos propositions sont génériques, leur pertinence se mesure en grande partie à travers la manière dont elles sont perçus par les autres. Beaucoup de travaux sur la simulation multi-agents prennent bien sûr en compte cette dimension, notamment à travers toutes les problématiques liées à la modélisation d'un SMA par des thématiciens non-spécialistes. Dans notre cas, nous nous intéressons plus spécifiquement à l'utilisateur informaticien, notamment aux développeurs. Ainsi, la manière dont sont perçus ou utilisés les modèles et les outils que nous proposons est pour nous une source très importante de réflexion.

Sur ce point, nous faisons d'ailleurs un usage intensif des outils que nous utilisons pour nos propres recherches dans le cadre de nos travaux pédagogiques à l'IUT. C'est pour nous un

moyen à la fois de promouvoir la programmation et la simulation de SMA, et de soumettre nos solutions logicielles à des étudiants qui n'ont aucune connaissance de ce paradigme, voire très peu d'expérience en programmation. La vision d'un esprit vierge de tout dogme permet non seulement de mesurer les qualités et les défauts des concepts que nous manipulons, mais elle est aussi souvent source de nouvelles idées très originales. En cela, notre démarche s'apparente totalement à celle employée par Mitchel Resnick au début des années 90 [Resnick, 1992], lorsqu'il attendait, de l'interaction avec ses étudiants, une exploration plus exhaustive des potentialités de ses propres outils.

## 1.4 Plan du document

Ce mémoire est structuré en deux parties. Sans forcément respecter un ordre chronologique strict, la première présente une synthèse des travaux et des coencadrements doctoraux passés. La seconde discute de nos recherches et développements en cours, avant de conclure sur les perspectives qui leur sont associées.

### Partie I

Le chapitre 2 présente les travaux liés au modèle IRM4S. Ceci nous amènera dans le chapitre 3 aux contributions qui traitent spécifiquement du concept d'environnement. Ces deux chapitres nous permettront par ailleurs de discuter notre point de vue sur certains aspects du paradigme multi-agents.

Les trois chapitres suivants mettent en perspective les travaux doctoraux que nous avons coencadrés jusqu'en 2013. Le chapitre 4 traite de segmentation d'image, le 5 d'IA et de jeux vidéo et le 6 de robotique mobile collective. Malgré la diversité thématique de ces applications, nous verrons qu'elles sont toutes source d'inspiration en ce qui concerne la manière dont on peut appréhender le concept d'environnement dans les SMA.

### Partie II

Le chapitre 7 présentent les plates-formes MaDKit et TurtleKit. Nous donnerons par la même occasion deux exemples de travaux directement liés à ces outils. Le premier en vie artificielle et le deuxième, effectué dans le cadre d'une collaboration industrielle, sur l'IA et le jeu vidéo.

Le chapitre 8 présentent nos travaux réalisés dans le cadre de l'utilisation du calcul haute performance sur GPU et les tout derniers développements qui en sont issus.

Le chapitre 9 présente les travaux qu'il nous semble important de mener à court et moyen termes, ainsi que les perspectives de recherche qui pourront leur être associées dans le futur.

# Première partie

# Travaux et coencadrements doctoraux passés

# Chapitre 2

# Le modèle IRM4S

Les premiers travaux que nous avons menés sont en lien direct avec notre thèse où nous avions utilisé le principe *influence/réaction* [Ferber et J.-P. Müller, 1996] pour modéliser la simultanéité des actions afin de traiter certains biais de simulation. Au fil de nos adaptations de ce modèle, il nous est apparu que la simultanéité n'est qu'une facette de cette théorie. Celle-ci a des implications beaucoup plus profondes qui concrétisent certains aspects du paradigme SMA tels que l'autonomie, la perception ou encore le rôle de l'environnement. Après la thèse, nous avons donc clarifié notre point de vue dans un modèle générique : IRM4S [Michel, 2007a,b]. Ces travaux nous ont par ailleurs amené à collaborer avec d'autres chercheurs sur la question [Helleboogh et al., 2007].

## 2.1 IRM4S : *Influence/Reaction Model for Simulation*

### Aux origines : le problème de la simultanéité

Le principe *influence/réaction* (IR) a été initialement proposé dans le livre de Jacques Ferber [Ferber, 1995] pour pallier les difficultés liées aux modélisations de l'action généralement envisagées en IA symbolique, c'est-à-dire par modification directe de l'environnement [1]. Trois constats sont développés : (1) le monde est considéré statique, seules les actions des agents le font évoluer (*postulat de staticité*), (2) impossibilité de représenter la simultanéité (*postulat de séquentialité*) et (3) une action n'est pas distinguée de ses conséquences et décrit à la fois *ce que fait l'agent* et *ce qui se passe* (*postulat d'universalité*). En une phrase, cette représentation de l'action ne fait aucun lien entre la dynamique de l'individu et la dynamique du monde.

1. Par exemple, Genesereth et Nilsson [1987] formalisent le cycle perception/délibération/action en le décomposant en trois applications. Soit $\sigma \in \Sigma$ l'état du monde, le comportement d'un agent $a$, $Behaviour_a : \Sigma \mapsto \Sigma$, est calculé en appliquant successivement $Percept_a : \Sigma \mapsto P_a$, qui calcule un percept à partir de l'état du système, $Mem_a : P_a \times S_a \mapsto S_a$, qui calcule le nouvel état interne de l'agent $s_a$ et $Decision_a : P_a \times S_a \mapsto \Sigma$, qui modifie le monde suivant l'action de $a$. On a donc une correspondance directe entre l'action et sa conséquence sur l'environnement (e.g. $\sigma = \{door(closed)\} \mapsto \sigma = \{door(open)\}$).

Dans [Ferber, 1995 ; Ferber et J.-P. Müller, 1996], le problème de la simultanéité est particulièrement mis en avant. Dans [Michel, 2007a], nous illustrions celui-ci à l'aide de la figure 2.1. Avec la modification directe de l'environnement, on en vient nécessairement à l'état 1 ou 2, et il est impossible de modéliser simplement une dynamique aboutissant à l'état 3.

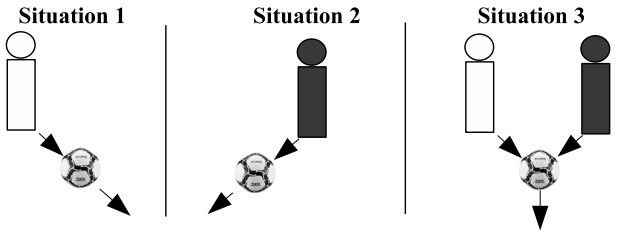



FIG. 2.1 – Robots footballeurs et simultanéité

Pour adresser cette difficulté, [Ferber, 1995 ; Ferber et J.-P. Müller, 1996] proposent des formalismes qui se fondent sur des principes d'influences et de réactions aux influences. Ainsi, un agent ne réalise plus des *actions*, mais **produit des influences** : une influence ne modifie pas directement l'environnement et rien ne garantit son résultat. L'idée majeure est de distinguer les gestes individuels (niveau agent) de ce qui se passe effectivement, c'est-à-dire la **réaction de l'environnement** aux influences (niveau multi-agent).

Cependant ces formalismes sont assez peu intuitifs et relativement complexes. Le problème tient dans ce qu'ils utilisent une extension du formalisme STRIPS [Fikes et Nilsson, 1972], c'est-à-dire précisément celui incriminé pour montrer les faiblesses de la représentation usuelle de l'action. Autrement dit, la solution proposée est formalisée dans un cadre qui ne s'y prête pas vraiment, ce qui complexifie notoirement sa notation. Les travaux qui s'en inspirent reposent d'ailleurs tous sur des adaptations, e.g. [Dàvila et Tucci, 2000 ; Demange, 2012 ; Simonin et al., 2011a ; Weyns et Holvoet, 2004]. Malgré cela, [Ferber et J.-P. Müller, 1996] continue d'être régulièrement cité dans des contextes très divers. En fait, le concept d'influence sous-tend un changement radical de perspective, dont l'intérêt va bien au-delà de la simultanéité.

### Le formalisme IRM4S

Avec IRM4S, notre objectif était précisément de mettre en lumière, grâce à un formalisme parcimonieux, les autres bénéfices implicitement associés à IR. En particulier, il s'agissait pour nous de montrer que IRM4S permet de concrétiser plusieurs aspects du paradigme multi-agents.

Ainsi, contrairement aux travaux qui adaptent IR en fonction d'un contexte applicatif spécifique, le but de IRM4S est d'être le plus générique possible. Dans cet optique, nous avons apporté un certain nombre de modifications au modèle original afin de (1) le simplifier et (2) d'en extraire les caractéristiques essentielles (tous les détails de ces modifications dans [Michel, 2007a]).

IRM4S définit un mécanisme à deux phases, (1) influence puis (2) réaction. Comme [Ferber et J.-P. Müller, 1996], nous utilisons le concept d'état dynamique, $\delta \in \Delta$, qui est une paire notée $\langle \sigma, \gamma \rangle$ où $\sigma \in \Sigma$ représente l'état du monde et $\gamma \in \Gamma$ les influences. L'évolution du système est calculée par une fonction *Evolution* telle que $\delta(t + dt) = \langle \sigma(t + dt), \gamma(t + dt) \rangle = Evolution(\langle \sigma(t), \gamma(t) \rangle)$. Celle-ci est décomposée en deux fonctions qui s'appliquent séquentiellement, *Influence* et *Reaction* :

$$Influence : \Sigma \times \Gamma \mapsto \Gamma' \quad puis \quad Reaction : \Sigma \times \Gamma' \mapsto \Sigma \times \Gamma \tag{2.1}$$

*Influence* calcule toutes les influences, $\gamma'(t) \in \Gamma'$, produites par le système (agents et environnement). *Reaction* calcule la manière dont le monde se transforme pour donner un nouvel état dynamique, $\delta(t + dt)$, à partir de $\sigma(t)$ et de $\gamma'(t)$. La fonction *Evolution* fait donc passer le SMA d'un état $\delta(t)$ au suivant $\delta(t + dt)$ en appliquant les équations suivantes :

$$\gamma'(t) = Influence(\sigma(t), \gamma(t)) \tag{2.2}$$

$$\langle \sigma(t + dt), \gamma(t + dt) \rangle = Reaction(\sigma(t), \gamma'(t)) \tag{2.3}$$

### Phase influence : agent et dynamique endogène de l'environnement

Le comportement d'un agent, $Behaviour_a : \Sigma \times \Gamma \mapsto \Gamma'$, est décomposé en trois fonctions qui s'appliquent séquentiellement :

$$p_a(t) = Perception_a(\sigma(t), \gamma(t)) \tag{2.4}$$

$$s_a(t + dt) = Memorization_a(p_a(t), s_a(t)) \tag{2.5}$$

$$\gamma'_a(t) = Decision_a(p_a(t), s_a(t + dt)) \tag{2.6}$$

Les influences dues à la dynamique endogène de l'environnement sont représentées par la fonction $Natural_w$ (objets en mouvement, évaporation d'une phéromone, lois physiques, etc.) :

$$\gamma'_w(t) = Natural_w(\sigma(t), \gamma(t)) \tag{2.7}$$

Contrairement à [Ferber et J.-P. Müller, 1996], où la dynamique de l'environnement est intégrée dans la réaction, nous considérons dans IRM4S que l'environnement produit lui aussi des influences. On a donc globalement $\gamma'(t) = Influence(\sigma(t), \gamma(t)) = \{\gamma(t) \cup \gamma'_w(t) \cup_{a=1}^{n} \gamma'_a(t)\}$. C'est un point qui contribue largement à la clarification du principe IR. L'idée est que les agents et l'environnement ne sont pas temporellement indépendants : toutes les influences sont simultanées, elles doivent donc être prises en compte globalement dans la réaction.

### Phase réaction : calcul du nouvel état dynamique du système

*Reaction* : $\Sigma \times \Gamma' \mapsto \Delta$ consiste à prendre en compte (1) toutes les influences et (2) l'état du monde courant afin de calculer le nouvel état dynamique du sytème :

$$\delta(t + dt) = Reaction(\sigma(t), \gamma'(t)) \tag{2.8}$$

IRM4S donne très peu d'indices sur le calcul de la réaction. Il s'agit en fait de rester générique. Rentrer dans les détails de ce calcul ne peut se faire en dehors du cadre d'une modélisation particulière. L'exemple suivant illustre le fonctionnement en deux phases du modèle pour un système composé de deux robots footballeurs :

— (1) État initial : $\delta(0) = \langle \sigma(0) = \{Bot_1, Bot_2, Ball\}, \gamma(0) = \{\}\rangle$
— (2) Influence : $\gamma'(0) = \{shoot_1(1,1), shoot_2(-1,1)\}$ : des vecteurs forces
— (3) Reaction : $\delta(1) = \langle \sigma(1) = \{Bot_1, Bot_2, Ball\}, \gamma(1) = \{move_{ball}(0,2)\}\rangle$
— (4) Influence : $\gamma'(1) = \{move_{ball}(0,2), slow_{ball}(0,-0.5)\}$
— (5) Reaction : $\delta(2) = \langle \sigma(2) = \{Bot_1, Bot_2, Ball\}, \gamma(2) = \{move_{ball}(0,1.5)\}\rangle$

L'évolution présentée correspond au scénario suivant : (1) $Bot_1$ et $Bot_2$ sont en position de frapper une balle. (2) Chacun décide de frapper la balle (*Influence*). (3) Les frappes sont combinées pour donner un mouvement à la balle (*Réaction*). (4) Ce mouvement entraîne la production d'une influence par l'environnement qui modélise les frictions auxquelles la balle est soumise (*Influence*). (5) La balle change de position et sa vitesse est ralentie (*Réaction*).

Cet exemple illustre plusieurs points. Tout d'abord, la modélisation de la simultanéité est effectivement simplifiée : dans l'étape 2, on dispose de toutes les informations nécessaires au calcul de la résultante des frappes aboutissant à l'état dynamique de l'étape 3. Deuxièmement, on voit que l'évolution endogène de l'environnement est facilement intégrée (étape 4). Enfin, les agents ont la possibilité de percevoir le fait que la balle est en train de rouler car cette dynamique est modélisée à l'aide d'une influence qui peut être perçue par les agents (état dynamique $\delta$). La figure 2.2 schématise le déroulement en deux phases d'un cycle IRM4S :

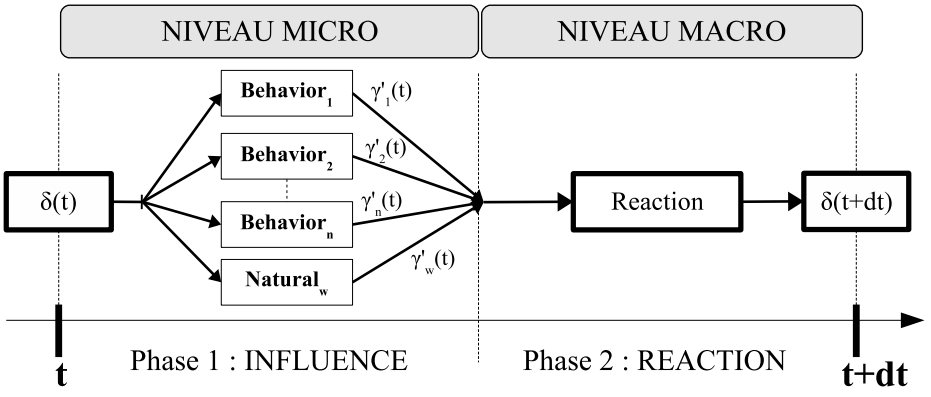



FIG. 2.2 – Évolution en deux phases d'un système modélisé avec le modèle IRM4S

## 2.2 Caractéristiques du modèle IRM4S

En premier lieu, IRM4S constitue bien sûr un cadre qui facilite la modélisation des actions simultanées [Badeig, 2010 ; Demange, 2012 ; Lebrun, 2012]. Cependant, comme nous l'avons dit, la représentation de la simultanéité n'est qu'une facette du principe IR. Dans cette section, nous discutons d'autres aspects qui sont rendus explicites par IRM4S (détaillés dans [Michel, 2007a]). Pour chacun d'entre eux, nous listons à la fin des sous-sections quelques travaux qui référencent IRM4S et qui traitent explicitement du sujet concerné.

### Contrainte de localité et contrainte d'intégrité environnementale

Comme dit en introduction (section 1.2), le fait qu'un agent réalise ses perceptions et ses actions localement est reconnu comme un point essentiel des SMA [2]. Cette contrainte est indissociable de IRM4S, et plus généralement de IR : elle est à la fois sa cause et sa conséquence. C'est elle qui rend nécessaire la proscription de la modification directe de l'environnement (*contrainte d'intégrité environnementale*). Précisément parce que la perception d'un agent est par définition locale, celui-ci *ne peut pas disposer des données* requises pour calculer le résultat de ses actions. Il faudrait pour cela qu'il ait une connaissance exhaustive de son environnement et de sa dynamique, ce qui serait en parfaite contradiction avec la contrainte de localité. D'où l'intérêt de remplacer *action* par *influence*, changement qui entraîne le respect systématique des deux contraintes que nous venons d'évoquer (voir [Galland, 2013 ; Logie et al., 2010 ; Saunier et Jones, 2014 ; Siebert, 2011 ; Simo Kanmeugne, 2014]).

### Concrétisation du concept d'autonomie

Le concept d'autonomie possède plusieurs interprétations dans la littérature SMA et reste depuis longtemps sujet à controverse [Castelfranchi, 1995] [3]. Si on se focalise sur le comportement, l'autonomie est souvent rapprochée du concept de proactivité : les décisions d'un agent ne sont pas uniquement mues par ses perceptions mais expriment aussi l'existence d'un but [Guessoum et Briot, 1999]. Si, au contraire, l'autonomie est étudiée d'un point de vue externe à l'agent, on peut la voir comme un concept relationnel qui marque le degré d'indépendance sociale [Sichman et al., 1994] ou la liberté d'action vis-à-vis du reste de la société (dans une organisation par exemple [Hubner et al., 2007]). Il peut alors s'agir de limiter l'autonomie d'un agent afin que son degré de liberté ne puisse pas affecter la bonne marche du système.

Appréhender l'autonomie dans les SMA est donc largement une question d'opinion et les avis divergent sur ce qui fait qu'un agent est *vraiment* autonome. Comme d'autres, nous pen-

2. Dans la littérature, notamment francophone, cet aspect est explicité sous l'appellation de *principe de localité* [Siebert et al., 2009 ; Thomas et al., 2007] ou *contrainte de localité* [Chevaillier et al., 1999].

3. Voir les post-proceedings du workshop, *Agents and Computational Autonomy*, qui était dédié à la question [Nickles et al., 2004]. On y trouve notamment une taxonomie de l'autonomie [Carabelea et al., 2004].

sons qu'il est intéressant de relier cette propriété à un critère qui ne souffre pas l'interprétation [Gouaich, 2004 ; Weiß et al., 2003]. Ainsi, nous considérons un agent autonome uniquement si les variables utilisées par son processus décisionnel ne peuvent être modifiées que par celui-ci : aucun autre processus ne doit y avoir accès. Ce qui est par exemple caractérisé par *la contrainte d'intégrité interne* de l'agent dans [Gouaich, 2004] [4].

Grâce au concept d'influence, dans IRM4S cette propriété est nécessairement concrétisée car c'est une conséquence du modèle : aucun agent ne peut violer l'intégrité interne d'un autre. Ce n'est pas le cas lorsque la modification directe de l'environnement est permise, car un agent peut éventuellement modifier l'état d'un autre, invalidant de fait son autonomie au sens pris ici (voir [Logie et al., 2010 ; Siebert, 2011]).

### Distinction esprit / corps explicite

La distinction esprit/corps est un sujet régulièrement évoqué dans la littérature SMA pour son intérêt conceptuel (e.g. [Platon et al., 2005 ; Tranier, 2007]). Elle est aussi très souvent mise en avant dans les simulations multi-agents comme un principe de conception qui permet plus de modularité dans la modélisation [Okuyama et al., 2005 ; Soulié, 2012]. Nous aurons d'ailleurs l'occasion d'y revenir car cette question est bien sûr fortement connexe avec celle de l'environnement.

Dans IRM4S, les traits physiques d'un agent font partie de l'environnement, car ils doivent être modifiés lors de la phase de réaction (e.g. sa localisation). À l'opposé, l'état interne d'un agent contient des variables directement modifiées et utilisées par le système décisionnel (e.g. les variables d'un mécanisme de renforcement). IRM4S nécessite donc de séparer clairement le modèle décisionnel (esprit) du modèle physique (corps) d'un agent (voir [Galland, 2013 ; Saunier et Jones, 2014 ; Simo Kanmeugne, 2014]).

### Réification des interactions dans l'environnement

L'interaction est évidemment un concept clé dans les SMA. Dans IRM4S, modéliser la dynamique des interactions devient une problématique incontournable qui ne peut pas être résolue au niveau des agents, mais doit l'être au niveau de l'environnement (combinaison des influences dans la réaction). C'est une conséquence du principe IR (voir par exemple la discussion sur l'interaction et IR dans l'habilitation de Vincent Chevrier [Chevrier, 2002]).

4. Cette perspective génie logiciel nous permet en particulier d'éviter l'épineuse question du but. Si cette notion peut être évidente dans les architectures cognitives, elle est beaucoup plus floue dans le cadre des architectures réactives. De pus, lier autonomie (agent) et proactivité (but), exclut de fait la modélisation sous forme d'agents de phénomènes tels une balle qui roule ou un feu de forêt ; ce qui nous paraît beaucoup trop restrictif : *tout peut être agentifié* [Kubera et al., 2010].

La modélisation de l'interaction était d'ailleurs un des thèmes centraux de notre thèse[5]. Dans les simulations multi-agents, les interactions ne sont généralement pas modélisées explicitement et résultent souvent directement de la façon dont les comportements et leurs actions sont concrétisés.

IRM4S nécessite de décrire *comment les influences interagissent* lors de la phase de réaction de l'environnement. Ce qui introduit une dichotomie claire, à la fois dans le modèle et dans l'implémentation, entre modélisation des comportements et modélisation des interactions (voir [Simo Kanmeugne, 2014 ; Steel et al., 2010]).

Avec le recul, il nous paraît ici intéressant de faire un lien avec la simulation orientée interaction (IODA [Kubera et al., 2011]). Celle-ci repose explicitement sur l'idée que la modélisation d'une interaction passe par la prise en compte de tous les agents impliqués, ce pourquoi elle doit être réifiée indépendamment des agents. Un aspect essentiel de IODA est donc de décorréler explicitement la modélisation des interactions, qu'on peut voir comme une réaction, des entités qui y participent. De fait, les comportements ne modifient pas directement l'environnement dans cette approche.

## 2.3 Apports de IRM4S pour la simulation multi-agents

### Prévention des biais de simulation dus à l'ordonnancement

Dans l'introduction, nous avons souligné que les modèles SMA sont très sensibles à leur implémentation. En particulier, l'ordre dans lequel les agents sont activés (perceptions et actions) a une influence certaine sur la dynamique globale, à l'image de la figure 2.1 où l'on peut aboutir à trois états du monde différents. De plus, les agents peuvent avoir des perceptions du monde très différentes pour une même estampille $t$, ce qui peut provoquer de subtiles artéfacts, même pour les modèles les plus simples [Chevrier et Fatès, 2008]. Concrètement, cela influence les perceptions, et par extension les actions produites.

Dans IRM4S, calculer la réaction nécessite de connaître toutes les influences produites pour un même instant $t$. Par conséquent, l'ordre dans lequel les agents perçoivent le monde et produisent leurs influences n'a aucun impact sur le résultat final (voir [Demange, 2012 ; Siebert, 2011]).

### Parallélisation de la phase influence

Cette indépendance du résultat vis-à-vis de l'ordre d'activation des agents a une autre conséquence, cette fois pratique. Il est possible de paralléliser tous les calculs de la phase

5. Pour implémenter IR, nous avions notamment utilisé le modèle formel $MIC^*$, {*Movement, Interaction, Calcul*}$^*$, proposé par Abdelkader Gouaïch dans sa thèse [Gouaïch, 2005]. Celui-ci définit des *objets d'interactions* (qu'on peut utiliser pour représenter les influences), qui évoluent selon des lois environnementales (réaction) propres à des *espaces d'interactions* (une matrice), cf. [Gouaïch et al., 2005].

influence : le comportement des agents et l'évolution endogène de l'environnement. C'est une caractéristique importante d'IRM4S dans la mesure où la parallélisation des simulations multi-agents est un sujet récurrent (large échelle). En général, la distribution d'une simulation soulève un certain nombre de questions quant à la cohérence globale de la simulation et les problèmes peuvent être nombreux [Cicirelli et al., 2015 ; Rihawi et al., 2013 ; Sébastien, 2010].

Même si la réaction constitue indubitablement un point de centralisation dont la distribution pose problème (nous en discutons dans [Michel, 2007a]), dans IRM4S la répartition du calcul des influences est assez simple car elle n'implique aucune gestion additionnelle concernant la cohérence de la simulation. Ce qui permet notamment de profiter des architectures multi-cœurs très aisément, sans renoncer à la cohérence globale.

### Modélisation de perceptions basées sur les influences

Dans l'exemple donné section 2.1, on voit qu'un agent peut percevoir une influence. Cela permet de modéliser des perceptions qui expriment la dynamique d'une situation, comme le fait qu'une balle roule. Une telle perception est difficile à calculer si on s'en tient uniquement à une description statique du monde $\Sigma$, car l'agent doit construire une représentation complexe de son environnement dans laquelle il lui faut stocker la position de la balle, état après état, pour pouvoir ensuite en déduire que celle-ci est en train de rouler.

Lorsqu'on permet aux agents de percevoir des influences, ils n'ont pas besoin de mettre en œuvre de tels calculs. Grâce au concept d'état dynamique $\Delta$ et à la réification des influences $\Gamma$, il est possible de concevoir des raisonnements directement à partir des influences : la balle roule.

Ceci ne s'arrête d'ailleurs pas à la dynamique environnementale. Il est effectivement simple de modéliser qu'un agent *perçoive le comportement* d'un autre agent. Si l'on réifie dans l'environnement le comportement d'un agent sous la forme d'une influence, celui-ci devient observable par les autres agents. Dans un modèle proie-prédateur, on peut représenter le comportement de fuite ou d'attaque sous la forme d'influences. La proie peut alors savoir si un prédateur en a après elle, car elle peut le *sentir* directement dans l'environnement. Les agents peuvent donc *percevoir les désirs des autres agents*, sans avoir à accéder aux états internes des autres (voir [Logie et al., 2010]). Nous n'avons pour l'instant pas exploité cet aspect dans nos travaux, mais il a sans doute un potentiel en ce qui concerne l'expressivité des comportements qu'il permet d'envisager.

Avant de conclure cette section, notons enfin les travaux réalisés par Stéphane Galland et ses collègues qui, dans le cadre de la modélisation de foule et de trafic, ont développé tout un ensemble de modèles et d'outils qui intègrent explicitement bon nombre des aspects que nous avons discutés (esprit/corps, autonomie, contrainte de d'intégrité, etc.) ; voir les travaux réalisés avec la plate-forme JaSIM [Galland, 2013]).

Pour ce qui est du formalisme IRM4S en particulier, notons qu'il a été étendu (IRM4MLS) pour la simulation multi-niveaux par Gildas Morvan et ses collègues [Morvan et al., 2011]. Une méthodologie associée à ce dernier est proposée dans [J.-B. Soyez et al., 2013].

## 2.4 Modéliser la dynamique de l'environnement

Le concept d'influence est sans aucun doute l'élément clé du principe IR. De lui dérivent quasiment toutes les propriétés que nous avons évoquées. Cependant, bien qu'il représente une évolution du vocabulaire liée au concept d'agent, ce changement a pour conséquence directe de mettre l'environnement au premier plan. Ainsi, malgré une disparité naturelle dans la manière d'interpréter IR, tous les travaux qui s'en réclament font de l'environnement une abstraction de premier ordre : définir la réaction ne peut se faire au sein des agents. La modélisation de la dynamique de l'environnement devient donc essentielle, indissociable du comportement global du système. Suivant la façon dont la réaction est modélisée, il est possible d'obtenir des dynamiques très différentes sans modifier en quoi que ce soit le comportement des agents, cela même pour des agents simplissimes comme les *turmites* (une extension des fourmis de Langton à plusieurs agents) [Chevrier et Fatès, 2008].

La modélisation de la dynamique de l'environnement peut prendre différentes formes dans la mesure où il est bien sûr possible de concevoir la réaction en utilisant des approches très diverses. Dans [Helleboogh et al., 2007], nous avons collaboré avec Alexander Helleboogh, Giuseppe Vizzari et Adelinde Uhrmacher dans le cadre de la définition d'un modèle formel, basé sur IR, qui fait reposer la dynamique de l'environnement sur le concept d'activité. Dans ce modèle, une activité décrit la façon dont un élément de la simulation évolue (que ce soit le corps d'un agent ou tout autre élément physique). Ainsi les influences des agents engendrent ou modifient des activités (e.g. déplacement). La modélisation de la dynamique environnementale consiste à spécifier la façon dont toutes les activités interfèrent entre elles (réaction) pour donner une évolution globale du système dans le temps.

Comme on le voit sur la figure 2.3, la modélisation de l'environnement est découpée en trois parties : (1) la description de la structure physique de l'environnement, c'est-à-dire l'ensemble des variables qui décrivent le monde de manière statique ; (2) la description de la dynamique des activités (début, fin, évolution, etc.) ; (3) le contrôle de la dynamique (gestion des activités et de leurs interférences). Si on se réfère aux articles qui s'inspirent de ce modèle, il est clair que cette décomposition constitue la principale contribution de ce travail, plus que le formalisme lui-même. En effet, même si ce dernier a pu être repris ou étendu dans d'autres travaux (e.g. [Ranathunga et al., 2012]), la modélisation complète d'un système s'avère assez fastidieuse. Décrire les lois de réaction et d'interférence, ainsi que la façon dont elles modifient les activités, est rapidement problématique lorsqu'on modélise des systèmes possédant des dynamiques non triviales, inspirées du réel (les robots footballeurs sont pris en exemple dans [Helleboogh et al., 2007]). Ce qui ne va pas sans perdre un peu de l'intelligibilité de la modélisation.

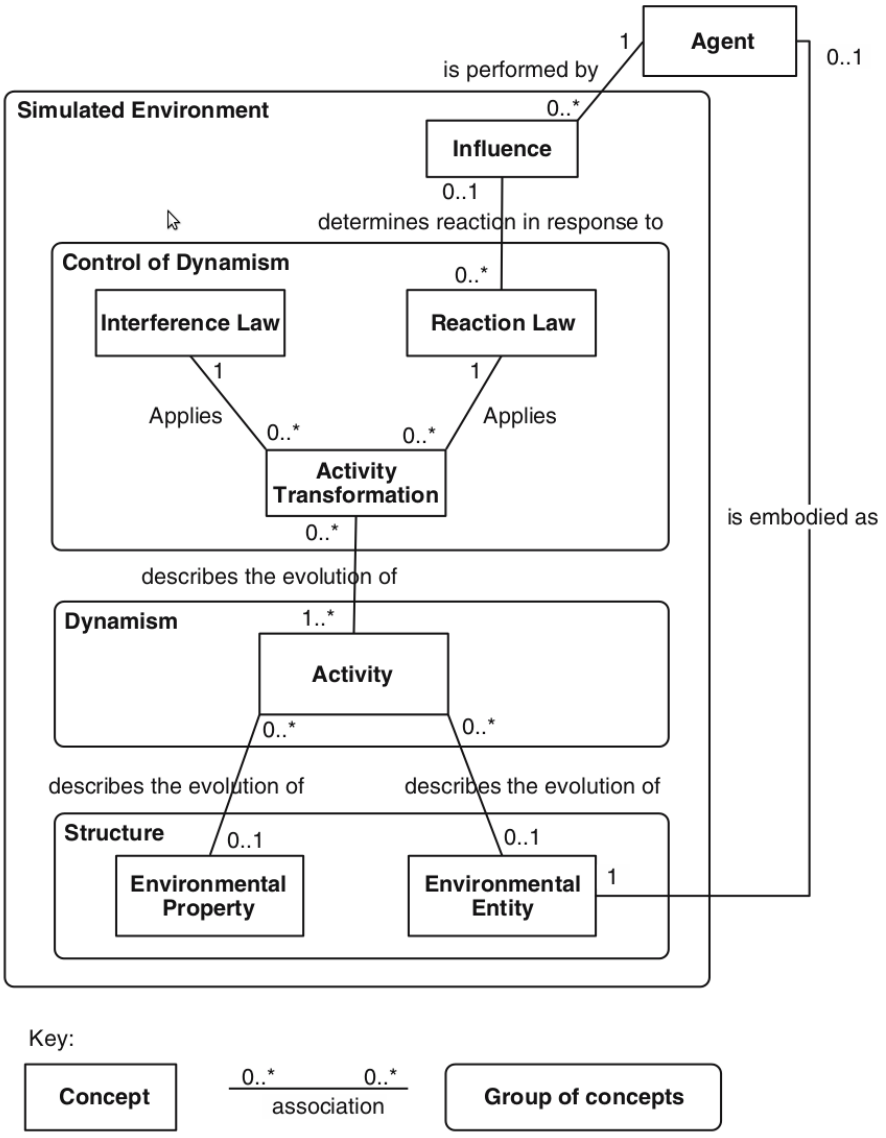



FIG. 2.3 – Décomposition de la modélisation de l'environnement dans [Helleboogh et al., 2007] : (1) structure, (2) dynamique et (3) contrôle de la dynamique

En fait, on atteint ici une des limites des approches basées sur IR, celle de la complexité de sa mise en œuvre. Si cette dernière contribution est très intéressante du point de vue de l'application de IR, elle montre aussi la difficulté de pousser le raisonnement dans ses derniers retranchements, c'est-à-dire en intégrant simultanément *toutes* les influences dans le calcul de la réaction. Ce qui est tout simplement impossible à partir d'un certain niveau de complexité du système. Il faut faire des choix.

Pour autant, cette difficulté n'empêche pas d'utiliser IR sur des SMA relativement complexes, comme le montre [Simonin et al., 2011b] qui propose une spécification formelle de [Ferber et J.-P. Müller, 1996], réalisée à l'aide du langage B, dont le but est de permettre la validation théorique de SMA réactifs (l'exemple est le *platooning*). Il est cependant clair

qu'un travail conséquent reste à faire au niveau méthodologique en ce qui concerne le calcul de la réaction, d'où l'intérêt de faire le lien avec les travaux sur l'interaction que nous avons mentionnés dans la section 2.2.

## 2.5 Bilan

Depuis ces travaux, nous n'avons pas retravaillé sur le modèle IRM4S. Nous verrons cependant dans les perspectives de ce mémoire qu'il reprend aujourd'hui, en 2015, une importance toute particulière dans nos recherches les plus récentes. En fait, fin 2007, le bilan que nous tirions de ces travaux était teinté à la fois de satisfaction et de frustration.

D'un côté, IRM4S est un modèle qui exhibe de bonnes propriétés d'un point de vue conceptuel. Sur ce plan, il répondait très bien aux objectifs que nous nous étions fixés dans le sens où il fournit un cadre qui permet d'expliciter des aspects importants de la modélisation et de la simulation multi-agents. Il reste donc un élément essentiel de notre réflexion et de notre vision des SMA. On peut dire que nous *pensons influence/réaction*. C'est d'ailleurs pour cette raison que nous avons étudié le concept d'environnement. Comme nous l'avons expliqué, IRM4S nécessite d'appréhender l'environnement comme une abstraction incontournable.

Dans le même temps, par rapport à nos objectifs liés à la conception d'outils génériques de simulation, sa mise en œuvre concrète nous paraissait trop difficile. Autrement dit, nous n'avions pas d'intuitions assez claires sur la manière d'intégrer IRM4S dans nos outils sans que cela ne complexifie notoirement leur usage. Proposer une plate-forme générique pour faire des *simulations IRM4S compatibles* nous paraissait tout simplement voué à l'échec en termes d'accessibilité et de réutilisabilité [6].

Dans cette analyse, nous avons sans doute aussi été très influencé par le fait que les performances de nos propres simulations étaient loin d'être exceptionnelles. Même pour des interactions simples, calculer la réaction de l'environnement s'avérait très coûteux. Il nous paraissait donc difficile de convaincre de l'intérêt d'utiliser des outils où les habituelles simulations de démonstrations vont trois fois moins vite que sur une autre plate-forme. C'est d'ailleurs l'une des raisons qui nous a poussé à nous tourner vers le calcul haute performance et les architectures massivement parallèles.

6. Intuition contredite par les travaux menés par Stéphane Galland [Galland, 2013] (voir la thèse de Jonathan Demange [Demange, 2012]), même si, de ce que nous avons compris des discussions que nous avons pu avoir, tout n'est bien sûr pas aussi simple : le calcul de la réaction reste un point difficile.

# Chapitre 3

# Sur le concept d'environnement dans les SMA

Ce chapitre présente nos travaux qui traitent explicitement du concept d'environnement dans les SMA. Dans nos recherches, cette orientation trouve ses racines dans des interactions que nous avons eues avec d'autres chercheurs, le plus souvent en dehors du contexte de la simulation, dans le cadre de l'AOSE. Ces collaborations ont débuté vers la fin de la thèse. Comme nous l'avons dit, le principe influence/réaction donne un rôle fondamental à la modélisation de l'environnement. C'est précisément pour cette raison que nous nous sommes rapproché de chercheurs plus spécialistes que nous en AOSE. Il nous fallait dépasser la vision purement physique que nous avions de l'environnement, telle que nous la pratiquions alors dans les simulations multi-agents.

## 3.1 Environments 4 Multi-Agent Systems

Au risque de nous répéter, il n'est pas anecdotique de mentionner que c'est notre intérêt pour influence/réaction qui nous a fait rencontrer Danny Weyns, et non le hasard. Son interprétation de [Ferber et J.-P. Müller, 1996] avait attiré notre attention car elle était très proche de la notre [Weyns et Holvoet, 2003]. Nous étions d'accord sur l'idée que l'environnement était un concept clé dans influence/réaction. Il était cependant clair que, à de rares exceptions près [Odell et al., 2003], l'environnement n'avait été finalement que très peu étudié pour lui-même, en dehors du contexte des agents situés et de la simulation. Autrement dit, alors qu'il était désigné comme une dimension fondamentale des SMA (e.g. dans *Voyelle* [Demazeau, 1995] ou l'introduction à l'IA de [Russell et Norvig, 1995]), il n'était pas dans les faits considéré comme une abstraction de premier ordre.

De ce constat est née une collaboration avec Danny Weyns et Van Parunak pour l'organisation d'une série de workshops intitulée *Environments for Multiagent Systems* : E4MAS. Dans la préface du premier volume, Parunak notait le fossé qui existait entre les approches

d'inspiration vie artificielle, où la dynamique de l'environnement joue un rôle crucial, et celles issues d'une vision IAD plus classique, où l'intelligence collective repose essentiellement sur la combinaison d'intelligences individuelles, et où l'environnement n'a quasiment aucun rôle[1]. Notre objectif était donc avant tout de clarifier le concept d'environnement, afin de promouvoir une meilleure intégration de celui-ci dans la conception des SMA, quelle que soit leur nature. Les trois premières éditions d'E4MAS[2] ont eu lieu en 2004, 2005 et 2006, dans le cadre de la conférence AAMAS.

Dans leur ensemble, les articles issus des trois premiers volumes traitent du concept d'environnement selon des angles extrêmement variés : AOSE (modèles conceptuels [Chang et al., 2005], architectures [Ricci et al., 2007b ; Weyns et Holvoet, 2007], méthodologie [Simonin et Gechter, 2006]), simulation multi-agents (langage [Okuyama et al., 2005], plate-forme [Steiner et al., 2005]), coordination et interaction médiées par l'environnement [Platon et al., 2006 ; Saunier et al., 2007], normes et institutions électroniques [Arcos et al., 2007], etc.

Dans une certaine mesure, c'était un résultat espéré qui montrait l'intérêt du sujet. De plus, l'évolution des propositions au fil des années est assez intéressante. Dans sa préface du premier volume, Parunak concluait sur le fait que les contributions mettaient en évidence l'intérêt d'une hybridation entre les approches issues de l'IAD et celles issues du vivant, à travers un renforcement du rôle de l'environnement. C'était effectivement le cas. Mais ces idées ont ensuite maturé et sont livrées de manière beaucoup plus explicite dans les volumes suivants. Par exemple dans le troisième volet, [Ricci et al., 2007a] avance le concept de *stigmergie cognitive*. [Arcos et al., 2007] propose de *situer* les institutions électroniques. [Gardelli et al., 2007] étudie la conception d'environnements dédiés à faciliter l'auto-organisation, etc.

Lors de la troisième édition, cinq groupes de travail furent constitués afin de consolider certaines idées à travers plusieurs publications simultanées pour la revue *Autonomous Agents and Multi-Agent Systems*. Celles-ci sont emblématiques des travaux liés à E4MAS :

— L'environnement : abstraction de premier ordre dans les SMA [Weyns et al., 2007a]
— Mécanismes liés à l'ingénierie de l'environnement [Platon et al., 2007]

1. Un extrait de la préface [Weyns et al., 2005a] : *All interaction among agents of any sort requires an environment. For an AI agent whose interactions with other agents are based on speech act theory, the environment consists of a computer network that can convey messages…For an ALife agent, the environment is whatever the agent's sensors sense and whatever its effectors try to manipulate. In most cases, AI agents (and their designers) can take the environment for granted. Error-correcting protocols ensure that messages once sent will arrive in due course. Message latency may lead to synchronization issues among agents, but these issues can be discussed entirely at the level of the agents themselves, without reasoning about the environment. As a result, the environment fades into the background, and becomes invisible. Not so for ALife agents. Simon observed long ago that the complex behavior of an ant wandering along the ground is best explained not by what goes on inside the ant, but by what happens outside, in the structure of the ground over which the ant moves [Simon, 1969]. When a termite interacts with other termites by depositing and sensing pheromones, the absorption and evaporation of the pheromone by the environment plays a critical role in the emergent structure of the colony's behavior. There are no error-correcting protocols to ensure that an agent who tries to push a rock from one place to another will in fact be able to realize that objective. From the ALife perspective, the environment is an active participant in agent dynamics, a first-class member of the overall system.*

2. Chaque édition a été suivie de la préparation d'un ouvrage collectif composé des versions étendues d'articles sélectionnés et réévalués, ainsi que d'articles invités. Ces ouvrages sont publiés comme des volumes de la collection LNCS : 3374 [Weyns et al., 2005a], 3830 [Weyns et al., 2006b], 4389 [Weyns et al., 2007b].

— Étude des infrastructures pour environnements [Viroli et al., 2007]
— Applications environnement-centrées [Valckenaers et al., 2007]
— Dynamiques environnementales et simulation [Helleboogh et al., 2007] (notre groupe).

Notamment, [Weyns et al., 2007a] (le groupe composé de Danny Weyns, Andrea Omicini et James Odell) donne pour l'environnement la définition suivante :

> *The environment is a* ***first-class abstraction*** *that* ***provides the surrounding conditions for agents to exist*** *and that* ***mediates both the interaction among agents and the access to resources****.*

L'objet de cette définition, qui se veut volontairement générale, est avant de tout de synthétiser trois idées récurrentes de la perspective E4MAS. Premièrement, l'environnement doit être considéré comme une abstraction de premier ordre, au sens génie logiciel [3]. À ce titre, il doit être conçu, indépendamment des agents, de telle sorte que ses responsabilités soient clairement identifiées. Deuxièmement, il fournit les conditions d'existence des agents (actions et perceptions) : tous les SMA en possèdent un. Enfin, parce qu'il est le lieu de l'interaction, l'environnement peut contenir des dynamiques additionnelles qui permettront de catalyser la coopération et la collaboration (e.g. par stigmergie).

Après ces trois éditions, nous avions alors estimé avoir atteint notre objectif dans le sens où les contributions montraient clairement l'intérêt de voir l'environnement comme une abstraction de premier ordre. Courant 2013 nous avons considéré qu'il était opportun d'organiser à nouveau cet événement afin de faire le point sur les nouvelles directions qui sont liées à cette perspective. Toujours dans le cadre de AAMAS, *E4MAS - 10 Years Later* fut en avril 2014 le quatrième workshop de la série [4]. Danny Weyns et moi-même compilons actuellement les contributions issues de cette édition afin qu'elles paraissent dans un nouveau volume. En particulier, cette édition a abordé des sujets tels que les environnements large échelle, la distinction esprit/corps ou encore l'introduction de l'humain dans la boucle.

Dans les sections qui suivent, nous revenons plus spécifiquement sur quelques-unes de nos contributions sous-tendues par une perspective E4MAS.

## 3.2 Modèle 3-tiers de l'environnement dans les SMA

Dans le cadre de la première édition de E4MAS, nous avons collaboré avec Danny Weyns, Van Parunak, Tom Holvoet et Jacques Ferber en vue d'établir un état de l'art sur la manière dont le concept d'environnement avait été envisagé dans les SMA [Weyns et al., 2005b]. L'une des contributions de l'article consiste dans une analyse des différentes approches alors dispo-

3. Au sens d'un composant logiciel fournissant une abstraction dont les mécanismes internes sont encapsulés de telle sorte qu'une modification de ce composant peut se faire sans qu'il soit nécessaire de modifier les autres composants du système.
4. Cette édition a vu le comité d'organisation renforcé par d'autres chercheurs : Olivier Boissier, Alessandro Ricci et Michael Schumacher.

nibles (modèles, implémentations et méthodologies AOSE). Cette partie avait abouti à deux principales conclusions. Premièrement, elle confirmait que l'environnement n'avait pas de rôle prépondérant en AOSE, aucune dynamique ne lui était généralement associée. Deuxièmement, elle mettait en évidence que le terme environnement recouvrait implicitement un très grand nombre de fonctionnalités différentes (communication, interaction, structure organisationnelle, infrastructure logicielle ou matérielle, etc.). Par conséquent, il était avant tout nécessaire de préciser les différents aspects que pouvait recouvrir le concept d'environnement dans les SMA.

De par sa réception dans la communauté, l'une des contributions majeures de cet article repose sur un modèle qui distingue trois niveaux d'environnements (figure 3.1) :

— *MAS application* : l'environnement applicatif du SMA, c'est-à-dire le modèle du monde, physique et/ou virtuel (e.g. une organisation), dans lequel les agents évoluent. Celui-ci est supporté par un cadriciel SMA (*MAS framework*) qui fournit les abstractions dont le développeur a besoin pour définir et faire fonctionner l'environnement applicatif (infrastructure pour les communications, modèle d'environnement physique, etc.)
— *Execution platform* : l'environnement d'exécution, qui fait référence aux possibilités offertes par le logiciel sur lequel tourne le cadriciel SMA. Un *agent Java* a par exemple potentiellement accès à toutes les fonctionnalités du langage.
— *Physical infrastructure* : l'infrastructure physique utilisée pour le déploiement (ressources réseaux, matérielles et computationnelles).

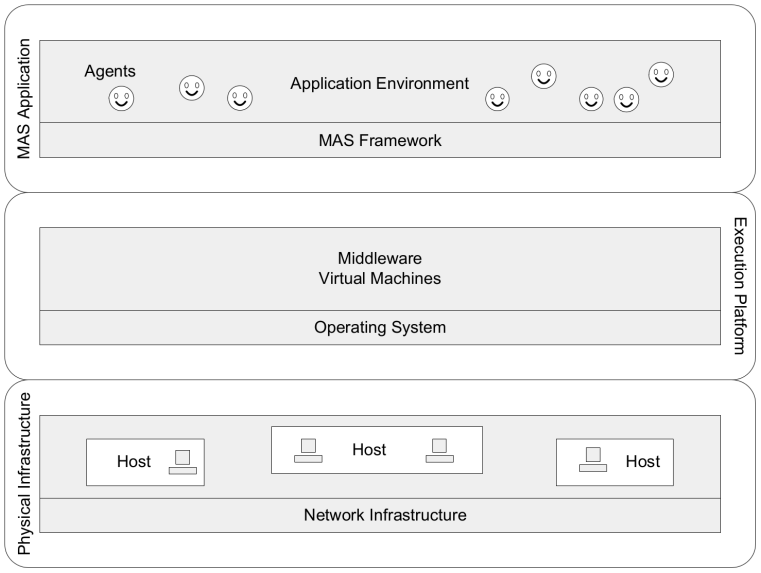



FIG. 3.1 – Modèle 3-tiers de l'environnement dans les SMA [Weyns et al., 2005b]

L'objectif de ce modèle n'est pas de dire que tous ces aspects doivent systématiquement être traités lorsqu'on conçoit un SMA. Il n'a d'ailleurs pas été reçu comme tel par la communauté. Par contre, il permet d'expliciter différentes problématiques suivant les niveaux (modélisation, outils logiciels, déploiement, etc.) qui, si elles sont prises comme un tout, peuvent apporter

beaucoup de confusion lors de la conception. De plus, cette représentation permet d'aborder des problématiques plus techniques telles que la mise en correspondance du niveau applicatif avec l'environnement d'exécution ou l'infrastructure physique (e.g. la distribution des agents en fonction du nombre de cœurs, un agent ou une plate-forme par site dans un réseau, etc.).

Plus généralement, cet article promeut l'adoption d'une perspective unificatrice où, quel que soit le cadre applicatif, quelle que soit la nature des agents, ces derniers peuvent être, et donc doivent être, considérés comme situés dans un environnement. Une telle perspective permet d'envisager que tout SMA peut potentiellement profiter de dynamiques implémentées dans l'environnement, pas seulement les SMA inspirés du vivant : ***tout agent est situé***.

## 3.3 Le modèle AGRE

En accord avec l'idée précédente, envisager la structure organisationnelle d'un SMA comme une forme d'environnement dans lequel les agents sont situés est immédiat. Il est par exemple pertinent d'assimiler les notions de groupe et de rôle, c'est-à-dire la position sociale de l'agent, à des coordonnées au sein d'une structure organisationnelle. Par exemple, Mathieu et al. [2014] montrent qu'il est possible d'utiliser un unique formalisme pour représenter à la fois la localisation spatiale d'un agent dans un monde physique et sa position dans une organisation. Ce qui permet d'unifier les deux points de vue.

Dans la même perspective, avec Jacques Ferber et José Báez, nous avons travaillé sur une extension du modèle Agent/Groupe/Rôle afin d'intégrer le concept d'environnement d'une manière générique dans celui-ci [Ferber et al., 2005].

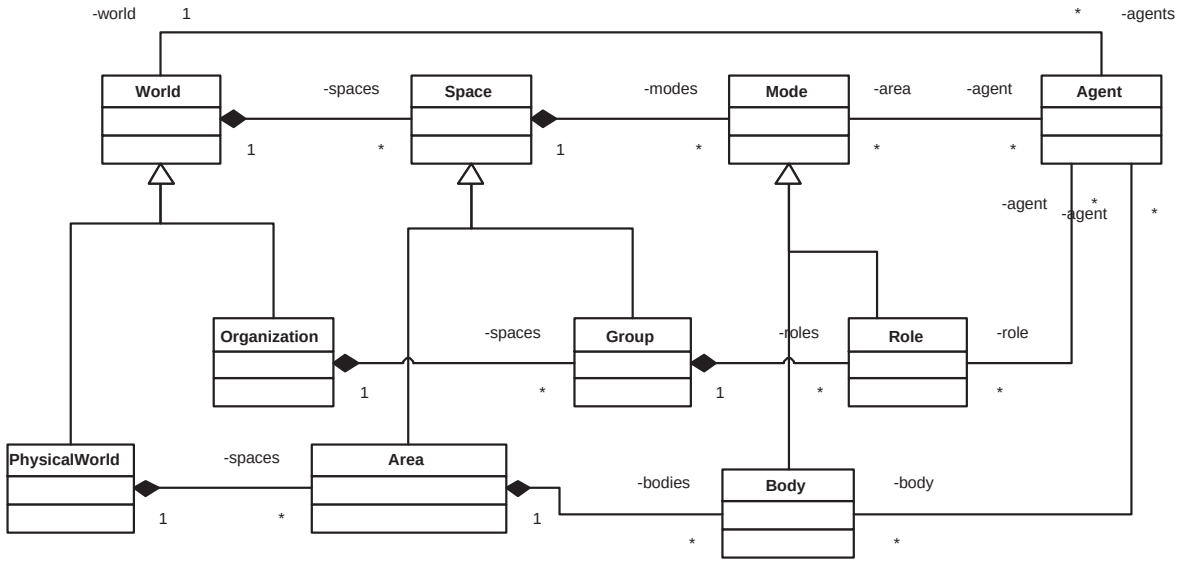



Fig. 3.2 – Diagramme UML du modèle AGRE

Sur la figure 3.2, on voit que le modèle AGRE (Agent/Groupe/Rôle/Environnement), intègre les environnements physiques et sociaux en les réifiant comme des sous classes de `Space` : `Group` pour les espaces sociaux et `Area` pour ceux inclus dans un monde physique. Le monde (`World`) est ainsi défini comme une agrégation d'espaces définissant des organisations ou des mondes physiques.

Les agents agissent dans ces espaces via des sous classes de `Mode` qui définissent la manière dont ils existent et interagissent dans un espace : via un `Role` pour un environnement social et via un `Body` dans un espace physique.

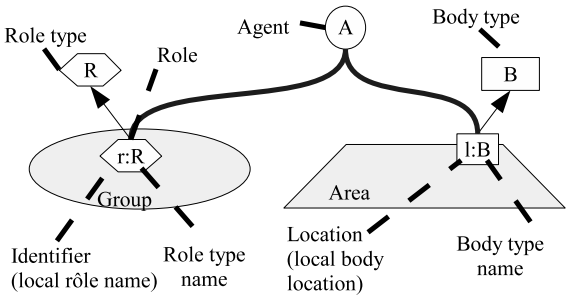



FIG. 3.3 – `Agent`, `Area` et `Group`

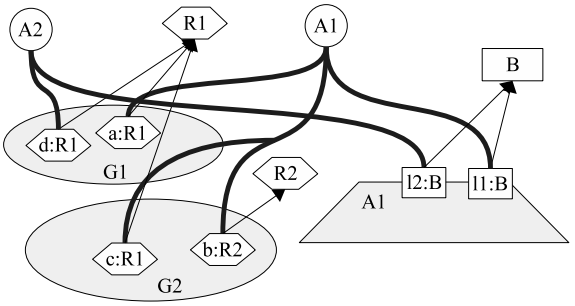



FIG. 3.4 – AGRE : exemple avec 2 agents

Techniquement, le code d'un agent consiste à récupérer son corps ou son rôle afin de pouvoir utiliser les primitives liées à l'environnement correspondant, par exemple pour effectuer des envois de messages dans un groupe grâce à un rôle, ou bien des mouvements dans un espace physique grâce à un corps. Voici une illustration du code correspondant :

```
// soit un un Agent a, une Organisation o et un PhisicalWorld pw
Role r = o.requestRole(GroupName, RoleType, RoleName, a);
r.sendMessage(RoleName,Message);
Body b = pw.requestBody(AreaName, BodyType, Location, a);
b.move(30,10);
```

Malgré le fait qu'il soit régulièrement référencé, le modèle lui-même n'a que très peu été repris [5]. En fait, sur un plan conceptuel, ce modèle contient quelques points difficiles qui nous apparaissent plus clairement avec le recul. La difficulté principale est en quelque sorte contenue dans le nom même du modèle. AGRE, de par sa dénomination, met sur un pied d'égalité des concepts organisationnels et la notion d'environnement. Cela alors que l'idée fondamentale sous-tendue par l'approche est qu'un espace social n'est qu'une sorte d'environnement parmi d'autres. Dans AGRE, cette idée est effectivement mise en pratique car un agent agit dans les deux types d'environnement grâce à un concept unique (`Mode`). Cependant, le vocabulaire employé peut laisser croire que l'unification n'est pas vraiment réalisée. En effet, `Body` n'est utilisé que pour accéder à l'environnement physique, alors que, dans l'*inconscient collectif* de

5. Notons que AGRE a été revisité dans [Báez-Barranco et al., 2007]. Le modèle AGREEN généralise certains points et en introduit d'autres (e.g. institution), notamment à travers des réflexions basées sur IR.

la communauté, *body* est un concept fortement corrélé à la notion d'environnement situé au sens large, c'est-à-dire au sens où nous l'entendions dans le cadre du modèle IRM4S (section 2.2). Autrement dit, il aurait été sans doute plus pertinent d'utiliser `Body` à un plus haut niveau, comme mère de tous les moyens d'accès aux différents environnements qui peuvent être envisagés. Sans cela, les deux types d'environnements envisagés apparaissent effectivement comme n'étant pas similaires, ni unifiés, principale critique adressée au modèle.

## 3.4 Intérêt d'une vision unifiée de l'environnement

Des sections précédentes, on voit l'intérêt pratique, voire la nécessité conceptuelle, de considérer l'environnement global d'un agent comme la composition d'un ensemble d'espaces possédant chacun leurs propres caractéristiques. En premier lieu, cela permet de séparer les différentes problématiques qui peuvent être rencontrées d'une manière claire (organisation, modélisation de mondes physiques, gestion des ressources matérielles, déploiement, etc.).

Dans le même temps, pour promouvoir cette idée, nous avons vu qu'il est important d'avoir une vue unifiée sur la manière dont un agent interagit avec différentes formes d'environnement. Dans cette optique, dans [Gouaïch et Michel, 2005] nous montrons l'intérêt de réifier les moyens de perceptions et d'actions d'un agent dans la modélisation de l'environnement.

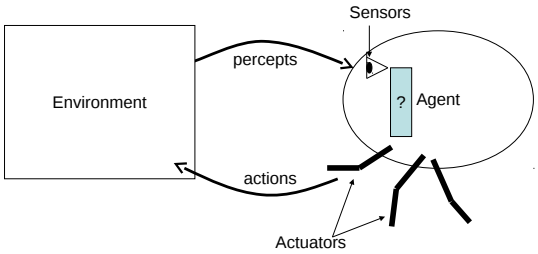



FIG. 3.5 – [Russell et Norvig, 1995]

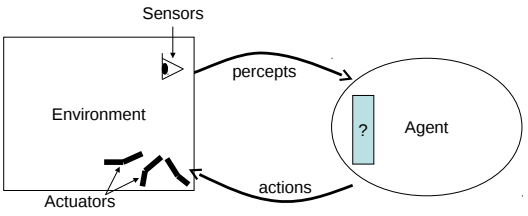



FIG. 3.6 – [Gouaïch et Michel, 2005]

La figure 3.5 illustre la représentation classique de la relation entre un agent et son environnement, telle qu'elle est proposée dans [Russell et Norvig, 1995]. On y voit que les moyens de perceptions et d'actions sont explicités, mais définis du côté de l'agent. De fait, la modélisation de l'environnement est très fortement dépendante du modèle agent utilisé.

Déplacer les moyens de perception et d'action dans le modèle de l'environnement permet de briser cette dépendance (figure 3.6). De là, on peut définir un environnement sans présupposer de l'architecture des agents qui l'utiliseront. De plus, il devient alors possible d'envisager qu'un agent interagit avec plusieurs environnements sans repenser leur modélisation (figure 3.7).

Cette vision permet aussi de considérer la composition d'environnements suggérée par l'approche 3-tiers (section 3.2), dans laquelle un agent accède à des ressources de l'infrastructure physique, soit directement, soit par l'intermédiaire de l'environnement applicatif (figure 3.8).

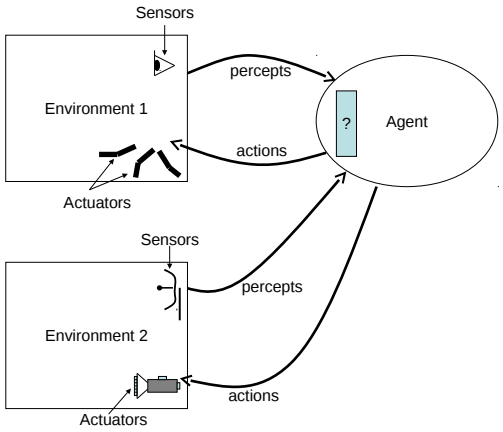



FIG. 3.7 – Multiplication de la relation entre agent et environnement

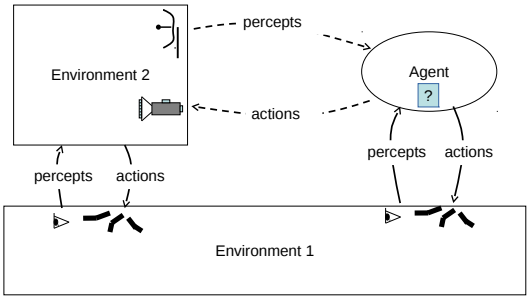



FIG. 3.8 – Composition d'environnements

Cette approche rend aussi explicite le fait qu'il est de la responsabilité de l'environnement de définir la manière dont il est accédé, ainsi que les possibilités d'actions qu'il offre. De plus, la distinction entre l'esprit et le corps de l'agent devient elle aussi explicite : le corps de l'agent (effecteurs et senseurs) fait partie intégrante de l'environnement et l'agent n'en a pas la maîtrise, seul le processus délibératif est sous son contrôle. Dans [Gouaïch et Michel, 2005], cela est illustré à l'aide du modèle $MIC^*$ [Gouaïch, 2005]. Dans ce modèle, le concept d'*objet d'interaction* est utilisé pour réifier les moyens de perceptions et d'actions dans l'environnement, lui-même découpé en *espaces d'interactions* (composition d'environnements) dans lesquels les objets d'interactions prennent place. [Gouaïch et al., 2005] montre par ailleurs comment $MIC^*$ peut aussi être utilisé pour modéliser un environnement social, là encore en utilisant les objets d'interactions, ce qui montre la généricité de cette approche.

Sur cette idée, les travaux de Jean-Christophe Soulié, effectués dans le cadre de la simulation multi-agent, sont précurseurs [Soulié, 2001] : les moyens d'action et de perception sont réifiés dans l'environnement pour permettre une approche multi-environnements (approche depuis formalisée en DEVS [Soulié, 2012]).

En fait, beaucoup des recherches portant sur le concept d'environnement font, d'une manière ou d'une autre, le même constat : déplacer les moyens d'actions et de perceptions dans l'environnement permet généricité (relation agent/environnement unifiée), modularité (composition d'environnements), extensibilité (construction itérative du système) et réutilisabilité (séparation claire entre problématiques agent et environnement).

Dans le cadre de l'AOSE, l'une des approches les plus abouties est celle des *artifacts* (*Agents & Artifacts*, A&A), élaborée par Alessandro Ricci et ses collègues depuis plus d'une dizaine d'années [Ricci et al., 2011]. L'environnement y est vu comme une collection dynamique d'*artifacts* qui définissent chacun des ressources et des outils que les agents utilisent pour accomplir diverses tâches. Les auteurs parlent d'*environnement programming* : concevoir les artifacts devient aussi important, si ce n'est plus, que d'élaborer les agents (voir la plate-forme

CArtAgO [Ricci et al., 2007b]). Ainsi dans [Ricci et al., 2011], l'idée majeure de l'approche est libellée de la façon suivante :

*programming MAS = programming Agents + programming Environment*

Cette dernière citation permet de mesurer le chemin parcouru depuis les premiers workshops E4MAS. En particulier, le rapprochement entre IAD et vie artificielle que Parunak, comme beaucoup d'autres chercheurs, appelait de ses vœux est devenu beaucoup plus palpable. Bien sûr, cette vision n'est pas unanimement partagée par la communauté [6]. Mais, pour nous, elle est fondatrice. Programmer l'environnement ne doit pas se résumer à modéliser l'espace dans lequel les agents évoluent, comme s'il n'était qu'une conséquence de leur existence. Au contraire, il doit être vu comme une partie de la solution, c'est-à-dire comme un élément dont la dynamique permet au système de remplir sa fonction.

Nous synthétisons ici notre vision des principaux aspects que nous avons évoqués dans ce chapitre.

— Tout d'abord, il n'y a pas d'agent sans environnement. Si cela peut paraître évident d'un point de vue conceptuel, l'environnement n'est pas toujours réifié dans le code, même pour des SMA où les agents sont spatialement localisés [7]. La réification de l'environnement est toujours pertinente, que le SMA soit ancré dans un monde physique ou composé d'agents purement communicants.
— L'environnement est multiple. Il peut notamment être physique et/ou social. Il est donc important de distinguer les différents aspects que l'on peut identifier pour le fonctionnement du SMA cible, en particulier afin de les séparer clairement dans l'implémentation, par exemple avec une approche de type *artifacts*.
— L'environnement est le lieu des perceptions, des actions et des interactions, il peut les catalyser et/ou les contraindre. Sa dynamique (réaction) est essentielle.
— La réification des moyens de perceptions et d'actions de l'agent, dans l'environnement, facilite la modularité, et donc la réutilisabilité.

## 3.5 Leçon majeure : déplacer la complexité

Au-delà des avantages dont nous avons discuté, notamment en lien avec le génie logiciel (auxquels nous sommes très sensible), nous tirons une leçon majeure de tous ces travaux. Dans la préface du troisième volume [Weyns et al., 2007b], nous mettions nous-même en exergue un point essentiel qui était, implicitement ou explicitement, présent dans *tous* les travaux

6. D'ailleurs, quel que soit le concept, *agent*, *environnement*, *interaction* ou *organisation*, rares sont les interprétations qui font consensus dans la communauté multi-agent. Ce qui est compréhensible étant donné le large spectre des recherches qu'elle couvre.

7. Il est tout à fait possible de faire une simulation où des agents se déplacent dans un espace en utilisant uniquement des attributs (x,y) qui leur sont propres. L'environnement est alors visible au niveau de l'affichage mais il n'est nullement réifié en tant que tel.

centrés sur l'environnement : **déplacer la complexité** du comportement **des agents vers l'environnement**, permet de bien mieux la gérer.

Parce qu'ils ont été validés dans un contexte industriel, les travaux de Danny Weyns sur la réalisation de véhicules automatisés (*Automated Guided Vehicles* AGV) sont emblématiques de cette perspective [8]. Dans [Weyns et al., 2006a], les AGV utilisent un environnement virtuel chargé de valider leurs mouvements futurs. Lorsque l'environnement détecte une possible collision à venir, il résout automatiquement les conflits spatiaux, sans que les agents n'aient besoin d'intervenir. Les agents n'ont donc pas besoin de traiter ce problème, ce qui permet de (1) diminuer la complexité de leur comportement et de (2) faire en sorte qu'ils se focalisent sur leur tâche principale, aller d'un point A à un point B.

C'est pourquoi, sans même parler de l'intérêt de modéliser des dynamiques environnementales, notamment celles qui catalysent la stigmergie, déplacer la complexité permet de ne pas affubler le comportement des agents (au final du code) avec des traitements qui peuvent avantageusement être effectués par ailleurs, dans l'environnement. Par exemple, dans [Dinont, 2007], dans le contexte d'architectures cognitives, l'introduction d'un "maximum d'artifcats"est vue comme un moyen de limiter la complexité de chacune des entités. Dans le cadre de la simulation multi-agents, on retrouve cette idée dans [Payet et al., 2006b] où il est précisément question de déléguer une partie des processus agents dans l'environnement afin de réduire la complexité du modèle, et ainsi permettre la réutilisabilité et l'intégration des différents processus définis au niveau de l'environnement. Toujours en simulation multi-agents, EASS (filtrage de perceptions) [Badeig et Balbo, 2012 ; Balbo et al., 2013] est un autre exemple d'approche environnement-centrée qui vise explicitement à déplacer la complexité du comportement des agents dans l'environnement.

Les applications présentées dans les prochains chapitres, même si elles n'ont pas été spécifiquement centrées autour de cette idée, reposent toutes sur des solutions où la modélisation de l'environnement est primordiale, que ce dernier soit physique, social ou les deux. C'est à travers ces applications que ces idées ont pris pour nous plus d'évidence.

8. Voir aussi les travaux sur les *delegate MAS* [Holvoet et al., 2009], les *polyagents* [Parunak et Brueckner, 2007] et récemment l'approche multi-environnements de [Galland et al., 2014].

# Chapitre 4

# Segmentation d'image par SMA

Lors de notre arrivée au CReSTIC de Reims en 2005, nous avons intégré le groupe SIC (Signal, Image et Connaissance). Celui-ci possède des compétences à la fois dans le domaine de l'IA mais aussi autour du signal et plus particulièrement du Traitement Numérique de l'Image (TNI). Suite à notre intégration, nous nous sommes donc naturellement rapproché de cette thématique. Dans ce cadre, nous avons eu l'opportunité de participer à l'encadrement de deux doctorants sous la direction de Zahia Guessoum : Jason Mahdjoub [1] et Smaine Mazouzi [2].

Les travaux réalisés dans ces thèses se placent dans le domaine de la vision artificielle et abordent en particulier des problématiques liées à la reconnaissance et à l'interprétation de formes. Ceux de Jason Mahdjoub ont eu pour contexte l'automatisation de la reconnaissance de pathologies pour l'aide à la décision médicale à partir d'images issues de scanner. Ceux de Smaine Mazouzi se sont focalisés sur un type d'image appelé *image de profondeur*, notamment utilisé pour l'interprétation de scènes naturelles (e.g. en robotique).

Dans les deux cas, ces travaux reposent sur une approche ascendante et utilisent les SMA pour explorer des alternatives aux techniques TNI classiques. Dans ce chapitre, nous nous focalisons sur la manière dont ces travaux ont abordé la question de la segmentation d'une image, partie où nous sommes le plus intervenu en terme d'encadrement.

## 4.1 Difficultés d'adaptation des méthodes TNI classiques

Segmenter [3] automatiquement une image est une étape fondamentale pour beaucoup de contextes applicatifs liés au TNI, notamment en vision artificielle où elle sert à détecter les objets d'une scène (routes dans une image satellite, visages/corps dans une foule, tumeur dans une radiographie, etc.). Une des difficultés du domaine tient dans l'évaluation objective

1. *Vers un système de vision auto-adaptatif à base de systèmes multi-agents* [Mahdjoub, 2011].
2. *Reconnaissance de formes par les systèmes auto-organisés. Application aux images de profondeur* [Mazouzi, 2008].
3. La segmentation correspond à l'extraction d'informations relatives aux différents contours et régions d'une image étant donné un critère (e.g. une forte variation des couleurs).

de la qualité d'une technique de segmentation [Estrada et Jepson, 2009 ; Unnikrishnan et al., 2007]. En particulier, il n'existe pas à proprement parler de solution optimale, même si elle est produite par l'homme [D. Martin et al., 2001]. Par exemple, pour une même image satellite, segmenter les routes n'a rien à voir avec délimiter un feu de forêt : il existe une bonne segmentation pour chaque cas, mais elles sont très différentes l'une de l'autre. Ainsi, l'évaluation d'une technique est souvent mise en rapport avec l'usage [Borra et Sarkar, 1997]. Une segmentation sera jugée comme qualitativement performante si elle facilite effectivement l'application de procédures de plus haut niveau (e.g. la reconnaissance de formes en vision artificielle). D'autres approches reposent sur l'idée qu'il est pertinent de comparer quantitativement les différentes techniques en les confrontant à une *vérité terrain* établie manuellement par l'homme, en dehors de tout contexte applicatif [4].

Ce qui ressort clairement de ces études, c'est qu'il n'existe pas de technique qui puisse être qualifiée d'optimale. Une technique peut être efficace sur une image et déficiente sur une autre. En d'autres termes, les différentes approches ont du mal à s'adapter à la multiplicité des contextes (type d'image + utilisation) [5]. Comme le discute Jason Mahdjoub dans sa thèse, cette difficulté des outils traditionnels à s'adapter à différents contextes peut être corrélée au fait qu'ils reposent sur des approches supervisées.

Les algorithmes TNI habituels utilisent majoritairement des opérateurs qui agissent de manière globale sur l'image. Par conséquent, ces opérateurs s'accommodent difficilement des spécificités locales que les différentes zones d'une image peuvent avoir. De fait, il est ensuite nécessaire de faire appel à des heuristiques afin de produire un résultat acceptable. Les algorithmes utilisés sont souvent très sophistiqués. En effet, ils sont eux aussi appliqués sur la totalité de l'image et doivent donc composer localement avec beaucoup de cas particuliers. Par exemple, pour décider si un pixel fait partie d'un contour, un critère de sélection peut s'avérer pertinent à un endroit de l'image et mauvais à un autre.

Ces heuristiques embarquent notamment des raisonnements de haut niveau (classification, apprentissage, etc.) qui impliquent des décisions basées sur un nombre de paramètres parfois très conséquent. C'est pourquoi elles traitent certains cas mieux que d'autres : elles ont été spécialisées par leurs concepteurs. D'ailleurs, pour la vision artificielle, il est difficile de se reposer sur une unique approche et certains travaux de recherche traitent de la sélection automatique, en ligne, du meilleur algorithme parmi un ensemble de plusieurs solutions, en fonction de leur performance sur l'instance courante [V. Martin, 2007].

4. Voir la très utilisée *Berkeley Segmentation Database*(BSD) [D. Martin et al., 2001]. Son intérêt est d'avoir été élaborée grâce à un protocole tel que chaque image a été segmentée par au moins quatre personnes. On a donc plusieurs segmentations correctes pour une image, et non pas une seule à laquelle on devrait systématiquement se référer. Sur cette base, des métriques permettent d'évaluer la qualité d'une segmentation, et donc de comparer les différents algorithmes objectivement [Estrada et Jepson, 2009].

5. On peut faire un lien avec la difficulté de mesurer la qualité d'une IA : l'adaptation et la résistance aux variations de l'environnement est un point important [Hernández-Orallo, 2014].

## 4.2 L'image : un environnement pour SMA

L'absence de solution optimale et la difficulté d'adaptation des méthodes classiques sont des arguments assez forts en faveur d'une approche émergentiste, basée sur des mécanismes non supervisés, où le résultat final est obtenu par l'interaction d'entités qui n'ont pas une vue globale du problème [Gleizes, 2004]. Cela permet de se positionner dans une perspective radicalement différente où la complexité de la tâche n'est pas abordée de manière frontale. Une approche ascendante, par essence non focalisée sur la recherche directe d'une solution optimale, doit notamment permettre d'éviter le piège d'une spécialisation trop forte.

Dans cette perspective, une idée ingénieuse a émergé à la fin des années 90 : utiliser les données de l'image pour définir un environnement concret dans lequel seront simulés des agents de faible grain, mobiles et de petites tailles (généralement un pixel). Cette vision a ouvert la voie a des approches très novatrices en TNI, notamment bio-inspirées : segmentation à l'aide d'une population d'agents évoluant suivant des principes évolutionnaires [Liu et Tang, 1999], reconnaissance de formes grâce à des fourmis virtuelles [Ramos et Almeida, 2000], ou encore détection de régions à l'aide d'agents inspirés par le comportement des araignées sociales [Bourjot et Chevrier, 2001].

Dans ces approches, aucun agent n'a connaissance de l'image dans son ensemble. Les décisions se font sur la base d'une perception locale : le pixel occupé par l'agent ou ses voisins immédiats. De plus, les actions des agents n'ont pas d'effet global et n'affectent que leur environnement immédiat.

## 4.3 Détection et reconstruction de contours par SMA

Dans sa thèse, Jason Mahdjoub a proposé un SMA pour la segmentation d'images médicales [Mahdjoub et al., 2006]. Les expérimentations ont notamment été effectuées sur des images 2D comportant des coupes de poumons humains (figure 4.1). Dans ce contexte, une difficulté importante réside dans la détection des arêtes qui se situent dans des zones où les changements de couleur ne sont pas abrupts mais répartis sur plusieurs pixels.

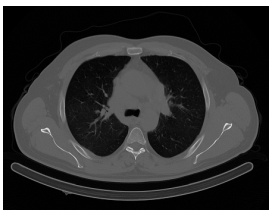


FIG. 4.1 – Coupe 2D d'une image acquise par un scanner médical (poumons)

Dans le même esprit que [Haroun et al., 2005], la démarche consiste à utiliser des agents au niveau microscopique afin d'améliorer les résultats obtenus par des algorithmes de TNI

conventionnels. L'objectif est de lisser les approximations inhérentes à une approche globale. La procédure de détection comporte ainsi deux étapes importantes : (1) la préparation de l'image grâce à des prétraitements TNI classiques, c'est-à-dire la construction de l'environnement des agents et (2) la reconstruction des contours par des agents évoluant au niveau des pixels.

### Prétraitement de l'image : construction de l'environnement

**Nettoyage de l'image** En TNI, il est fréquent d'appliquer, préalablement à tout autre traitement, un premier filtre (opérateur matriciel) pour lisser le bruit présent dans une image. Ici, une variante du filtre de Nagao [Demigny et al., 1993] a été utilisée, car elle a la particularité de réduire le bruit sans détruire l'information qui pourrait correspondre à des contours.

**Détection des contours potentiels** Sur une image classique, un contour est caractérisé par un changement, plus ou moins brusque, de l'intensité lumineuse entre pixels voisins. Ainsi, les filtres TNI qui sont dédiés à la détection de contours consistent dans le calcul du gradient correspondant. La figure 4.2 montre le résultat obtenu grâce au filtre de Kirsch [6]. On voit que, bien que les contours principaux soient détectés, il reste de nombreuses zones de bruit. C'est précisément ces zones que le SMA doit permettre d'éliminer dans un second temps.

**Création d'un champ vectoriel dans l'environnement** Le GVF (Gradient Vector Flow) [Xu et Prince, 1997] est un outil TNI qui permet de créer un champ de vecteurs modélisant une force d'attraction vers les contours potentiels détectés par le premier filtre. La figure 4.3 en montre une représentation pour l'image précédente. L'objectif est d'enrichir l'environnement avec des informations qui vont améliorer les perceptions que les agents ont de l'image, et donc faciliter leur comportement. L'environnement fournit maintenant directement la réponse à la question que tous les agents se posent initialement : *vers où dois-je me déplacer pour trouver des pixels appartenant potentiellement à un contour* ?

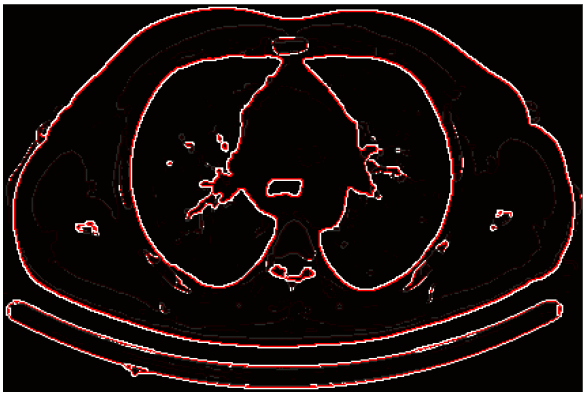


FIG. 4.2 – Image filtrée par Kirsch

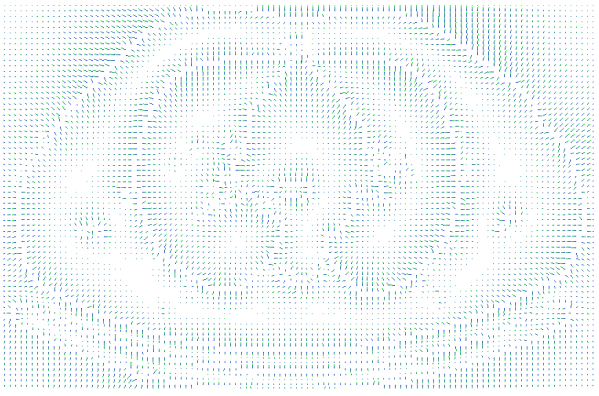


FIG. 4.3 – Gradient construit par le GVF

6. Le filtre de Kirsch est un opérateur assez précis qui calcule le gradient dans les huit directions cardinales afin que la valeur la plus pertinente soit ensuite sélectionnée.

**Marquage de l'environnement** Les deux structures de données précédentes sont uniquement utilisées pour la perception des agents et ne sont pas modifiées au cours du processus de simulation. À celles-ci s'ajoute une troisième structure de données booléennes qui permet aux agents de signifier qu'un pixel a déjà été exploré.

Ce qui est ici remarquable, c'est que l'environnement n'est pas modélisé de façon monolithique, c'est-à-dire par les seules données de l'image initiale. Au contraire, il possède de multiples représentations bien distinctes les unes des autres. On a donc au total trois environnements : (1) l'image traitée avec le filtre de Kirsch, (2) le GVF et (3) la carte d'exploration, les deux premiers pour la perception et le troisième pour l'action.

### Comportements des agents

Les agents utilisés possèdent trois comportements : (1) exploration, (2) suivi de contour et (3) fermeture de contour par négociation. Initialement, tous les agents sont dans un état où ils doivent explorer l'image, grâce au GVF, à la recherche d'un contour potentiel. Lorsqu'un agent en détecte un, il détermine les directions possibles pour ce contour et crée des agents de suivi dont le but est de représenter le contour à l'aide d'une succession de segments.

Dès que la portion du contour qu'un agent explore ne peut plus être approximée par une droite, celui-ci stoppe et devient un agent nœud puis crée un nouvel agent de suivi. Lorsqu'un contour est déjà exploré par un autre agent, une fusion est opérée par négociation. Enfin, dans le cas où un agent de suivi n'a plus aucune direction à suivre, il devient un agent de fin de contour. Ce dernier a pour but d'opérer la fermeture d'un contour en se connectant à son pair le plus proche. La figure 4.4 schématise ces différents comportements sur un unique contour.

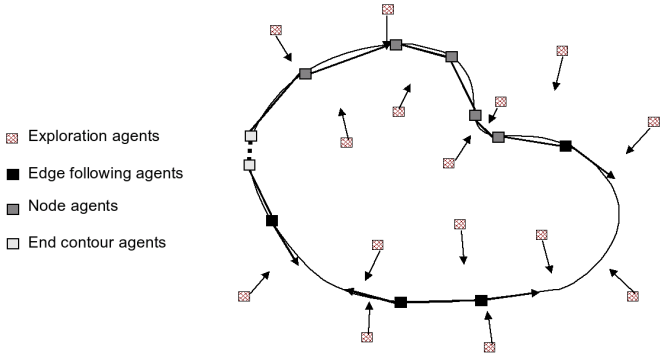



FIG. 4.4 – Vue schématique des différents comportements agents

Comme le montre la figure 4.5, le SMA montre de bonnes performances qualitatives avec une détection des contours jugée correcte et une forte diminution des zones de bruit. Cette figure montre aussi que la qualité du résultat peut être pondérée par la précision utilisée pour la longueur maximale d'un segment [Mahdjoub et al., 2006].

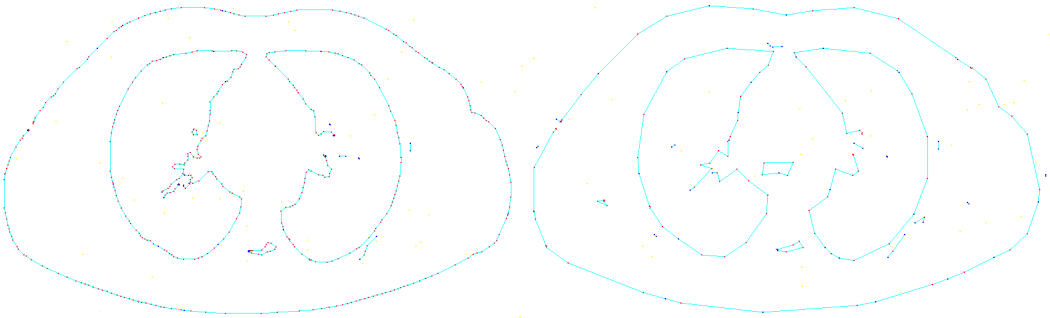


FIG. 4.5 – Résultats de segmentation obtenus avec différents seuils de précision

### Limites de l'approche proposée

Dans une perspective E4MAS, nous pensons qu'une contribution importante de ce travail repose sur la façon dont l'environnement est construit. En particulier, nous souhaitons mettre l'accent sur la construction des environnements à l'aide du prétraitement de l'image. Grâce à ce dernier, les agents travaillent directement sur des données environnementales qui leur permettent d'avoir un comportement plus efficace et, surtout, plus simple. Un tel prétraitement est une opération classique en TNI.

Sans remettre en cause l'idée de prétraitement, on peut cependant critiquer la manière dont celui-ci est effectué. Par rapport aux problématiques soulevées précédemment, on peut considérer que l'utilisation de filtres TNI classiques réintroduit in fine les biais dus à une approche globale. Et, en effet, même si l'utilisation d'agents de niveau microscopique permet de résoudre une partie importante du bruit, ces derniers ne font que suivre les contours détectés par une approche globale et les caractéristiques locales de l'image ne sont finalement pas vraiment prises en compte. Sur cet aspect, une piste intéressante peut être de multiplier les prétraitements, en utilisant d'autres filtres, afin que les agents raisonnent localement sur des données plus contrastées. Cependant, cela ne résout pas vraiment le problème dans la mesure où les agents restent cantonnés à l'utilisation de données calculées de manière globale.

Avec le recul nous pensons que le vrai problème ne se situe pas dans l'utilisation d'outils qui travaillent au niveau global, mais tient au fait que la modélisation de l'environnement est trop pauvre. Notamment, les actions que les agents peuvent effectuer sur l'environnement sont trop limitées. Ceci a deux conséquences importantes. Premièrement, l'espace d'états de la solution est effectivement restreint à des données calculées globalement : les caractéristiques de l'image restent statiques. Deuxièmement, cela élimine toute communication indirecte entre les agents, par l'environnement, et prive donc le système des possibilités offertes par la stigmergie. Ce qui a aussi pour effet de rendre les communications directes plus complexes dans la mesure où elles doivent gérer tous les cas. En particulier, nous n'avons pas détaillé la manière dont les agents communiquent mais la fermeture de contour ainsi que la fusion entre agents impliquent des échanges de messages qui complexifient notablement le comportement des agents.

En résumé, cette approche est trop centrée sur les comportements des agents qui sont par conséquent trop complexes. Le GVF est créé pour enlever une partie de la complexité, mais par la suite aucune dynamique environnementale n'est utilisée pour aider à la résolution de la tâche. Dans l'approche que nous allons maintenant présenter, ce point est traité de manière radicalement différente : aucun agent ne communique directement avec ses pairs et la solution est intégralement construite grâce à des principes stigmergiques.

## 4.4 Segmentation d'images de profondeur par stigmergie

Une image de profondeur se caractérise par le fait que chaque pixel (x,y) est associé à une donnée Z(x,y) correspondant à la distance entre le point d'une scène et le capteur utilisé. Essentiellement utilisées en robotique pour la reconnaissance d'objets tridimensionnels [Park et N. DeSouza, 2005], ces images sont connues pour être fortement bruitées et difficiles à segmenter [Hoover et al., 1996]. Comme pour le SMA précédent, l'objectif est de réaliser la segmentation à l'aide d'agents réactifs évoluant au niveau microscopique. Cependant, les agents vont cette fois avoir l'opportunité de modifier directement les données de l'image.

### Modélisation et construction de l'environnement

**Représentation des données de l'image** Deux filtres (gaussien et médian) sont d'abord appliqués afin de réduire le bruit. Un calcul est ensuite effectué afin d'associer à chaque pixel les paramètres (a, b,c et d) de l'équation du plan tangent à la surface en (x,y). Tout l'enjeu de la segmentation consiste à lisser les données de l'image de manière à ce qu'elles ne contiennent plus que quelques plans, c'est-à-dire les surfaces des objets présents dans la scène.

**Marquage de l'environnement** Ici encore, une structure de données supplémentaire est utilisée pour stocker l'état du pixel : lissé (équation de plan lissée à l'intérieur d'une région), aligné (équation de plan modifiée sur les valeurs d'une autre région) ou inchangé. Dans ce modèle les agents évoluent donc dans deux environnements simultanément. Par rapport au modèle précédent, la différence majeure est la capacité que les agents ont de pouvoir influencer l'ensemble des variables environnementales. En particulier, les agents vont agir en modifiant directement les paramètres des équations de plan.

### Comportement des agents

Afin de lisser et d'aligner les pixels de l'image, le comportement d'un agent est principalement basé sur deux attributs : (1) son équation de plan et (2) sa capacité à altérer un pixel, un booléen noté $C_t$.

Initialement, l'équation de plan d'un agent n'a pas de valeur : celui-ci doit d'abord s'*adapter* à une région. Pour ce faire, il mémorise les $L$ derniers pixels visités afin de tester s'il se trouve dans une région plane. Ce qui intervient lorsque $L$ pixels possèdent des équations de plan jugées proches, selon un seuil. Si le test est positif, l'agent mémorise une équation plan, pondérée par les valeurs observées, et acquiert la capacité d'altérer les pixels de sa région : il y est adapté.

Commence alors pour l'agent une phase de lissage. Pour chaque pixel rencontré, l'agent modifie à la fois son équation de plan et celle du pixel, en fonction d'une moyenne pondérée par la distance parcourue par l'agent dans sa région : plus un agent évolue dans une même région, plus son équation de plan prévaut. En quelque sorte, l'agent accumule de l'énergie au fur et à mesure de son déplacement dans une même région.

Enfin, lorsqu'un agent rencontre un pixel jugé comme faisant partie d'une autre région, il l'aligne sur sa propre région suivant un coefficient d'altération prédéfini noté $\xi$ [7]. En contrepartie, il perd sa capacité d'altération et commence un nouveau processus d'adaptation à une région. La figure 4.6 schématise le comportement global correspondant.

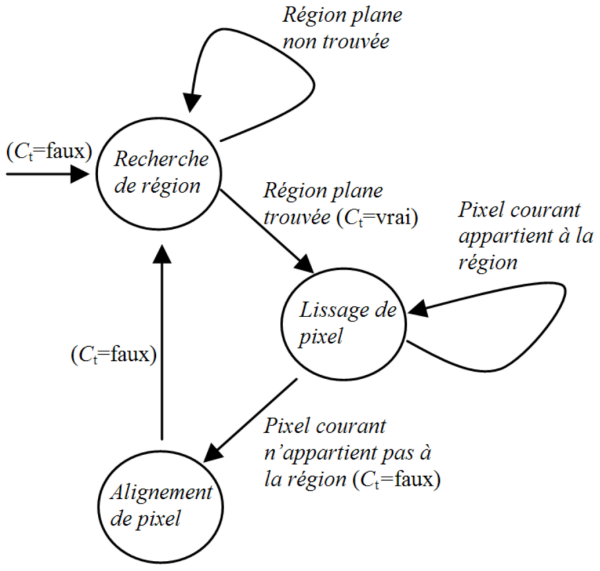



FIG. 4.6 – Comportement agent pour la segmentation d'une image de profondeur

La manière dont la segmentation est obtenue au niveau global repose sur trois phénomènes : (1) le lissage des régions, c'est-à-dire l'alignement, pour une région, des paramètres des plans sur des valeurs équivalentes, (2) l'élimination des zones de bruit pour lesquelles les agents n'ont pas le temps de s'adapter et (3) la compétition entre agents qui prend place par émergence au niveau des contours, c'est-à-dire le fait que les pixels se trouvant entre deux plans distincts

7. La force d'altération $\xi$ joue un rôle important dans le résultat. Plus cette force est élevée, plus la segmentation se fait rapidement, au détriment de la qualité. Cette possibilité d'ajuster le temps de résolution est une caractéristique très intéressante, notamment dans un contexte temps réel. On peut par exemple imaginer modifier ce paramètre en fonction des ressources matérielles disponibles et des contraintes temporelles. Il y a ici un lien évident avec l'approche par niveaux de détail (LOD) présentée dans le chapitre suivant.

vont être tour à tour alignés sur des paramètres très éloignés, de telle sorte qu'ils restent distingués dans le résultat final [Mazouzi et al., 2008b].

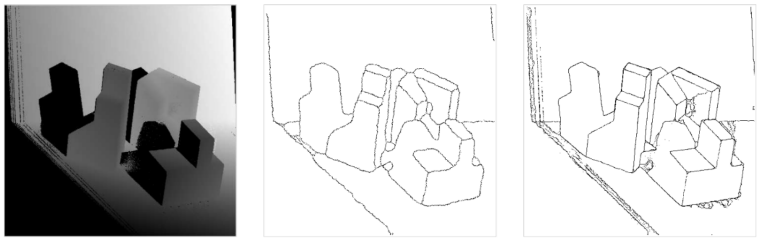
### Ajout d'une dynamique environnementale de type champ de potentiel

Inspirée des nombreux travaux ayant montré l'intérêt des champs de potentiels dans l'environnement (e.g. [Mamei et Zambonelli, 2005 ; Simonin, 2010]), une approche similaire a été utilisée pour catalyser l'émergence des phénomènes d'élimination du bruit et de détection de contours. Fondamentalement, l'objectif est le même qu'avec le GVF dans le modèle précédent : orienter les agents vers les points d'intérêt, ici les pixels de contour ou de bruit.

Mais à la grande différence de l'approche précédente, où le GVF résulte d'un calcul global, ce sont cette fois les agents qui vont, par leurs actions au niveau microscopique, créer des champs de potentiels locaux. Plus précisément, et la distinction est importante, lorsqu'un pixel est aligné par un agent, c'est la dynamique de l'environnement qui répond à cette action par la création d'un champ de potentiel qui attire les autres agents vers cet endroit.

De plus, l'environnement possède une autre dynamique endogène qui réduit l'influence du champ de potentiel au fur et à mesure des altérations. Plus le nombre de modifications sur un pixel est important, moins le champ de potentiel correspondant est fort. Si bien que la fin de la segmentation est détectée automatiquement au niveau de l'environnement : elle correspond à la disparition totale de tous les champs de potentiel.

### Résultats sur des objets polyédriques



(a) (b) (c)

FIG. 4.7 – (a) Image de profondeur rendue réaliste ; (b) et (c) segmentations pour différentes valeurs de $\xi$

La figure 4.7 présente un exemple de segmentation sur des objets polyédriques. Ce modèle a été évalué en utilisant le framework proposé dans [Hoover et al., 1996] [8]. Comparés à quatre autres méthodes de segmentation publiées dans la littérature, les résultats obtenus sont au

8. Celui-ci contient une base (ABW) contenant 40 images de profondeur et définit un ensemble de métriques qui permettent des comparaisons par rapport à une réalité terrain déterminée manuellement.

moins équivalents voire meilleurs, à la fois en ce qui concerne la qualité de la segmentation et le temps d'exécution [9] [Mazouzi et al., 2009b].

### Extension aux objets courbés

La plupart des approches se limitent exclusivement aux images contenant des objets polyédriques. En fait, les méthodes proposées dans la littérature supportent assez mal l'introduction d'objets courbés dans la scène [Jiang et Bunke, 1999]. La segmentation est très ardue à cause d'une importante déformation des surfaces dans les données brutes récupérées.

Pour que le SMA puisse fonctionner sur des images contenant des objets de toutes formes, seules deux modifications ont été nécessaires. La première consiste à modifier le critère d'homogénéité d'une région (plan) afin de l'étendre aux surfaces courbes [10]. La deuxième concerne la manière dont les agents lissent les pixels. Au lieu d'opérer sur un seul élément, le lissage est cette fois effectué en considérant le pixel courant et tous ses voisins. Ce qui revient à utiliser un filtre de 3x3 local qui permet de mieux prendre en compte la courbure d'une région. L'alignement reste effectué sur un seul pixel, car supposé faisant partie d'un contour.

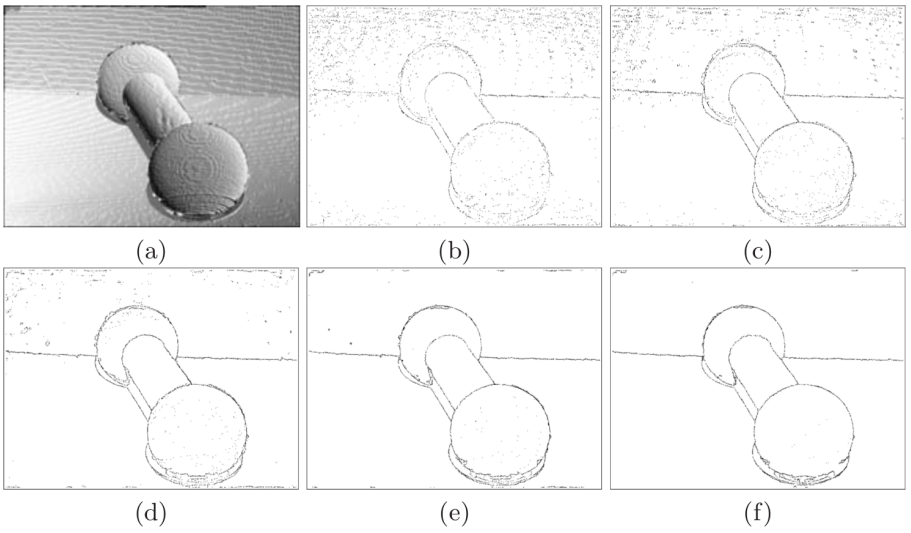



FIG. 4.8 – (a) Image de profondeur rendue réaliste ; segmentation à (b) t=500, (c) t=3500 ; (d) t=7500 ; (e) t=10500 ; (f) t=13500. [Mazouzi et al., 2008a]

La figure 4.8 montre le résultat de la segmentation à différents stades du processus de simulation. Cette deuxième version a été testée avec le framework précédent sur les bibliothèques ABW et K2T (objets courbés). Les résultats obtenus pour les objets polyédriques

9. Comme nous l'avons déjà souligné, il s'agit bien sûr d'une qualité importante, notamment si on se place dans le contexte de la vision artificielle où le temps réel peut être une contrainte forte.

10. Les équations de plan sont remplacées par le calcul de deux valeurs, Ch et Cv, représentant les courbures directionnelles respectivement selon un plan horizontal et un plan vertical. Ainsi, un ensemble de pixels contigus est considéré comme faisant partie d'une même région si les courbures sont équivalentes étant donné un seuil.

sont restés équivalents voire meilleurs. Pour K2T, seule une autre méthode était disponible pour la comparaison : le SMA est légèrement meilleur en moyenne [Mazouzi et al., 2009a].

L'absence de solution optimale et la difficulté d'adaptation des méthodes supervisées sont des arguments assez forts en faveur de solutions ascendantes. De ce point de vue-là, les bons résultats obtenus, en termes à la fois de qualité et de performance, sont encourageants. Mais le plus remarquable reste la facilité avec laquelle le SMA initial a pu être étendu aux objets courbés. Contrairement aux approches classiques qui supportent mal les changements de contexte, le SMA s'avère plus flexible. Cette caractéristique est sans aucun doute liée aux bénéfices d'une approche de résolution par émergence [Gleizes, 2004]. Le comportement des agents ne fait aucune supposition sur la nature de l'image, ni n'utilise de données globales, ce qui le rend très générique et permet à l'approche plus de robustesse. On a donc une vraie rupture avec les algorithmes supervisés qui reposent, eux, principalement sur l'intelligence de leurs concepteurs.

## 4.5 Perspective E4MAS

Nous avons peu d'expérience en ce qui concerne la résolution de problèmes complexes par auto-organisation de SMA, c'est pourquoi cette interaction avec le TNI fut très enrichissante. Dans une perspective E4MAS, elle montre une fois de plus l'importance du couplage entre dynamiques environnementales et comportements individuels : l'environnement est au cœur de l'ingénierie des systèmes auto-organisés et auto-adaptatifs [Hassas, 2006].

Le dernier modèle illustre par ailleurs l'intérêt de déplacer la complexité des agents vers l'environnement. En effet, abstraction faite des changements liés au critère d'homogénéité et à la méthode de lissage, il faut remarquer que le comportement d'un agent est resté strictement le même, celui présenté figure 4.6. C'est donc le changement de nature des données environnementales qui permet à ce modèle de s'exporter sur différents types d'images, et non une amélioration qu'on aurait apportée au comportement des agents afin de les rendre *plus intelligents*. La modification du code d'un agent s'est donc limitée au changement des structures de données sur lesquelles la perception et l'action travaillent.

Que manque-t-il alors pour finir de rendre ce comportement complètement générique, et ainsi ne plus avoir à y toucher ? Si on adopte une approche environnement-centrée, la solution est immédiate. Si on déporte le calcul du lissage et de l'alignement dans l'environnement et qu'on réifie le critère d'homogénéité, le code d'un agent n'aura plus besoin d'être modifié, quel que soit le type d'image. Pour que cela soit le cas, il suffit que l'environnement soit capable de gérer la perception et l'action suivantes :

1. perception : le pixel courant répond-il au critère d'homogénéité mémorisé dans l'agent ?
2. action : lisser/aligner le pixel en fonction du critère d'homogénéité mémorisé.

Dans une telle perspective, le comportement d'un agent se réduit à l'essentiel et son implémentation peut ainsi être codée sur seulement quelques lignes. Sa simplicité n'en mérite d'ailleurs pas plus. Déporter au maximum le comportement des agents dans des procédures environnementales montre ici tout son intérêt. On obtient un comportement agent complètement décorrélé du critère d'homogénéité. Cela facilite sa compréhension et son codage et permet de se concentrer sur ses aspects importants, et éventuellement d'apporter une *vraie* amélioration. On pourrait donc a priori l'appliquer à tout type d'image, et même imaginer l'utiliser sur d'autres problèmes, par exemple pour de la fouille de données.

Pour conclure sur cette expérience dans le TNI, notons enfin que le prétraitement systématique de l'image nous a particulièrement marqué. Cette démarche permet de ne pas se contenter des seules données initiales du problème. Il s'agit de construire un environnement adapté aux agents que l'on souhaite mettre en place. Cela afin de ne pas introduire un bruit inutile dans leur comportement, générateur de complexité. Autrement dit, ce n'est pas aux agents de rendre les données intelligibles, l'environnement doit s'en charger. C'est une démarche qui contribue très efficacement au déplacement de la complexité des agents vers l'environnement.

# Chapitre 5

# Gestion par niveaux de détail des IA dans un jeu vidéo

En 2008, nous avons rejoint l'équipe SMILE du LIRMM. Dans celle-ci, la thématique jeu vidéo tient une place importante, notamment à travers les travaux menés par Abdelkader Gouaïch sur, entre autres, l'adaptation dans les jeux sérieux [Hocine et al., 2015]. Par ailleurs, le contexte local est aussi très favorable dans la mesure où l'agglomération montpelliéraine tend a devenir une place importante du jeu vidéo en France [1]. De plus, les jeux vidéo représentent un domaine applicatif naturel pour la simulation, car ils reposent en grande partie sur la mise en œuvre effective des techniques qui s'y rapportent.

Ce contexte nous a donc amené à nous rapprocher de cette thématique, dès notre arrivée. Ainsi, avec Abdelkader Gouaïch et Stefano Cerri, nous avons participé à l'encadrement de la thèse de Ghulam Mahdi : *Level of Detail in Agent Societies for Games* [Mahdi, 2013]. Celle-ci a consisté dans la proposition d'une méthodologie pour la réalisation d'IA dont la qualité peut être adaptée en fonction des ressources matérielles disponibles, ceci grâce à une approche par niveaux de détails (level of detail LOD).

## 5.1 Vers plus d'agents autonomes dans les jeux vidéo

Un jeu vidéo n'a d'objet que si un joueur y joue. Tous les développements doivent converger vers un seul objectif : maximiser l'expérience utilisateur (notée ici *Quality of Experience* QoE) [2]. Dans cette optique, la cohérence du monde virtuel dans lequel le joueur évolue est

1. La présence d'une antenne d'Ubisoft, un géant du secteur, et l'organisation depuis 2010, par l'agglomération, de la manifestation annuelle *Montpellier In Game* (MIG), forum principalement dédié aux professionnels, témoignent d'une dynamique régionale sur le sujet.

2. Il s'agit évidemment d'une problématique très vaste. On y trouve en particulier la notion d'*immersion* [Ermi et Mäyrä, 2005], et plus généralement de *flow* [Chen, 2007], un concept qu'on peut lier au degré d'implication du joueur. On retrouve le problème de l'évaluation que nous avons rencontré dans le cadre du TNI. Avec l'humain dans la boucle, celle-ci est fortement liée au contexte et difficile à réaliser [Jennett et al., 2008].

cruciale [McMahan, 2003]. Par cohérence, il faut comprendre crédible, ou mieux, intelligible. En particulier, il ne s'agit pas forcément de simuler un monde à l'image de notre réalité, mais plutôt de proposer à l'utilisateur un environnement dans lequel il puisse se projeter en tant que joueur, grâce à une absence de contradictions rédhibitoires. Il n'y a pas de règle en la matière. Seule l'expérience utilisateur compte. Ainsi la détection des collisions peut être un prérequis dans un jeu et être superflue dans un autre.

Longtemps restée à la marge derrière les graphismes, la qualité de l'IA est aujourd'hui considérée comme une dimension importante d'un jeu. Notamment, pour rendre encore plus convaincants les mondes virtuels, les comportements des *entités non joueur* (PNJ) doivent être le plus crédibles possible [Hingston, 2012].

Dans le même temps, pour être jouable, un jeu vidéo doit nécessairement produire un minimum d'images par seconde (Frame Per Second FPS). Le calcul d'une *frame* comprend à la fois (1) l'exécution des IA (et l'évolution du monde) et (2) l'affichage graphique (le rendu du monde). Chacun de ces aspects s'inscrit dans un contexte temps réel et les calculs correspondants doivent impérativement être terminés en quelques millisecondes, sans quoi le nombre de FPS peut devenir trop faible et dégrader la QoE.

Dans ce contexte, il est fréquent que l'IA utilisée dans les jeux soit source de ralentissement. Il est en effet difficile de borner les ressources qui seront finalement nécessaires pour traiter les diverses situations qui peuvent se présenter (complexité de la situation, nombre d'IA présentes, etc.), surtout lorsqu'on considère des architectures cognitives [Dinont, 2007 ; Funge et al., 1999]. La gestion temporelle de l'exécution des IA constitue donc un problème difficile et fortement contraint.

Il existe d'ailleurs une différence notable entre l'IA pratiquée dans le monde académique et la philosophie employée pour l'élaboration des IA d'un jeu vidéo. Dans *Artifcial Intelligence for Games* [Millington et Funge, 2009] (la bible de l'ingénieur sur le sujet), les auteurs expliquent éviter le terme d'*agent*, car il sous-entend un design ascendant (*bottom-up*) où le comportement global du système résulte de la composition d'agents autonomes élaborés individuellement, ce qui renvoie effectivement à l'idée de perte de contrôle. De fait, bien qu'ils ne s'interdisent aucune approche (la démarche est pragmatique), les développeurs de jeux vidéo privilégient des designs *top-down* où les comportements des *game characters* sont produits de façon supervisée, notamment pour simuler des centaines de personnages. Ce qui permet de borner plus facilement le temps d'exécution correspondant à l'IA.

Cependant, plusieurs travaux ont montré les limites des architectures trop simples [Niederberger et Gross, 2005 ; Routier, 2005]. De plus, les approches supervisées sont souvent basées sur une connaissance exhaustive de l'environnement, ce qui rend difficile l'élaboration de PNJ exhibant des comportements autonomes convaincants (non-repsect de la contrainte de localité).

## 5.2 LOD pour la gestion adaptative des IA d'agents autonomes

À l'instar de [Dinont, 2007], la thèse de Ghulam propose des contributions qui visent à permettre une approche multi-agents ascendante qui intègre des contraintes temporelles. Dans le contexte des jeux vidéo, la solution proposée repose sur une approche par niveau de détail (*level of detail* LOD)[3] qui permet l'implémentation d'agents dont l'IA peut être dynamiquement adaptée, de manière bidirectionnelle (dégradée ou améliorée), en fonction des ressources matérielles disponibles [Mahdi et al., 2013].

L'idée générale du LOD consiste à allouer le plus de ressources aux entités les plus importantes, ici pour le joueur. Pour appliquer le LOD sur des IA, l'approche retenue, inspirée de [Niederberger et Gross, 2005], consiste à modifier dynamiquement le comportement d'un agent, suivant son importance courante (*behavioral LOD*). Il s'agit donc de définir un modèle d'agent permettant de moduler le comportement en fonction du temps accordé par un système de distribution des ressources.

Dans [Niederberger et Gross, 2005], la répartition des ressources est entièrement supervisée. Un algorithme calcule l'importance de chaque agent (ici en rapport avec sa visibilité dans la scène) et lui alloue un certain temps d'exécution. Toute l'intelligence de la répartition est concentrée dans cet algorithme (finalement assez complexe) : les agents n'ont donc aucune influence sur la politique globale d'exécution. Ainsi, les besoins qui pourraient être identifiés au niveau microscopique, par les agents eux-mêmes, ne peuvent être pris en compte.

## 5.3 Utilisation d'un environnement social

Dans une perspective E4MAS, une des contributions importantes de la thèse repose précisément sur l'utilisation d'un environnement social qui permet aux agents d'influencer la répartition globale des ressources. Dans ce but, le modèle AGR (cf. section 7.1) est utilisé pour spécifier l'importance des agents. Sur cette base, le travail du concepteur consiste à prédéfinir la manière dont les ressources (*tokens*) sont partagées entre les rôles.

Pour appliquer le LOD, l'idée est de définir plusieurs modes de fonctionnement possibles pour la répartition des ressources (tableau 5.1) : *normal*, *dégradé* ou *optimisé*. En mode dégradé, on peut par exemple décider de réduire à zéro les ressources données aux agents faisant partie du décor, et inversement les activer lorsque plus de ressources deviennent disponibles.

3. Cette perspective peut être rapprochée des travaux qui utilisent le LOD dans le cadre de la simulation multi-agents, notamment pour des systèmes large échelle comme les simulations de foule (e.g. [Navarro et al., 2011]) ou la simulation multi-niveaux [Picault et Mathieu, 2011 ; J.-B. Soyez et al., 2013]. Même si le contexte applicatif et les objectifs peuvent être assez différents, il s'agit souvent d'être capable d'introduire des contraintes temporelles sur les durées de calcul.

| Agent Role | Tokens in Optimal Mode | Tokens in Medium Mode | Tokens in Basic Mode |
|---|---|---|---|
| Targets | 37.5% | 41.6% | 50% |
| Ennemies | 25% | 25% | 25% |
| Goomba | 18.7% | 25% | 25% |
| Clouds | 12.5% | 8.3% | 0% |
| Trees | 6.3% | 0% | 0% |

TAB. 5.1 – Modes de fonctionnement pour le LOD

Cette modélisation de l'importance grâce à une structure organisationnelle a plusieurs avantages. Cela permet en premier lieu de ne pas corréler l'importance à un concept prédéfini, comme la distance par exemple. Au contraire, il est possible d'utiliser plusieurs critères simultanément dans une approche unifiée : dans AGR, les rôles définissent de manière abstraite la fonction qu'un agent remplit dans la société, et donc son importance relative au sein de celle-ci.

Deuxièmement, il est simple pour un agent de modifier proactivement son importance, car il lui suffit de changer de rôle. Un agent peut aussi avoir plusieurs rôles et cumuler les ressources. Ce qui permet d'obtenir un nombre de configurations important pour les priorités entre agents, sans que cela ne nécessite de négociations directes entre les agents (voir [Dinont, 2007] pour une solution de ce type).

Enfin, l'algorithme de scheduling correspondant est extrêmement simple et se résume à une division des ressources entre les groupes : aucune autre connaissance n'est nécessaire. Ce qui est rendu possible par le fait que les agents participent proactivement à la solution, ce qui permet d'introduire des considérations issues du niveau microscopique, de manière ascendante. Ici, c'est donc bien grâce à la définition d'un environnement social que les agents peuvent rentrer dans la boucle, et ainsi influencer le résultat global de la répartition des ressources.

Notamment, si nous employons les termes *environnement social*, plutôt que *organisation*, c'est précisément pour insister sur le fait que c'est le moyen par lequel les agents *existent* dans le système, c'est la structure organisationnelle qui *médie les interactions* et l'*accès aux ressources*. On a là tous les ingrédients de la définition d'un environnement, telle que nous l'avons présentée dans la section 3.1.

## 5.4 Délégation des moyens de perceptions et d'actions

Une fois les ressources distribuées (*tokens*), reste à définir un moyen pour que les agents puissent les utiliser sans les dépasser. Les *tokens* pouvant être assimilés à une monnaie virtuelle, l'idée est de valuer, en *tokens*, les différents algorithmes que les agents peuvent utiliser, ce qui permet notamment de s'abstraire du contexte matériel [4]. Parallèlement, on définit pour chaque

4. Même si on ne peut en effet pas connaître la puissance du matériel utilisée et le temps que prendra finalement un algorithme, il est relativement simple de calculer des relations mathématiques entre algorithmes. Si l'algorithme A est trois fois plus coûteux que B alors A nécessitera trois fois plus de jetons que B.

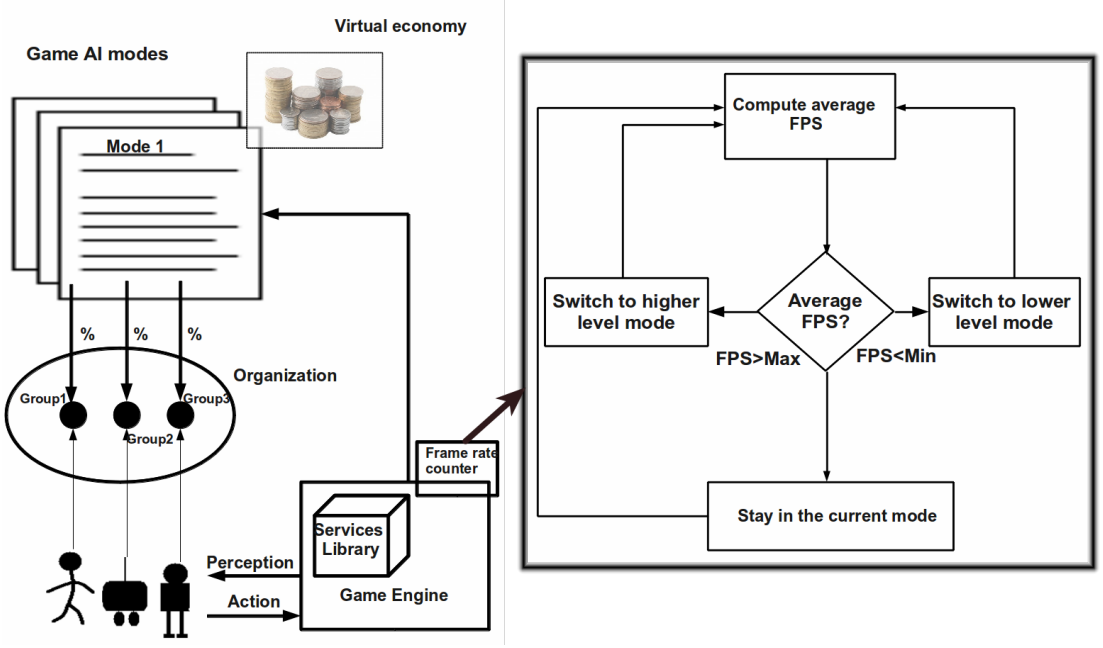



FIG. 5.1 – Vue globale du mécanisme de LOD adaptatif pour les IA

mode un nombre de *tokens*. Ainsi, lors de la distribution des ressources, chaque agent reçoit un certain nombre de *tokens*, en fonction de son importance, qu'il peut dépenser pour effectuer des perceptions et des actions. La figure 5.1 illustre le fonctionnement global du système.

Sur ce point, un aspect essentiel de l'approche proposée tient dans la délégation des moyens de perception et d'action au moteur de jeu (figure 5.2). Dans cette application, c'est grâce à ce mécanisme qu'on peut garantir que les agents ne dépasseront pas les ressources allouées, malgré leur autonomie. Sans entrer dans les détails, il s'agit de vérifier que les agents ne déclenchent pas des comportements pour lesquels ils n'ont pas assez de *tokens* (un comportement autonome déviant est si vite arrivé).

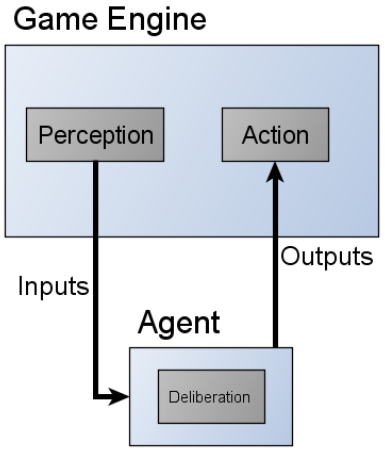



FIG. 5.2 – Délégation des moyens de perceptions et d'actions des agents au moteur de jeu

C'est donc bien grâce à la délégation des moyens de perception et d'action qu'il est ici possible de concilier *autonomie des agents* et *contrôle du résultat global*. Contrôle, non pas dans le sens où le résultat global est supervisé ou dirigé, mais plutôt dans le sens où l'on peut garantir que la solution produite reste bornée, ce qui est ici un impératif applicatif incontournable étant donné les contraintes temporelles très fortes. Autrement dit, *les agents proposent* et *l'environnement dispose*.

On voit bien ici que le principe influence/réaction n'est pas loin. Les agents ne proposent que des influences et ne peuvent donc pas mettre en péril le fonctionnement du système du point de vue des performances. On retrouve aussi par conséquent l'intérêt de déplacer les moyens d'actions et de perceptions dans l'environnement (cf. section 3.4).

Pour conclure sur ces travaux, notons qu'un jeu de tir, *My Duck Hunt* (tir aux canards inspiré du jeu de Nintendo *Duck Hunt*), a été réalisé à l'aide du framework AGDE (création et exécution de jeux vidéo) développé par Abdelkader Gouaïch. Ceci afin d'évaluer statistiquement l'intérêt de l'approche en évaluant son influence sur la qualité de la QoE. Rappelons ici que la seule métrique valable est le ressenti du joueur pendant le jeu.

Ainsi, l'expérimentation a consisté à faire jouer différents utilisateurs avec deux versions du jeu (avec et sans LOD) et à analyser leur ressenti (sans les informer du test). Les résultats ont montré une influence positive du LOD. Au niveau technique, cela s'explique non seulement par une meilleure stabilité effective du nombre de FPS (figures 5.3 et 5.4), mais aussi par le fait que l'ajustement des IA, en fonction de leur importance, permet d'optimiser l'utilisation des ressources sans que cela n'impacte trop les aspects qui sont essentiels pour le joueur.

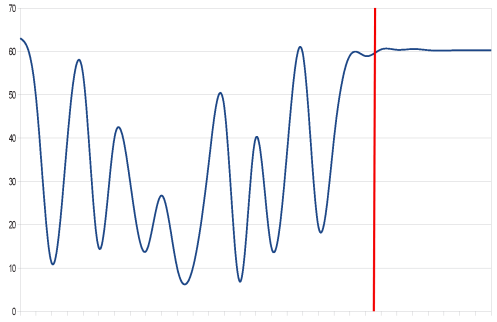



FIG. 5.3 – FPS sans LOD

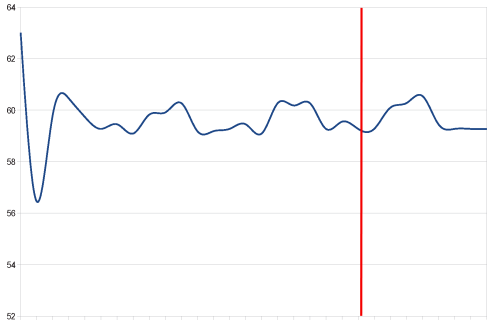



FIG. 5.4 – FPS avec LOD

# Chapitre 6

# Robotique mobile collective en milieu sous-marin

Pour compenser l'absence d'un an de Jacques Ferber (année sabatique 2010/2011), fin 2010 nous avons intégré le coencadrement de la thèse de Nicolas Carlési, débutée fin 2009 et codirigée par Bruno Jouvencel de l'équipe EXPLORE du LIRMM (robotique). Dans sa thèse, intitulée *Coopération entre véhicules sous-marins autonomes : une approche organisationnelle réactive multi-agent*, Nicolas a étudié la coopération et la coordination entre véhicules hétérogènes en milieu sous-marin [Carlési, 2013]. Sa contribution majeure repose sur l'utilisation conjointe du modèle organisationnel AGR [Ferber et Gutknecht, 1998] avec des techniques de coordinations réactives utilisant des messages extrêmement simples [**simon2001** ]. Malgré la simplicité des messages, la structuration des interactions obtenue grâce à l'organisation permet d'obtenir une coordination efficace entre des véhicules sous-marins autonomes (*Autonomous Underwater Vehicule* AUV) hétérogènes [Carlési et al., 2011]. Dans ce cadre, Nicolas a proposé une méthodologie de conception et une architecture logicielle qui sont indépendantes de la nature des AUV et du type de mission considérés.

## 6.1 Vers des flottilles de robots sous-marins autonomes

Du fait d'un milieu très hostile pour l'homme, l'exploration des profondeurs marines représente un vrai défi et les robots sont couramment utilisés pour réaliser divers types de missions. Dans ce contexte, l'utilisation de flottilles d'AUV est motivée par deux objectifs courants en robotique mobile collective :

— minimiser les coûts d'une mission en répartissant les risques sur plusieurs unités, de sorte que le dysfonctionnement ou la perte d'un appareil ne remettent pas en cause la mission dans son ensemble.
— permettre la réalisation plus rapide d'une mission grâce au nombre d'appareils (e.g. exploration ou couverture de zone).

Dans l'eau de mer, la célérité d'une onde sonore, principal moyen de communication utilisé [1], n'est que d'environ 1500 m/s. Il y a donc une latence importante entre les messages. De plus, du fait de la forte dynamique du milieu sous-marin, les messages peuvent être très bruités et les pertes fréquentes. Enfin, la nature des ondes acoustiques induit une bande passante faible. Communiquer est donc difficile. Ce qui nécessite de considérer des approches où les communications et les quantités d'informations échangées restent faibles. Par ailleurs, il est difficile de connaître la position absolue d'un appareil, notamment car il est impossible de communiquer avec un dispositif satellitaire (e.g. GPS), même pour de petites profondeurs. Un AUV embarque quasiment systématiquement un dispositif destiné à évaluer sa dérive.

La littérature consacrée à la conception de flottilles d'AUV montre que ces contraintes limitent fortement la portée des applications envisagées. En particulier, coordonner le déplacement reste un problème difficile sur lequel la plupart des travaux se focalisent (navigation en formation). D'une manière générale, les solutions proposées reposent sur des flottilles homogènes car cela facilite la coordination.

Cependant la possibilité d'intégrer des AUV hétérogènes est l'un des objectifs du domaine. En effet, les AUV sont souvent conçus pour des tâches spécialisées car ils ne peuvent pas embarquer beaucoup de matériel (consommation énergétique). Par conséquent la réalisation de missions plus complexes nécessite de faire l'hypothèse qu'ils seront potentiellement hétérogènes. Les approches proposées ne passent pas ce critère car elles sont trop dépendantes de l'homogénéité : un AUV moins rapide compromet la coordination de toute la flottille. De plus, bien qu'elles prennent en compte autant que possible les contraintes du milieu, les solutions proposées s'inspirent malgré tout de ce qui se fait dans le domaine terrestre ou aérien et s'appuient par conséquent sur des hypothèses où la fiabilité des communications est une donnée importante. Notamment, elles nécessitent un partage d'information conséquent et des échanges de messages fréquents. Autrement dit, en s'appuyant sur des communications de haut niveau, elles ne prennent pas suffisamment en compte la réalité du domaine applicatif. Les deux objectifs poursuivis par Nicolas ont donc été les suivants :

1. permettre la conception de flottilles composées d'AUV hétérogènes, à la fois en termes de fonctions et de capacités.
2. réduire au minimum les communications ainsi que la quantité d'informations échangée à travers celles-ci. De plus, le bruit et la perte de messages ne doivent pas compromettre le fonctionnement de la flottille.

## 6.2 Approche organisationnelle réactive multi-agent

Le premier objectif nécessite de faire abstraction des caractéristiques individuelles des AUV. Dans cette optique, le modèle AGR a été utilisé car il permet de définir des organisations indépendamment des caractéristiques individuelles [Ferber et al., 2004].

1. Les ondes électromagnétiques se propagent très mal dans l'eau et n'ont une portée que de quelques mètres.

L'une des contributions de Nicolas repose ainsi sur la proposition d'un modèle organisationnel pour la spécification de flottilles d'AUV. Celui-ci permet de définir des missions variées sans faire de supposition sur les AUV participants. La figure 6.1 montre un exemple d'organisation hiérarchique et la 6.2 donne un exemple des rôles qui peuvent être tenus dans un groupe.

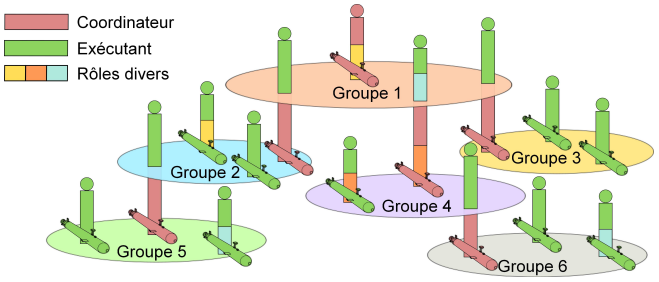



FIG. 6.1 – Organisation de la flotille

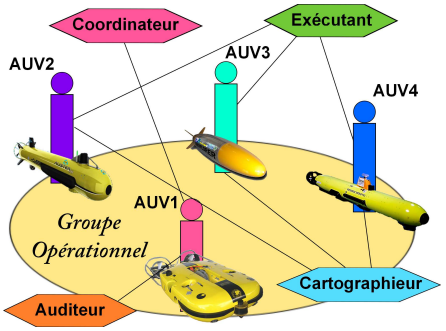



FIG. 6.2 – Exemple de groupe

En ce qui concerne la réduction des communications, Nicolas avait été orienté par Jacques Ferber vers le modèle *satisfaction/altruisme* (SAT-ALT) proposé par Olivier Simonin [2001]. Inspiré du principe d'éco-résolution [Ferber, 1995], ce modèle définit une solution réactive où les agents émettent des signaux d'attraction et de répulsion très simples (quelques bits peuvent suffirent) qui déclenchent chez eux différents comportements suivant le contexte. Pour l'avoir vu fonctionner sur des robots réels, nous connaissions nous aussi bien le potentiel de ce modèle.

Nicolas a cependant remarqué le problème associé à la faible sémantique des messages, c'est-à-dire la nécessité, au niveau de l'agent, de reconstruire le contexte associé aux messages afin qu'ils puissent être exploitables. Ce qui avait pour conséquence de rapidement introduire de la complexité dans les comportements dès que les agents étaient hétérogènes [Chapelle et al., 2002]. Autrement dit, lors de la réception d'un signal, un agent doit pouvoir répondre à la question : *repousser/attirer, par qui ? pourquoi ?*

C'est là que l'utilisation de la structure organisationnelle, en tant qu'*environnement médiant les interactions*, a joué un rôle décisif. Non seulement elle permet de faire le lien entre la mission et la structure de la flottille nécessaire au niveau global mais, au niveau microscopique, elle permet de spécifier d'une manière simple le contexte des interactions, c'est-à-dire la façon dont les AUV doivent réagir aux messages en fonction des rôles qu'ils tiennent dans l'organisation.

Le principe consiste à préciser, pour chaque message, le rôle de l'émetteur et le rôle du destinataire. Les messages sont donc liés au contexte interactionnel défini par la spécification de l'organisation. Ainsi, lorsqu'un agent reçoit un message, il en connaît immédiatement le contexte : il sait *de qui* (rôle émetteur) et *pourquoi* (rôle cible) [Carlési et al., 2011].

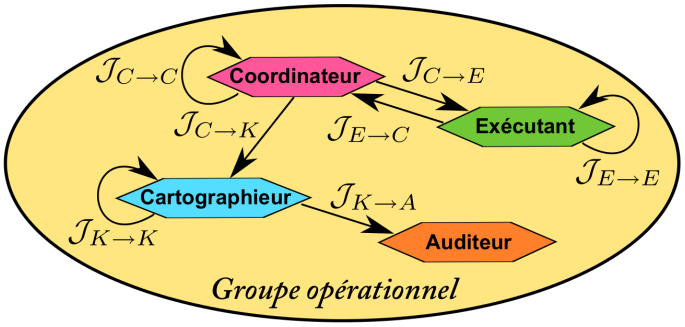



FIG. 6.3 – Spécification des interactions au sein d'un groupe

La figure 6.3 montre un exemple des interactions définies au sein d'un groupe (l'orientation des interactions définie qui réagit au signal). À partir de là, le concepteur spécifie comment un signal doit être interprété suivant la situation interactionelle.

La table 6.1 donne deux exemples. La première interaction spécifie que lorsqu'un cartographieur émet un signal d'intensité positive (attraction) à destination d'un coordinateur, celui-ci réagit ($I_{K\to C}$) en se dirigeant vers lui afin de l'aider à se repositionner. La deuxième définit une interaction due à un signal négatif (répulsion) envoyé par un cartographieur à ses pairs, par exemple afin d'optimiser l'occupation de l'espace et d'éviter les perturbations mutuelles.

| Interaction $\mathcal{I}_{coordinateur\to cartographieur}$ | | |
|---|---|---|
| Situation d'interaction | Signal $I_{C\to K}$ | but coordinateur |
| erreur de positionnement estimée > précision désirée | $I > 0$ | rejoindre l'émetteur et transmettre position |
| **Interaction $\mathcal{I}_{cartographieur\to cartographieur}$** | | |
| Situation d'interaction | Signal $I_{K\to K}$ | but destinataire |
| suivi auditeur \|\| acquisition en cours | $I < 0$ | éloignement de l'émetteur |

TAB. 6.1 – Interactions : *coordinateur* - *cartographieur* et *cartographieur* - *cartographieur*

Ce principe permet de définir aisément un grande nombre d'interactions sans que cela ne nécessite des messages coûteux. Dans les scénarios utilisés par Nicolas, seule une dizaine de bits est nécessaire pour encoder l'intensité du signal et les rôles impliqués. Ce qui permet à cette approche d'adresser très efficacement les contraintes pesant sur les communications en milieu sous-marin. De plus, un des intérêts de SAT-ALT est qu'il n'est pas nécessaire que tous les signaux soient reçus pour que le système puisse fonctionner. L'approche proposée est donc très résistante au bruit et aux pertes.

## 6.3 Simulation avec TurtleKit 3

Nicolas a utilisé notre plate-forme TurtleKit (cf. section 7.2) tout au long de sa thèse pour élaborer et tester son modèle. Il a d'ailleurs effectué un travail remarquable en ce qui concerne la modélisation des contraintes du milieu sous-marin (3e dimension, vitesse des messages, pertes, dérive des AUV, distance de sécurité, etc.).

Ainsi, différentes stratégies de fonctionnement de flotille ont pu être testées. Nicolas a notamment proposé une stratégie collective où les AUV se regroupent (relocalisation, nouvelle organisation, etc.) avant de passer par une phase autonome, puis de se regrouper à nouveau (figure 6.4). Ce fonctionnement a été testé à l'aide de simulations d'un scénario de recherche de boîte noire.

La figure 6.5 montre les trajectoires suivies par les différents AUV durant la mission (le fond marin est en niveau de gris). Ces simulations ont permis de montrer comment des AUV hétérogènes (4 rôles différents) arrivent à se coordonner et à coopérer en utilisant des messages de seulement 14 bits : 8 bits pour les signaux d'interaction, 2 bits par rôle (émetteur et destinataire) et 2 bits pour l'identifiant d'un AUV.

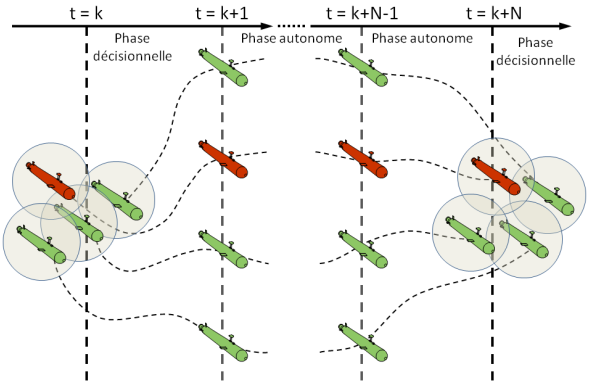



Fig. 6.4 – Stratégie de fonctionnement

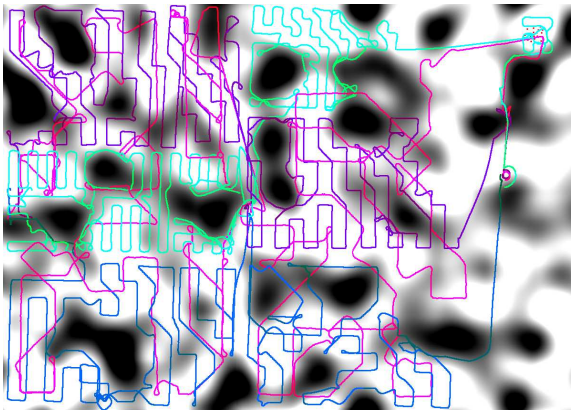


Fig. 6.5 – Exploration systématique de l'environnement par des AUV hétérogènes : recherche de boîte noire

Le travail de Nicolas a eu une influence assez importante sur l'implémentation de MaDKit 5 et de TurtleKit 3, que nous discutons dans le chapitre suivant. Il en fut le premier utilisateur alors qu'elles n'étaient qu'en version *pre-alpha*. Notamment, dans MaDKit 4, aussi étrange que cela puisse paraître, il n'était pas possible d'envoyer un message en spécifiant le rôle de l'émetteur (contextualisation de l'interaction). Par ailleurs, cela nous a permis de tester et de retravailler la modularité de TurtleKit 3. Nicolas a pu ainsi redéfinir tous les aspects nécessaires à l'implémentation de ses simulations.

## 6.4 Combiner environnements physique et social

Dans les deux chapitres précédents, nous avons vu tour à tour l'intérêt des dynamiques environnementales physiques puis sociales. Ici, le couplage entre un modèle organisationnel et un modèle réactif montre l'intérêt de combiner différents types d'environnements. C'est grâce à cette combinaison que le modèle SAT-ALT a pu être étendu de manière aussi efficace. Ce qui permet un mariage très intéressant entre une approche réactive (SAT-ALT) et une approche cognitive (ici les AUV contiennent des procédures décisionnelles qui font appel à de la planification).

De plus, si dans les approches réactives la structure sociale est souvent considérée a posteriori, en tant que phénomène émergent, cet exemple montre l'intérêt de rajouter, a priori, un support social aux interactions. Il n'y a pas de raison pour ne pas faire en sorte que l'environnement médie les interactions de la façon la plus efficace possible. Combiner environnements physique et social est un moyen concret de le faire.

Une idée similaire est développée dans l'article récent de [Galland et al., 2014], intitulé *Contextualiser l'interaction entre agents en combinant dimensions sociale et physique au sein de l'environnement*[2]. Comme le note les auteurs, dimension physique et dimension sociale, rarement réunies, sont souvent vues comme des problématiques décorélées, la première déidée aux actions ancrées dans le réel, et la seconde dédiée aux communications par envoi de messages. De plus, lorsqu'elles le sont, c'est au sein de l'agent que la combinaison est traitée. Ainsi, les auteurs proposent un modèle qui unifie ces deux dimensions dans l'environnement pour *contextualiser l'interaction*. Dans celui-ci, basé sur influence/réaction, les agents émettent des influences qui sont combinées en fonction des positions sociales et physiques des agents.

Combiner environnements physique et social constitue aujourd'hui pour nous une piste très intéressante. En particulier, nous avons l'intention de tester ce principe sur un modèle de type fourmis, en essayant des simulations où plusieurs dizaines de phéromones seront associées à des rôles (ce qui devrait être possible grâce à nos derniers travaux sur l'utilisation du calcul haute performance). Cela devrait permettre de contextualiser l'interaction et ainsi d'augmenter les fonctionnalités du système, sans pour autant aboutir à des *fourmis cognitives*. De notre point de vue, cette combinaison est essentiellement un moyen de plus de déplacer la complexité des agents dans l'environnement.

2. Prix du meilleur papier des JFSMA 2014.

# Deuxième partie

# Recherches en cours et perspectives

# Chapitre 7

# Développements logiciels : MaDKit et TurtleKit

Ce chapitre présente nos principales contributions logicielles : MaDKit, un outil logiciel générique pour le développement de SMA, et TurtleKit, une plate-forme de simulation multi-agents. Nous présentons ensuite deux collaborations issues de l'utilisation de TurtleKit, dans un cadre académique pour la première et industriel pour la seconde.

## 7.1 MaDKit : Multiagent Development Kit

### Approche organisation-centrée

Initiée en 1997 par Jacques Ferber et Olivier Gutknecht, MaDKit est une plate-forme logicielle générique pour le développement et la simulation de SMA. Elle est basée sur une implémentation du métamodèle organisationnel Agent/Groupe/Rôle AGR (figure 7.1), anciennement AALAADIN [Ferber et Gutknecht, 1998].

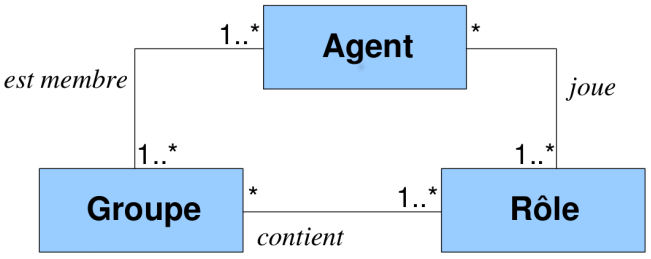



Fig. 7.1 – Le métamodèle organisationnel Agent/Groupe/Rôle AGR

Son développement est donc sous-tendu par une approche de conception organisation-centrée. En substance, il s'agit de considérer que la modélisation explicite de l'organisation,

et de sa dynamique, permet de faciliter le développement de SMA, notamment en permettant de ne pas faire reposer la responsabilité de la structuration du système sur les seuls agents. La littérature est très féconde sur le sujet et de nombreux aspects ont été abordés : abstractions conceptuelles [Zambonelli et al., 2001], dynamique [Mathieu et al., 2002], langage de modélisation [Boissier et al., 2007], spécification fonctionnelle [Sims et al., 2008], déontique [Hubner et al., 2007], pour ne citer que ceux-là.

La perspective utilisée pour MaDKit repose sur une approche minimaliste et générique [Ferber et al., 2004]. AGR ne décrit pas les aspects fonctionnels d'une organisation. Il propose uniquement de caractériser leur structure avec un minimum de concepts et de relations. Une organisation est vue comme un ensemble de groupes dans lesquels les agents jouent des rôles.

Sachant que le développement de MaDKit se fonde sur une architecture de type micro-noyau [1] [Gutknecht et al., 2001], son cœur se résume à très peu de fonctions-clés : gestion du cycle de vie des agents et maintenance de la structure organisationnelle à travers laquelle ils interagissent. Tout le reste de la plate-forme est construit suivant un principe d'agentification des services (réseau, simulation, interface graphique, etc.). Par ailleurs, aucune architecture comportementale n'est définie au niveau de MaDKit. De même, à part la structure organisationnelle, aucun environnement applicatif (modèle du monde) n'est défini par défaut. Cette approche minimaliste permet de poser peu de contraintes sur les SMA envisageables. En contrepartie, il est donc nécessaire de développer le modèle d'agent et d'environnement désiré. MaDKit 4 contient un certain nombre de *plugins* dédiés à cet effet.

### Historique des développements et motivations

Nous avons intégré le projet MaDKit en 1999, d'abord comme utilisateur en DEA, puis en tant que développeur lors de la conception des outils de simulation [Michel et al., 2001]. À la suite du départ d'Olivier Gutknecht en 2001, notre rôle a pris plus d'importance avec la prise en charge de l'évolution du noyau vers la version 3 (2001), puis 4 (2003), que nous avons maintenue jusqu'en 2010. Durant cette période, de nombreux autres développements ont été réalisés, notamment avec les travaux de Saber Mansour sur l'intégration de la mobilité dans MaDKit [Mansour et Ferber, 2007] et ceux de Sebastian Rodriguez et al. [2007] pour la simulation distribuée.

MaDKit 4 est encore utilisée mais plusieurs problèmes empêchent de faire évoluer le projet à partir de cette version. La vieillesse de certaines parties du code, combinée aux nombreuses ramifications qu'il comporte, rend impossible toute modification profonde du noyau.

En 2010, nous avons donc entièrement refondu le code afin de produire la version 5 [2]. Nous avons investi beaucoup de temps dans cet effort et il convient de dire l'état d'esprit

1. Seules les fonctions fondamentales sont d'abord implémentées, avant d'être complétées par des services construits à partir des fonctionnalités basiques [Liedtke, 1996].

2. Pour ce développement, nous nous sommes bien sûr inspiré des versions précédentes et nous remercions au passage toutes les personnes qui ont pu contribuer de près ou de loin à MaDKit.

dans lequel nous l'avons mené. Bien entendu, nous sommes nous-même convaincu, par la pratique et l'expérience, qu'une structure organisationnelle simplifie la programmation agent [3]. Mais au-delà de ça, MaDKit est avant tout, pour nous, un outil expérimental dont nous connaissons tous les rouages et qui, grâce à sa généricité, nous a permis de prototyper très rapidement beaucoup de nouvelles idées. Plus important, il constitue le socle de notre plate-forme de simulation TurtleKit, et est donc au cœur de nos travaux de recherche. In fine, c'est MaDKit qui nous permet d'éprouver la pertinence des approches et modèles que nous proposons, une fois confrontés à la réalité du développement et, plus important encore, de l'utilisation. Il nous paraît crucial que ce socle soit le plus solide et performant possible.

### Philosophie et caractéristiques de la version 5

Cette nouvelle version s'accompagne d'un changement de philosophie dans la manière dont MaDKit se présente aux utilisateurs. En version 4, MaDKit est distribué comme un ensemble de codes source organisés en modules (*plugins*). Développer avec MaDKit 4 nécessite de se conformer à cette architecture et de réaliser son propre *plugin*. De plus, cette version est elle-même dépendante d'un certain nombre de librairies additionnelles. Finalement, une fois décompressée, la version 4 ne pèse pas moins de 80 Mo. Bien sûr, tout n'est pas indispensable mais il est très difficile de le voir et de s'y retrouver pour un utilisateur novice.

Cette complexité de mise en œuvre est aujourd'hui loin des pratiques de développement couramment utilisées, notamment dans l'écosystème Java. Pour la vesion 5, il nous fallait faire un choix. Soit élaborer un environnement de développement complet destiné à guider la programmation *sous* MaDKit (type *plugin* Eclipse), à l'image de ce qui se fait pour beaucoup de cadriciels multi-agents (e.g. Jason [Bordini et al., 2007], JaCaMo [Boissier et al., 2013], MAY [Noel et al., 2010]). Soit créer une librairie indépendante qui puisse être utilisée sans contrainte dans un projet de développement lambda, pour programmer *avec* MaDKit. C'est cette dernière approche que nous avons suivie.

MaDKit 5 se présente donc sous la forme d'une librairie Java composée d'un unique fichier. Cette version est principalement destinée aux développeurs : elle ne contient pas d'outil dédié à faciliter son usage, si ce n'est la documentation exhaustive du code sous forme de Javadoc, quelques tutoriels et démonstrations jouet. Sous le capot, les changements apportés sont nombreux (implémentation d'AGR, réseau, optimisation multi-cœurs pour la simulation, traces, échange de messages liés à une conversation, etc.) et il serait difficile d'en expliquer tous les tenants et aboutissants sans rentrer dans des détails techniques. Cependant, il nous faut mentionner l'importance que les tests unitaires ont prise pour cette version. Toutes les fonctions-clés sont systématiquement testées (les comptes rendus sont disponibles sur www.madkit.net/madkit/test). Cela nous permet d'éviter toute régression due à la maintenance et, surtout, d'avoir une grande confiance dans le fonctionnement de

3. C'est l'école de pensée dans laquelle nous avons fait nos armes, notre *environnement primordial*.

la plate-forme, point essentiel pour nos travaux en simulation. En particulier, MaDKit 5 constitue le socle logiciel utilisé pour développer la plate-forme TurtleKit 3.

### Diffusion

MaDKit est disponible sur www.madkit.net. Le code source est disponible sous licence libre sur GitHub. Le nombre de téléchargements effectués via le site officiel, comptabilisé depuis février 2011 sur un serveur du LIRMM, s'élève aujourd'hui à un peu plus de 10000 (à peu près répartis à parts égales entre les versions 4.x et 5.x). La véritable taille de la communauté d'utilisateurs est difficile à évaluer dans la mesure où nous ne maîtrisons pas l'usage qui peut en être fait. Pour ce qui est des utilisations récentes, MaDKit a servi pour des travaux portant sur des thèmes extrêmement variés [4]. Nous savons aussi que beaucoup de téléchargements sont effectués dans un cadre pédagogique.

## 7.2 TurtleKit : simulation de systèmes complexes

### Simulation multi-agents basée sur le langage Logo

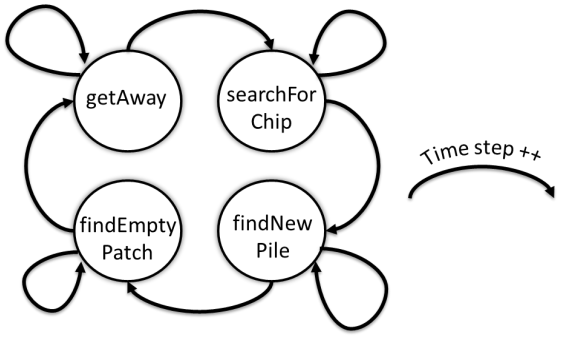
TurtleKit [Michel et al., 2005] est une plate-forme de simulation multi-agents dont nous avons commencé le développement en 1999. Dans nos travaux de thèse, elle constituait un moyen d'éprouver expérimentalement les outils de simulation de MaDKit. Son développement a été inspiré par Swarm [Minar et al., 1996] et StarLogo [Resnick, 1994], les deux principales plates-formes alors disponibles.

Développée au Santa Fe Institute, Swarm était très orientée génie logiciel et proposait des abstractions génériques pour la simulation. Élaborée par Mitchel Resnick durant son doctorat [Resnick, 1992], StarLogo (*Logo, évolution à plusieurs agents du langage Logo [5]) fournissait quant à elle un modèle d'agent et d'environnement prédéfinis mais très puissant en termes d'expressivité : une combinaison entre automates cellulaires (environnement) et agents mobiles. NetLogo [Sklar, 2007], dont le développement a commencé à la même période que TurtleKit, en est la descendante directe la plus connue.

4. Modélisation multi-niveaux de systèmes de systèmes (SdS) [J. B. Soyez, 2013], patrons de conception [Couturier et al., 2012], simulation de processus cognitifs (émotions) [Larue et al., 2013], outils e-learning collaboratifs P2P [Mazyad et al., 2013], segmentation d'images médicales [Pereira et al., 2012], étude de la communication dans les organisations multi-agents [Bouslimi et al., 2014], apprentissage multi-agents par renforcement [Aissani et al., 2012], répartition de ressources dans une grille de calcul [Thabet et al., 2011], étude de la délégation des actions/interactions dans l'environnement [Balbo et al., 2013], modélisation des émotions dans les SMA [Saunier et Jones, 2014], résolution de contraintes [Methlouthi et Bouamama, 2011], suivi de traces en e-learning [Talon et al., 2013].

5. Langage créé à la fin des années 60, notamment sous l'impulsion de Seymour Papert, cofondateur avec Marvin Minsky du MIT. L'idée était de créer un outil pédagogique novateur et interactif pour l'enseignement des mathématiques, notamment auprès des enfants [Feurzeig et al., 1969]. C'est dans le même esprit que StarLogo a été créée : pour promouvoir la *pensé décentralisée* auprès des étudiants. Resnick s'intéresse d'ailleurs plus aux processus cognitifs mis en jeu qu'aux simulations en elles-mêmes, travaux à la base de son célèbre livre *Turtles, termites, and traffic jams : Explorations in massively parallel microworlds* [Resnick, 1994].

À l'instar de StarLogo, TurtleKit définit un environnement de type automate cellulaire (une grille discrétisée en cellules, des *patchs*), dans lequel des agents (*turtles*) évoluent à l'aide de primitives atomiques très simples (avancer, tourner, modifier les variables des *patchs*, etc.). La conception d'un agent consiste à élaborer des comportements de plus haut niveau à l'aide des primitives atomiques et à définir la manière dont ils s'enchaînent. La figure 7.2 donne l'exemple du comportement de la *termite* tiré du livre de Resnick. Les agents peuvent aussi émettre et percevoir des phéromones digitales possédant des dynamiques de diffusion et d'évaporation, ce qui permet de créer des champs de gradients utilisés par les agents pour divers comportements, comme le suivi de traces par une fourmi.




```
public String findNewPile()
{
  if (getPatchColor() == yellow)
  {  // I am on a pile
    return("findEmptyPatch");
  }
  else
  {
    wiggle(); // random walk
    return("findNewPile");
  }
}
```

Fig. 7.2 – La termite de Resnick

Étant donné que TurtleKit repose sur un modèle très proche, pour ne pas dire similaire, à celui de NetLogo, il convient de préciser son positionnement. En effet, même si en 1999 NetLogo n'était pas très aboutie (première version finalisée distribuée en 2002), aujourd'hui elle est internationalement reconnue pour sa qualité et offre une batterie d'outils assez exceptionnels. Pourquoi donc ne pas se tourner vers elle. Comme nous l'avons dit, TurtleKit est aussi inspirée de Swarm. Cette plate-forme a la particularité de réifier les fonctions-clés d'un moteur de simulation (ordonnancement, observation, etc.), ce qui permet une grande généricité pour le fonctionnement d'une simulation. Non pas que NetLogo soit hermétique à toute manipulation (elle est open source), mais son architecture logicielle n'est pas faite pour permettre une modification du moteur. C'est notamment aussi pour cela que TurtleKit propose une transcription du langage Logo en Java, au contraire de NetLogo qui a gardé la syntaxe originelle, dont il est du coup impossible de se départir. En restant purement dans le monde de la programmation objet, TurtleKit s'adresse principalement aux développeurs. L'idée est de faciliter la modification et l'extension de ses fonctionnalités.

### Historique et architecture

TurtleKit a subi de nombreuses évolutions. Elle fut tout d'abord intégrée à MadKit et a donc suivi ses différents changements architecturaux. En particulier, la version 2 consiste dans un *plugin* pour MadKit 4. TurtleKit 2 a été en grande partie le fait des améliorations

apportées par Grégory Beurier pour les besoins de ses travaux de thèse [Beurier, 2007] (cf. section suivante). En 2009 nous avons d'ailleurs recruté Grégory en tant qu'ingénieur de recherche pour consolider TurtleKit 2. Cette version est toujours fonctionnelle, mais TurtleKit a subi la même évolution que MaDKit et a été entièrement refondue en 2012 sur les mêmes principes. La version 3 est aujourd'hui le support de nos derniers travaux de recherche.

TurtleKit 3 est développée selon les mêmes principes de micro-noyau que MaDKit et très peu de classes sont nécessaires à son fonctionnement. Elle est distribuée sous la forme d'une librairie Java indépendante (une seule archive). Un seul fichier peut suffire à créer une simulation. Étant basée sur MaDKit, TurtleKit a aussi la particularité que tous les services de simulation sont agentifiés. De telle sorte qu'il est possible de communiquer facilement avec d'autres agents de type MaDKit pour effectuer des tâches spécifiques (connexion BD, traces, etc.). Dans le détail, TurtleKit réifie les éléments suivants (figure 7.3) :

— **`Turtle`** : le modèle d'agent générique.
— **`Environnement`** : le modèle d'environnement et sa dynamique (plus les *patchs*).
— **`Scheduler`** : agent qui contrôle l'ordre d'exécution des différents agents (moteur et simulation).
— **`Observer`** : agent permettant l'observation, éventuellement la modification, des données de la simulation : visualisations, traces, initialisations spécifiques, etc.
— **`Pheromone`** : diffusion et évaporation de données dans l'environnement.

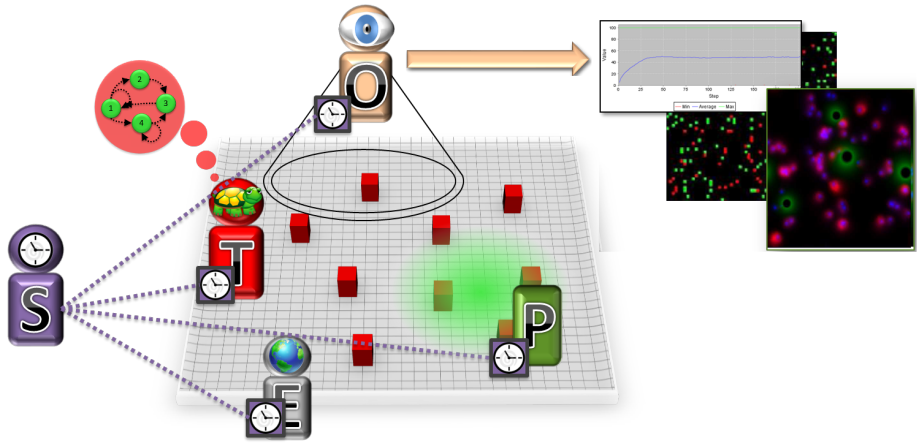



Fig. 7.3 – TurtleKit : **`Scheduler` `Turtle` `Environnement` `Observer` `Pheromone`**

## Diffusion

TurtleKit est disponible sur www.turtlekit.org et son code source sur GitHub. La dernière version n'a pas encore le niveau de finition de MaDKit et reste pour l'instant alpha ;

son API reste très changeante. Le nombre de téléchargements effectués via le site officiel depuis février 2011 est d'environ 900 (serveur Lirmm). Un peu moins de 100 pour la version 3, pour l'instant principalement utilisée localement. Dans le cadre de travaux doctoraux, outre Grégory et Nicolas, Gildas Morvan a utilisé TurtleKit (aide à la décision en entomologie médico-légale) [Morvan, 2009] et Arnaud Glad pour prototyper le modèle EVAP (*patrolling* en robotique) [Glad, 2011]. Nous avons eu aussi connaissance des travaux suivants : simulation de trafic [Kammoun et al., 2014], robotique mobile collective [Simonin et al., 2014] (*foraging*) et [Pépin et al., 2009] (environnement intelligent pour agents situés).

## 7.3 Morphogenèse et embryogénie artificielle avec TurtleKit

Grégory a démarré sa thèse (*Codage indirect de la forme dans les systèmes multi-agents : émergence multi-niveaux, évolution et morphogenèse* [Beurier, 2007]) avant que nous partions pour Reims. Nous étions donc très proche de ses travaux, et pour tout dire assez fasciné par les résultats. Après notre départ, nous avons donc continuer à collaborer, notamment pour une publication commune [Beurier et al., 2006].

Dans cet article, une modélisation multi-agents du processus de morphogenèse est proposée. En s'inspirant du développement naturel des embryons, l'objectif est de simuler l'évolution d'organismes artificiels afin de produire une forme définie par l'utilisateur. Les questions de recherche sous-jacentes sont les suivantes :

— Comment reproduire, artificiellement, les mécanismes qui permettent à une unique cellule d'évoluer, par division et différenciation, dans un organisme multi-cellulaires ?
— Comment, grâce à un processus évolutionnaire, trouver le code génétique de cette cellule afin qu'il produise un organisme fonctionnellement adéquat ?

Le modèle utilisé est très directement inspiré des processus biologiques connus en embryogenèse. Le code génétique d'une cellule (agent), représenté sur seulement quelques bits, permet de définir un comportement de division et de différenciation grâce à deux types de gènes. Les *gènes de segmentation*, responsables de la production dans l'environnement de morphogènes (une substance chimique modélisée par des phéromones TurtleKit). Et les *gènes sélecteurs* qui codent des fonctions comportementales spécifiques aux caractéristiques de l'organisme recherché : colorisation de la cellule, auto-réplication, apoptose, etc.

L'environnement joue ici un rôle fondamental car il contient un *gradient maternel* initial qui brise la symétrie du système et *oriente* la croissance de l'organisme. En effet l'expression des gènes est déterminée par la quantité et les gradients de morphogènes présents dans le voisinage immédiat de la cellule. La simulation permet ainsi de réaliser un système initialement composé d'un seul agent qui, par subdivisions successives, produit une forme. Celle-ci est donc codée indirectement dans le génome de l'agent. Le bon code génétique est obtenu grâce à un algorithme évolutionnaire : chaque code génétique d'une population initiale est automatiquement évalué par des agents du moteur de simulation, en fonction de la ressemblance avec la forme

désirée (figure 7.4). Des croisements sont ensuite effectués pour faire évoluer la population de génération en génération, jusqu'à trouver un génome satisfaisant.

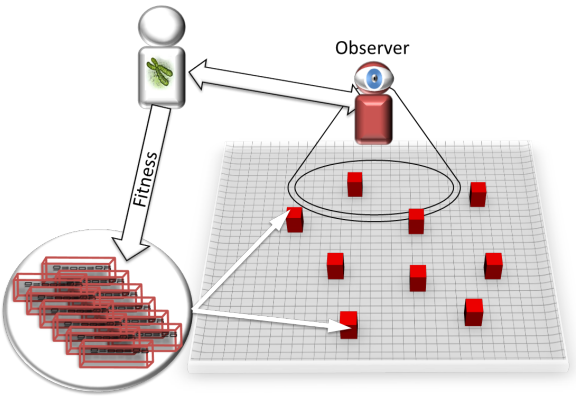



FIG. 7.4 – Évaluation et gestion de la population génétique par des agents

FIG. 7.5 – Phases de croissance d'un organisme formant le mot "Lirmm"

Dans [Beurier et al., 2006], la forme étudiée est le drapeau français, un classique de l'embryogénie artificielle en biologie théorique [Wolpert, 1968]. Mais les résultats les plus impressionnants ont été uniquement publiés dans la thèse. Dans celle-ci, Grégory est arrivé à reproduire des formes très complexes (mots). La figure 7.5 présente l'évolution d'un organisme qui donne le mot *Lirmm* (plusieurs centaines de générations sont nécessaires).

## 7.4 Programmation situationnelle : *l'IA pour les nuls*

Dans le cadre d'une collaboration avec l'industrie du jeu vidéo, le moteur de simulation de TURTLEKIT a servi de support pour le développement d'un jeu où des joueurs sont mis en compétition à travers l'élaboration d'IA d'agents autonomes. Alors que nous nous apprêtons à présenter nos derniers travaux dans le prochain chapitre, il est important de mentionner que

les recherches associées à cette collaboration ont joué un rôle déterminant dans les orientations que nous avons récemment suivies, c'est pourquoi nous les détaillons ici.

### De la programmation compétitive aux jeux vidéo

La programmation compétitive (*competitive programming*), c'est-à-dire la mise en concurrence de programmes sur un problème défini, est depuis longtemps utilisée en recherche pour évaluer la pertinence des approches et favoriser leur création. Un exemple emblématique est celui de la RoboCup [Kitano et al., 1997], une compétition qui confronte des équipes de robots autonomes dans des parties de football réelles (avec différentes normes par catégorie) ou virtuelles (simulations) [6].

Dans un cadre pédagogique, durant notre thèse nous avions participé à l'élaboration du logiciel Warbot [7], un jeu destiné à l'évaluation et à l'analyse de techniques de coordination entre agents, dans une situation de compétition où deux équipes de robots hétérogènes s'affrontent dans une simulation, exactement dans le même esprit que la RoboCup, mais pour l'apprentissage de la programmation multi-agents en troisième cycle. D'un point de vue pédagogique, c'est une approche très intéressante qui permet aux étudiants de mesurer la difficulté d'établir des stratégies collectives et de *penser décentralisé*.

Ayant fait une thèse au Lirmm, Pierre-Alain Laur (alors codirigeant de Feerik), connaissait l'existence de Warbot (il y avait pris beaucoup de plaisir en tant qu'étudiant) et souhaitait transposer cette idée au monde du jeu vidéo en ligne : si des étudiants prenaient beaucoup de plaisir à créer des IA, pourquoi pas des joueurs ?

Ce sujet nous a particulièrement motivé car le défi n'était pas mince : comment faire pour permettre à un joueur lambda, non-informaticien, de s'amuser à *programmer* des IA ? La première difficulté est qu'on ne peut pas se reposer sur les concepts de variable, de structure de données, de boucle ou encore de structure conditionnelle, et encore moins d'*autonomie*, de *croyances*, de *désirs* ou d'*intentions*. Bref, tout ce par quoi pourrait passer la *programmation* de l'IA d'un agent. Le jeu doit être simple, accessible en quelques minutes, et il est hors de question d'enseigner tous ces concepts au joueur avant qu'il ne puisse jouer.

Le premier élément incontournable était donc de pouvoir élaborer les IA à l'aide d'une interface graphique (GUI). Dans la littérature SMA, il existe en fait un certain nombre d'approches pour faire de la programmation visuelle de comportement, sans avoir à éditer du code, notamment dans le cadre de la simulation (e.g. la plate-forme SeSAm [Klügl et al., 2006] ou les outils d'édition de RePast [M. North et al., 2007]). Lors de ce travail, nous avions fait trois constats. Le premier concernait le manque d'expressivité des outils et la

6. Beaucoup de conférences proposent des compétitions qui permettent de confronter les idées : par exemple dans le cadre de IJCAI 2015 avec la *Angry Birds AI Competition* (jeu vidéo de tir simple), ou encore de AAMAS, qui héberge maintenant annuellement la *Trading Agent Competition*, une compétition où des agents doivent négocier et vendre aux mieux des ressources dans un marché concurrentiel simulé en temps réel.

7. www.madkit.net/warbot, projet géré par Jacques Ferber. Warbot utilise aujourd'hui TurtleKit 3.

grande difficulté à obtenir des comportements de haut niveau. Le deuxième était les limites des représentations graphiques utilisées, spatialement trop gourmandes et du coup difficiles à appréhender. Enfin, et c'était le plus important, toutes ces approches n'étaient in fine que des représentations graphiques de *programmes*. En d'autres termes, elles font explicitement ou implicitement l'hypothèse que leur utilisateur comprend ce qu'est une structure conditionnelle, une variable, ou encore une boucle, autrement dit *comment se déroule un programme*.

### Principe de la programmation situationnelle

Dans [Michel et al., 2010, 2011], nous avons proposé le principe de *programmation situationnelle* qui permet à un utilisateur lambda de programmer (ou plutôt élaborer) des IA, grâce à une GUI simple, sans connaissance de la programmation. L'idée consiste à reproduire le cycle perception / délibération / action grâce à la mise en relation, via une GUI, de *situations* (combinaisons de percepts) avec l'*exécution d'actions* de très haut niveau (plans).

L'élaboration d'une IA consiste ainsi à définir une *situation* en piochant dans les percepts disponibles (ennemi proche, énergie faible, etc.) et à la relier à un plan, qu'on peut, de la même façon, paramétrer à l'aide d'actions de haut niveau : patrouiller une zone (plan) avec un mouvement particulier (paramètre), attaquer à vue (plan) suivant un seuil d'ennemis (paramètre), etc. La figure 7.6 schématise cette approche et la 7.7 en montre une instance.

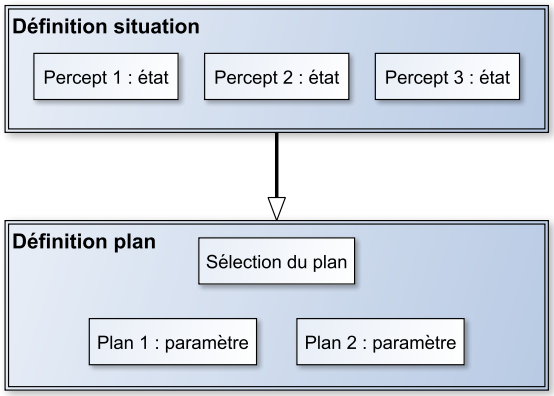



FIG. 7.6 – Programmation situationelle

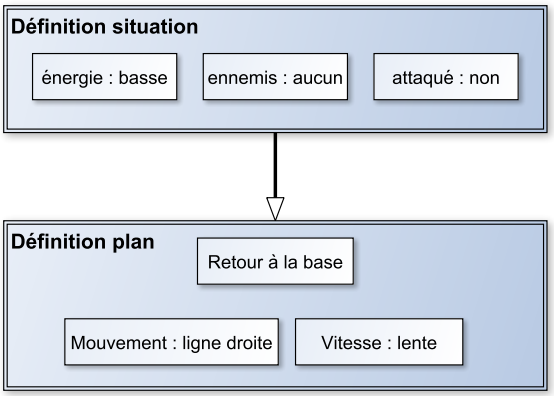



FIG. 7.7 – Exemple

Par ailleurs, l'utilisateur définit autant de situations qu'il le souhaite, mais les édite une à la fois, ce qui simplifie la compréhension et l'élaboration de l'IA et permet un grand nombre de configurations. La figure 7.8 montre des parties du prototype de GUI développée par Feerik (une page web). Les conflits entre situations sont résolus, par le joueur, via un éditeur de priorité (zone 4). Le point fort de cette approche réside dans le fait que le processus de délibération n'a pas besoin d'être explicité : il reste dans la tête du joueur ! Autrement dit, c'est la partie la plus complexe de l'IA qui est éliminée de la représentation graphique. Nul

besoin de la programmer, ni même de la justifier. C'est précisément l'intérêt du jeu : il définit une compétition entre raisonnements humains, et non un concours de programmation [8].

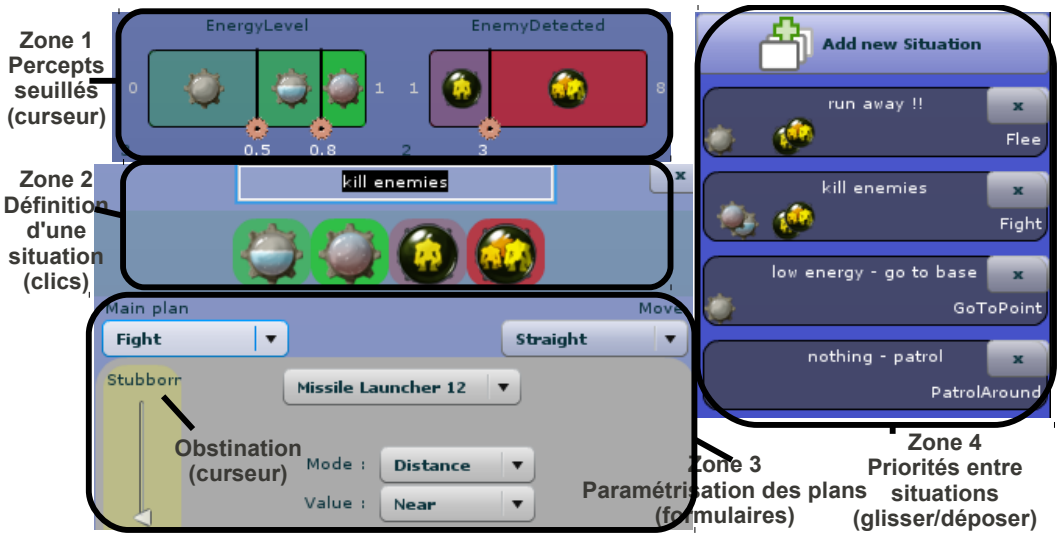



FIG. 7.8 – Délégation des moyens de perceptions et d'actions des agents au moteur de jeu

Dans une perspective plus large, tout l'intérêt de cette approche est de faire remonter les concepts les plus importants (ici *perception* et *action*), de sorte que l'utilisateur peut se concentrer sur les aspects essentiels. Cette démarche doit donc être rapprochée d'autres travaux SMA qui s'efforcent de faire disparaître les difficultés de la programmation afin de faire ressortir les concepts les plus importants (e.g. la *coopération* dans ADELFE pour la réalisation de SMA adaptatifs [Gleizes, 2004 ; Picard, 2004] ou l'*interaction* dans IODA via JEDI-Builder pour la simulation [Kubera et al., 2011 ; Picault, 2013]).

### De l'environnement à l'agent : percepts et actions de haut niveau

Pour conclure, il nous faut souligner que l'idée de la programmation situationnelle repose sur une approche environnement-centrée, inspirée de [Chang et al., 2005]. À travers une architecture cognitive composée de trois niveaux, (1) *reality*, (2) *concept* et (3) *mind*, dans cet article les auteurs montrent l'intérêt de fournir au *mind* les *concepts* les plus évolués possibles. Autrement dit, pour pouvoir modéliser et coder efficacement un comportement, il est important que l'environnement fournisse directement les percepts qui permettent de faire la décision, et non des données brutes qu'il faut traiter pour rendre intelligible une situation (*reality*). Ceci s'avère particulièrement pertinent pour le jeu que nous venons de décrire. On ne peut pas présenter à l'agent, c'est-à-dire ici au joueur, des données telles que la position des ennemis ou des alliés, idem pour les actions.

8. Bien qu'il fût quasiment terminé (3D, tutoriels, game design, etc.), le jeu n'a malheureusement jamais vu le jour suite à une restructuration de Feerik qui s'est alors concentré sur ses deux licences phares.

Ceci nous ramène à des conclusions que nous avons déjà rencontrées précédemment. Notamment, dans le cadre du TNI, nous sommes parti du comportement des agents et avons argumenté sur l'intérêt d'en transformer une partie en dynamiques environnementales. Ici, nous avons commencé par identifier la nécessité d'élaborer un environnement doté de moyens de perception et d'action de haut niveau. Mais, fondamentalement, il s'agit bien des deux faces d'une même pièce. La modélisation de l'environnement permet de simplifier la programmation des comportements, grâce au déplacement de la complexité des agents vers l'environnement. Celle-ci doit donc être le plus poussé possible, qu'on commence par les agents ou par l'environnement n'a pas d'importance, il suffit de garder à l'esprit ce principe de conception.

# Chapitre 8

# Simulation multi-agents et calcul haute performance sur GPU

Depuis 2010, notre projet de recherche s'est axé autour du calcul haute performance (*High Performance Computing* HPC) pour la simulation multi-agents. Nous nous sommes tourné en particulier vers l'utilisation de la puissance computationelle des cartes graphiques via le GPGPU (*General-Purpose computing on Graphics Processing Units*). Notre motivation est double. Dans un premier temps, il s'agit pour nous d'apporter des solutions aux problèmes de performances qu'on peut rencontrer pour simuler les modèles multi-agents. Plus fondamentalement, nous souhaitons explorer les conséquences théoriques liées à l'utilisation des paradigmes de programmation massivement parallèles, le GPGPU n'étant qu'une instance de ces derniers. Dans ce cadre, nos contributions actuelles [1] reposent sur une approche environnement-centrée qui consiste à transformer une partie des calculs liés aux comportements agent en dynamiques environnementales, dont l'exécution est déportée sur des architectures massivement parallèles. Autrement dit, il s'agit précisément de déplacer la complexité des agents vers l'environnement. Cette approche nous a permis d'adresser les deux principaux verrous liés à la technologie GPGPU : l'accessibilité et la réutilisabilité.

## 8.1 GPGPU et simulation multi-agents : intérêts et limites

### Contexte et motivations

Les modèles multi-agents sont souvent très gourmands en ressources de calcul. Cela ne serait rien si la question revenait uniquement à un problème de temps, et qu'il suffisait d'attendre le résultat de la simulation pour l'analyser. Ce n'est pas le cas avec la simulation multi-agents, au moins pendant la phase de conception du modèle. En effet, les modèles multi-agents se construisent le plus souvent par interaction avec les simulations qu'ils produisent.

1. [Hermellin et Michel, 2015a,b ; Michel, 2012, 2013a,b, 2014].

Comme on ne sait pas a priori ce que les comportements du niveau microscopique vont globalement produire, l'approche ascendante nécessite un retour quasi permanent sur les modifications qu'on apporte au modèle au fur et à mesure de sa conception : il faut observer sa dynamique pour le concevoir. Par conséquent, le temps de réponse d'une simulation est extrêmement important, car c'est lui qui détermine votre capacité à explorer un modèle. C'est une contrainte très forte sur la manière dont les modèles multi-agents sont conçus. Lorsqu'un modèle consomme trop de ressources de calcul, il est très vite nécessaire de le simplifier, et donc de faire des compromis (parfois drastiques) sur la modélisation, voire de l'abandonner pour un autre.

La thématique du HPC constitue ainsi une part grandissante des recherches menées en simulation multi-agents (voir Repast HPC par exemple [Collier et M. North, 2012]). Dans notre cas, c'est à la suite de plusieurs collaborations, académiques et industrielles, qu'elle est devenue une évidence.

Tout d'abord dans le contexte de l'organisation de sessions spéciales liées à la simulation en générale (SIM@SAC 2008 et 2009, cf. cv). Organisés conjointement avec Giuseppe Vizzari (Milan), ces événements nous ont permis de constater l'importante proportion de travaux liés au HPC. Ensuite, dans le cadre de deux collaborations nationales (projet EPIS [2] [Blanchart et al., 2011], cf. cv) où nous avons travaillé directement avec des personnes impliquées dans le HPC, puis via une interaction avec Frédéric Gruau, qui travaille sur une vision décentralisée du parallélisme basée sur les automates cellulaires [Gruau et Michel, 2011]. Mais c'est, début 2010, le montage du dossier d'incubation de la société MEZOA, au sein du Languedoc-Roussillon Incubation, qui nous a permis de lancer concrètement nos premières recherches sur le thème du GPGPU pour la simulation multi-agents [3].

Fin 2010, ce dossier s'est concrétisé par une collaboration au cours de laquelle nous avons supervisé deux stages (quatrième année à polytech Montpellier, en 2011). L'objectif était de programmer sur GPU, la dynamique d'évaporation et de diffusion de phéromones digitales, avec en arrière plan une réutilisation dans TurtleKit. Sur ce point, ces stages ont été des échecs dans le sens où la complexité des solutions produites n'a permis aucune réutilisation. D'un autre point de vue, ce résultat nous a permis d'identifier, par l'expérience, les deux verrous majeurs liés au GPGPU : l'accessibilité d'une part, c'est-à-dire la difficulté de programmer pour le GPU, et la réutilisabilité du code d'autre part.

### Verrous majeurs : accessibilité et réutilisabilité

Le GPGPU [4] repose sur l'utilisation de l'architecture massivement parallèle des GPU (*Grahics Processing Units*). Un GPU contient un nombre bien plus important d'unités de

2. Proposition d'une infrastructure dédiée à automatiser la réalisation de multiples simulations en parallèle sur un *cluster*, par exemple pour aider à une analyse de sensibilité.
3. MEZOA voulait utiliser le GPGPU pour réaliser des simulations multi-agents de croissance de plantes.
4. Voir la thèse de [Bourgoin, 2013] pour une introduction très bien faite sur le sujet.

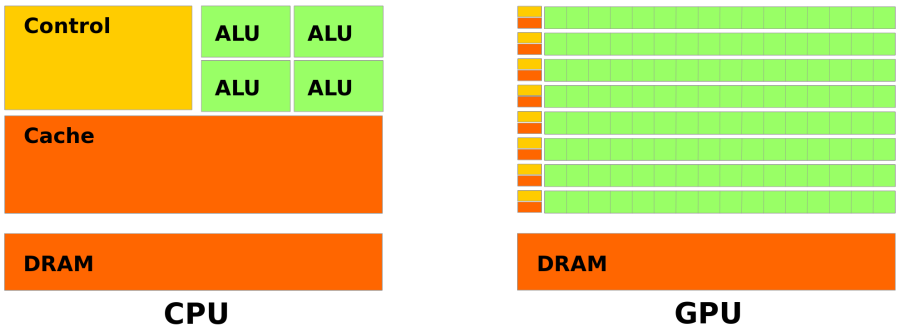



FIG. 8.1 – CPU versus GPU

calcul (ALU) q'un CPU (figure 8.1). Ces dernières sont beaucoup moins puissantes que celles des CPU mais leur nombre fait leur force. Les GPU de dernière génération sont composées de plusieurs milliers d'ALU et leurs performances théoriques explosent littéralement (figure 8.2). Outre les performances, l'avantage de cette technologie est d'être disponible sur quasiment tous les ordinateurs grand public, toutes les dernières cartes 3D en sont équipées.

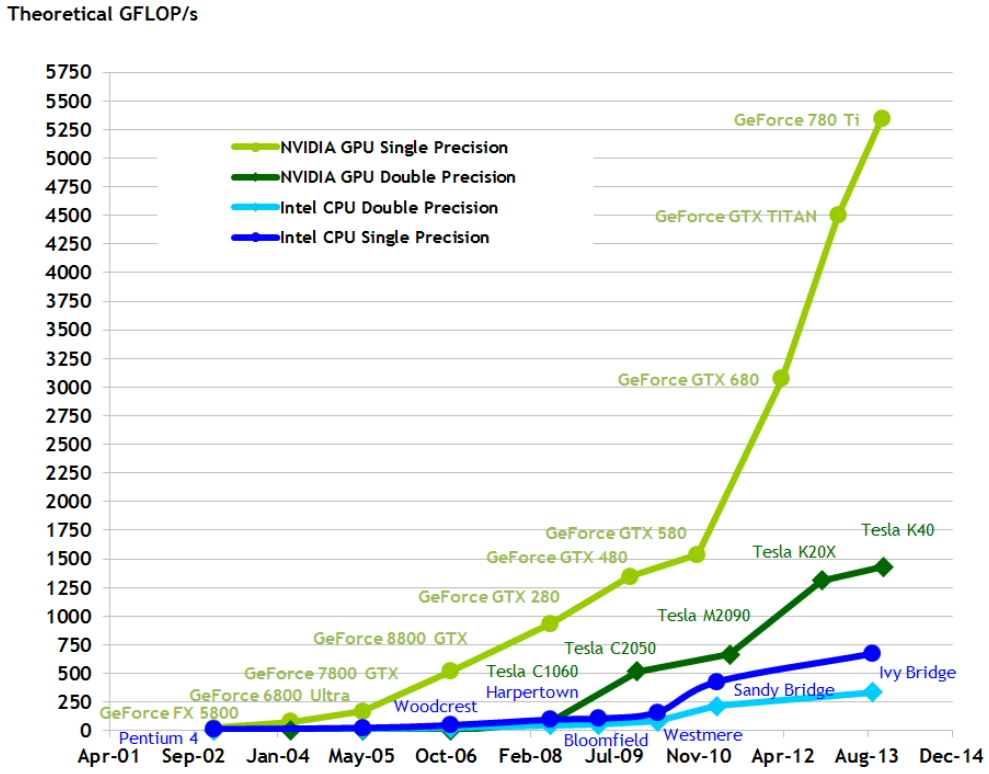



FIG. 8.2 – Évolution des performances théoriques des GPU

Utiliser le GPGPU est cependant difficile. L'architecture spécifique des GPU impose des contraintes très fortes sur à la fois, ce qui peut être parallélisé, et la manière de le faire [Che et al., 2008 ; Owens et al., 2007]. En particulier, le GPGPU s'appuie sur un parallélisme

de type SIMD (*Single Instruction Multiple Data*) [M. J. Flynn et Rudd, 1996][5], où une même opération est répétée sur des données différentes. Cette *parallélisation des données*, s'oppose au *parallélisme de traitement* où plusieurs instructions différentes sont exécutées simultanément. En particulier, le contexte est très différent de celui de la distribution des simulations sur plusieurs machines[6]. Le SIMD nécessite notamment de suivre les principes de la programmation par traitement de flot de données (*stream processing paradigm*) : étant donné un flot de données, une série d'opérations (*kernel*) est appliquée à chaque élément du flot.

Ainsi, non seulement le GPGPU est difficile d'accès, mais il est aussi si contraint qu'il oblige à repenser la modélisation multi-agents, notamment car il n'est pas possible d'adopter une conception orientée objet classique [Lysenko et D'Souza, 2008]. Les modèles multi-agents usuels ne peuvent donc être simulés sur GPU sans un effort de traduction conséquent et non trivial. Certains travaux tentent de pallier cette difficulté en encapsulant l'usage du GPGPU dans un langage de plus haut niveau (e.g. le XML dans [Richmond et al., 2010]), mais ces solutions restent compliquées du point de vue de l'accessibilité.

Il est donc compréhensible que peu de travaux soient enclins à investir du temps dans l'utilisation du GPGPU car la pérennité du code produit est difficile à obtenir. Pour l'instant, les travaux mêlant GPGPU et SMA sont majoritairement liés à des expérimentations très spécifiques, ce qui ne permet pas leur réutilisation. Aucune des plates-formes génériques connues que sont par exemple NetLogo [Sklar, 2007], RePast [M. North et al., 2013] ou encore MASON [Luke et al., 2005], n'intègrent aujourd'hui de GPGPU.

## 8.2 Délégation GPU des perceptions agents

Les premiers travaux réalisés mettent cependant en évidence la correspondance entre les modèles d'environnement de type grille de cellules (e.g. TurtleKit), et l'architecture des GPU, qu'on peut voir comme une grille d'ALU. Ainsi, notre intuition était que, grâce à une séparation explicite entre le modèle de l'environnement et le modèle de l'agent, il devait être possible, d'une manière ou d'une autre, de profiter du GPGPU dans TurtleKit sans être obligé d'en arriver à une implémentation *tout-sur-GPU*, difficilement réutilisable.

Lorsque nous avons commencé l'intégration du GPGPU dans TurtleKit en 2012, notre objectif majeur était donc d'obtenir des gains de performances tout en évitant les écueils de la programmation GPU, c'est-à-dire en conservant l'accessibilité et la réutilisabilité de la plate-forme, et en particulier son interface de programmation (API) orientée objet.

5. Flynn a proposé cette taxonomie du parallélisme dans [M. Flynn, 1972], qui est souvent référencée pour cela. Ici, le lecteur lui préférera [M. J. Flynn et Rudd, 1996] qui contient une conclusion très intéressante sur la façon dont Flynn envisageait le parallélisme comme une dimension essentielle des architectures informatiques à venir.

6. Dans ce cas les problématiques peuvent être liées à la gestion de la cohérence, de la synchronisation ou de la répartition de charges [Rihawi et al., 2013 ; Sébastien, 2010 ; Suryanarayanan et Theodoropoulos, 2013]. De ce point de vue-là le GPGPU est aussi très intéressant car il ne nécessite pas de disposer d'un *cluster*.

Dans [Michel, 2014], nous revenons dans le détail sur la méthode employée pour atteindre cet objectif. En particulier, nous nous sommes focalisé sur l'optimisation de simulations du modèle MLE [Beurier et al., 2003], un modèle simple qui génère des structures complexes de manière fractale à partir d'un unique comportement (figure 8.3) [7].

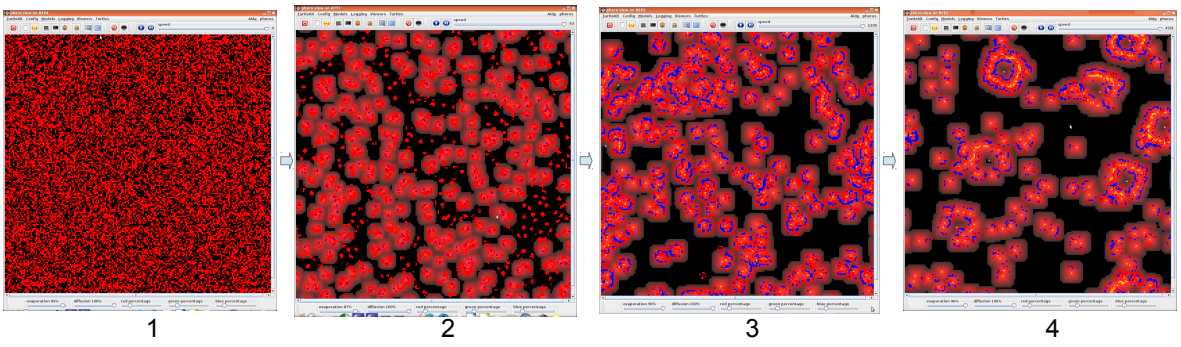



FIG. 8.3 – Exemple d'évolution du modèle MLE

Outre son intérêt intrinsèque d'un point de vue vie artificielle, ce modèle présentait un intérêt pour nous car il requiert énormément de ressources de calcul, notamment car il nécessite la diffusion et l'évaporation d'un grand nombre de phéromones digitales, ce qui avait rendu très difficile son étude lors de sa conception.

À partir d'une décomposition claire entre agent et environnement, l'idée consiste à utiliser une approche hybride (figure 8.4) : les agents sont exécutés par le CPU (code inchangé) et les dynamiques environnementales sont déportées sur le GPU (phéromones).

FIG. 8.4 – Principe de l'approche hybride

7. Des vidéos de simulations du modèle MLE sont disponibles sur http ://www.lirmm.fr/ fmichel/mle/

Une illustration du code correspondant à la dynamique d'évaporation permet de saisir l'intérêt de la programmation GPU pour réaliser ces calculs. La double boucle présente en séquentiel disparaît dans la version GPU [8]. Cependant, elle n'est pas remplacée par du *code parallèle* : c'est l'architecture matérielle qui la réalise. En GPGPU, *le code est dans la structure.*

**Algorithme 1 :** évaporation en séquentiel

**Entrées** : Une *grille* de cellules représentant une phéromone, sa *hauteur*, sa *largeur* et le coefficient d'évaporation à lui appliquer, *evapCoef*.

```
début
    pour i=1 à largeur faire
        pour j=1 à hauteur faire
            grille[i][j] ← grille[i][j] × evapCoef ;
        fin
    fin
fin
```

**Algorithme 2 :** évaporation en GPU

**Entrées** : Une *grille* de cellules représentant une phéromone, sa *hauteur*, sa *largeur* et le coefficient d'évaporation à lui appliquer, *evapCoef*.

```
début
    i ← blockIdx.x × blockDim.x + threadIdx.x ;
    j ← blockIdx.y × blockDim.y + threadIdx.y ;
    si i < largeur et j < hauteur alors
        grille[i][j] ← grille[i][j] × evapCoef ;
    fin
fin
```

Grâce à une distinction explicite entre les agents et l'environnement, nous avons pu facilement intégré dans TurtleKit un *module GPU* correspondant aux dynamiques d'évaporation et de diffusion (*GPU diffusion*), sans avoir à changer quoi que ce soit à l'API du modèle d'agent, et donc en préservant l'accessibilité et la réutilisabilité de la plate-forme. La figure 8.5 montre comment ce module a été intégré à TurtleKit de manière transparente.

8. Dans la version GPU, $i$ et $j$ permettent de faire la correspondance entre une donnée de la grille et la localisation d'une unité de calcul (*thread*) sur le GPU. Le test permet de savoir si les coordonnées du *thread* ne sont pas en dehors de la grille de données.

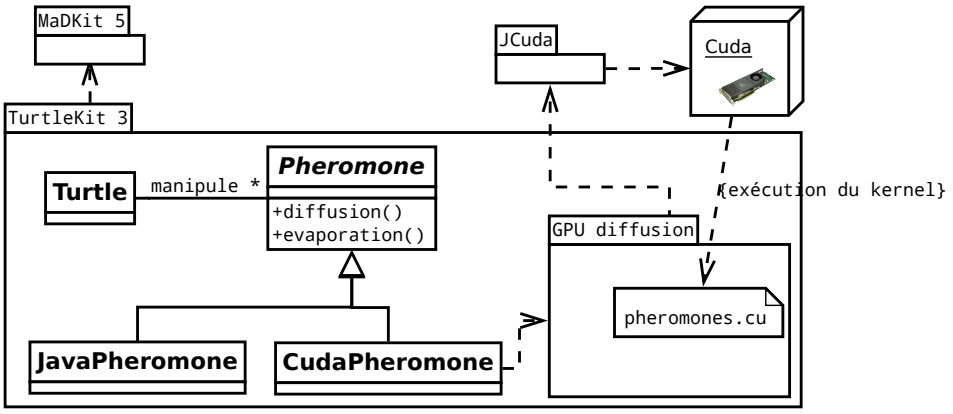



FIG. 8.5 – Intégration transparente du module *GPU diffusion* dans TURTLEKIT 3

Nous avons rapidement obtenu de bons résultats quantitatifs. Le modèle MLE nécessite la simulation de beaucoup de phéromones et l'amélioration des performances était attendue. En revanche, les calculs liés aux comportements des agents représentaient clairement un goulot d'étranglement qu'il nous fallait adresser. C'est là qu'une approche environnement-centrée nous a permis d'avancer.

En fait, nous avons remarqué qu'une partie des perceptions effectuées par les agents n'impliquait nullement leur état interne. En effet, dans le modèle MLE, les agents calculent les gradients locaux des phéromones qui les entourent afin de choisir une direction. Ce calcul, extrêmement coûteux, peut être fait sans connaître l'état d'un agent, autrement dit le résultat est le même quel que soit l'agent qui le réalise. Par conséquent nous avons pu déporter ces calculs sur le GPU, en les transformant en dynamique environnementale (*module GPU gradients*). Autrement dit, c'est maintenant l'environnement qui précalcule, pour toute la grille, le résultat des perceptions dont les agents auront besoin. De cette démarche, nous avons généralisé un principe de conception appelé *délégation GPU des perceptions agents* :

> *Tout calcul de perception agent qui n'implique pas l'état de l'agent peut être transformé dans une dynamique endogène de l'environnement, et ainsi considéré pour une implémentation dans un module GPU indépendant.*

Ce principe implémente plusieurs aspects de l'approche environnement-centrée :

— tout d'abord il s'agit de déplacer la complexité des agents vers l'environnement. En l'occurrence, il s'agit à la fois de la (1) complexité du code comportemental et de la (2) complexité algorithmique des calculs effectués. En effet, il permet de simplifier l'écriture du comportement des agents, et ainsi de se focaliser sur l'essentiel (1) et de déplacer la complexité, en termes de temps calcul, du CPU (agents) vers le GPU (environnement) (2). Dans le contexte du GPGPU, il permet donc de faire d'une pierre deux coups.

— deuxièmement, ce principe repose clairement sur la modélisation de percepts de haut niveau, gérés par l'environnement, qui ont pour but de faciliter la délibération des agents.

— Ce principe suggère par ailleurs un prétraitement systématique des données environnementales : les données de perceptions ne sont pas calculées *à la demande* des agents, mais consistent plutôt dans la nature même de l'environnement et de sa dynamique. Pour l'instant cet aspect n'est rentable que dans le contexte GPGPU (calculer toutes les perceptions n'est pas efficace en CPU), mais nous pensons que le contexte *massivement parallèle* qu'offre le GPGPU est une dimension qui sera de plus en plus présente dans les systèmes informatiques à venir, comme déjà suggéré dans [M. J. Flynn et Rudd, 1996].

Pour les travaux qui prennent en compte les problématiques d'accessibilité et de réutilisabilité du GPGPU, on retrouve des approches environnement-centrées. Par exemple dans [Laville, 2014], les auteurs proposent une librairie contenant un ensemble d'outils qui permettent de lancer des dynamiques multi-agent (le plus souvent environnementales) sur le GPU de manière transparente. Cela est rendu possible grâce à une distinction nette entre agents et environnement.

## 8.3 GPGPU et SMA : un état des lieux

Nous avons obtenus nos premiers résultats courant 2012 [Michel, 2012]. Sur cette base, fin 2012 nous avons défendu un sujet de thèse sur ce thème (HPC + SMA) afin d'obtenir un financement ministériel. Grâce à un bon classement du sujet, nous avons eu l'opportunité de recruter Emmanuel Hermellin, qui a commencé sa thèse en octobre 2013. Ce sujet étant l'expression de notre propre projet de recherche, nous codirigeons officiellement [9] Emmanuel avec Jacques Ferber.

Le premier travail entrepris a consisté à effectuer un état de l'art exhaustif des travaux mêlant GPGPU et SMA. Ce domaine évolue vite et il s'agissait de réexaminer les directions de recherche prises récemment. Cet état de l'art [Hermellin et al., 2014, 2015] montre clairement que la plupart des travaux restent focalisés sur les performances brutes dans un contexte applicatif. Ils n'ont pas pour objectif d'être accessibles ou réutilisables. On voit par contre clairement un intérêt montant pour les approches hybrides (CPU + GPU), comme le montre la figure 8.6 (les références des travaux positionnés sur cette frise sont dans l'article). Celles-ci permettent effectivement d'envisager des SMA de différentes natures et de mélanger architectures réactives et cognitives, au contraire des approches *tout-sur-GPU* [Pavlov et J. Müller, 2013].

9. Grâce à une demande d'accréditation effectuée auprès de l'école doctorale I2S en juillet 2013, procédure qui permet aux enseignants chercheurs proches de l'HDR de codiriger officiellement des travaux doctoraux.

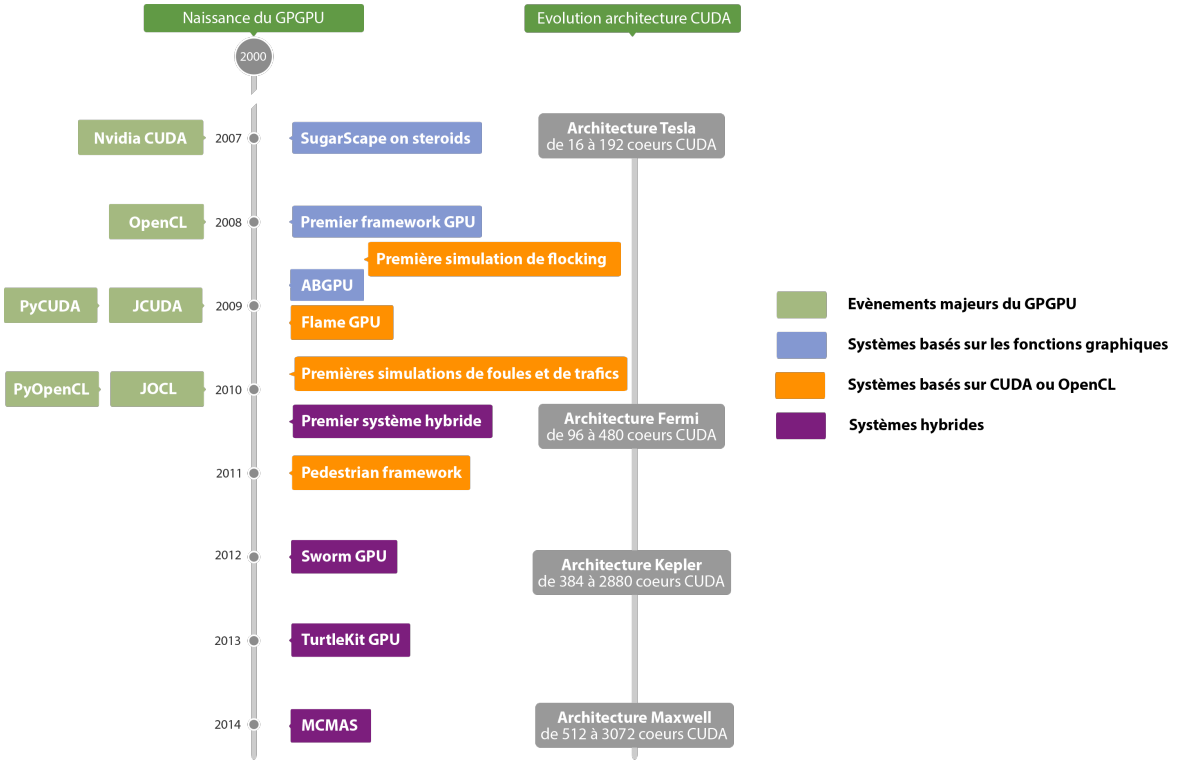



FIG. 8.6 – Évolution des approches mêlant GPGPU et SMA [Hermellin et al., 2014]

Cette évolution, du *tout-sur-GPU* vers des approches hybrides nous a conforté dans l'idée de poursuivre notre démarche. Ainsi, les tout derniers travaux que nous poursuivons actuellement consistent à éprouver le principe de délégation GPU sur d'autres modèles multi-agents, cela afin de mesurer son pouvoir de généricité et sa pertinence.

## 8.4 Délégation GPU avec les *boids* de Reynolds

Ainsi, après MLE, le deuxième modèle que nous avons étudié est un classique des multi-agents : les *boids* de Reynolds. Reynolds avait pour objectif de réaliser des animations réalistes de nuées d'oiseaux artificiels (*flocking*) [Reynolds, 1987]. Reynolds s'est rendu compte qu'il n'était pas possible d'utiliser des scripts supervisés pour réaliser ce genre d'animations et a donc proposé un modèle multi-agents basé sur trois règles comportementales pour les *boids* :

— **R.1** *Collision Avoidance* : éviter les collisions entre entités ;
— **R.2** *Flock Centering* : rester le plus proche possible des autres entités ;
— **R.3** *Velocity matching* : adapter sa vitesse à celles des autres entités.

Dans [Hermellin et Michel, 2015a,b], nous avons effectué une revue des différentes implémentations que nous avons pu trouver dans les principales plates-formes. Il est intéressant de noter qu'il n'en existe pas deux de semblables et que, même si les différents comportements globaux observés peuvent être qualifiés de qualitativement équivalents, dans les détails ils

sont tout de même assez différents d'une plate-forme à l'autre. De plus, les codes source correspondants sont relativement complexes et il est assez difficile d'y retrouver les règles de Reynolds de manière explicite.

Ainsi, dans notre propre implémentation des boids dans TurtleKit nous nous sommes inspirés des implémentations existantes en essayant d'expliciter les trois règles de Reynolds dans le code. Des vidéos du comportement global obtenu sont disponibles sur le site d'Emmanuel [10].

Une fois ce modèle établi, nous avons alors appliqué le principe de délégation GPU. En l'occurrence, il est possible de déporter une partie importante des perceptions. En effet, dans le comportement de cohésion, chaque agent effectue une perception qui consiste à récupérer l'ensemble des orientations de ses voisins afin d'en calculer la moyenne. Il est facile de voir que le résultat de cette perception n'a rien à voir avec l'état de l'agent qui la réalise. Nous avons donc pu créer un nouveau module GPU (*average*) qui implémente cette opération en tout point de l'espace, c'est-à-dire sous la forme d'une nouvelle dynamique environnementale qui calcule les moyennes à chaque cycle de simulation.

De par la simplicité des calculs concernés, le codage de ce nouveau module a été rapide. Ce qui est un résultat important pour nous car il montre l'intérêt de notre méthode du point de vue de l'accessibilité, nul besoin d'être un expert en programmation GPU pour écrire les algorithmes que nous décrivons. De plus, du point de vue de la réutilisabilité, cette deuxième expérience confirme l'intérêt du principe de délégation GPU. Premièrement l'API agent est restée inchangée (orientée objet). Deuxièmement, nous avons pu utiliser le même principe architectural que lors de l'intégration des modules précédemment évoqués, comme le montre la figure 8.7.

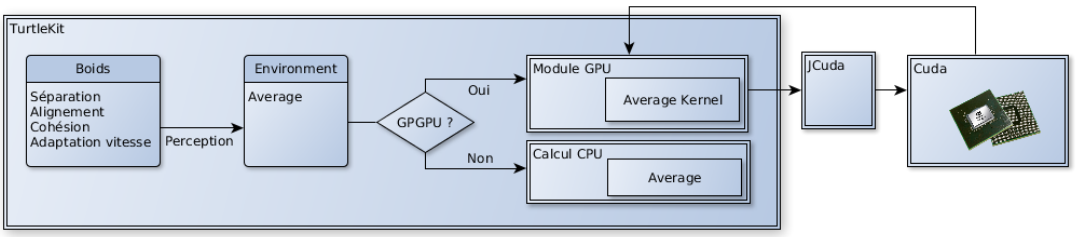



Fig. 8.7 – Intégration du module *GPU average* dans TurtleKit 3

Ces travaux sont donc des premières contributions intéressantes par rapport aux deux problématiques identifiées en début de chapitre. Le principe de délégation GPU promeut effectivement **accessibilité** et **réutilisabilité**, en particulier dans un contexte mêlant GPGPU et SMA où elles sont très difficiles à obtenir.

10. Les Boids de Reynolds avec TurtleKit 3.

Pour conclure sur ces travaux, notons qu'une piste de recherche que nous souhaitons explorer concerne la désynchronisation entre le CPU et le GPU. Par défaut la simulation évolue en alternant les calculs agents (CPU) avec les dynamiques environnementales (GPU) de manière synchrone. Lorsqu'on supprime cette synchronisation, on parallélise de fait le CPU et le GPU si bien que l'exécution devient encore plus rapide. Mais le plus intéressant est que, malgré le fait que la simulation ne respecte plus un ordre causal prédéfini (la réplication est impossible), les comportements globaux obtenus dans MLE et dans le *flocking* continuent de donner des dynamiques globales qualitativement équivalentes. On peut donc tester la robustesse et l'adaptativité des comportements face aux changements de dynamiques environnementales, et ainsi leur potentiel de résilience. Ici, les dynamiques collectives restent robustes pour différentes vitesses d'évolution de l'environnement.

# Chapitre 9

# Perspectives et conclusion

Les recherches que nous menons depuis plusieurs années visent à faire avancer l'état de l'art dans le domaine de la modélisation et de la simulation multi-agents. Nous nous positionnons dans un contexte générique. En particulier, nous avons vu que la simulation multi-agents peut être utilisée pour des objectifs très divers : étude de systèmes complexes (e.g. vie artificielle), résolution de problèmes (e.g. segmentation d'image), conception de SMA ancrés dans le réel (e.g. robotique mobile collective) ou encore le développement de logiciels basés sur des simulations (e.g. jeux vidéo). Les modèles et les outils que nous proposons ont donc pour objectif d'être utilisables quel que soit le contexte applicatif. À long terme, il s'agit pour nous de faire évoluer les simulations multi-agents de sorte qu'elles permettent d'explorer de nouvelles dynamiques collectives, qualitativement différentes de celles qui existent aujourd'hui, grâce à des approches et des outils originaux.

## 9.1 Positionnement

Le projet de recherche que nous avons démarré sur le calcul haute performance et la simulation multi-agents constitue en soi une thématique qui sera bien sûr au cœur de nos préoccupations dans les années à venir. Les performances constituent un verrou majeur et les orientations qui sont prises en direction du calcul haute performance par différents groupes de recherche le montrent clairement. *IBM Research* parle par exemple de simulations comptant des milliards d'agents, distribuées sur des grappes de serveurs comptant des centaines de nœuds [Suzumura et Kanezashi, 2012].

Dans le même temps, la grande diversité des approches existantes témoigne à la fois de la richesse du paradigme multi-agents mais aussi de la complexité liée à la conception et à l'utilisation des simulations multi-agents. En cela, il nous semble que nous sommes actuellement à un tournant crucial pour les MABS : l'augmentation des capacités de calcul ainsi que la qualité des outils existants ont permis une envolée spectaculaire du nombre de simulations multi-agents, mais nous sommes dans une situation où la complexité des modèles manipulés nous échappe

en grande partie. Comprendre les dynamiques des modèles les plus simples n'est toujours pas une évidence. Le travail mené par Chevrier et Fatès [2008] sur le modèle influence/réaction le montre bien. Même avec seulement deux agents aux comportements minimalistes, plongés dans un environnement simple, il est possible d'obtenir des dynamiques très différentes suivant la manière dont le reste du modèle est élaboré (perception, action, temps, interaction, etc.). Ainsi, de la même manière que M. J. Flynn et Rudd [1996] expliquaient que les architectures parallèles évolueraient sans doute plus vite que nos capacités à les utiliser, il semble que ce soit aussi le cas avec les MABS. Ne prendrons-nous pas un risque épistémologique en tirant des conclusions à partir de simulations impliquant un milliard d'agents si certaines subtilités nous échappent encore lorsqu'ils ne sont que deux ?

En soi, ce n'est pas un problème d'un point de vue applicatif. Par exemple, en robotique collective, si la simulation peut aider concrètement à concevoir une architecture multi-agents, alors elle aura pleinement joué son rôle. Ce qui est aussi valable dans le cadre de la résolution de problèmes : nul besoin de formaliser, de valider ou même de comprendre le processus de simulation s'il permet de répondre effectivement aux objectifs. Et, même dans le cadre de l'étude des systèmes complexes, un modèle est avant tout un moyen d'explorer des hypothèses ; son objet n'est pas de représenter la réalité. Dans ce cadre, il constitue un moyen de se poser des questions sur un phénomène, au sens énoncé par Minsky [1965] : "*To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A*".

Ce raisonnement a cependant ses limites dans le sens où il ne permet pas de remettre en cause nos modèles et nos pratiques, freinant ainsi leur évolution et leur potentiel à générer des connaissances nouvelles. Notamment, en tant qu'informaticien, nous ne devons pas nous arrêter à l'idée qu'un modèle de simulation n'existe que pour répondre à des questions sur un phénomène $A$. Si les modèles de simulation sont souvent destinés à appréhender le monde qui nous entoure, il est aussi important de considérer qu'ils deviennent à leur tour des phénomènes qui ont leur propre réalité au sein de nos ordinateurs. Ils sont, eux aussi, des objets de type $A$. La grande différence avec les systèmes complexes naturels est que la manière dont ils fonctionnent nous est pleinement accessible. Il est donc possible de l'étudier dans les détails et il est fondamental de ne pas s'en priver, sans quoi nous nous privons aussi d'une partie de leur potentiel.

C'est pourquoi, si nous avons pris le risque de nous tourner vers le GPGPU pour adresser le problème des performances, c'est précisément parce que notre but n'a jamais été uniquement d'accélérer les simulations existantes. Contrairement à la distribution sur plusieurs machines, le GPGPU offre un contexte technologique très novateur, radicalement différent de celui qu'on trouve dans les plates-formes usuelles ; nous y voyons une formidable opportunité de repenser les modèles multi-agents et ainsi d'étudier de nouvelles formes d'environnements, d'agents et d'interactions. Cette technologie est donc essentiellement pour nous un moyen d'aller *plus vite* vers ce qu'il sera possible de faire avec la simulation multi-agents dans les années à venir, GPGPU ou pas.

De fait, dans les solutions que nous apportons pour la simulation multi-agents et le calcul haute performance, il s'agit avant tout pour nous d'élaborer des approches qui ne remettent pas en cause notre capacité à traiter d'autres problématiques. La réutilisabilité et la généricité restent donc des points essentiels de notre projet. C'est notamment la raison pour laquelle notre démarche s'oppose aux approches mêlant GPGPU et MABS dans le seul but d'obtenir les meilleures performances possibles. Si celles-ci sont pertinentes dans un contexte applicatif spécifique, en revanche elles permettent peu de faire évoluer l'expressivité des modèles multi-agents grâce à une capitalisation des efforts de modélisation et d'implémentation.

Ainsi, après avoir décrit nos objectifs à court terme, nous présenterons ici deux perspectives de recherche (niveau agent et niveau multi-agents) dans lesquelles nous devrions pouvoir obtenir des résultats originaux, grâce à nos travaux sur le GPGPU, sans qu'ils soient pour autant limités à ce contexte. La première se situe plus spécifiquement au niveau de la modélisation des comportements agents et repose sur une exploration plus approfondie du principe de délégation. La deuxième concerne l'étude de nouvelles dynamiques collectives, notamment grâce à une réintégration plus systématique du modèle IRM4S dans nos recherches.

## 9.2 Vers une méthodologie pour la simulation sur GPU

Sur la thématique de la simulation multi-agents sur GPU, même s'ils sont encourageants, les premiers résultats que nous avons obtenus pour l'instant ne sont que préliminaires. Ils représentent une preuve de concept mais un certain nombre d'étapes restent à franchir afin de constituer un ensemble cohérent de modèles, d'outils et de principes méthodologiques associés au contexte du GPGPU. Ils doivent donc être consolidés de manière à ce que nous puissions par la suite nous appuyer sur eux pour développer les autres pistes de recherche que nous souhaitons aborder.

À court et moyen termes, les directions à prendre sont assez claires. Il s'agit d'abord de travailler sur les outils logiciels, afin de les rendre réellement opérationnels et accessibles, puis d'établir une méthodologie associée à notre démarche de conception de simulations hybrides, c'est-à-dire fondée sur le principe de délégation GPU.

### Court terme : utilisation GPGPU transparente dans TurtleKit 3

Dans l'immédiat, pour améliorer l'accessibilité et la réutilisabilité des outils de TurtleKit 3 qui sont liés au GPGPU, il est important de finaliser son architecture. Pour l'instant, TurtleKit 3 est toujours en version alpha et la réalisation de simulations accélérées grâce au GPU nécessite de manipuler son noyau. Pour atteindre cet objectif, il nous faut tout d'abord appliquer le principe de délégation sur plus de modèles multi-agents. À l'heure actuelle, nous n'avons pas encore assez de recul sur son utilisation et donc sur la meilleure façon de l'implémenter. Ceci nous permettra par ailleurs d'évaluer expérimentalement la capacité

de notre approche à produire des éléments facilement réutilisables dans d'autres simulations et par là ses avantages et ses limites.

Nous pensons en particulier à élaborer une bibliothèque de dynamiques environnementales que nous construirons de manière itérative et modulaire. Par exemple, les modules GPU (*diffusion*, *gradients*, *average*) présentés dans le chapitre précédent ne sont pas liés à un modèle multi-agents spécifique. Ces modules sont donc suffisamment génériques pour pouvoir être utilisés dans d'autres modèles. Nous devrons donc faire évoluer l'architecture de TurtleKit afin d'en simplifier l'usage. Ici, notre objectif sera de rendre le GPGPU accessible au plus grand nombre en facilitant l'implémentation de simulations l'utilisant de manière transparente, grâce aux dynamiques environnementales que nous aurons prédéfinies.

### Moyen terme : faciliter la programmation GPU et les simulations hybrides

L'expérience que nous accumulerons au fur et à mesure de nos expérimentations devrait nous permettre d'avoir suffisamment de recul pour pouvoir élaborer une véritable méthodologie de conception associée au principe de délégation GPU et à la réalisation de simulations hybrides.

Ici, il ne s'agit plus du tout de cacher la programmation GPU. Au contraire, notre objectif sera de permettre à un développeur n'ayant aucune expérience dans ce domaine de l'aborder d'une manière simple, et surtout itérative. Il est aujourd'hui difficile d'aborder la simulation multi-agents sur GPU sans avoir des connaissances relativement pointues, notamment car il n'est pas évident de voir ce qui peut être déporté sur le GPU dans un modèle multi-agents.

Grâce au principe de délégation GPU, et à l'approche hybride sous-jacente, l'objectif de cette méthodologie sera de permettre le développement de modules GPU dont le code sera très simple à concevoir. L'aspect itératif et modulaire de notre approche devrait aussi permettre de faire en sorte que cette méthodologie soit facilement accessible et simple à mettre en place. Une itération du processus de délégation GPU ne compte en effet que quelques étapes : (1) identification d'une délégation possible puis (2) implémentation du module GPU et enfin (3) intégration de ce dernier dans le moteur de simulation.

## 9.3 Vers plus de délégation dans les comportements agents

Dans le cadre de l'accessibilité et de la réutilisabilité des comportements agents, un verrou important repose sur la complexité des implémentations finalement produites, notamment du fait de la focalisation excessive sur l'individu dont nous avons parlé en introduction [Picault, 2013]. De fait, il est souvent difficile de retrouver ce qui fait l'essence du comportement d'un agent, par exemple dans le but de le reproduire dans un autre contexte.

Nous pensons que ce problème n'est pas forcément lié uniquement à un manque d'abstractions adéquates. Le fait est que, s'il est toujours possible de concevoir sur le papier des

règles comportementales explicites a priori, la programmation itérative des modèles, induite par l'approche ascendante, fait que les comportements sont parasités, au fur et à mesure, par du code qui n'a rien à voir avec la délibération. En particulier, à plusieurs reprises dans ce mémoire (sections 4.5 et 7.4), nous avons souligné l'intérêt de fournir à l'agent des percepts et des moyens d'action de haut niveau, afin de simplifier son comportement et d'en garder l'essentiel. Cela constitue pour nous un point fondamental dans la perspective du déplacement de la complexité des agents vers l'environnement.

Dans le cadre des simulations multi-agents, il existe en fait de nombreuses approches qui visent à répartir la complexité du code dans différents modules afin (1) d'extraire ce qui fait l'essence du comportement d'un agent et de (2) modulariser son implémentation. IODA [Kubera et al., 2011] (séparation déclaratif/procédural), EASS (filtrage des perceptions) [Badeig et Balbo, 2012] ou encore [Payet et al., 2006b] (réduction de la complexité du modèle basée sur une approche environnement-centrée) sont autant d'exemples qui permettent de réduire la complexité des agents. Ces approches s'appuient sur une séparation entre les calculs liés aux comportements, aux perceptions, aux actions ou aux interactions. Plus généralement, on peut rapprocher ces démarches des travaux dédiés à l'AOSE. Par exemple, nous avons vu que Ricci et al., 2011 proposent une simplification du comportement des agents grâce à un codage explicite de services réifiés sous la forme d'*artifacts* (environnement).

Cependant, si ces approches permettent effectivement de décomposer et donc de simplifier le codage des comportements, nous pensons qu'elles pourraient être avantageusement complétées par une méthodologie permettant d'identifier ce que doit être une perception ou une action d'un point de vue qualitatif. En effet, quelle que soit la décomposition qui est opérée a priori, si les percepts sont in fine trop complexes, on a inévitablement une complication du code correspondant au comportement de l'agent.

Dans cette perspective, l'intérêt du principe de délégation que nous avons proposé est de fournir un critère générique, indépendant de l'interprétation, qui permet une analyse a priori, mais aussi a posteriori, une fois le code produit, de ce qui peut être déplacé en dehors de l'agent. En cela, il constitue un moyen d'analyse supplémentaire dans le cadre du déplacement de la complexité. Ainsi, nous comptons étudier dans quelle mesure il est possible d'étendre ce principe à d'autres aspects du comportement de l'agent, grâce à de nouveaux critères similaires, c'est-à-dire basés sur des propriétés indépendantes de l'interprétation, et donc du contexte applicatif. En première approche, il sera par exemple intéressant d'essayer d'étendre le principe de délégation aux actions.

Dans ce cadre, nos objectifs seront donc en partie liés aux résultats que nous obtiendrons avec la méthodologie décrite dans la section précédente. Si le principe de délégation GPU confirme son intérêt sur une grande diversité de modèles multi-agents et que nous arrivons à en extraire une méthodologie convaincante, nous pourrons alors étudier dans quelle mesure nos conclusions pourront être étendues aux MABS en général.

Dans le même ordre d'idée, nous serons aussi intéressé à reprendre les travaux que nous avons effectués dans le cadre de la programmation situationnelle (cf. section 7.4). En effet, dans ce travail il s'agissait précisément de fournir des moyens de se focaliser sur l'élaboration de la délibération, grâce à une vision haut niveau des perceptions et des actions, dont le traitement est délégué à d'autres modules. L'idée serait donc de reprendre le concept, non pas dans sa version graphique, mais directement au niveau des interfaces de programmation.

Pour conclure sur cette perspective, on peut s'appuyer sur une remarque intéressante formulée à propos du principe de délégation et de notre vision environnement-centrée. Celle-ci concerne les limites liées à son application. Plus précisément, la question est de savoir jusqu'à quel point nous considérons que le code des agents doit être transféré dans l'environnement et si, ce faisant, les agents ne perdent pas de leur autonomie dans le sens où l'environnement finirait par *décider* de tout. La question est donc celle-ci : *déléguer, oui, mais jusqu'où ?*

De notre point de vue, tout ce qui peut être extrait du processus décisionnel doit l'être [1]. Lorsqu'on code le comportement d'un agent, rendre exploitables des données environnementales brutes (construction des percepts), ou implémenter les détails d'une action, sont autant de points qui nous éloignent de la question essentielle qui se pose au niveau de l'agent : l'élaboration de la délibération. À titre d'exemple, idéalement, des comportements comme les *boids* de Reynolds, qui ne comptent sur le papier que trois règles comportementales très simples, devraient pouvoir s'écrire sur seulement quelques lignes de codes, ce qui n'est aujourd'hui pas le cas dans les plates-formes usuelles. Ce qui complique singulièrement le partage d'expériences ainsi que la possibilité de comparer les approches qui sont réalisées dans différents contextes logiciels.

Pour autant, étant donné la multiplicité des travaux qui traitent de la réutilisabilité et de la simplification des comportements, la question précédente en soulève en fait un certain nombre d'autres. Par exemple, la modélisation du processus de perception peut prendre différentes formes (active, passive, filtrée, par interaction, etc.). Il sera donc important de mettre en perspective le principe de délégation, tel que nous l'avons pour l'instant énoncé, par rapport aux approches que nous avons citées et qui s'inscrivent dans la même démarche. Notre idée est que le principe de délégation est un moyen d'aller plus loin dans les différentes voies ouvertes par ces travaux.

## 9.4 Vers de nouveaux modèles et dynamiques multi-agents

### Le retour d'influence/réaction

Nos simulations du *flocking* donnent en fait des comportements collectifs très différents suivant qu'on les exécute avec ou sans GPU. Sans GPU, les dynamiques globales sont satis-

1. Ce qui ne remet pas en cause l'autonomie des agents au sens discuté dans la section 2.2. Les limites de la délégation se situent simplement au niveau des processus qui manipulent les variables internes de l'agent.

faisantes dans le sens où le *flocking* fonctionne bien, mais elles sont finalement assez pauvres en surprises. Avec le GPU, on voit apparaître des comportements collectifs beaucoup plus riches et surprenants, en un mot plus *complexes*. Cela s'explique en fait très bien, avec influence/réaction.

Dans la version sans GPU, où les *boids* perçoivent et agissent directement l'un après l'autre, un pas de simulation aboutit à la convergence du système vers une valeur commune pour l'orientation des agents. Le premier *boid* s'aligne et modifie sa direction, le deuxième en fait autant mais perçoit la nouvelle direction du premier et la prend en compte, et ainsi de suite. Au final, le dernier *boid* qui agit se retrouve dans un monde pour le moins très simple : tout le monde va dans la même direction, modulo une petite variation aléatoire sur les directions individuelles [2].

Dans la version avec GPU, toutes les perceptions sont précalculées par l'environnement et les agents perçoivent donc *tous* le même état du monde pour un même instant $t$. Ainsi, on obtient localement des comportements de *flocking* mais le système ne converge pas vers une valeur unique pour l'orientation au fil de la simulation.

En fait, l'approche hybride que nous utilisons impose de séparer les calculs liés aux agents (CPU), des calculs liés à l'environnement (GPU). Sans que cela n'implique d'utiliser influence/réaction dans toute la modélisation, on a en revanche nécessairement un mécanisme en deux phases qui se met en place sur les parties du modèle concernées, d'où une manifestation des effets liés à une approche de type IRM4S, avec une suppression des artéfacts dus à l'ordonnancement séquentiel des agents [3].

Il s'agit pour nous d'un résultat crucial. Dans un contexte classique, pour obtenir un *flocking plus vivant*, nous serions sans doute parti sur une modification des comportements (plus d'aléatoire, plus de perceptions, modification des règles comportementales, etc.), jusqu'à obtenir une dynamique qui nous satisfasse, ce qui aurait invariablement augmenté la complexité des agents. Ici, avec strictement le même comportement, ce sont les dynamiques collectives que nous pouvons rendre plus complexes, grâce à influence/réaction.

Si nous pouvons aller plus loin dans la simplification du comportement des *boids*, tout en conservant des dynamiques collectives complexes et que, par la suite, nous parvenons à généraliser ce résultat sur d'autres modèles, nous aurons alors un argument fort sur l'intérêt que peut avoir une approche environnement-centrée basée sur IRM4S.

Aujourd'hui, grâce aux architectures massivement parallèles, nous avons une vision plus claire de la manière dont IRM4S peut être incorporé plus systématiquement dans la modé-

2. L'aléatoire, dont les utilisateurs de MABS sont friands, n'est ici finalement rien d'autre qu'un moyen de donner *artificiellement* plus de vie au modèle. Par ailleurs, brasser la liste d'activation des agents n'apporte aucun changement : le système converge toujours vers une valeur moyenne.

3. En fait, nous sommes apparemment retombé, par une autre voie, sur des résultats similaires à ceux de [Charrier et al., 2009] qui, via une approche environnement-centrée basée sur influence/réaction (avec des précalculs de perceptions), montre une différence qualitative pour le *flocking*. Dans sa thèse, Rodolphe Charrier [2009] parle d'*une nouvelle voie de simulation du flocking* et situe le phénomène dans la synchronisation des états internes des agents. Dans son modèle, ces derniers sont réifiés directement dans l'environnement physique.

lisation et dans l'implémentation[4]. Nous serons donc particulièrement intéressé à réintégrer le modèle IRM4S dans nos travaux. Nous voyons ici la possibilité de revisiter les modèles multi-agents classiques et il y a là une piste de recherche qui nous paraît très prometteuse.

### Nouveau contexte, nouveaux modèles

Comme nous l'avons suggéré, dans le cadre de l'évolution des modèles multi-agents, nous considérons que l'intérêt des architectures massivement parallèles ne se limitent pas uniquement aux gains de performances. En fait, nous pensons que le verrou lié aux architectures classiques est si important qu'il conditionne la nature des modèles multi-agents qui viennent à l'esprit. Autrement dit, dans un contexte séquentiel, nous faisons le tri entre ce que nous sommes habitués à pouvoir faire et tout le reste ; nous savons ce qui est possible et ce qui ne l'est pas. Ce qui nous restreint énormément du point de vue des modèles multi-agents que nous envisageons. Nous nous permettons ici d'illustrer ce point de vue à l'aide d'une anecdote personnelle.

En 2012, nous avons assisté à une conférence de Guy Theraulaz sur la construction des nids de termites [Theraulaz et al., 2012] lors de la conférence SASO. Nous commencions alors tout juste à avoir nos premiers résultats avec les phéromones sur GPU et son exposé nous a amené à nous poser une question, empreinte d'une arrière-pensée comptable d'informaticien : *combien de phéromones différentes les fourmis utilisent-elles* ? Lorsque nous avons pu lui poser la question, sa réponse de biologiste fut : "*Oh, tu sais, les fourmis vivent dans un monde chimique*". Nous ne nous attendions pas à une autre réponse dans le sens où nous avions l'intuition que les fourmis utilisent "beaucoup" de phéromones, mais celle-ci était livrée d'une manière qui nous a donné beaucoup à réfléchir. Non seulement le nombre de phéromones est important, mais la diversité des dynamiques liées à chacune d'elles permet sans doute des subtilités importantes dans les communications chimiques, et donc dans les dynamiques collectives possibles.

Avec les architectures classiques, les phéromones digitales coûtent très cher à mettre en œuvre. Lorsqu'on les utilise, il faut se restreindre à quelques unités, avec des dynamiques simples, dans des environnements de faible taille, sans quoi le modèle reste inexploitable. Dans ce contexte, il est donc tout simplement hors de question d'envisager de simuler le modèle multi-agents suggéré par la réponse de Theraulaz. De fait, nous sommes contraints de travailler avec des modèles limités en termes de dynamiques environnementales, et donc de nous focaliser sur les comportements individuels pour monter en complexité.

Autrement dit, nous pensons que l'une des raisons de la focalisation excessive sur l'individu tient aussi au coût prohibitif que peuvent avoir les dynamiques environnementales dans un

4. En particulier, comme dit précédemment, nous comptons étendre le principe de délégation aux actions des agents. Ce qui doit être lié à l'application du principe influence/réaction. Calculer le résultat d'une influence consiste précisément à déporter celui-ci dans la réaction. Autrement dit, il s'agit bien de convertir certains calculs habituellement concrétisés dans le comportement des agents en dynamiques environnementales (réaction).

contexte séquentiel. C'est pourquoi nous nous les interdisons, ou les réduisons à leur strict minimum. Cela alors que nous pourrions sans doute obtenir des dynamiques collectives originales et plus complexes, tout en conservant des comportements agents simples. À l'heure où il est question d'augmenter toujours plus le nombre d'agents simulés, il est tout aussi important de ne pas se restreindre sur ce qu'il est possible de faire au niveau de l'environnement [5].

Pour aller plus loin dans l'évolution des modèles multi-agents, une autre perspective intéressante sera donc d'étudier des dynamiques environnementales calculées grâce à des algorithmes issus d'autres disciplines. Par exemple, dans le cadre du traitement numérique de l'image, il existe de très nombreux opérateurs (filtres) destinés à calculer différentes sortes de gradients dans une image [6] (Sobel, Kirsch, etc.). Pour l'instant, pour le calcul du gradient dans le modèle MLE (cf. section 8.2), nous avons simplement transcrit le comportement des agents en code GPGPU alors que les opérateurs TNI reposent sur des calculs matriciels qui sont bien mieux adaptés aux architectures des GPU.

Ainsi, il existe toute une diversité de dynamiques environnementales qui peuvent être explorées grâce à des algorithmes déjà existants, les outils issus du traitement du signal ne sont qu'un premier exemple. Il sera donc intéressant d'étudier ce qu'elles peuvent apporter aux SMA en termes de comportements collectifs. Dans une démarche environnement-centrée, une idée sera par exemple de travailler sur des comportements agents très simples (fourmi de Langton, MLE, *boids*, etc.), fixés, et de mesurer l'impact que peuvent avoir différentes dynamiques environnementales sur le comportement global du système en termes d'originalité ou d'efficacité.

Plus généralement, le GPGPU offre un contexte de programmation extrêmement différent qui oblige à repenser les modèles multi-agents mais, en fait, nous ne considérons pas cela uniquement comme une pure contrainte qu'il nous faudrait intégrer aux modélisations existantes. Au contraire, parce que les architectures massivement parallèles obligent à penser la modélisation différemment, notre intuition est qu'elles peuvent être une source très féconde d'idées originales et de dynamiques collectives novatrices. En particulier, dans le cadre du déplacement de la complexité des agents vers l'environnement, le GPGPU nous pousse à considérer et à explorer des dynamiques environnementales originales que nous n'aurions jamais envisagées dans un contexte séquentiel. Ainsi, il ne s'agit pas tant de possibilités technologiques nouvelles, que de nous ouvrir à de nouvelles manières de voir les modèles multi-agents.

5. Les travaux de Charrier [2009] sont d'ailleurs très intéressants de ce point de vue là, car ils reposent sur tout un ensemble d'outils mathématiques basés sur la notion de champ dans l'environnement. Dans son modèle SMAL (*SMA Logistique*), le comportement des agents est directement relié aux dynamiques environnementales. Charrier s'inspire de la théorie synergétique de Haken (outils pour l'étude de l'auto-organisation dans les systèmes complexes) et parle de l'asservissement du comportement des agents par les champs présents dans l'environnement.

6. Nous en avons vu quelques uns dans le cadre des travaux présentés dans le chapitre 4.

## 9.5 En conclusion

> " *Science initiated a successful dialogue with nature. On the other hand, the first outcome of this dialogue was the discovery of a silent world. This is the paradox of classical science. It revealed to men a dead, passive nature, a nature that behaves as an automaton which, once programmed, continues to follow the rules inscribed in the program... We have discovered that far from being an illusion, irreversibility plays an essential role in nature and lies at the origin of most processes of self-organization. We find ourselves in a world in which reversibility and determinism apply only the limiting, simple cases, while irreversibility and randomness are the rules*"
>
> Prigogine et Stengers, 1984, *Order out of chaos*, p.6, p.8

Avancer dans la compréhension des systèmes complexes constitue aujourd'hui un enjeu scientifique majeur. Pour preuve, la littérature qui leur est consacrée abonde chaque jour un peu plus et le nombre de domaines concernés ne fait que croître. Rien d'étonnant à cela dans la mesure où, parce qu'ils sont définis de manière générique comme des systèmes composés d'un grand nombre d'entités en interaction, il est possible d'en voir à peu près partout. On trouve non seulement des systèmes complexes dans la nature et les sociétés humaines, mais l'ère informatique a ouvert la voie à la création de systèmes artificiels physiques (e.g. robotique collective), virtuels (e.g. les réseaux sociaux) ou même les deux (e.g. Internet des objets), dont la complexité est tout aussi concrète.

Jusqu'à sa mort, Ilia Prigogine a défendu l'idée que de nombreux travaux, notamment ceux liés à la théorie du chaos [Gleick, 1989 ; Motter et Campbell, 2013], imposent un changement de paradigme scientifique profond, sans précédent, loin des lois où l'universalité temporelle est un postulat. L'irréversibilité des phénomènes, que nous constatons par l'expérience au niveau macroscopique, serait en fait valable à toutes les échelles concevables. Par conséquent, on ne peut se passer d'une dissymétrie dans les lois de la physique : passé et futur ne sont pas équivalents et introduire la *flèche du temps* n'est pas une option, elle joue un rôle constructif. Et si le monde nous apparaît complexe, c'est parce qu'il *est* complexe, irréductible aux trajectoires individuelles des éléments qui le compose[7].

L'une des implications épistémologiques importantes de cette révolution est qu'elle fait voler en éclats les frontières disciplinaires, elle les rend obsolètes. Pour Prigogine, l'introduction de la *flèche du temps* remet profondément en cause la dichotomie entre d'un côté les sciences naturelles, dont l'idéal serait d'atteindre la certitude associée à une description déterministe, et de l'autre les sciences humaines, où incertitude et choix dominent. Elles font toutes deux face

7. Comme le suggéraient déjà les travaux de Poincaré. De Prigogine, voir aussi *Les lois du chaos* pour une introduction très concise publiée en français, extraite de deux conférences à destination des scientifiques [Prigogine, 1994]. On y trouve notamment que les travaux d'Alan Turing [1952] sur la formation des structures stationnaires de non-équilibre sont considérés comme fondamentaux.

à des systèmes complexes qui se construisent dans le temps, grâce aux interactions. Dans cette déconstruction des mentalités, les modèles multi-agents, ainsi que leurs simulations, auront sans aucun doute un rôle important, sinon fondamental.

.

# Bibliographie


Aissani, N., Bekrar, A., Trentesaux, D. et Beldjilali, B. (2012). «Dynamic scheduling for multi-site companies : a decisional approach based on reinforcement multi-agent learning». English. *Journal of Intelligent Manufacturing* 23.6, p. 2513–2529 (cf. p. 66).

Amblard, F. et Phan, D. (2006). *Modélisation et simulation multi-agents*. science informatique et SHS. Hermes Lavoisier (cf. p. 3).

Arcos, J., Noriega, P., Rodríguez-Aguilar, J. et Sierra, C. (2007). «E4MAS Through Electronic Institutions». English. In : *Environments for Multi-Agent Systems III*. Sous la dir. de D. Weyns, V. Parunak et F. Michel. T. 4389. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 184–202 (cf. p. 28). (doi livre)

Axelrod, R. M. (1997). *The complexity of cooperation : Agent-based models of competition and collaboration*. Princeton University Press (cf. p. 3).

Badeig, F. (2010). «Un environnement actif pour la simulation multi-agents : application à la gestion de crise dans les transports». Thèse de doctorat. Université Paris-Dauphine (cf. p. 19).

Badeig, F. et Balbo, F. (2012). «Définition d'un cadre de conception et d'exécution pour la simulation multi-agent.» *Revue d'Intelligence Artificielle* 26.3, p. 255–280 (cf. p. 8, 9, 36, 91).

Báez-Barranco, J.-A., Stratulat, T. et Ferber, J. (2007). «A Unified Model for Physical and Social Environments». English. In : *Environments for Multi-Agent Systems III*. Sous la dir. de D. Weyns, V. Parunak et F. Michel. T. 4389. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 41–50 (cf. p. 32). (doi livre)

Balbo, F., Saunier, J. et Badeig, F. (2013). «Evaluation of Environment Contextual Services in Multiagent Systems». English. In : *Agents and Artificial Intelligence*. Sous la dir. de J. Filipe et A. Fred. T. 271. Communications in Computer and Information Science. Springer Berlin Heidelberg, p. 192–207 (cf. p. 8, 36, 66).

Bandini, S., Manzoni, S. et Vizzari, G. (2009). «Agent Based Modeling and Simulation : An Informatics Perspective». *Journal of Artificial Societies and Social Simulation* 12.4, p. 4 (cf. p. 8).

Béhé, F., Galland, S., Gaud, N., Nicolle, C. et Koukam, A. (2014). «An ontology-based metamodel for multiagent-based simulations». *Simulation Modelling Practice and Theory* 40, p. 64–85 (cf. p. 8).

Beslon, G. (2008). «Apprivoiser la vie - Modélisation individu-centrée de systèmes biologiques complexes». Habilitation à Diriger des Recherches. INSA-Lyon (cf. p. 7).

Beurier, G. (2007). «Codage indirect de la forme dans les systèmes multi-agents : émergence multi-niveaux, évolution et morphogénèse». Thèse de doctorat. Université Montpellier II (cf. p. 68, 69).

Beurier, G., Michel, F. et Ferber, J. (2006). «A morphogenesis model for multiagent embryogeny». In : *ALIFE X : Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems, Bloomington, IN USA, June 3-7, 2006*. Sous la dir. de L. M. Rocha, L. S. Yaeger, M. A. Bedau, D. Floreano, R. L. Goldstone et A. Vespignani. Cambridge, MA, USA : MIT Press, p. 84–90 (cf. p. 69, 70). pdf

Beurier, G., Simonin, O. et Ferber, J. (2003). «Un modèle de système multi-agents pour l'émergence multi-niveaux». In : *11eme journées Francophones sur les Systèmes Multi-Agents*. Revue des Sciences et Technologies de l'Information. Hermès, p. 235–247 (cf. p. 79).

Blanchart, E., Cambier, C., Canape, C., Gaudou, B., Ho, T.-N., Ho, T.-V., Lang, C., Michel, F., Marilleau, N. et Philippe, L. (2011). «EPIS : A Grid Platform to Ease and Optimize Multi-agent Simulators Running».

In : *Proceedings of the 9th International Conference on Practical Applications of Agents and Multiagent Systems, PAAMS 2011, Salamanca, Spain, 6-8th April, 2011*. Sous la dir. d'Y. Demazeau, M. Pechoucek, J. Corchado et J. Pérez. T. 88. Advances in Intelligent and Soft Computing. Springer Berlin Heidelberg, p. 129–134 (cf. p. 76). pdf

Boissier, O., Bordini, R. H., Hübner, J. F., Ricci, A. et Santi, A. (2013). «Multi-agent oriented programming with JaCaMo». *Science of Computer Programming* 78.6, p. 747–761 (cf. p. 65).

Boissier, O., Hübner, J. et Sichman, J. (2007). «Organization Oriented Programming : From Closed to Open Organizations». English. In : *Engineering Societies in the Agents World VII*. Sous la dir. de G. O'Hare, A. Ricci, M. O'Grady et O. Dikenelli. T. 4457. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 86–105 (cf. p. 64).

Bommel, P. (2009). «Definition of a methodological framework for designing multi-agents models adapted to the renewable resources management». Theses. Université Montpellier II - Sciences et Techniques du Languedoc (cf. p. 10).

Bonabeau, E. (2002). «Agent-based modeling : Methods and techniques for simulating human systems». *Proceedings of the National Academy of Sciences* 99.suppl 3, p. 7280–7287 (cf. p. 9).

Bonabeau, E. et Theraulaz, G. (1994). «Why do we need artificial life ?» *Artificial Life* 1.3, p. 303–325 (cf. p. 3).

Bonté, B. (2011). «Modélisation et simulation de l'interdépendance entre l'objet, l'observateur et le modèle de l'objet dans la Triade de Minsky. Application à la surveillance épidémiologique en santé animale.» Thèse de doctorat. Université Montpellier II (cf. p. 7).

Bordini, R. H., Hübner, J. F. et Wooldridge, M. (2007). *Programming multi-agent systems in AgentSpeak using Jason*. T. 8. John Wiley & Sons (cf. p. 65).

Borra, S. et Sarkar, S. (1997). «A framework for performance characterization of intermediate-level grouping modules». *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19.11, p. 1306–1312 (cf. p. 38).

Bourgoin, M. (2013). «Abstractions performantes pour cartes graphiques». Thèse de doctorat. LIP6 (cf. p. 76).

Bourjot, C. et Chevrier, V. (2001). «De la simulation de construction collective à la détection de régions dans des images à niveaux de gris : L'inspiration des araignées sociales». In : *Fondements des systèmes multi-agents : modèles, spécifications formelles et vérification - JFIADSMA 01 - neuvième journées francophones d'Intelligence Artificielle et systèmes multi-agents, Montréal, Québec, Canada, November 12-14, 2001*. Sous la dir. d'A. El Fallah Seghrouchni et L. Magnin. Hermès Lavoisier Editions, p. 253–263 (cf. p. 39).

Bouslimi, I., Hanachi, C. et Ghedira, K. (2014). «An Experimental Evaluation of Communication in an Organization-Based Multi-agent System». In : *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*. T. 3, p. 72–78 (cf. p. 66).

Brandouy, O., Mathieu, P. et Veryzhenko, I. (2013). «On the Design of Agent-Based Artificial Stock Markets». English. In : *Agents and Artificial Intelligence*. Sous la dir. de J. Filipe et A. Fred. T. 271. Communications in Computer and Information Science. Springer Berlin Heidelberg, p. 350–364 (cf. p. 3).

Camps, V. et Gleizes, M.-P. (1996). «Attitudes coopératives individuelles pour l'adaptation collective». In : *Actes des quatrièmes Journées Francophones IAD SMA* (cf. p. 5).

Carabelea, C., Boissier, O. et Florea, A. (2004). «Autonomy in Multi-agent Systems : A Classification Attempt». English. In : *Agents and Computational Autonomy*. Sous la dir. de M. Nickles, M. Rovatsos et G. Weiss. T. 2969. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 103–113 (cf. p. 19).

Carlési, N. (2013). «Coopération entre véhicules sous-marins autonomes : une approche organisationnelle réactive multi-agent». Thèse de doctorat. Université Montpellier II (cf. p. 55).

Carlési, N., Michel, F., Jouvencel, B. et Ferber, J. (2011). «Generic Architecture for Multi-AUV Cooperation Based on a Multi-Agent Reactive Organizational Approach». Anglais. In : *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2011 San Francisco, California, Sept. 25-30, 2011*. San Francisco, California, USA, p. 5041–5047 (cf. p. 55, 57). pdf

Castelfranchi, C. (1995). «Guarantees for autonomy in cognitive agent architecture». English. In : *Intelligent Agents*. Sous la dir. de M. Wooldridge et N. Jennings. T. 890. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 56–70 (cf. p. 19).

Chang, P. H., Chen, K.-T., Chien, Y.-H., Kao, E. et Soo, V.-W. (2005). «From reality to mind : A cognitive Middle Layer of Environment Concepts for Believable Agents». In : *Environments for Multi-Agent Systems, First International Workshop, E4MAS 2004, New York, NY, USA, July 19, 2004, Revised Selected Papers*. Sous la dir. de D. Weyns, V. Parunak et F. Michel. T. 3374. LNAI. New York, USA : Springer, p. 57–73 (cf. p. 28, 73). (doi livre)

Chapelle, J., Simonin, O. et Ferber, J. (2002). «How Situated Agents can Learn to Cooperate by Monitoring their Neighbors' Satisfaction». In : *Proceedings of the 15th Eureopean Conference on Artificial Intelligence, ECAI'2002, Lyon, France, July 2002*. Sous la dir. de F. van Harmelen. IOS Press, p. 68–72 (cf. p. 57).

Charrier, R. (2009). «L'intelligence en essaim sous l'angle des systèmes complexes : étude d'un système multi-agent réactif à base d'itérations logistiques couplées». Thèse de doctorat. Université Nancy II (cf. p. 93, 95). pdf

Charrier, R., Bourjot, C. et Charpillet, F. (2009). «Study of self-adaptation mechanisms in a swarm of logistic agents». In : *Self-Adaptive and Self-Organizing Systems, 2009. SASO'09. Third IEEE International Conference on*. IEEE, p. 82–91 (cf. p. 93).

Che, S., Boyer, M., Meng, J., Tarjan, D., Sheaffer, J. W. et Skadron, K. (2008). «A performance study of general-purpose applications on graphics processors using CUDA». *Journal of Parallel and Distributed Computing* 68.10. General-Purpose Processing using Graphics Processing Units, p. 1370–1380 (cf. p. 77).

Chen, J. (2007). «Flow in games (and everything else)». *Communications of the ACM* 50.4, p. 31–34 (cf. p. 49).

Chevaillier, P., Harrouet, F., Reignier, P. et Tisseau, J. (1999). «oRis : un environnement pour la simulation multi-agents des systemes manufacturiers de production». *Proceedings MOSIN* 99, p. 225–230 (cf. p. 19).

Chevrier, V. (2002). «Contributions au domaine des systemes multi-agents». Habilitation à Diriger des Recherches. Université Henri Poincaré, Nancy (cf. p. 20).

Chevrier, V. et Fatès, N. (2008). *Multi-agent Systems as Discrete Dynamical Systems : Influences and Reactions as a Modelling Principle*. Research Report. LORIA (cf. p. 21, 23, 88).

Cicirelli, F., Giordano, A. et Nigro, L. (2015). «Efficient environment management for distributed simulation of large-scale situated multi-agent systems». *Concurrency and Computation : Practice and Experience* 27.3, p. 610–632 (cf. p. 22).

Collier, N. et North, M. (2012). «Repast HPC : A Platform for Large-Scale Agent-Based Modeling». In : *Large-Scale Computing*. Sous la dir. de B. S. Werner Dubitzky Krzysztof Kurowski. John Wiley  Sons, Inc., p. 81–109 (cf. p. 76).

Couturier, V., Huget, M.-P. et Telisson, D. (2012). «Patterns for Agent-Based Information Systems : A Case Study in Transport». In : *Innovative Information Systems Modelling Techniques*. Sous la dir. de C. Kalloniatis. InTech, p. 49–72 (cf. p. 66).

David, N. (2013). «Validating Simulations». In : *Simulating Social Complexity*. Sous la dir. de B. Edmonds et R. Meyer. Understanding Complex Systems. Springer Berlin Heidelberg, p. 135–171 (cf. p. 10).

Dàvila, J. et Tucci, K. (2000). «Towards a logic-based, multi-agent simulation theory». In : *International Conference on Modeling, Simulation and Neural Networks MSNN'2000*. Merida, Venezuela (cf. p. 16).

DeAngelis, D. L. et Grimm, V. (2014). «Individual-based models in ecology after four decades». *F1000Prime Reports* 6.39 (cf. p. 4).

Decker, K. S. (1996). «Distributed artificial intelligence testbeds». In : *Foundations of distributed artificial intelligence*. Sous la dir. de G. M. P. O'Hare et N. R. Jennings. John Wiley & Sons, Inc. Chap. 3, p. 119–138 (cf. p. 5).

Demange, J. (2012). «Un modèle d'environnement pour la simulation multiniveau-Application à la simulation de foules». Thèse de doctorat. Université de Technologie de Belfort-Montbeliard (cf. p. 16, 19, 21, 25).

Demazeau, Y. (1995). «From interactions to collective behaviour in agent-based systems». In : *Proceedings of the 1st. European Conference on Cognitive Science, Saint-Malo, France* (cf. p. 5, 27).

Demigny, D., Devars, J., Kessal, L. et Quesne, J. F. (1993). «Implantations temps réel du filtre de lissage d'images de Nagao». *Traitement du signal* 10.4, p. 319–330 (cf. p. 40).

Dinont, C. (2007). «Calculs longs et partage des ressources processeur dans les systèmes multi-agents cognitifs». Thèse de doctorat. Lille 1 (cf. p. 36, 50–52).

Drogoul, A. (1993). «De la simulation multi-agent à la résolution collective de problèmes». Thèse de doctorat. Université Pierre et Marie Curie, Paris VI (cf. p. 3, 8).

Drogoul, A. (2000). «Systèmes multi-agents situés». Habilitation à Diriger des Recherches. Université Pierre et Marie Curie, Paris VI (cf. p. 5).

Drogoul, A., Amouroux, E., Caillou, P., Gaudou, B., Grignard, A., Marilleau, N., Taillandier, P., Vavasseur, M., Vo, D.-A. et Zucker, J.-D. (2013). «Gama : multi-level and complex environment for agent-based models and simulations». In : *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents et Multiagent Systems, p. 1361–1362 (cf. p. 10).

Drogoul, A., Vanbergue, D. et Meurisse, T. (2003). «Multi-agent Based Simulation : Where Are the Agents ?» In : *Multi-Agent-Based Simulation II*. Sous la dir. de J. Simão Sichman, F. Bousquet et P. Davidsson. T. 2581. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 1–15 (cf. p. 9, 10).

Duboz, R. (2004). «Intégration de modèles hétérogènes pour la modélisation et la simulation de systèmes complexes. Application à la modélisation multi-échelles en écologie marine». Thèse de doctorat. Calais : Université du Littoral, Côte d'Opale (cf. p. 6).

Edmonds, B. et Hales, D. (2003). «Replication, replication and replication : Some hard lessons from model alignment». *Journal of Artificial Societies and Social Simulation* 6.4 (cf. p. 10).

Epstein, J. M. et Axtell, R. L. (1996). *Growing Artificial Societies*. Brookings Institution Press, Washington D.C. (cf. p. 3).

Erman, L. D., Hayes-Roth, F., Lesser, V. R. et Reddy, D. R. (1980). «The Hearsay-II Speech-Understanding System : Integrating Knowledge to Resolve Uncertainty». *ACM Computing Surveys* 12.2, p. 213–253 (cf. p. 4).

Ermi, L. et Mäyrä, F. (2005). «Fundamental components of the gameplay experience : Analysing immersion». *Worlds in play : International perspectives on digital games research* 37 (cf. p. 49).

Estrada, F. J. et Jepson, A. D. (2009). «Benchmarking image segmentation algorithms». *International Journal of Computer Vision* 85.2, p. 167–181 (cf. p. 38).

Ferber, J. (1995). *Les systèmes multi-agents : vers une intelligence collective*. InterEditions (cf. p. 4, 8, 15, 16, 57).

Ferber, J. et Gutknecht, O. (1998). «A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems». In : *Proceedings of the Third International Conference on Multiagent Systems, ICMAS 1998, Paris, France, July 3-7, 1998*. Sous la dir. d'Y. Demazeau. IEEE Computer Society, p. 128–135 (cf. p. 55, 63).

Ferber, J., Gutknecht, O. et Michel, F. (2004). «From Agents to Organizations : an Organizational View of Multi-Agent Systems». In : *Agent-Oriented Software Engineering IV*. Sous la dir. de P. Giorgini, J. P. Müller et J. Odell. T. 2935. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 214–230 (cf. p. 56, 64). pdf (doi livre)

Ferber, J., Michel, F. et Bàez-Barranco, J.-A. (2005). «AGRE : Integrating Environments with Organizations». In : *Environments for Multi-Agent Systems*. Sous la dir. de D. Weyns, H. V. D. Parunak et F. Michel. T. 3374. Lecture Notes in Computer Science. New York, USA : Springer Berlin Heidelberg, p. 48–56 (cf. p. 9, 31). pdf (doi livre)

Ferber, J. et Müller, J.-P. (1996). «Influences and Reaction : a Model of Situated Multi-agent Systems». In : *2nd International Conference on Multi-agent Systems (ICMAS-96)*. Sous la dir. de M. Tokoro. Soraku-gun, Kyoto, Japan : The AAAI Press, p. 72–79 (cf. p. 15–17, 24, 27).

Feurzeig, W. et al. (1969). *Programming-Languages as a Conceptual Framework for Teaching Mathematics. Final Report on the First Fifteen Months of the LOGO Project*. Research Report 1889. Distributed by ERIC Clearinghouse (cf. p. 66).

Fikes, R. E. et Nilsson, N. J. (1972). «STRIPS : A new approach to the application of theorem proving to problem solving». *Artificial intelligence* 2.3, p. 189–208 (cf. p. 16).

Flynn, M. (1972). «Some computer organizations and their effectiveness». *Computers, IEEE Transactions on* 100.9, p. 948–960 (cf. p. 78).

Flynn, M. J. et Rudd, K. W. (1996). «Parallel architectures». *ACM Computing Surveys (CSUR)* 28.1, p. 67–70 (cf. p. 78, 82, 88).

Fortino, G. et North (2013). «Simulation-based development and validation of multi-agent systems». *J Simulation* 7.3, p. 137–143 (cf. p. 5).

Funge, J., Tu, X. et Terzopoulos, D. (1999). «Cognitive modeling : knowledge, reasoning and planning for intelligent characters». In : *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., p. 29–38 (cf. p. 50).

Galán, J. M., Izquierdo, L. R., Izquierdo, S. S., Santos, J. I., Olmo, R. del, López-Paredes, A. et Edmonds, B. (2009a). «Errors and Artefacts in Agent-Based Modelling». *Journal of Artificial Societies and Social Simulation* 12.1, p. 1 (cf. p. 10).

Galán, J. M., Izquierdo, L. R., Izquierdo, S. S., Santos, J. I., Olmo, R. del, López-Paredes, A. et Edmonds, B. (2009b). «Errors and Artefacts in Agent-Based Modelling». *Journal of Artificial Societies and Social Simulation* 12.1, p. 1 (cf. p. 10).

Galland, S. (2013). «Méthodologie et outils pour la simulation multiagent dans des univers virtuels». Habilitation à Diriger des Recherches. Université de Franche-Comté (cf. p. 3, 19, 20, 22, 25).

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 : *Principe de Parcimonie - JFSMA 14 - Vingt-deuxièmes Journées Francophones sur les Systèmes Multi-Agents, Loriol-sur-Drôme, France, Octobre 8-10, 2014*. Sous la dir. de R. Courdier et J. Jamont. Cepadues Editions, p. 65–74 (cf. p. 36, 60).

Gardelli, L., Viroli, M., Casadei, M. et Omicini, A. (2007). «Designing Self-organising MAS Environments : The Collective Sort Case». English. In : *Environments for Multi-Agent Systems III*. Sous la dir. de D. Weyns, V. Parunak et F. Michel. T. 4389. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 254–271 (cf. p. 28). (doi livre)

Gause, G. F. (1934). *The struggle for existence*. Williams & Wilkins, Baltimore (cf. p. 3).

Genesereth, M. R. et Nilsson, N. J. (1987). *Logical foundations of artificial intelligence*. Morgan Kaufmann Publishers Inc. (cf. p. 15).

Gilbert, N. et Conte, R. (1995). *Artificial Societies : The Computer Simulation of Social Life*. Bristol, PA, USA : Taylor & Francis, Inc. (cf. p. 3).

Glad, A. (2011). «Étude de l'auto-organisation dans les algorithmes de patrouille multi-agent fondés sur les phéromones digitales». Thèse de doctorat. Université Nancy II (cf. p. 69).

Gleick, J. (1989). *La théorie du chaos : vers une nouvelle science*. Albin Michel Paris (cf. p. 1, 96).

Gleizes, M.-P. (2004). «Vers la résolution de problèmes par émergence». Habilitation à Diriger des Recherches. Université Paul Sabatier, Toulouse III (cf. p. 5, 9, 39, 47, 73).

Gouaich, A. (2004). «Requirements for Achieving Software Agents Autonomy and Defining Their Responsibility». English. In : *Agents and Computational Autonomy*. Sous la dir. de M. Nickles, M. Rovatsos et G. Weiss. T. 2969. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 128–139 (cf. p. 20).

Gouaïch, A. (2005). «Movement, interaction, calculation as primitives for everywhere and anytime computing». Thèse de doctorat. Montpellier : Université Montpellier II (cf. p. 21, 34).

Gouaïch, A. et Michel, F. (2005). «Towards a Unified View of the Environment(s) within Multi-Agent Systems». *Informatica, Special Issue : Hot Topics in European Agent Research I* 29.4. Sous la dir. d'A. Omicini, P. Petta et M. Gams, p. 423–432 (cf. p. 9, 33, 34). pdf

Gouaïch, A., Michel, F. et Guiraud, Y. (2005). «MIC* : A Deployment Environment for Autonomous Agents». In : *Environments for Multi-Agent Systems*. Sous la dir. de D. Weyns, H. V. D. Parunak et F. Michel. T. 3374. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 109–126 (cf. p. 21, 34). pdf (doi livre)

Gruau, F. et Michel, F. (2011). «Rekindling Parallelism». In : *Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2011 Fifth IEEE Conference on*. Washington, DC, USA : IEEE Computer Society, p. 55–60 (cf. p. 76). pdf

Guessoum, Z. et Briot, J.-P. (1999). «From Active Objects to Autonomous Agents». *IEEE Concurrency, Special series on Actors and Agents* 7.3. Sous la dir. de D. Kafura et J.-P. Briot, p. 68–76 (cf. p. 19).

Gutiérrez, T. N. (2012). «Une architecture de contrôle de systèmes complexes basée sur la simulation multi-agent.» Thèse de doctorat. Nancy I (cf. p. 4).

Gutknecht, O., Ferber, J. et Michel, F. (2001). «Integrating tools and infrastructures for generic multi-agent systems». In : *Proceedings of the Fifth International Conference on Autonomous Agents, Agents'01*. Montreal, Quebec, Canada : ACM press, p. 441–448 (cf. p. 64). pdf

Haroun, R., Boumghar, F., Hassas, S. et Hamami, L. (2005). «A Massive Multi-agent System for Brain MRI Segmentation». English. In : *Massively Multi-Agent Systems I*. Sous la dir. de T. Ishida, L. Gasser et H. Nakashima. T. 3446. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 174–186 (cf. p. 39).

Hassas, S. (2006). «Engineering Complex Adaptive Systems Using Situated Multi-agents». English. In : *Engineering Societies in the Agents World VI*. Sous la dir. d'O. Dikenelli, M.-P. Gleizes et A. Ricci. T. 3963. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 125–141 (cf. p. 47).

Helleboogh, A., Vizzari, G., Uhrmacher, A. et Michel, F. (2007). «Modeling dynamic environments in multi-agent simulation». English. *Autonomous Agents and Multi-Agent Systems* 14.1, p. 87–116 (cf. p. 9, 15, 23, 24, 29). pdf

Hermellin, E. et Michel, F. (2015a). «Délégation GPU des perceptions agents : Application aux Boids de Reynolds». In : *JFSMA 15 - Vingt-troisièmes Journées Francophones sur les Systèmes Multi-Agents, Rennes, France, 29 juin - 1er juillet*. Sous la dir. de L. Vercouter et G. Picard, p. 185–194 (cf. p. 75, 83).

Hermellin, E. et Michel, F. (2015b). «GPU Environmental Delegation of Agent Perceptions : Application to Reynolds's Boids». In : *accepté dans Thirteenth Multi-Agent-Based Simulation, MABS XIII @ AAMAS* (cf. p. 75, 83).

Hermellin, E., Michel, F. et Ferber, J. (2015). «État de l'art sur les Simulations Multi-Agents et le GPGPU : évolution et perspectives de recherches». *Revue d'Intelligence Artificielle* 29.3-4, p. 423–449 (cf. p. 82).

Hermellin, E., Michel, F. et Ferber, J. (2014). «Systèmes multi-agents et GPGPU : état des lieux et directions pour l'avenir». In : *Principe de Parcimonie - JFSMA 14 - Vingt-deuxièmes Journées Francophones sur les Systèmes Multi-Agents, Loriol-sur-Drôme, France, Octobre 8-10, 2014*. Sous la dir. de R. Courdier et J. Jamont. Cepadues Editions, p. 97–106 (cf. p. 82, 83).

Hernández-Orallo, J. (2014). «On environment difficulty and discriminating power». English. *Autonomous Agents and Multi-Agent Systems*, p. 1–53 (cf. p. 38).

Hewitt, C. (1977). «Viewing control structures as patterns of passing messages». *Artificial intelligence* 8.3, p. 323–364 (cf. p. 4).

Hingston, P. (2012). *Believable Bots : Can Computers Play Like People ?* Springer Science & Business Media (cf. p. 50).

Hocine, N., Gouaïch, A., Cerri, S., Mottet, D., Froger, J. et Laffont, I. (2015). «Adaptation in serious games for upper-limb rehabilitation : an approach to improve training outcomes». English. *User Modeling and User-Adapted Interaction* 25.1, p. 65–98 (cf. p. 49).

Holland, J. H. (1995). *Hidden order : how adaptation builds complexity*. Addison Wesley Longman Publishing Co., Inc. (cf. p. 3).

Holvoet, T., Weyns, D. et Valckenaers, P. (2009). «Patterns of Delegate MAS». In : *Self-Adaptive and Self-Organizing Systems, 2009. SASO '09. Third IEEE International Conference on*, p. 1–9 (cf. p. 36).

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D. B., Bowyer, K., Eggert, D. W., Fitzgibbon, A. et Fisher, R. B. (1996). «An experimental comparison of range image segmentation algorithms». *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 18.7, p. 673–689 (cf. p. 43, 45).

Hubner, J. F., Sichman, J. S. et Boissier, O. (2007). «Developing organised multiagent systems using the MOISE+ model : programming issues at the system and agent levels». *International Journal of Agent-Oriented Software Engineering* 1.3, p. 370–395 (cf. p. 19, 64).

Jennett, C., Cox, A. L., Cairns, P., Dhoparee, S., Epps, A., Tijs, T. et Walton, A. (2008). «Measuring and defining the experience of immersion in games». *International journal of human-computer studies* 66.9, p. 641–661 (cf. p. 49).

Jiang, X. et Bunke, H. (1999). «Edge detection in range images based on scan line approximation». *Computer Vision and Image Understanding* 73.2, p. 183–199 (cf. p. 46).

Kammoun, H. M., Kallel, I., Casillas, J., Abraham, A. et Alimi, A. M. (2014). «Adapt-Traf : An adaptive multiagent road traffic management system based on hybrid ant-hierarchical fuzzy model». *Transportation Research Part C : Emerging Technologies* 42, p. 147–167 (cf. p. 69).

Kiesling, E., Günther, M., Stummer, C. et Wakolbinger, L. (2012). «Agent-based simulation of innovation diffusion : a review». English. *Central European Journal of Operations Research* 20.2, p. 183–230 (cf. p. 3).

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E. et Matsubara, H. (1997). «RoboCup : A challenge problem for AI». *AI magazine* 18.1, p. 73 (cf. p. 71).

Klügl, F., Herrler, R. et Fehler, M. (2006). «SeSAm : implementation of agent-based simulation using visual programming». In : *AAMAS '06 : Proceedings of the fifth international joint conference on Autonomous Agents and Multiagent Systems*. Hakodate, Japan : ACM, p. 1439–1440 (cf. p. 71).

Kravari, K. et Bassiliades, N. (2015). «A Survey of Agent Platforms». *Journal of Artificial Societies and Social Simulation* 18.1, p. 11 (cf. p. 10).

Kubera, Y., Mathieu, P. et Picault, S. (2010). «Everything can be Agent !» In : *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems : volume 1-Volume 1*. International Foundation for Autonomous Agents et Multiagent Systems, p. 1547–1548 (cf. p. 20).

Kubera, Y., Mathieu, P. et Picault, S. (2009). «How to Avoid Biases in Reactive Simulations». English. In : *7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009)*. Sous la dir. d'Y. Demazeau, J. Pavón, J. Corchado et J. Bajo. T. 55. Advances in Intelligent and Soft Computing. Springer Berlin Heidelberg, p. 100–109 (cf. p. 10).

Kubera, Y., Mathieu, P. et Picault, S. (2011). «IODA : an interaction-oriented approach for multi-agent based simulations». English. *Autonomous Agents and Multi-Agent Systems* 23.3, p. 303–343 (cf. p. 8, 9, 21, 73, 91).

Langton, C. G. (1986). «Studying artificial life with cellular automata». *Physica D : Nonlinear Phenomena* 22.1, p. 120–149 (cf. p. 3).

Larue, O., Poirier, P. et Nkambou, R. (2013). «Emotional Emergence in a Symbolic Dynamical Architecture». English. In : *Biologically Inspired Cognitive Architectures 2012*. Sous la dir. d'A. Chella, R. Pirrone, R. Sorbello et K. R. Jóhannsdóttir. T. 196. Advances in Intelligent Systems and Computing. Springer Berlin Heidelberg, p. 199–204 (cf. p. 66).

Laville, G. (2014). «Exécution efficace de systèmes multi-agents sur GPU». Thèse de doctorat. Université de Franche-Comté (cf. p. 82).

Le Page, C., Becu, N., Bommel, P. et Bousquet, F. (2012). «Participatory agent-based simulation for renewable resource management : The role of the cormas simulation platform to nurture a community of practice». *Journal of artificial societies and social simulation* 15.1, p. 10 (cf. p. 4).

Lebrun, Y. (2012). «Architecture multi-agent pour la gestion d'objets tangibles et virtuels sur Table Interactive». Thèse de doctorat. Université de Valenciennes et du Hainaut-Cambresis (cf. p. 19).

Lesser, V. R. et Corkill, D. D. (1983). «The Distributed Vehicle Monitoring Testbed : A Tool for Investigating Distributed Problem Solving Networks». *AI Magazine* 4.3, p. 15–33 (cf. p. 4).

Liedtke, J. (1996). «Toward Real Microkernels». *Communications of the ACM* 39.9, p. 70–77 (cf. p. 64).

Liu, J. et Tang, Y.-Y. (1999). «Adaptive image segmentation with distributed behavior-based agents». *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 21.6, p. 544–551 (cf. p. 39).

Logie, R., Hall, J. G. et Waugh, K. G. (2010). «Investigating Agent Influence and Nested Other-Agent Behaviour». *International Journal On Advances in Intelligent Systems* 3.1 and 2, p. 106–120 (cf. p. 19, 20, 22).

Lotka, A. J. (1920). «Analytical note on certain rhythmic relations in organic systems». *Proceedings of the National Academy of Sciences of the United States of America* 6.7, p. 410–415 (cf. p. 3).

Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K. et Balan, G. (2005). «Mason : A multiagent simulation environment». *Simulation* 81.7, p. 517–527 (cf. p. 78).

Lysenko, M. et D'Souza, R. M. (2008). «A Framework for Megascale Agent Based Model Simulations on Graphics Processing Units». *Journal of Artificial Societies and Social Simulation* 11.4, p. 10 (cf. p. 78).

Mahdi, G. (2013). «Level of Detail in Agent Societies for Games». Thèse de doctorat. Université Montpellier II (cf. p. 49).

Mahdi, G., Francillette, Y., Gouaïch, A., Michel, F. et Hocine, N. (2013). «Level Of Detail Based AI Adaptation for Agents in Video Games». Anglais. In : *ICAART 2013 - Proceedings of the 5th International Conference on Agents and Artificial Intelligence, Volume 2, Barcelona, Spain, 15-18 February, 2013*. Sous la dir. de J. Filipe et A. L. N. Fred. SciTePress, p. 182–194 (cf. p. 51). pdf

Mahdjoub, J. (2011). «Vers un système de vision auto-adaptatif à base de systèmes multi-agents». Thèse de doctorat. Université Reims Champagne-Ardenne (cf. p. 37).

Mahdjoub, J., Guessoum, Z., Michel, F. et Herbin, M. (2006). «A multi-agent approach for the edge detection in image processings». In : *Proceedings of the Fourth European Workshop on Multi-Agent Systems, EUMAS'06, Lisbon, Portugal, December 14-15, 2006*. Sous la dir. de B. Dunin-Keplicz, A. Omicini et J. Padget. T. 223. CEUR-WS.org (cf. p. 39, 41). pdf

Mamei, M. et Zambonelli, F. (2005). «Motion Coordination in the Quake 3 Arena Environment : A Field-Based Approach». English. In : *Environments for Multi-Agent Systems*. Sous la dir. de D. Weyns, V. Parunak et F. Michel. T. 3374. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 264–278 (cf. p. 45). (doi livre)

Mandiau, R., Champion, A., Auberlet, J.-M., Espié, S. et Kolski, C. (2008). «Behaviour based on decision matrices for a coordination between agents in a urban traffic simulation». English. *Applied Intelligence* 28.2, p. 121–138 (cf. p. 3).

Mansour, S. et Ferber, J. (2007). «MAGR : Integrating mobility of agents with organizations.» In : *International Conference Intelligent Systems and Agents IADIS*. (Cf. p. 64).

Marcenac, P. et Giroux, S. (1998). «GEAMAS : A Generic Architecture for Agent-Oriented Simulations of Complex Processes». English. *Applied Intelligence* 8.3, p. 247–267 (cf. p. 10).

Martin, D., Fowlkes, C., Tal, D. et Malik, J. (2001). «A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics». In : *Proc. 8th Int'l Conf. Computer Vision*. T. 2, p. 416–423 (cf. p. 38).

Martin, V. (2007). «Vision cognitive : apprentissage supervisé pour la segmentation d'images et de videos». Thèse de doctorat. Université Nice Sophia Antipolis (cf. p. 38).

Mathieu, P., Picault, S. et Routier, J.-C. (2015). «La force du collectif». *Dossier Pour la Science*. Les robots en quête d'humanité 87, p. 12–17 (cf. p. 4).

Mathieu, P., Picault, S. et Secq, Y. (2014). «Les environnements : en avoir ou pas ? Formalisation du concept et patterns d'implémentation». In : *Principe de Parcimonie - JFSMA 14 - Vingt-deuxièmes Journées Francophones sur les Systèmes Multi-Agents, Loriol-sur-Drôme, France, Octobre 8-10, 2014*. Sous la dir. de R. Courdier et J. Jamont. Cepadues Editions, p. 55–64 (cf. p. 31).

Mathieu, P., Routier, J.-C. et Secq, Y. (2002). «Principles for Dynamic Multi-agent Organizations». English. In : *Intelligent Agents and Multi-Agent Systems*. Sous la dir. de K. Kuwabara et J. Lee. T. 2413. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 109–122 (cf. p. 64).

Mazouzi, S. (2008). «Reconnaissance de formes par les systèmes auto-organisés. Application aux images de profondeur». Thèse de doctorat. Université de Constantine, Algérie (cf. p. 37).

Mazouzi, S., Guessoum, Z. et Michel, F. (2009a). «A Distributed and Collective Approach for Curved Object-Based Range Image Segmentation». In : *Progress in Pattern Recognition, Image Analysis, Computer Vision,*

*and Applications, Proceedings of the 14th Iberoamerican Conference on Pattern Recognition, CIARP 2009, Guadalajara, Jalisco, Mexico, November 15-18, 2009*. Sous la dir. d'E. Bayro-Corrochano et J.-O. Eklundh. T. 5856. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 201–208 (cf. p. 47). pdf

Mazouzi, S., Guessoum, Z. et Michel, F. (2008a). «Segmentation collective d'images à base d'objets polyédriques et courbés». Français. *I3 Information - Interaction - Intelligence* 8.2. Sous la dir. de G. I3/CNRS, p. 229–251 (cf. p. 46). pdf

Mazouzi, S., Guessoum, Z. et Michel, F. (2009b). «Une approche multi-agent pour la segmentation d'images de profondeur à base d'objets polyédriques. Une nouvelle approche de segmentation d'images». Français. *Technique et Science Informatiques* 28.3, p. 365–393 (cf. p. 46). pdf

Mazouzi, S., Guessoum, Z., Michel, F. et Batouche, M. (2008b). «An Agent-Based Approach for Range Image Segmentation». In : *Massively Multi-Agent Technology*. Sous la dir. de P. S. Nadeem Jamali et T. Sugawara. T. 5043. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 146–161 (cf. p. 45). pdf (doi livre)

Mazyad, H., Tnazefti-Kerkeni, I. et Basson, H. (2013). «A Multi-agent System to Implement a Collaborative Learning Method». In : *ICIW 2013, The Eighth International Conference on Internet and Web Applications and Services*, p. 266–271 (cf. p. 66).

McMahan, A. (2003). «Immersion, engagement and presence». *The video game theory reader*, p. 67–86 (cf. p. 50).

Methlouthi, I. et Bouamama, S. (2011). «A New Honeybee Optimization for Constraint Reasoning : Case of Max-CSPs». English. In : *Knowlege-Based and Intelligent Information and Engineering Systems*. Sous la dir. d'A. König, A. Dengel, K. Hinkelmann, K. Kise, R. Howlett et L. Jain. T. 6882. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 325–334 (cf. p. 66).

Michel, F. (2014). «Délégation GPU des perceptions agents : intégration itérative et modulaire du GPGPU dans les simulations multi-agents. Application sur la plate-forme TurtleKit 3». *Revue d'Intelligence Artificielle* 28.4, p. 485–510 (cf. p. 75, 79).

Michel, F. (2004). «Formalisme, outils et éléments méthodologiques pour la modélisation et la simulation multi-agents». Thèse de doct. Université Montpellier II (cf. p. 2). pdf

Michel, F. (2012). «GPU Environmental Delegation of Agent Perceptions for MABS». In : *ICCS'12, IEEE International Conference on Complex Systems, Agadir, Morocco, November 5-6*. Sous la dir. de M. Essaaidi et M. Nemiche. IEEE Computer Society, p. 1–6 (cf. p. 75, 82). pdf

Michel, F. (2013a). «Intégration du calcul sur GPU dans la plate-forme de simulation multi-agent générique TurtleKit 3». Français. In : *Dynamiques, couplages et visions intégratives - JFSMA 13 - Vingt-et-unièmes journées francophones sur les systèmes multi-agents, Lille, France, Juillet 3-5, 2013*. Sous la dir. de S. Hassas et M. Morge. Cépaduès Editions, 189–198 (best paper) (cf. p. 75). pdf

Michel, F. (2007a). «Le modèle IRM4S, de l'utilisation des notions d'influence et de réaction pour la simulation de systèmes multi-agents». *Revue d'Intelligence Artificielle* 21.5-6. Sous la dir. d'Y. Demazeau, p. 757–779 (cf. p. 9, 15, 16, 19, 22). pdf

Michel, F. (2007b). «The IRM4S Model : The Influence/Reaction Principle for Multiagent Based Simulation (short paper)». In : *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, USA, May 14-18, 2007*. Sous la dir. d'E. H. Durfee, M. Yokoo, M. N. Huhns et O. Shehory. AAMAS '07. Honolulu, Hawaii : ACM, p. 903–905 (cf. p. 9, 15). pdf

Michel, F. (2013b). «Translating Agent Perception Computations into Environmental Processes in Multi-Agent-Based Simulations : A means for Integrating Graphics Processing Unit Programming within Usual Agent-Based Simulation Platforms». *Systems Research and Behavioral Science, Special Issue : Conceptual Models Of Complex Systems* 30.6, p. 703–715 (cf. p. 75). pdf

Michel, F., Beurier, G. et Ferber, J. (2005). «The TurtleKit Simulation Platform : Application to Complex Systems». In : *Workshops Sessions of the Proceedings of the 1st International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2005, November 27 - December 1, 2005, Yaoundé,*

*Cameroon*. Sous la dir. d'A. Akono, E. Tonyé, A. Dipanda et K. Yétongnon. Yaoundé, Cameroon : IEEE, p. 122–128 (cf. p. 66). pdf

Michel, F., Ferber, J. et Drogoul, A. (2009). «Multi-Agent Systems and Simulation : a Survey From the Agents Community's Perspective». Anglais. In : *Multi-Agent Systems : Simulation and Applications*. Sous la dir. d'A. Uhrmacher et D. Weyns. Computational Analysis, Synthesis, and Design of Dynamic Systems. CRC Press - Taylor & Francis, p. 3–52 (cf. p. 2, 5, 8). pdf (doi livre)

Michel, F., Ferber, J. et Gutknecht, O. (2001). «Generic Simulation Tools Based on MAS Organization». In : *Proceedings of the Tenth European Workshop on Modelling Autonomous Agents in a Multi Agent World, MAAMAW'2001, Annecy, France, 2-4 May, 2001*. Sous la dir. d'Y. Demazeau et F. J. Garijo. Annecy, France (cf. p. 64). pdf

Michel, F., Ferber, J., Laur, P.-A. et Aleman, F. (2010). «Programmation situationnelle : programmation visuelle de comportements agents pour non informaticiens (présentation courte)». Français. In : *Systèmes Multi-agents, Défis Sociétaux - JFSMA 10 - Dix-huitièmes journées francophones sur les systèmes multi-agents, Mahdia, Tunisia, October 18-20, 2010*. Sous la dir. de M. Occello et L. Rejeb. Cépaduès Editions, p. 65–74 (cf. p. 72). pdf

Michel, F., Ferber, J., Laur, P.-A. et Aleman, F. (2011). «Situational Programming : Agent Behavior Visual Programming for MABS Novices». In : *Multi-Agent-Based Simulation XI*. Sous la dir. de T. Bosse, A. Geller et C. Jonker. T. 6532. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 1–15 (cf. p. 72). pdf (doi livre)

Michel, F., Gouaïch, A. et Ferber, J. (2004). «Weak Interaction and Strong Interaction in Agent Based Simulations». In : *Multi-Agent-Based Simulation III*. Sous la dir. de D. Hales, B. Edmonds, E. Norling et J. Rouchier. T. 2927. Lecture Notes in Computer Science. Melbourne, Australia : Springer Berlin Heidelberg, p. 43–56 (cf. p. 8, 10). pdf (doi livre)

Millington, I. et Funge, J. (2009). *Artificial intelligence for games*. CRC Press (cf. p. 50).

Minar, N., Burkhart, R., Langton, C. et Askenazi, M. (1996). *The Swarm Simulation System : A Toolkit for Building Multi-Agent Simulations*. Rapp. tech. 96-06-042. Santa Fe, NM, USA : Santa Fe Institute (cf. p. 3, 66).

Minsky, M. (1965). «Matter, mind and models». In : *Proc. International Federation of Information Processing Congress*. T. 1. Spartan Books, p. 45–50 (cf. p. 7, 88).

Morvan, G. (2009). «Approche multi-agents d'un système d'aide à la décision en environnement dynamique incertain-Application à l'entomologie médico-légale». Thèse de doctorat. Université d'Artois (cf. p. 69).

Morvan, G. (2012). «Multi-level agent-based modeling - Bibliography». *CoRR, arXiv :1205.0561v7 [cs.MA], URL http ://arxiv.org/abs/1205.0561v7* abs/1205.0561 (cf. p. 11).

Morvan, G., Veremme, A. et Dupont, D. (2011). «IRM4MLS : The Influence Reaction Model for Multi-Level Simulation». English. In : *Multi-Agent-Based Simulation XI*. Sous la dir. de T. Bosse, A. Geller et C. Jonker. T. 6532. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 16–27 (cf. p. 23).

Motter, A. E. et Campbell, D. K. (2013). «Chaos at fifty». *Physics Today* 66.5, p. 27 (cf. p. 1, 96).

Navarro, L., Flacher, F. et Corruble, V. (2011). «Dynamic level of detail for large scale agent-based urban simulations». In : *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. International Foundation for Autonomous Agents et Multiagent Systems, p. 701–708 (cf. p. 51).

Nickles, M., Rovatsos, M. et Weiss, G. (2004). *Agents and Computational Autonomy*. English. T. 2969. Lecture Notes in Computer Science. Springer Berlin Heidelberg (cf. p. 19).

Niederberger, C. et Gross, M. (2005). «Level-of-detail for cognitive real-time characters». *The Visual Computer* 21.3, p. 188–202 (cf. p. 50, 51).

Noel, V., Arcangeli, J.-P. et Gleizes, M.-P. (2010). «Component-Based Agent Architectures to Build Dedicated Agent Frameworks (regular paper)». anglais. In : *International Symposium 'From Agent Theory to Agent Implementation' (AT2AI), Vienna - Austria*. Sous la dir. de R. Trappl. Austrian Society for Cybernetic Studies, p. 483–488 (cf. p. 65).

North, M., Collier, N., Ozik, J., Tatara, E., Macal, C., Bragen, M. et Sydelko, P. (2013). «Complex adaptive systems modeling with Repast Simphony». English. *Complex Adaptive Systems Modeling* 1.1, 3 (cf. p. 10, 78).

North, M., Tatara, E., Collier, N. et Ozik, J. (2007). «Visual agent-based model development with Repast Simphony». In : *Agent 2007 Conference on Complex Interaction and Social Emergence*. Argonne, IL, USA : Argonne National Laboratory, p. 173–192 (cf. p. 71).

Odell, J., Parunak, V., Fleischer, M. et Brueckner, S. (2003). «Modeling Agents and Their Environment». English. 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 Berlin Heidelberg, p. 16–31 (cf. p. 27).

Okuyama, F. Y., Bordini, R. H. et Rocha Costa, A. C. da (2005). «ELMS : An Environment Description Language for Multi-Agent Simulations». In : *Environments for Multi-Agent Systems, First International Workshop, E4MAS 2004, New York, NY, USA, July 19, 2004, Revised Selected Papers*. Sous la dir. de D. Weyns, V. Parunak et F. Michel. T. 3374. LNAI. New York, USA : Springer, p. 91–108 (cf. p. 20, 28). (doi livre)

Orcutt, G. H. (1957). «A new type of socio-economic system». *Review of economics and statistics* 39.2, p. 116–123 (cf. p. 3).

Owens, J. D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A. E. et Purcell, T. J. (2007). «A Survey of General-Purpose Computation on Graphics Hardware». *Computer Graphics Forum* 26.1, p. 80–113 (cf. p. 77).

Park, J. et N. DeSouza, G. (2005). «3-D Modeling of Real-World Objects Using Range and Intensity Images». English. In : *Machine Learning and Robot Perception*. Sous la dir. de B. Apolloni, A. Ghosh, F. Alpaslan, L. C. Jain et S. Patnaik. T. 7. Studies in Computational Intelligence. Springer Berlin Heidelberg, p. 203–264 (cf. p. 43).

Parry, H. et Bithell, M. (2012). «Large Scale Agent-Based Modelling : A Review and Guidelines for Model Scaling». English. In : *Agent-Based Models of Geographical Systems*. Sous la dir. d'A. J. Heppenstall, A. T. Crooks, L. M. See et M. Batty. Springer Netherlands, p. 271–308 (cf. p. 11).

Parunak, V. (1997). «"Go to the ant" : Engineering principles from natural multi-agent systems». English. *Annals of Operations Research* 75, p. 69–101 (cf. p. 9).

Parunak, V. et Brueckner, S. (2007). «Concurrent Modeling of Alternative Worlds with Polyagents». English. In : *Multi-Agent-Based Simulation VII*. Sous la dir. de L. Antunes et K. Takadama. T. 4442. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 128–141 (cf. p. 36).

Parunak, V., Savit, R. et Riolo, R. (1998). «Agent-Based Modeling vs. Equation-Based Modeling : A Case Study and Users' Guide». English. In : *Multi-Agent Systems and Agent-Based Simulation*. Sous la dir. de J. Sichman, R. Conte et N. Gilbert. T. 1534. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 10–25 (cf. p. 2).

Pavlov, R. et Müller, J. (2013). «Multi-Agent Systems Meet GPU : Deploying Agent-Based Architectures on Graphics Processors». In : *Technological Innovation for the Internet of Things*. Sous la dir. de L. Camarinha-Matos, S. Tomic et P. Graça. T. 394. IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, p. 115–122 (cf. p. 82).

Payet, D., Courdier, R., Ralambondrainy, T. et Sébastien, N. (2006a). «Le modèle à Temporalité : pour un équilibre entre adéquation et optimisation du temps dans les simulations agent». In : *Systemes Multi-Agents, Articulation entre l'individuel et le collectif - JFSMA 2006 - Quatorzieme journees francophones sur les systemes multi-agents, Annecy, France, October 18-20, 2006*. Sous la dir. de V. Chevrier et M. Huget. Lavoisier, p. 63–76 (cf. p. 8).

Payet, D., Courdier, R., Sébastien, N. et Ralambondrainy, T. (2006b). «Environment as support for simplification, reuse and integration of processes in spatial MAS». In : *Proceedings of the 2006 IEEE International Conference on Information Reuse and Integration, IRI - 2006 : Heuristic Systems Engineering, September 16-18, 2006, Waikoloa, Hawaii, USA*. IEEE Systems, Man, et Cybernetics Society, p. 127–131 (cf. p. 8, 10, 36, 91).

Pépin, N., Simonin, O. et Charpillet, F. (2009). «Intelligent Tiles-Putting Situated Multi-Agents Models in Real World.» In : *ICAART*, p. 513–519 (cf. p. 69).

Pereira, C., Mahdjoub, J., Guessoum, Z., Gonçalves, L. et Ferreira, M. (2012). «Using MAS to Detect Retinal Blood Vessels». English. In : *Highlights on Practical Applications of Agents and Multi-Agent Systems*. Sous la dir. de J. B. Pérez, M. A. Sánchez, P. Mathieu, J. M. C. Rodríguez, E. Adam, A. Ortega, M. N. Moreno, E. Navarro, B. Hirsch, H. Lopes-Cardoso et V. Julián. T. 156. Advances in Intelligent and Soft Computing. Springer Berlin Heidelberg, p. 239–246 (cf. p. 66).

Picard, G. (2004). «Méthodologie de développement de systèmes multi-agents adaptatifs et conception de logiciels à fonctionnalité émergente». Thèse de doctorat. Université Paul Sabatier Toulouse III (cf. p. 9, 73).

Picault, S. (2013). «De la simulation multi-agents à la simulation multi-niveaux. Pour une réification des interactions.» Habilitation à Diriger des Recherches. Université des Sciences et Technologie de Lille-Lille I (cf. p. 9, 10, 73, 90).

Picault, S. et Mathieu, P. (2011). «An interaction-oriented model for multi-scale simulation». In : *IJCAI'2011–Barcelona (Spain)–July, 16-22 2011*. AAAI Press, p. 332–337 (cf. p. 11, 51).

Platon, E., Mamei, M., Sabouret, N., Honiden, S. et Parunak, V. (2007). «Mechanisms for environments in multi-agent systems : Survey and opportunities». English. *Autonomous Agents and Multi-Agent Systems* 14.1, p. 31–47 (cf. p. 28).

Platon, E., Sabouret, N. et Honiden, S. (2006). «Overhearing and Direct Interactions : Point of View of an Active Environment». English. In : *Environments for Multi-Agent Systems II*. Sous la dir. de D. Weyns, V. Parunak et F. Michel. T. 3830. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 121–138 (cf. p. 28). (doi livre)

Platon, E., Sabouret, N. et Honiden, S. (2005). «Tag Interactions in MultiAgent Systems : Environment Support». In : *EUMAS 2005 - Proceedings of the Third European Workshop on Multi-Agent Systems, Brussels, Belgium, December 7-8, 2005*. Sous la dir. de M. P. Gleizes, G. A. Kaminka, A. Nowé, S. Ossowski, K. Tuyls et K. Verbeeck. Koninklijke Vlaamse Academie van Belie voor Wetenschappen en Kunsten, p. 270–281 (cf. p. 20).

Poincaré, H. (1902). *La science et l'hypothèse*. Flammarion (cf. p. 1).

Prigogine, I. (1994). *Les lois du chaos*. Flammarion (cf. p. 96).

Prigogine, I. et Stengers, I. (1984). *Order out of chaos : Man's new dialogue with nature*. Bantam books New York, p. 5–6, 8 (cf. p. 96).

Ramos, V. et Almeida, F. (2000). «Artificial ant colonies in digital image habitats-a mass behaviour effect study on pattern recognition». In : *Proc. of ANTS 2000 - 2nd Int. Workshop on Ant Algorithms (From Ant Colonies to Artificial Ants)*. Sous la dir. de T. S. Marco Dorigo Martin Middendorf, p. 113 (cf. p. 39).

Ranathunga, S., Cranefield, S. et Purvis, M. (2012). «Identifying events taking place in Second Life virtual environments». *Applied Artificial Intelligence* 26.1-2, p. 137–181 (cf. p. 23).

Rao, A. S. et Georgeff, M. P. (1992). «An Abstract Architecture for Rational Agents». In : *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*. Sous la dir. de B. Nebel, C. Rich et W. R. Swartout. Cambridge, Massachusetts, USA, p. 439–449 (cf. p. 8).

Resnick, M. (1992). «Beyond the centralized mindset–explorations in massively-parallel microworlds». Thèse de doct. Massachusetts Institute of Technology (cf. p. 12, 66).

Resnick, M. (1994). *Turtles, termites, and traffic jams : Explorations in massively parallel microworlds*. Mit Press (cf. p. 3, 9, 66).

Reynolds, C. W. (1987). «Flocks, herds and schools : A distributed behavioral model». In : *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM, p. 25–34 (cf. p. 3, 83).

Ricci, A., Omicini, A., Viroli, M., Gardelli, L. et Oliva, E. (2007a). «Cognitive Stigmergy : Towards a Framework Based on Agents and Artifacts». English. In : *Environments for Multi-Agent Systems III*. Sous la dir. de D. Weyns, V. Parunak et F. Michel. T. 4389. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 124–140 (cf. p. 28). (doi livre)

Ricci, A., Piunti, M. et Viroli, M. (2011). «Environment programming in multi-agent systems : an artifact-based perspective». *Autonomous Agents and Multi-Agent Systems* 23.2, p. 158–192 (cf. p. 34, 35, 91).

Ricci, A., Viroli, M. et Omicini, A. (2007b). «CArtA gO : A Framework for Prototyping Artifact-Based Environments in MAS». English. In : *Environments for Multi-Agent Systems III*. Sous la dir. de D. Weyns, V. Parunak et F. Michel. T. 4389. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 67–86 (cf. p. 28, 35). (doi livre)

Richmond, P., Walker, D., Coakley, S. et Romano, D. (2010). «High performance cellular level agent-based simulation with FLAME for the GPU». *Briefings in Bioinformatics* 11.3, p. 334–347. eprint : http://bib.oxfordjournals.org/content/11/3/334.full.pdf+html (cf. p. 78).

Rihawi, O., Secq, Y. et Mathieu, P. (2013). «Relaxing Synchronization Constraints in Distributed Agent-based Simulations». *Jurnal Teknologi* 63.3, p. 65–76 (cf. p. 22, 78).

Rodriguez, S., Hilaire, V. et Koukam, A. (2007). «A Holonic Approach to Model and Deploy Large Scale Simulations». English. In : *Multi-Agent-Based Simulation VII*. Sous la dir. de L. Antunes et K. Takadama. T. 4442. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 112–127 (cf. p. 64).

Routier, J.-C. (2005). «Conception par agent orientée compétences». Habilitation à Diriger des Recherches. Lille I (cf. p. 50).

Russell, S. J. et Norvig, P. (1995). *Artificial Intelligence : A Modern Approach*. 2e éd. Prentice Hall (cf. p. 27, 33).

Sargent, R. G. (2001). «Verification and Validation : Some Approaches and Paradigms for Verifying and Validating Simulation Models». In : *Proceedings of the 33Nd Conference on Winter Simulation*. WSC '01. Arlington, Virginia : IEEE Computer Society, p. 106–114 (cf. p. 10).

Saunier, J., Balbo, F. et Badeig, F. (2007). «Environment as Active Support of Interaction». English. In : *Environments for Multi-Agent Systems III*. Sous la dir. de D. Weyns, V. Parunak et F. Michel. T. 4389. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 87–105 (cf. p. 8, 28). (doi livre)

Saunier, J. et Jones, H. (2014). «Mixed agent/social dynamics for emotion computation». In : *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents et Multiagent Systems, p. 645–652 (cf. p. 19, 20, 66).

Schelling, T. C. (1971). «Dynamic models of segregation». *Journal of mathematical sociology* 1.2, p. 143–186 (cf. p. 3).

Sébastien, N. (2010). «Distribution et Parallélisation de Simulations Orientées Agents». Français. Thèse de doctorat. Université de la Réunion (cf. p. 22, 78). pdf

Sébastien, N., Courdier, R., Hoareau, D., Huget, M.-P. et Payet, D. (2009). «Ordonnancement parallèle de simulations orientées agent : Une approche basée sur l'analyse des dépendances temporelles des influences et perceptions». *Revue d'intelligence artificielle* 23.5-6, p. 673–696 (cf. p. 8).

Sichman, J. S., Conte, R., Castelfranchi, C. et Demazeau, Y. (1994). «A Social Reasoning Mechanism Based On Dependence Networks». In : *Proceedings of the Eleventh European Conference on Artificial Intelligence, ECAI 94*, p. 188–192 (cf. p. 19).

Sichman, J. S., Conte, R. et Gilbert, N., éds. (1998). *Multi-Agent Systems and Agent-Based Simulation, First International Workshop, MABS '98, Paris, France, July 4-6, 1998, Proceedings*. T. 1534. Lecture Notes in Computer Science. Springer (cf. p. 3).

Siebert, J. (2011). «Approche multi-agent pour la multi-modélisation et le couplage de simulations. Application à l'étude des influences entre le fonctionnement des réseaux ambiants et le comportement de leurs utilisateurs.» Thèse de doctorat. Université Henri Poincaré-Nancy I (cf. p. 19–21).

Siebert, J., Ciarletta, L. et Chevrier, V. (2009). «De l'intérêt du couplage de modèles pour appréhender les interactions utilisateurs/réseaux dynamiques». *Revue d'intelligence artificielle* 23.5-6, p. 749–776 (cf. p. 19).

Simo Kanmeugne, P. (2014). «Simulation crédible des déplacements de piétons en temps réel : modèle microscopique à influence macroscopique». Thèse de doctorat. Université Pierre et Marie Curie, Paris VI (cf. p. 19–21).

Simon, H. A. (1969). «The sciences of the artificial». *Cambridge, MA* (cf. p. 28).

Simonin, O. (2010). «Contribution à la résolution collective de problème (Modèles d'auto-organisation par interactions directes et indirectes dans les SMA réactifs et robotiques)». Habilitation à Diriger des Recherches. Université Henri Poincaré-Nancy I (cf. p. 45).

Simonin, O. (2001). «Le modèle satisfaction-altruisme : coopération et résolution de conflits entre agents situés réactifs, application à la robotique». Thèse de doctorat. Université Montpellier II (cf. p. 8, 57).

Simonin, O., Charpillet, F. et Thierry, E. (2014). «Revisiting wavefront construction with collective agents : an approach to foraging». English. *Swarm Intelligence* 8.2, p. 113–138 (cf. p. 5, 69).

Simonin, O. et Gechter, F. (2006). «An Environment-Based Methodology to Design Reactive Multi-agent Systems for Problem Solving». In : *Environments for Multi-Agent Systems II, Second International Workshop, E4MAS 2005, Utrecht, The Netherlands, July 25, 2005, Selected Revised and Invited Papers*. T. 3830. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 32–49 (cf. p. 28). (doi livre)

Simonin, O., Lanoix, A., Scheuer, A. et Charpillet, F. (2011a). «Specifying in B the Influence/Reaction Model to Study Situated MAS : Application to vehicles platooning». Anglais. In : *V2CS : First International workshop on Verification and Validation of multi-agent models for complex systems*. France, 15 pages (cf. p. 16). pdf

Simonin, O., Lanoix, A., Scheuer, A. et Charpillet, F. (2011b). «Specifying in B the Influence/Reaction Model to Study Situated MAS : Application to vehicles platooning». In : *V2CS : First International workshop on Verification and Validation of multi-agent models for complex systems*. France, 15 pages (cf. p. 24).

Sims, M., Corkill, D. et Lesser, V. (2008). «Automated organization design for multi-agent systems». English. *Autonomous Agents and Multi-Agent Systems* 16.2, p. 151–185 (cf. p. 64).

Sklar, E. (2007). «NetLogo, a Multi-agent Simulation Environment». *Artificial Life* 13.3, p. 303–311 (cf. p. 10, 66, 78).

Smith, R. G. (1980). «The Contract Net Protocol : High-Level Communication and Control in a Distributed Problem Solver». *IEEE Transactions on Computers* C29.12, p. 1104–1113 (cf. p. 4).

Soulié, J.-C. (2012). «Multi-modélisation et simulation de systèmes. Complexes : de la théorie à l'application». Habilitation à Diriger des Recherches. Université du Littoral Côte d'Opale (cf. p. 6, 20, 34).

Soulié, J.-C. (2001). «Vers une approche multi-environnements pour les agents». Thèse de doctorat. Université de la Réunion (cf. p. 8, 34).

Soyez, J. B. (2013). «Conception et modélisation de systèmes de systèmes : une approche multi-agents multi-niveaux». Thèse de doctorat. Université de Lille 1 (cf. p. 66).

Soyez, J.-B., Morvan, G., Dupont, D. et Merzouki, R. (2013). «A Methodology to Engineer and Validate Dynamic Multi-level Multi-agent Based Simulations». English. In : *Multi-Agent-Based Simulation XIII*. Sous la dir. de F. Giardini et F. Amblard. T. 7838. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 130–142 (cf. p. 23, 51).

Steel, T., Kuiper, D. et Wenkstern, R. (2010). «Context-Aware Virtual Agents in Open Environments». In : *Autonomic and Autonomous Systems (ICAS), 2010 Sixth International Conference on*, p. 90–96 (cf. p. 21).

Steiner, R., Leask, G. et Mili, R. (2005). «An Architecture for MAS Simulation Environments». In : *Environments for Multi-Agent Systems II, Second International Workshop, E4MAS 2005, Utrecht, The Netherlands, July 25, 2005, Selected Revised and Invited Papers*. T. 3830. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 50–67 (cf. p. 28). (doi livre)

Sun, R. (2006). *Cognition and multi-agent interaction : From cognitive modeling to social simulation*. Cambridge University Press (cf. p. 8).

Suryanarayanan, V. et Theodoropoulos, G. (2013). «Synchronised Range Queries in Distributed Simulations of Multiagent Systems». *ACM Trans. Model. Comput. Simul.* 23.4, 25 :1–25 :25 (cf. p. 78).

Suzumura, T. et Kanezashi, H. (2012). «Highly Scalable X10-Based Agent Simulation Platform and Its Application to Large-Scale Traffic Simulation». In : *Proceedings of the 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications*. DS-RT '12. Dublin, Ireland : IEEE Computer Society, p. 243–250 (cf. p. 87).

Szer, D. et Charpillet, F. (2006). «Point-based dynamic programming for DEC-POMDPs». *AAAI* 6, p. 1233–1238 (cf. p. 5).

Talon, B., Kerkeni, I., Tliche, S. et Belaid-Ajroud, H. (2013). «Tracking a Collaborative Web2.0 E-Learning Environment». English. In : *Scaling up Learning for Sustained Impact*. Sous la dir. de D. Hernández-Leo, T. Ley, R. Klamma et A. Harrer. T. 8095. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 577–580 (cf. p. 66).

Thabet, I., Bouslimi, I., Hanachi, C. et Ghédira, K. (2011). «A Multi-agent Organizational Model for Grid Scheduling». English. In : *Agent and Multi-Agent Systems : Technologies and Applications*. Sous la dir. de J. O'Shea, N. Nguyen, K. Crockett, R. Howlett et L. Jain. T. 6682. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 148–158 (cf. p. 66).

Theraulaz, G., Perna, A. et Kuntz, P. (2012). «L'art de la construction chez les insectes sociaux». *Pour la science* 420, p. 28–35 (cf. p. 94).

Thiele, J. C. (2015). «Towards Rigorous Agent-Based Modelling». Thèse de doct. Georg-August-Universität Göttingen (cf. p. 10).

Thomas, V., Bourjot, C. et Chevrier, V. (2007). «Construction de systèmes multi-agents par apprentissage collectif à base d'interactions». *Revue d'Intelligence Artificielle* 21.5-6, p. 643–672 (cf. p. 19).

Topin, X., Fourcassié, V., Gleizes, M. P., Theraulaz, G. et Régis, C. (1999). «Theories and experiments on emergent behaviour : From natural to artificial systems and back». In : *European Conference on Cognitive Science, Siena* (cf. p. 5).

Tranier, J. (2007). «Vers une vision intégrale des systèmes multi-agents : Contribution à l'intégration des concepts d'agent, d'environnement, d'organisation et d'institution.» Thèse de doctorat. Université Montpellier II (cf. p. 20).

Treuil, J.-P., Drogoul, A. et Zucker, J.-D. (2008). *Modélisation et simulation à base d'agents : exemples commentés, outils informatiques et questions théoriques*. Dunod (cf. p. 4).

Troitzsch, K. (2009). «Multi-agent systems and simulation : a survey from an application perspective». In : *Multi-Agent Systems : Simulation and Applications*. Sous la dir. d'A. Uhrmacher et D. Weyns. Computational Analysis, Synthesis, and Design of Dynamic Systems. CRC Press - Taylor & Francis, p. 53–75 (cf. p. 3).

Turing, A. M. (1952). «The chemical basis of morphogenesis». *Philosophical Transactions of the Royal Society of London B : Biological Sciences* 237.641, p. 37–72 (cf. p. 96).

Unnikrishnan, R., Pantofaru, C. et Hebert, M. (2007). «Toward objective evaluation of image segmentation algorithms». *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29.6, p. 929–944 (cf. p. 38).

Valckenaers, P., Sauter, J., Sierra, C. et Rodriguez-Aguilar, J. (2007). «Applications and environments for multi-agent systems». English. *Autonomous Agents and Multi-Agent Systems* 14.1, p. 61–85 (cf. p. 29).

Viroli, M., Holvoet, T., Ricci, A., Schelfthout, K. et Zambonelli, F. (2007). «Infrastructures for the environment of multiagent systems». English. *Autonomous Agents and Multi-Agent Systems* 14.1, p. 49–60 (cf. p. 29).

Volterra, V. (1926). *Variations and fluctuations of the number of individuals in animal species living together*. McGraw-Hill (cf. p. 3).

Weiß, G., Rovatsos, M. et Nickles, M. (2003). «Capturing agent autonomy in roles and XML». In : *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. ACM, p. 105–112 (cf. p. 20).

Weyns, D., Boucké, N. et Holvoet, T. (2006a). «Gradient field-based task assignment in an AGV transportation system». In : *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. AAMAS '06. Hakodate, Japan : ACM, p. 842–849 (cf. p. 36).

Weyns, D. et Holvoet, T. (2007). «A Reference Architecture for Situated Multiagent Systems». English. In : *Environments for Multi-Agent Systems III*. Sous la dir. de D. Weyns, V. Parunak et F. Michel. T. 4389. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 1–40 (cf. p. 28). (doi livre)

Weyns, D. et Holvoet, T. (2004). «Formal Model for Situated Multi-Agent Systems». *Fundamenta Informaticae* 63. Sous la dir. de B. Dunin-Keplicz, p. 1–34 (cf. p. 16).

Weyns, D. et Holvoet, T. (2003). «Model for Simultaneous Actions in Situated Multi-agent Systems». English. In : *Multiagent System Technologies*. Sous la dir. de M. Schillo, M. Klusch, J. Müller et H. Tianfield. T. 2831. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 105–118 (cf. p. 27).

Weyns, D., Omicini, A. et Odell, J. (2007a). «Environment as a first class abstraction in multiagent systems». *Autonomous Agents and Multi-Agent Systems* 14.1, p. 5–30 (cf. p. 28, 29).

Weyns, D., Parunak, H. V. D. et Michel, F. (2005a). *Environments for Multi-Agent Systems*. T. 3374. Lecture Notes in Computer Science. Springer Berlin Heidelberg (cf. p. 28). (doi livre)

Weyns, D., Parunak, H. V. D. et Michel, F. (2006b). *Environments for Multi-Agent Systems II*. T. 3830. Lecture Notes in Computer Science. Springer Berlin Heidelberg (cf. p. 28). (doi livre)

Weyns, D., Parunak, H. V. D. et Michel, F. (2007b). *Environments for Multi-Agent Systems III*. T. 4389. Lecture Notes in Computer Science. Springer Berlin Heidelberg (cf. p. 28, 35). (doi livre)

Weyns, D., Parunak, H. V. D., Michel, F., Holvoet, T. et Ferber, J. (2005b). «Environments for Multiagent Systems : State-of-the-Art and Research Challenges». In : *Environments for Multi-Agent Systems*. Sous la dir. de D. Weyns, H. V. D. Parunak et F. Michel. T. 3374. Lecture Notes in Computer Science. New York, USA : Springer Berlin Heidelberg, p. 1–47 (cf. p. 9, 29, 30). pdf (doi livre)

Wolpert, L. (1968). «The French Flag Problem : A Contribution to the Discussion on Pattern Development and Regulation». In : *Towards a Theoretical Biology. Volume I*. Sous la dir. de C. H. Waddington. Edinburgh University Press Edinburgh, p. 125–133 (cf. p. 70).

Wooldridge, M. et Jennings, N. R. (1995). «Intelligent agents : Theory and practice». *Knowledge engineering review* 10.2, p. 115–152 (cf. p. 4).

Xu, C. et Prince, J. L. (1997). «Gradient vector flow : A new external force for snakes». In : *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE, p. 66–71 (cf. p. 40).

Zambonelli, F., Jennings, N. et Wooldridge, M. (2001). «Organisational Abstractions for the Analysis and Design of Multi-agent Systems». English. In : *Agent-Oriented Software Engineering*. Sous la dir. de P. Ciancarini et M. Wooldridge. T. 1957. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 235–251 (cf. p. 64).

Zeigler, B. P., Kim, T. G. et Praehofer, H. (2000). *Theory of modeling and simulation : integrating discrete event and continuous complex dynamic systems*. Academic Press (cf. p. 6, 7).

Zhao, C., Kaulakis, R., Morgan, J., Hiam, J., Ritter, F., Sanford, J. et Morgan, G. (2014). «Building social networks out of cognitive blocks : factors of interest in agent-based socio-cognitive simulations». English. *Computational and Mathematical Organization Theory*, p. 1–35 (cf. p. 3).

# Curriculum Vitæ

## 1. Identité et parcours professionnel

### État-civil et coordonnées


| | |
|---|---|
| Nom et prénom : | **Fabien Michel** |
| Date de naissance : | **11 décembre 1975** |
| Grade : | **maître de conférences depuis septembre 2005** |
| Affectation : | **IUT de Montpellier (dept. informatique)** |
| Unité de recherche : | **LIRMM, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (UMR 5506)** |
| Équipe de recherche : | **SMILE : System, Multi-agent, Interaction, Language, Evolution** |
| Adresse postale : | **LIRMM**<br>**161 rue Ada**<br>**34392 Montpellier Cedex 5** |
| Téléphone / télécopie : | ☎ **(+33) 4 67 41 86 75** 🖷 **(+33) 4 67 41 85 00** |
| Courriel : | **fmichel@lirmm.fr** |
| Site professionnel : | http://www.lirmm.fr/~fmichel |


### Parcours professionnel

| | |
|---|---|
| 2008 - : | **Maître de conférences** en informatique à l'**IUT de Montpellier**.<br>Laboratoire **LIRMM**, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, UMR 5506. Équipe SMILE |
| 2005-2008 : | **Maître de conférences** en informatique à l'**IUT de Reims**.<br>Laboratoire **CReSTIC**, Centre de Recherche en Sciences et Technologies de l'Information et de la Communication, EA 3804. Équipe MODECO |
| 2004-2005 : | **Chercheur contractuel** (LIRMM) |
| 2003-2004 : | **1/2 ATER en informatique** à l'université **Montpellier II** |
| 2002-2003 : | **Chercheur contractuel** (LIRMM) |
| 1999-2002 : | **Allocataire de recherche** (LIRMM) |

### Formation

1999-2004 : **Doctorat en informatique**, Université Montpellier II.
Titre : *Formalisme, outils et éléments méthodologiques pour la modélisation et la simulation multi-agents.*
Directeur : Jacques Ferber (Pr., Université Montpellier II). Rapporteurs : A. Drogoul (DR IRD), D.R.C. Hill (Pr. Université Blaise Pascal Clermont-Ferrand), H.V.D Parunak (DR Altarum Institute, Ann Arbor, MI, USA). Président du jury : Y. Demazeau (CR HDR. Institut IMAG, Grenoble). Examinateurs : J-P. Müller DR HDR CIRAD Montpellier). Document pdf

1998-1999 : **DEA d'informatique**, Université **Montpellier II**

1997-1998 : **Maîtrise d'informatique**, Université Claude Bernard **Lyon I**

1996-1997 : **Licence d'informatique**, Université Claude Bernard **Lyon I**

1993-1996 : **DEUG A**, filière 1, Université Claude Bernard **Lyon I**

## 2. Publications

### Résumé quantitatif des publications depuis 2005

Comme cela se fait parfois, pour différencier les conférences, nous avons utilisé la classification proposée par *The Computing Research and Education Association of Australasia* (CORE/ERA). On peut y lire :

*The portal uses the following ranking categories, derived primarily from earlier CORE ranking exercises :*

- A* - flagship conference
- A - excellent conference, determined by a mix of indicators
- B - good conference, determined by a mix of indicators
- C - other ranked conference venues
- Unranked - insufficient information is available to judge ranking (ici classé en workshops internationaux)

Par ailleurs, nous avons utilisé les catégories de publications proposées dans les recommandations du CNU : livre, article invité, revue, chapitre, conférence, workshop. Exceptés les rapports techniques, tous ces articles ont été évalués par un comité de lecture.

**Depuis 2005 :**

- livres édités : 3
- article invité : 1
- revues internationales : 3
- revues francophones : 5
- chapitres : 5
- conférence A* : 1 short paper
- conférences A : 2
- conférence B : 1
- conférences C : 6
- workshops internationaux : 7
- conférences francophones : 6 (un *best paper*)
- workshops francophones : 3 + 2 posters

## Liste exhaustive des publications (depuis 2000) :

### - direction d'ouvrages

1. Weyns, D., Parunak, H. V. D. et Michel, F. (2007). *Environments for Multi-Agent Systems III*. T. 4389. Lecture Notes in Computer Science. Springer Berlin Heidelberg. (doi livre)
2. Weyns, D., Parunak, H. V. D. et Michel, F. (2006). *Environments for Multi-Agent Systems II*. T. 3830. Lecture Notes in Computer Science. Springer Berlin Heidelberg. (doi livre)
3. Weyns, D., Parunak, H. V. D. et Michel, F. (2005). *Environments for Multi-Agent Systems*. T. 3374. Lecture Notes in Computer Science. Springer Berlin Heidelberg. (doi livre)

### - article invité

1. Michel, F., Ferber, J. et Drogoul, A. (2009). Multi-Agent Systems and Simulation : a Survey From the Agents Community's Perspective. Anglais. In : *Multi-Agent Systems : Simulation and Applications*. Sous la dir. d'A. Uhrmacher et D. Weyns. Computational Analysis, Synthesis, and Design of Dynamic Systems. CRC Press - Taylor & Francis, 3–52. pdf (doi livre)

### - revues internationales

1. Michel, F. (2013). Translating Agent Perception Computations into Environmental Processes in Multi-Agent-Based Simulations : A means for Integrating Graphics Processing Unit Programming within Usual Agent-Based Simulation Platforms. *Systems Research and Behavioral Science, Special Issue : Conceptual Models Of Complex Systems*, 30(6) :703–715. pdf
2. Helleboogh, A., Vizzari, G., Uhrmacher, A. et Michel, F. (2007). Modeling dynamic environments in multi-agent simulation. English. *Autonomous Agents and Multi-Agent Systems*, 14(1) :87–116. pdf
3. Gouaïch, A. et Michel, F. (2005). Towards a Unified View of the Environment(s) within Multi-Agent Systems. *Informatica, Special Issue : Hot Topics in European Agent Research I*, 29(4) :423–432. Sous la dir. d'A. Omicini, P. Petta et M. Gams. pdf

*Impact factors* : (1) 0.474 (ISI JCR 2012) (2) 1.108 (ISI JCR 2013) (3) 1.1 (SJR 2007)

### - revues francophones

1. Hermellin, E., Michel, F. et Ferber, J. (2015). État de l'art sur les Simulations Multi-Agents et le GPGPU : évolution et perspectives de recherches. *Revue d'Intelligence Artificielle*, 29(3-4) :423–449.
2. Michel, F. (2014). Délégation GPU des perceptions agents : intégration itérative et modulaire du GPGPU dans les simulations multi-agents. Application sur la plate-forme TurtleKit 3. *Revue d'Intelligence Artificielle*, 28(4) :485–510.
3. Mazouzi, S., Guessoum, Z. et Michel, F. (2009). Une approche multi-agent pour la segmentation d'images de profondeur à base d'objets polyédriques. Une nouvelle approche de segmentation d'images. Français. *Technique et Science Informatiques*, 28(3) :365–393. pdf
4. Mazouzi, S., Guessoum, Z. et Michel, F. (2008). Segmentation collective d'images à base d'objets polyédriques et courbés. Français. *I3 Information - Interaction - Intelligence*, 8(2) :229–251. Sous la dir. de G. I3/CNRS. pdf
5. Michel, F. (2007). Le modèle IRM4S, de l'utilisation des notions d'influence et de réaction pour la simulation de systèmes multi-agents. *Revue d'Intelligence Artificielle*, 21(5-6) :757–779. Sous la dir. d'Y. Demazeau. pdf

**- chapitres de livre (post proceedings d'articles sélectionnés et révisés avec second processus de relecture)**

1. Michel, F., Ferber, J., Laur, P.-A. et Aleman, F. (2011). Situational Programming : Agent Behavior Visual Programming for MABS Novices. In : *Multi-Agent-Based Simulation XI*. Sous la dir. de T. Bosse, A. Geller et C. Jonker. T. 6532. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1–15. pdf (doi livre)
2. Mazouzi, S., Guessoum, Z., Michel, F. et Batouche, M. (2008). An Agent-Based Approach for Range Image Segmentation. In : *Massively Multi-Agent Technology*. Sous la dir. de P. S. Nadeem Jamali et T. Sugawara. T. 5043. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 146–161. pdf (doi livre)
3. Weyns, D., Parunak, H. V. D., Michel, F., Holvoet, T. et Ferber, J. (2005b). Environments for Multiagent Systems : State-of-the-Art and Research Challenges. In : *Environments for Multi-Agent Systems*. Sous la dir. de D. Weyns, H. V. D. Parunak et F. Michel. T. 3374. Lecture Notes in Computer Science. New York, USA : Springer Berlin Heidelberg, 1–47. pdf (doi livre)
4. Gouaïch, A., Michel, F. et Guiraud, Y. (2005). MIC* : A Deployment Environment for Autonomous Agents. In : *Environments for Multi-Agent Systems*. Sous la dir. de D. Weyns, H. V. D. Parunak et F. Michel. T. 3374. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 109–126. pdf (doi livre)
5. Ferber, J., Michel, F. et Bàez-Barranco, J.-A. (2005). AGRE : Integrating Environments with Organizations. In : *Environments for Multi-Agent Systems*. Sous la dir. de D. Weyns, H. V. D. Parunak et F. Michel. T. 3374. Lecture Notes in Computer Science. New York, USA : Springer Berlin Heidelberg, 48–56. pdf (doi livre)
6. Michel, F., Gouaïch, A. et Ferber, J. (2004). Weak Interaction and Strong Interaction in Agent Based Simulations. In : *Multi-Agent-Based Simulation III*. Sous la dir. de D. Hales, B. Edmonds, E. Norling et J. Rouchier. T. 2927. Lecture Notes in Computer Science. Melbourne, Australia : Springer Berlin Heidelberg, 43–56. pdf (doi livre)
7. Ferber, J., Gutknecht, O. et Michel, F. (2004). From Agents to Organizations : an Organizational View of Multi-Agent Systems. In : *Agent-Oriented Software Engineering IV*. Sous la dir. de P. Giorgini, J. P. Müller et J. Odell. T. 2935. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 214–230. pdf (doi livre)

**- conférences internationales**

**A***

1. Michel, F. (2007). The IRM4S Model : The Influence/Reaction Principle for Multiagent Based Simulation (short paper). In : *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, USA, May 14-18, 2007*. Sous la dir. d'E. H. Durfee, M. Yokoo, M. N. Huhns et O. Shehory. AAMAS '07. Honolulu, Hawaii : ACM, 903–905. pdf

**A**

1. Carlési, N., Michel, F., Jouvencel, B. et Ferber, J. (2011). Generic Architecture for Multi-AUV Cooperation Based on a Multi-Agent Reactive Organizational Approach. Anglais. In : *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2011 San Francisco, California, Sept. 25-30, 2011*. San Francisco, California, USA, 5041–5047. pdf
2. Beurier, G., Michel, F. et Ferber, J. (2006). A morphogenesis model for multiagent embryogeny. In : *ALIFE X : Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems, Bloomington, IN USA, June 3-7, 2006*. Sous la dir. de L. M. Rocha, L. S. Yaeger, M. A. Bedau, D. Floreano, R. L. Goldstone et A. Vespignani. Cambridge, MA, USA : MIT Press, 84–90. pdf
3. Gutknecht, O., Ferber, J. et Michel, F. (2001). Integrating tools and infrastructures for generic multi-agent systems. In : *Proceedings of the Fifth International Conference on Autonomous Agents, Agents'01*. Montreal, Quebec, Canada : ACM press, 441–448. pdf

**B**

1. Mazouzi, S., Guessoum, Z., Michel, F. et Batouche, M. (2007). A Multi-agent Approach for Range Image Segmentation with Bayesian Edge Regularization. In : *Proceedings of the* $9^{th}$ *International Conference on Advanced Concepts for Intelligent Vision Systems, ACIVS 2007, Delft, The Netherlands, August 28-31, 2007.* Sous la dir. de J. Blanc-Talon, W. Philips, D. Popescu et P. Scheunders. T. 4678. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 449–460. pdf

**C**

1. Mahdi, G., Francillette, Y., Gouaïch, A., Michel, F. et Hocine, N. (2013). Level Of Detail Based AI Adaptation for Agents in Video Games. Anglais. In : *ICAART 2013 - Proceedings of the 5th International Conference on Agents and Artificial Intelligence, Volume 2, Barcelona, Spain, 15-18 February, 2013.* Sous la dir. de J. Filipe et A. L. N. Fred. SciTePress, 182–194. pdf
2. Blanchart, E., Cambier, C., Canape, C., Gaudou, B., Ho, T.-N., Ho, T.-V., Lang, C., Michel, F., Marilleau, N. et Philippe, L. (2011). EPIS : A Grid Platform to Ease and Optimize Multi-agent Simulators Running. In : *Proceedings of the 9th International Conference on Practical Applications of Agents and Multiagent Systems, PAAMS 2011, Salamanca, Spain, 6-8th April, 2011.* Sous la dir. d'Y. Demazeau, M. Pechoucek, J. Corchado et J. Pérez. T. 88. Advances in Intelligent and Soft Computing. Springer Berlin Heidelberg, 129–134. pdf
3. Mahdi, G., Gouaïch, A. et Michel, F. (2010). Towards an Integrated Approach of Real-Time Coordination for Multi-agent Systems. In : *Agent and Multi-Agent Systems : Technologies and Applications, Proceedings of the* $4^{th}$ *KES International Symposium, KES-AMSTA 2010, Gdynia, Poland, June 23-25, 2010.* Sous la dir. de P. Jędrzejowicz, N. Nguyen, R. Howlet et L. Jain. T. 6070. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 253–262. pdf
4. Mazouzi, S., Guessoum, Z. et Michel, F. (2009). A Distributed and Collective Approach for Curved Object-Based Range Image Segmentation. In : *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Proceedings of the 14th Iberoamerican Conference on Pattern Recognition, CIARP 2009, Guadalajara, Jalisco, Mexico, November 15-18, 2009.* Sous la dir. d'E. Bayro-Corrochano et J.-O. Eklundh. T. 5856. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 201–208. pdf
5. Mahdjoub, J., Guessoum, Z., Michel, F. et Herbin, M. (2006). A multi-agent approach for the edge detection in image processings. In : *Proceedings of the Fourth European Workshop on Multi-Agent Systems, EUMAS'06, Lisbon, Portugal, December 14-15, 2006.* Sous la dir. de B. Dunin-Keplicz, A. Omicini et J. Padget. T. 223. CEUR-WS.org. pdf
6. Mazouzi, S., Guessoum, Z., Michel, F. et Batouche, M. (2006). A potential field-based multi-agent model for contour detection in range images. In : *Proceedings of the* $2^{nd}$ *IEEE International Conference on Intelligent Computer Communication and Processing, ICCP 2006, Cluj-Napoca, Romania, 1-2 September, 2006.* IEEE Computer Society, 69–75. pdf
7. Michel, F., Ferber, J. et Gutknecht, O. (2001). Generic Simulation Tools Based on MAS Organization. In : *Proceedings of the Tenth European Workshop on Modelling Autonomous Agents in a Multi Agent World, MAAMAW'2001, Annecy, France, 2-4 May, 2001.* Sous la dir. d'Y. Demazeau et F. J. Garijo. Annecy, France. pdf

**- workshops internationaux**

1. Hermellin, E. et Michel, F. (2015). GPU Environmental Delegation of Agent Perceptions : Application to Reynolds's Boids. In : *accepté dans Thirteenth Multi-Agent-Based Simulation, MABS XIII @ AAMAS.*
2. Michel, F. (2012). GPU Environmental Delegation of Agent Perceptions for MABS. In : *ICCS'12, IEEE International Conference on Complex Systems, Agadir, Morocco, November 5-6.* Sous la dir. de M. Essaaidi et M. Nemiche. IEEE Computer Society, 1–6. pdf
3. Gruau, F. et Michel, F. (2011). Rekindling Parallelism. In : *Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2011 Fifth IEEE Conference on.* Washington, DC, USA : IEEE Computer Society, 55–60. pdf

4. Mahdi, G., Gouaïch, A. et Michel, F. (2010). Real-time Coordination of Multi-Agent Systems : An Integrated View. In : *Agents in Real-time and Dynamic Environments Workshop ARDE @ AAMAS'10, Toronto, Canada, May 10-14 2010*. Toronto Canada. pdf
5. Mazouzi, S., Guessoum, Z., Michel, F. et Batouche, M. (2007). A Multi-agent Approach for Range Image Segmentation. In : *Multi-Agent Systems and Applications V, Proceedings of the $5^{th}$ International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2007, Leipzig, Germany, September 25-27, 2007*. Sous la dir. de H.-D. Burkhard, G. Lindemann, R. Verbrugge et L. Varga. T. 4696. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1–10. pdf
6. Michel, F., Beurier, G. et Ferber, J. (2005). The TurtleKit Simulation Platform : Application to Complex Systems. In : *Workshops Sessions of the Proceedings of the 1st International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2005, November 27 - December 1, 2005, Yaoundé, Cameroon*. Sous la dir. d'A. Akono, E. Tonyé, A. Dipanda et K. Yétongnon. Yaoundé, Cameroon : IEEE, 122–128. pdf
7. Beurier, G., Michel, F. et Ferber, J. (2005). Towards an evolution model of multiagent organisms. In : *Multi-Agents for modeling Complex Systems, Satellite Workshop of the European Conference on Complex Systems (MA4CS'05 @ ECCS'05)*. pdf
8. Gouaïch, A., Guiraud, Y. et Michel, F. (2003b). MIC* : An Agent Formal Environment. In : *7th World Multiconference on Systemics, Cybernetics and Informatics, SCI 2003, Orlando, Florida, USA, July 2003*. Orlando, USA : IIIS. pdf
9. Ferber, J., Michel, F. et Gutknecht, O. (2003). Agent/Group/Roles : Simulating with organizations. In : *Fourth International Workshop on Agent-Based Simulation (ABS 4), 28-30 Avril, Montpellier, France*. Sous la dir. de J.-P. Müller. Montpellier, France, 12.
10. Gouaïch, A., Guiraud, Y. et Michel, F. (2003a). Building Safe Multi-Agent Applications by Formalising the Deployment Environment. In : *Safe Agents Workshop @ AAMAS 2003, Melbourne, Australia, July 14-18, 2003*. Melbourne, Australia, 5. pdf
11. Michel, F., Beurier, G., Gouaïch, A. et Ferber, J. (2003a). The TurtleKit platform : application to multi-level emergence (poster). In : *Fourth International Workshop on Agent-Based Simulation (ABS 4), Montpellier, 28-30 Avril, France*. Sous la dir. de J.-P. Müller. Montpellier, France.
12. Michel, F., Gouaïch, A. et Ferber, J. (2003b). Weak and strong interactions in agent-based simulation (poster). In : *Fourth International Workshop on Agent-Based Simulation (ABS 4), Montpellier, 28-30 Avril, France*. Sous la dir. de J.-P. Müller. Montpellier, France.

**- conférences francophones**

1. Hermellin, E. et Michel, F. (2015). Délégation GPU des perceptions agents : Application aux Boids de Reynolds. In : *JFSMA 15 - Vingt-troisièmes Journées Francophones sur les Systèmes Multi-Agents, Rennes, France, 29 juin - 1er juillet*. Sous la dir. de L. Vercouter et G. Picard, 185–194.
2. Hermellin, E., Michel, F. et Ferber, J. (2014). Systèmes Multi-Agents et GPGPU : état des lieux et directions pour l'avenir. In : *Principe de parcimonie - JFSMA 14 - Vingt-deuxièmes Journées Francophones sur les Systèmes Multi-Agents, Valence, France, 8-10 octobre*. Sous la dir. de R. Courdier et J.-P. Jamont. Cépaduès Editions. pdf
3. Michel, F. (2013). Intégration du calcul sur GPU dans la plate-forme de simulation multi-agent générique TurtleKit 3. Français. In : *Dynamiques, couplages et visions intégratives - JFSMA 13 - Vingt-et-unièmes journées francophones sur les systèmes multi-agents, Lille, France, Juillet 3-5, 2013*. Sous la dir. de S. Hassas et M. Morge. Cépaduès Editions, 189–198 (best paper). pdf

4. Michel, F., Ferber, J., Laur, P.-A. et Aleman, F. (2010). Programmation situationnelle : programmation visuelle de comportements agents pour non informaticiens (présentation courte). Français. In : *Systèmes Multi-agents, Défis Sociétaux - JFSMA 10 - Dix-huitièmes journées francophones sur les systèmes multi-agents, Mahdia, Tunisia, October 18-20, 2010*. Sous la dir. de M. Occello et L. Rejeb. Cépaduès Editions, 65–74. pdf
5. Mazouzi, S., Guessoum, Z. et Michel, F. (2008). Une approche multi-agent pour la segmentation d'images de profondeur. In : *16e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle RFIA 2008*. Sous la dir. d'I. Bloch et P. Marquis. Amiens : AFRIF-AFIA. pdf
6. Michel, F. (2006). Le modèle IRM4S : le principe Influence/Réaction pour la simulation de systèmes multi-agents. In : *14 èmes Journées Francophones sur les Systèmes Multi-Agents, JFSMA 2006, Annecy, France, 18-20 octobre 2006*. Sous la dir. de V. Chevrier et M.-P. Huget. Annecy : Hermès, Paris, 9–22. pdf
7. Simonin, O., Michel, F., Chapelle, J. et Ferber, J. (2002). Un simulateur de systèmes multi-robots dans MadKit (paper court et démonstration). In : *Systèmes multi-agents et systèmes complexes : ingénierie, résolution de problèmes et simulation - JFSMA 02 - Dixièmes journées francophones sur les systèmes multi-agents, Lille, France, October 28-30, 2002*. Sous la dir. de P. Mathieu et J.-P. Müller. Lille : Hermès Lavoisier Editions, 167–170. pdf
8. Michel, F., Bommel, P. et Ferber, J. (2002). Simulation Distribuée Interactive sous MadKit (paper court et démonstration). In : *Systèmes multi-agents et systèmes complexes : ingénierie, résolution de problèmes et simulation - JFSMA 02 - Dixièmes journées francophones sur les systèmes multi-agents, Lille, France, October 28-30, 2002*. Sous la dir. de P. Mathieu et J.-P. Müller. Lille : Hermès Lavoisier Editions, 175–178. pdf
9. Michel, F. (2001). Le modèle Influence/Réaction pour la Simulation Multi-Agents (poster et papier). In : *Actes des Premières Journées Francophones des Modèles Formels de l'Interaction, MFI' 01, Toulouse, France, 21-23 mai 2001*. Sous la dir. de B. Chaib-draa et P. Enjalbert. T. 3. Toulouse, 391–406.
10. Gutknecht, O., Ferber, J. et Michel, F. (2000b). MadKit : Une expérience d'architecture de plate-forme multi-agent générique. In : *Systèmes multi-agents : Méthodologie, technologie et expériences - JFIADSMA 00 - huitième journées francophones d'Intelligence Artificielle et systèmes multi-agents, Saint-Jean-la-Vêtre, Loire, France, 2-4 octobre, 2000*. Sous la dir. de S. Pesty et C. Sayettat-Fau. Saint-Jean-la-Vêtre, Loire, France : Hermès Lavoisier Editions, 223–236. pdf
11. Gutknecht, O., Ferber, J. et Michel, F. (2000a). La plate-forme MadKit et l'outil de conception Sedit (démonstration). In : *Systèmes multi-agents : Méthodologie, technologie et expériences - JFIADSMA 00 - huitième journées francophones d'Intelligence Artificielle et systèmes multi-agents, Saint-Jean-la-Vêtre, Loire, France, October 2-4, 2000*. Sous la dir. de S. Pesty et C. Sayettat-Fau. Saint-Jean-la-Vêtre, Loire, France : Hermès Lavoisier Editions, 281–284.

**- workshops francophones**

1. Hermellin, E., Michel, F. et Ferber, J. (2014). État de l'art des simulations multi-agents sur GPU. In : *12e Rencontres des Jeunes Chercheurs en Intelligence Artificielle*.
2. Michel, F. (2007). Le modèle IRM4S : utilisation du principe Influence/Réaction pour la simulation de systèmes multi-agents (Poster). In : *Colloque ARCo-07 : Cognition - Complexité - Collectif, Loria, Nancy, France, novembre 2007*. pdf
3. Mahdjoub, J., Michel, F. et Gusessoum, Z. (2007). Segmentation dynamique d'images : Application à des signaux unidimentionnels (Poster). In : *Colloque ARCo-07 : Cognition - Complexité - Collectif, Loria, Nancy, France, novembre 2007*. pdf
4. Mazouzi, S., Guessoum, Z., Michel, F. et Batouche, M. (2007). Un modèle d'interaction multi-agent basé sur le champ de potentiel appliqué à la segmentation d'images de profondeur. In : *3èmes Journées internationales sur l'Information Graphique, JIG'2007, Constantine, Algérie, 29-30 Octobre, 2007*. Constantine, Algérie, 258–266. pdf

5. Michel, F., Daniel, G., Ferber, J. et Phan, D. (2005). Projet integration Moduleco/Madkit : premiers résultats. In : *Joint Conference on Multi-Agent Modelling for Environmental Management SMAGET 05*. Sous la dir. de N. Ferrand, D. Batten, P. Perez, J.-P. Müller, M. L. Bars, A. Dray et A. Drogoul. short paper. pdf
6. Gouaïch, A., Guiraud, Y. et Michel, F. (2003). Structure algébrique pour des environnements à entités interactionnelles et mobiles. In : *Sixièmes rencontres des Jeunes Chercheurs en Intelligence Artificielle, RJCIA'2003, Laval, France, 1-2 juillet, 2003*. Sous la dir. de F. D. de Saint-Cyr. AFIA. Laval, France : Association Française pour l'Itelligence Artificielle AFIA, 12. pdf
7. Michel, F. (2003). Interaction forte et interaction faible dans les simulations multi-agents. In : *Le statut épistémologique de la simulation - 10èmes journées de Rochebrune : rencontre interdisciplinaire sur les systèmes complexes naturels et artificiels*. Sous la dir. de G. Deffuant, J.-L. Dessalles, T. Fuhs, H. Glotin et J.-P. Müller. Rochebrune, Megève, France : Telecom Paris, 163–176. pdf
8. Michel, F. (2002). Modèles d'action et simulations multi-agents distribuées. In : *Dixièmes Doctorantes de l'Ecole Doctorale I2S JDOC'02*. Sous la dir. de M. Robert. Université des sciences et technioques du Languedoc. Montpellier : Ecole Doctorale Informations Structures et Systèmes, 93–96. pdf
9. Michel, F., Gutknecht, O. et Ferber, J. (2001). Une méthodologie pour la conception de simulateur multi-agents basée sur l'organisation. In : *Plate-forme AFIA 2001 - Atelier : Méthodologies et Environnements pour les Systèmes Multi-agents*. Sous la dir. de Z. Guessoum et M. Occello. Grenoble, France, 115–124. pdf
10. Michel, F. (2000). Une approche méthodologique pour la conception et l'analyse de simulateur multi-agents. In : *Cinquièmes rencontres des Jeunes Chercheurs en Intelligence Artificielle, RJCIA'00, Lyon, France, 11-13 septembre, 2000*. Sous la dir. de M. Ayel et J.-M. Fouet. AFIA. Lyon, France : Association Française pour l'Itelligence Artificielle AFIA, 269–279. pdf

**- thèse de doctorat**

1. Michel, F. (2004). Formalisme, outils et éléments méthodologiques pour la modélisation et la simulation multi-agents. Thèse de doct. Université Montpellier II. pdf

**- rapports techniques**

1. Bàez-Barranco, J.-A., Ferber, J. et Michel, F. (2005). *Extension du Modèle AGR et Propositions d'Application aux Standards FIPA*. Rapp. tech. RR LIRMM 00106617. Montpellier : Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier, LIRMM, CNRS.
2. Bàez-Barranco, J.-A., Ferber, J. et Michel, F. (2004b). *Propositions pour la Modélisation Organisationnelle*. Rapp. tech. RR LIRMM 00109191. Montpellier : Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier, LIRMM, CNRS.
3. Bàez-Barranco, J.-A., Ferber, J. et Michel, F. (2004a). *État de l'Art sur les Organisations dans les Systèmes Multi-Agents*. Rapp. tech. RR LIRMM 00109192. Montpellier : Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier, LIRMM, CNRS, 50 p.
4. Michel, F. (2002). *An Introduction to TurtleKit : a Platform for Building Logo Based Multi-Agent Simulations with MadKit*. Rapp. tech. RR LIRMM 02215. Montpellier : Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier, LIRMM, CNRS. pdf
5. Gutknecht, O., Ferber, J. et Michel, F. (2000). *Madkit : Une architecture de plate-forme multi-agent générique*. Rapp. tech. RR LIRMM 00061. Montpellier : Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier, LIRMM, CNRS. pdf

**Références web**

HAL archives-ouvertes.fr dblp computer science bibliography SpringerLink Google scholar

# 3. Activités scientifiques

## - Encadrement doctoral

Résumé : 1 codirection en cours, 4 thèses soutenues (coencadrant).

1. Emmanuel Hermellin, *Simulation multi-agents et calcul haute performance*, débutée en octobre 2013 (allocation ministérielle). Codirecteur (70 %) [8] avec Jacques Ferber.
2. Nicolas Carlési, *Coopération entre véhicules sous-marins autonomes : une approche organisationnelle réactive multi-agent*, soutenue le 19 décembre 2013 (projet ANR C-FLAM). Direction : Bruno Jouvencel et Jacques Ferber, coencadré à 33 %.
3. Ghulam Mahdi, *Level of Detail in Agent Societies for Games*, soutenue le 21 mai 2013 (financement géré par la Société Française d'Exportation de Ressources Éducatives SFERE) en partenariat avec le gouvernement pakistanais). Direction : Stefano Cerri, coencadré à 35% avec Abdelkader Gouaïch (35%).
4. Jason Mahdjoub, *Vers un système de vision auto-adaptatif à base de systèmes multi-agents*, soutenue le 15 décembre 2011 (allocation ministérielle). Direction : Zahia Guessoum, coencadré à 30 %.
5. Smaine Mazouzi, *Reconnaissance de formes par les systèmes auto-organisés. Application aux images de profondeur*, soutenue le 18 mai 2008 (cotutelle franco-algérienne). Direction : Zahia Guessoum et Mohamed Batouche, coencadré à 20 %.

## - Coorganisation d'événements scientifiques internationaux

— E4MAS workshop series @ AAMAS : 2004, 2005, 2006, 2014.
— Sixth International Workshop "From Agent Theory to Agent Implementation" AT2AI-6 @ AAMAS 2008
— Advances in computer simulation track - SIM@SAC 2009 et 2010
— MAS4SCM (*Multi-Agent Systems for Supply Chain Management*) special session @ ICSSSM 2006

## - Comités de programme internationaux et nationaux

- Résumé : A* (6), B (3), C (10), autres et francophones (35)

— **AAMAS**, International Conference on Autonomous Agents and Multiagent Systems, **2005, 2009, 2010, 2013, 2014** ($13^{th}$).
— **IJCAI 2011**, $22^{nd}$ International Joint Conference on Artificial Intelligence
— **Wintersim 2012**, $44^{th}$ Winter Simulation Conference.
— **SIM@SAC 2008**, Special Track on Advances in Computer Simulation @ ACM SAC ($23^{rd}$).
— **APSLA@SAC 2008**, Special Track on Agent-Oriented Programming, Systems, Languages, and Applications.
— **PAAMS**, International Conference on Practical Applications of Agents and Multi-Agent Systems, de **2009 à 2015** ($13^{th}$), pour les sept dernières éditions.
— **EUMAS**, European Workshop on Multi-Agent Systems, **2006, 2007, 2008** ($6^{th}$).
— **AISSAI 2010**, Artificial Intelligence for Simulation | Simulation for Artificial Intelligence Workshop @ ECAI 2010, $19^{th}$ European Conference on Artificial Intelligence.
— **ACSys**, $8^{th}$ à $11^{th}$ Workshop on Agents for Complex Systems @ SYNASC, **2011 à 2014**.
— **SIMUTools**, Conference on Simulation Tools and Techniques, de **2012 à 2015** ($8^{th}$).

8. Par accréditation de l'école doctorale I2S.

— **ABModSim**, Symposium on Agent Based Modeling and Simulation @ EMCSR **2006, 2008, 2010, 2012**, $18^{th}$ $19^{th}$ $20^{th}$ and $21^{st}$ European Meetings on Cybernetics and Systems Research.
— **Agent Cities**, International Workshop on Agent-based Modeling and Simulation of Cities @ ANT 2014, 2015 $5^{th}$ and $6^{th}$ Int. Conference on Ambient Systems, Networks, and Technologies.
— **SASO**, International Conference on Self-Adaptive and Self-Organizing Systems, demo selection committee, **2012 et 2013** ($7^{th}$).
— **AT2AI-7**, $7^{th}$ International Workshop "From Agent Theory to Agent Implementation" @ EMCSR 2010.
— **ESAW**, International Workshop "Engineering Societies in the Agents World", **2006, 2007, 2008** ($9^{th}$).
— **IBERAMIA 2014**, $14^{th}$ Ibero-American Conference on Artificial Intelligence.
— **EUSPN 2013**, $4^{th}$ International Conference on Emerging Ubiquitous Systems and Pervasive Networks, session "Multiagent Systems and Intelligent Computing".
— **MASW' 13**, First International Workshop on Multi-level Agent-based Simulation in Smart Cities @ ANT 2013, $4^{th}$ Int. Conference on Ambient Systems, Networks, and Technologies..
— **CTAACS** (**2012, 2013**), First and Second Conference on Theoretical and Applicative Aspects of Computer Science.
— **UbiTheraGames 2011**, $1^{st}$ International Workshop on Therapeutic Serious Games and Pervasive Computing @ $5^{th}$ International Conference on Pervasive Computing Technologies for Healthcare.
— **ANT 2010**, First International Conference on Ambient Systems, Networks and Technologies.
— **ECOSOA 2008**, International Workshop on Environment-Mediated Coordination in Self-Organizing and Self-Adaptive Systems (@ SASO 2008).
— **EEMMAS 2007**, Workshop "Engineering Environment-Mediated Multiagent Systems" @ $4^{th}$ European Conference on Complex Systems ECCS 2007.

— **JFSMA**, Journées Francophones sur les Systèmes Multi-Agents, de **2012 à 2015** ($23^{e}$).
— **RFIA 2014**, Dix-neuvième congrès francophone sur la Reconnaissance des Formes et l'Intelligence Artificielle.

## - Évaluation d'articles soumis à des revues

— JAAMAS, Journal of Autonomous Agents and Multi-Agent Systems. (3)
— TSC, IEEE Transactions on Services Computing. (1)
— TOMACS, ACM Transactions on Modeling and Computer Simulation. (3)
— TAAS, ACM Transactions on Autonomous Adaptive Systems. (3)
— JPDC, Journal of Parallel and Distributed Computing. (1)
— SCS simulation journal (special issue on environment and biology). (1)

— RIA, Revue d'Intelligence Artificielle, numéro spécial "modèles multi-agents pour des environnements complexes". (1)

## - Jurys de thèse

1. Nicolas Carlési, *Coopération entre véhicules sous-marins autonomes : une approche organisationnelle réactive multi-agent*, 2013, université de Montpellier 2 (coencadrant). Directeurs : Bruno Jouvencel et Jacques Ferber.
2. Jean-Baptiste Soyez, *Conception et modélisation de systèmes de systèmes : une approche multi-agents multi-niveaux*, 2013, université Lille 1, (examinateur). Directeurs : Rochdi Merzouki et Daniel Dupont. Coencadrant : Gildas Morvan.
3. Ghulam Mahdi, *Level of Detail in Agent Societies for Games*, 2013, université de Montpellier 2 (coencadrant). Directeur : Stefano Cerri. Coencadrant : Abdelkader Gouaïch.

4. Jason Mahdjoub, *Vers un système de vision auto-adaptatif à base de systèmes multi-agents*, 2011, université de Reims (coencadrant). Directeur : Zahia Guessoum.
5. Bruno Bonté, *Modélisation et simulation de l'interdépendance entre l'objet, l'observateur et le modèle de l'objet dans la Triade de Minsky. Application à la surveillance épidémiologique en santé animale*, 2011, CIRAD Montpellier (examinateur). Directeur : Jean-Pierre Müller. Coencadrant : Raphaël Duboz
6. Nicolas Sébastien, *Distribution et Parallélisation de Simulations Orientées Agent*, 2009, université de la Réunion (examinateur). Directeur : Rémy Courdier. Coencadrant : Marc-Philippe Huget.
7. John Tranier, *Vers une vision intégrale des systèmes multi-agents ; Contribution à l'intégration des concepts d'agent, d'environnement, d'organisation et d'institution*, 2007, université Montpellier 2 (examinateur). Directeur : Jacques Ferber.
8. Grégory Beurier, *Codage indirect de la forme dans les Systèmes Multi-Agents ; Emergence multi-niveaux, Morphogenèse et Evolution*, 2007, université Montpellier 2 (examinateur). Directeur : Jacques Ferber.

### - Participation à des projets et groupes de recherche nationaux

— participation au projet ANR C-FLAM, 10/2009 → 10/2012, (cf. thèse Carlési 6).
— participation au projet EPIS (Easy, Parallel and Interactive Sworm), 01/2010 → 12/2010, réalisé dans le cadre d'une demande de soutien aux projets informatiques dans les équipes scientifiques de l'IRD. Ce projet a réuni plusieurs antennes de l'IRD (Bondy, Dakar, Hanoï, Montpellier) ainsi que les universités de Toulouse, de Franche-Comté et de Montpellier.
— participation au projet "Intégration – MMM", financé par le CNRS (2004-2005). MMM a eu pour objectif de développer un système ouvert et modulaire de modélisation et simulation multi-agents à portée interdisciplinaire, par intégration de solutions préexistantes. En collaboration avec Denis Phan (économiste, université Paris IV), Gilles Daniel (ETH Zurich) et Jacques Ferber.
— membre du réseau AEGSST (01/2010 → 06/2013), Approche Énactive pour la Gouvernance de Systèmes Socio-Techniques, appartenant au Réseau National des Systèmes Complexes, RNSC.
— de 2001 à 2005, membre du groupe de travail *(modélisation et simulation individus-centrées)* du thème *systèmes multi-agents et simulation* du *GDR* $I^3$, responsable Jean-Pierre Müller (CIRAD).

### - Développements logiciels

— MaDKit, *Multi-Agent Development Kit*
  — site : www.madkit.net
  — sources : github.com/fmichel/MaDKit

— TurtleKit, *simulation de systèmes complexes*
  — site : www.turtlekit.org
  — sources : github.com/fmichel/TurtleKit

## 4. Responsabilités scientifiques et pédagogiques

### - Contrats de recherche

— Responsable du contrat 044009 avec l'entreprise FEERIK, 01/07/2009 → 31/12/2011 : prototypage d'un jeu vidéo en ligne basé sur une simulation de bataille entre entités artificielles intelligentes.
— Responsable du contrat 056427 avec l'entreprise Mezoa (suite incubation LRI), 01/12/2010 → 01/05/2011 : utilisation du GPGPU pour la simulation multi-agent.
— Responsable du contrat 060542, avec l'entreprise Feerik, 01/12/2010 → 30/06/2011 : prototypage d'une intelligence artificielle pour un jeu de cartes en ligne (Eredan).
— Coresponsable avec Abdelkader Gouaïch du contrat 100929 avec l'entreprise Altera Group, 01/12/2010 → 01/12/2011 : expertise pour la réalisation d'un module d'adaptation pour le système Kodro (logiciel pour personnes atteintes d'Alzheimer).

### - Participation à des comités de sélection (MCF)

1. MC 4317, Lyon I, 2015.
2. MC 4258, Université Paris 6, 2014.
3. MC 0726, Université Toulouse II-Le Mirail, 2013.
4. MC 0172, Université de la Réunion, 2012.
5. MC 4054, Lyon I, 2012.
6. MC 0869 et 2053 (deux postes), Lyon I, 2011.
7. MC 0345, Université de la Réunion, 2011.

### - Responsabilité au sein d'une équipe de recherche

De septembre 2012 à septembre 2014 : responsable adjoint de l'équipe de recherche SMILE (LIRMM), dirigée par Jacques Ferber.

### - Vue synthétique de l'expérience d'enseignement

Nous avons effectué environ 2500 heures d'enseignement (équivalent TD) dans diverses matières liées à la programmation et à l'analyse (fonctionnel, orienté objet, multi-agent, système, réseaux, web, génie logiciel, base de données, architecture) pour tous niveaux (de équivalent L1 à M2). Le tableau suivant résume les différentes fonctions dans lesquelles nous avons assuré ces enseignements.

| Année | Activité | Lieu d'exercice |
|---|---|---|
| 08- | **Maître de conférences** en informatique | **IUT de Montpellier**, département informatique **Université Montpellier II** |
| 05-08 | **Maître de conférences** en informatique | **IUT de Reims**, département informatique **Université de Reims** |
| 03-04 | 1/2 **ATER** en informatique | **Université Montpellier II** |
| 99-02 | **Allocataire de recherche** | **Université Montpellier II** |

### - Responsabilités de modules d'enseignement

— module de POO (S3, IUT de Reims en 2006/2007 et 2007/2008).
— module de POO avancée (S4, IUT de Reims en 2006/2007 et 2007/2008).
— Coresponsable, module de génie logiciel avancé (S4, IUT de Reims en 2007/2008).
— Responsable, module IA (S4, IUT de Reims en 2007/2008).
— Coresponsable, module SMA (Master 2, Reims en 2005/2006 et 2006/2007).
— module de POO avancée, licence professionelle (LP) PGI depuis 2008/2009.
— module de programmation Web, LP API DAE depuis 2008/2009.
— module de POO avancée, LP API DAE depuis 2010/2011.
— module de programmation mobile, année spéciale depuis cette année.

### - Responsabilités pédagogiques en lien avec la recherche

— coresponsabilité du cours sur les SMA en master 2 recherche lors des années universitaires 2005/2006 et 2006/2007 à l'université de Reims (2 × 14h).
— À montpellier, nous avons dispensé des cours sur les SMA en master 2 lors des années universitaires 2008/2009 et 2009/2010 (2 × 9h).

### - Coencadrements de stages de M2 recherche

— Jason Mahdjoub (CReSTIC), 2006 (50 % avec Z. Guessoum). Thème : Une approche multi-agent pour la détection de contours. Stage suivi d'un doctorat à l'université de Reims.
— Samir Hamichi (CReSTIC), 2006 (50 % avec Z. Guessoum). Thème : Développement d'une ontologie pour la gestion des chaînes logistiques. Stage suivi d'un doctorat au LIP6, Paris.

### - Divers

— de 2005 à 2006, membre du conseil du département informatique de l'IUT de Reims en tant qu'élu au sein du corps des enseignants chercheurs.

# Résumé

Les travaux de recherche synthétisés dans ce mémoire s'inscrivent principalement dans le domaine de la modélisation et de la simulation de systèmes multi-agents (SMA). La simulation multi-agents met en œuvre des modèles où les individus, leur environnement et leurs interactions sont directement représentés. Dans ces modèles, chaque individu – agent autonome– possède son propre comportement et produit ses actions en fonction d'une perception locale de son environnement. Ainsi, la simulation multi-agents est utilisée pour étudier des systèmes naturels comme les colonies de fourmis, les dynamiques de foules ou le trafic urbain, mais aussi pour concevoir des systèmes artificiels, par exemple dans le cadre de la robotique collective ou le développement de logiciels basés sur de l'intelligence artificielle distribuée. Dans ce cadre, nos recherches ont porté sur des problématiques liées à la modélisation de simulations multi-agents, avec la proposition de modèles formels et conceptuels (e.g. le modèle IRM4S) et d'outils logiciels génériques (plates-formes MaDKit et TurtleKit), et sur leur utilisation dans divers domaines tels que le jeu vidéo, le traitement numérique de l'image ou la robotique collective. Contrairement aux approches centrées sur la conception des comportements individuels, dans ces travaux l'environnement des agents est considéré comme une abstraction de premier ordre. Dans ce mémoire, nous dressons tout d'abord un bilan de nos recherches en argumentant l'intérêt d'une telle démarche pour les modèles multi-agents. Nous montrons ensuite comment celle-ci nous a récemment permis de proposer une approche originale dans le cadre de l'utilisation du calcul haute performance sur carte graphique (GPGPU) pour la simulation de SMA, avant de présenter les perspectives de recherche associées à notre positionnement.

# Abstract

This habilitation thesis synthesizes research works which are mainly related to the field of Multi-Agent Based Simulation (MABS). MABS is a general framework for modeling and experimenting with systems in which the dynamics emerges from local interactions among individuals (autonomous agents). Examples of use range from the study of natural systems (e.g. ant colonies, crowds or traffic jams) to the engineering of artificial ones (e.g., collective robotics, distributed artificial intelligence-based softwares). To this end, MABS modeling represents the behavior of individuals, their environment and interactions, so that global dynamics can be computed and studied from the bottom up. In this context, we have been investigating research on the theory and practice of MABS from two different perspectives : (1) the design of generic abstractions dedicated to the modeling of multi-agent dynamics (e.g., the IRM4S model) and (2) the engineering of MABS (MaDKit and TurtleKit platforms). Besides, we have been experimenting with MABS in different application domains such as image processing, video games, and collective robotics. Contrary to approaches that put the emphasis on the agent behaviors, all these works have been done by considering the environment of the agents as a first order abstraction. In this thesis, we first reflect upon the research we have conducted according to this perspective. Next, we show how we actually use this perspective to propose an original approach for using General-Purpose processing on Graphics Processing Units (GPGPU) within MABS, and then present the research perspectives related to our positioning.