

## AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur : ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite de ce travail expose à des poursuites pénales.

Contact : [portail-publi@ut-capitole.fr](mailto:portail-publi@ut-capitole.fr)

## LIENS

Code la Propriété Intellectuelle – Articles L. 122-4 et L. 335-1 à L. 335-10

Loi n° 92-597 du 1<sup>er</sup> juillet 1992, publiée au *Journal Officiel* du 2 juillet 1992

<http://www.cfcopies.com/V2/leg/leg-droi.php>

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>



Université Toulouse 1 Capitole

École Doctorale Mathématiques, Informatique et  
Télécommunications de Toulouse

# VERS UNE MEILLEURE INTÉGRATION DES DIMENSIONS SPATIALES, COMPORTEMENTALES ET PARTICIPATIVES EN SIMULATION À BASE D'AGENTS

Patrick Taillandier

Manuscrit en vue de l'obtention de  
l'Habilitation à Diriger des Recherches

présenté et soutenu publiquement le 20 juin 2019 devant :

Frédéric AMBLARD	Professeur, UMR IRIT, Université de Toulouse 1
Olivier BARRETEAU	Directeur de Recherche, UMR G-EAU, IRSTEA
Olivier BOISSIER	Professeur, UMR LHC, Mines Saint-Etienne
Alexis DROGOUL	Directeur de Recherche, UMI UMMISCO, IRD
Thomas HOUET	Directeur de Recherche, UMR LETG, CNRS
Thomas NESME	Professeur, UMR IPSA, Bordeaux Sciences Agro
Pascal PEREZ	Professeur, SMART, University of Wollongong
Cécile TANNIER	Directrice de Recherche, UMR THEMA, CNRS

sur la base des rapports rédigés par M. Olivier Barreteau, M. Olivier Boissier et Mme Cécile Tannier.





# Résumé

Mes travaux de recherche s'intéressent à la modélisation à base d'agents des systèmes complexes. Cette approche, qui connaît aujourd'hui un succès de plus en plus important, consiste à modéliser le système étudié sous la forme d'une collection d'entités informatiques autonomes interagissant, appelées agents. Mes travaux, qui ont plus particulièrement porté sur le développement de méthodes et outils visant à faciliter l'accès à cette approche de modélisation pour le plus grand nombre, se structurent autour de 3 principaux axes de recherche : l'intégration de la dimension spatiale dans les modèles, la modélisation de comportements d'acteurs complexes et les outils pour la modélisation et simulation participative.

L'intégration de la composante spatiale dans les modèles constitue l'une de mes plus importantes contributions lors de ces dernières années. L'objectif de ces travaux était de permettre à des utilisateurs non informaticiens de pouvoir intégrer de façon naturelle dans leurs modèles des données spatiales et de leur associer des comportements. Cette recherche à la fois théorique et opérationnelle a nécessité la définition d'un méta-modèle permettant de gérer différents types de topologies spatiales (grille, graphe, espace continu) et de proposer un langage permettant de manipuler celui-ci. Ces recherches sur la dimension spatiale des modèles m'ont également amené à proposer des méthodes et outils permettant de définir simplement des actions de déplacements complexes pour les agents, en particulier dans un contexte de simulation de trafic routier. J'ai enfin travaillé sur la problématique de la spatialisation de populations synthétiques pour initialiser des simulations.

Un second axe de recherche a concerné la modélisation de la cognition des acteurs. Mes travaux dans le domaine m'ont amené à proposer deux architectures d'agents. La première, qui a été développée dans le cadre du projet MAELIA, propose d'utiliser la théorie des fonctions de croyance pour modéliser la prise de décision des acteurs. La seconde, qui a été au cœur du projet ANR ACTEUR, dont je suis le porteur, est basée sur le paradigme BDI (Belief-Desire-Intention) et permet de prendre en compte des aspects cognitifs, émotionnels, sociaux et normatifs dans les comportements des agents. Ce second travail a été en grande partie réalisé dans le cadre de la thèse de Mathieu Bourgeois que j'ai co-encadré entre 2015 et 2018.

Un dernier axe de mes recherches a concerné la modélisation et la simulation participative. S'il existe de longue date des outils adaptés à ce type de démarche pour des modèles simples (KISS), arriver à fournir des méthodes et outils permettant de faire de même pour des modèles plus complexes (KIDS) est un vrai challenge, surtout si l'on souhaite rendre ces outils facilement accessibles. J'ai dans ce cadre travaillé sur un langage de modélisation graphique utilisable en modélisation participative ainsi que sur des outils dédiés à la conception de jeux sérieux. Enfin, en partant de méthodes développées durant ma thèse, j'ai travaillé sur l'apprentissage de comportements d'agents par la simulation participative.

Après la présentation de ces axes de recherche, ce mémoire se conclut en donnant ma vision des avancées attendues pour la modélisation à l'horizon 2030 et en présentant mes perspectives de recherche. Ces dernières, qui se situent en grande partie dans la continuité de mes travaux actuels, s'intègrent dans deux de mes trois axes de recherche. Ainsi, un premier axe de mes perspectives, en lien avec la dimension comportementale, vise à une meilleure modélisation de la décision individuelle et à la prise en compte de processus d'argumentation entre agents. Le second axe de mes perspectives, prolongeant les travaux sur la dimension participative, concerne le développement de simulations interactives intégrant des interfaces tangibles et une physicalisation des résultats de simulation. Enfin, une dernière perspective s'intéresse à la place de la simulation à base d'agents dans les démarches de prospective.

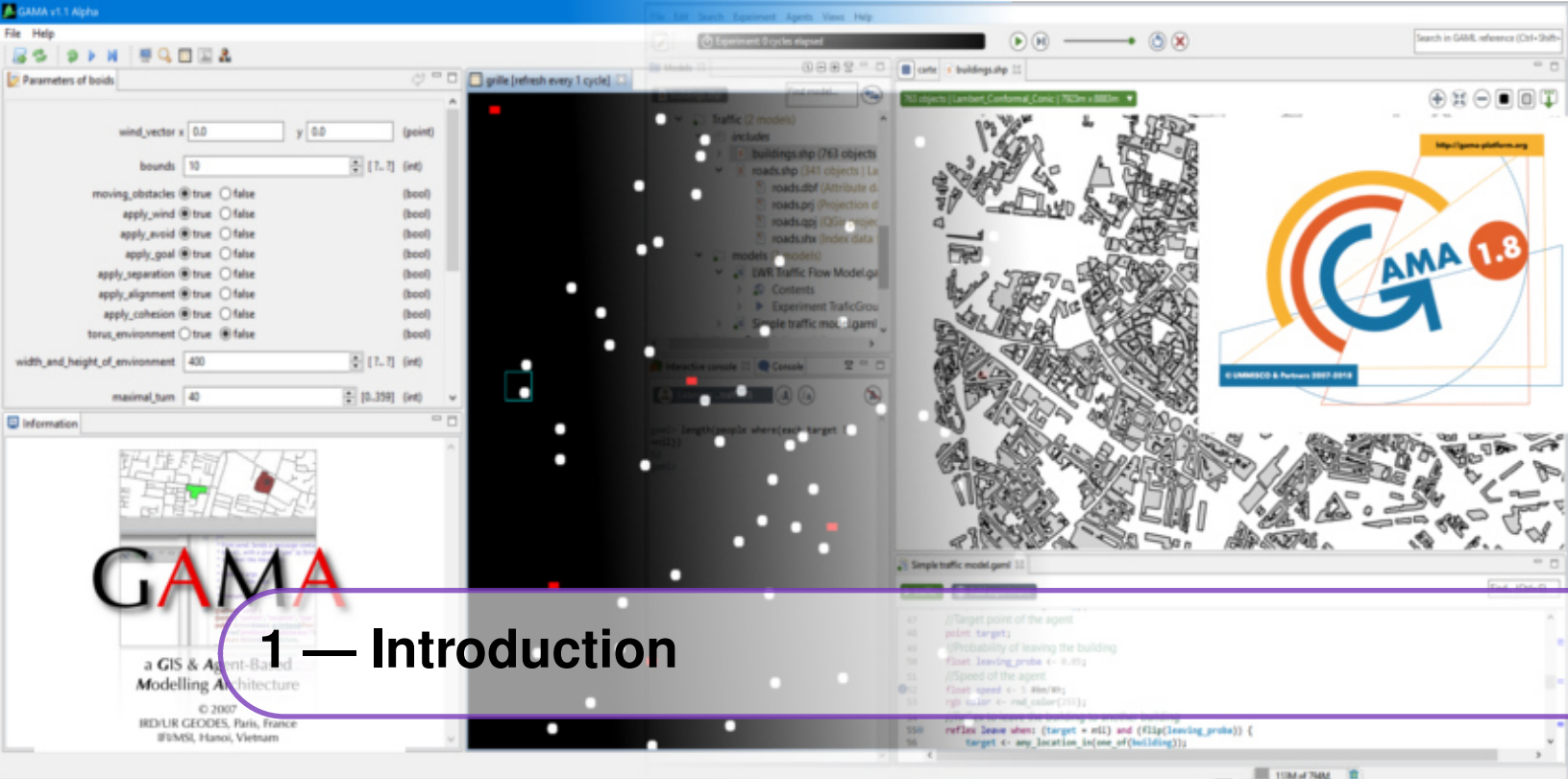
# Table des matières

<b>1</b>	<b>Introduction</b> .....	<b>1</b>
<b>1.1</b>	<b>Contexte</b> .....	<b>1</b>
<b>1.2</b>	<b>La modélisation à base d'agents de systèmes complexes</b> .....	<b>3</b>
1.2.1	Concept de base .....	3
1.2.2	Validation, utilisation et diffusion des modèles .....	5
1.2.3	Construire et exploiter un modèle à base d'agents .....	6
1.2.4	Les plates-formes de simulation en 2008 .....	8
<b>1.3</b>	<b>La plate-forme GAMA</b> .....	<b>10</b>
1.3.1	Généralité sur la plate-forme GAMA .....	10
1.3.2	Le langage de modélisation de GAMA .....	10
1.3.3	Utilisation de la plate-forme GAMA .....	11
<b>1.4</b>	<b>Contributions et organisation du mémoire</b> .....	<b>14</b>
<b>2</b>	<b>L'espace dans la simulation à base d'agents</b> .....	<b>17</b>
<b>2.1</b>	<b>Prise en compte de l'espace dans la plate-forme GAMA</b> .....	<b>17</b>
2.1.1	Contexte .....	17
2.1.2	Prise en compte de différentes topologies spatiales .....	19
2.1.3	Intégration des données géographiques dans les simulations .....	21
2.1.4	Déplacement des agents dans l'espace : le <i>moving skill</i> .....	22
<b>2.2</b>	<b>Modélisation du trafic routier avec GAMA</b> .....	<b>23</b>
2.2.1	Contexte .....	23
2.2.2	Le <i>advanced driving skill</i> de GAMA .....	24
2.2.3	Validation .....	26
2.2.4	Application .....	29
<b>2.3</b>	<b>Spatialisation de populations synthétiques</b> .....	<b>30</b>
2.3.1	Contexte .....	30
2.3.2	Le module <i>SpII</i> de la bibliothèque Gen* .....	31
2.3.3	Bilan sur <i>SpII</i> .....	36
<b>2.4</b>	<b>Conclusion</b> .....	<b>38</b>
<b>3</b>	<b>La représentation de la cognition des acteurs</b> .....	<b>39</b>
<b>3.1</b>	<b>Contexte général</b> .....	<b>39</b>

<b>3.2</b>	<b>Utilisation de la théorie des fonctions de croyance</b>	<b>40</b>
3.2.1	Contexte .....	40
3.2.2	Formalisme proposée .....	41
3.2.3	Application pour le modèle MAELIA .....	44
3.2.4	Discussion .....	49
<b>3.3</b>	<b>Architecture BEN</b>	<b>50</b>
3.3.1	Contexte .....	50
3.3.2	Formalisation .....	51
3.3.3	Architecture .....	52
3.3.4	Discussion .....	57
<b>3.4</b>	<b>Conclusion</b>	<b>59</b>
<b>4</b>	<b>Modélisation et simulation participative .....</b>	<b>61</b>
<b>4.1</b>	<b>Introduction</b>	<b>61</b>
<b>4.2</b>	<b>Modélisation participative</b>	<b>62</b>
4.2.1	Contexte .....	62
4.2.2	Modélisation graphique avec GAMA .....	63
4.2.3	Discussion .....	68
<b>4.3</b>	<b>Simulation participative</b>	<b>68</b>
4.3.1	Contexte .....	68
4.3.2	Fonctionnalités pour la conception de jeux sérieux avec GAMA .....	70
4.3.3	Apprentissage de comportements par des jeux sérieux .....	74
<b>4.4</b>	<b>Conclusion</b>	<b>77</b>
<b>5</b>	<b>Conclusion et perspectives .....</b>	<b>79</b>
<b>5.1</b>	<b>Résumé de mes travaux et projets de recherche</b>	<b>79</b>
5.1.1	Bilan de mes travaux de recherche .....	79
5.1.2	La modélisation à base d'agents à l'horizon 2030 .....	80
5.1.3	Projet de recherche .....	83
<b>A</b>	<b>Travaux et publications .....</b>	<b>103</b>
<b>A.1</b>	<b>Etat civil</b>	<b>103</b>
<b>A.2</b>	<b>Formation</b>	<b>103</b>
<b>A.3</b>	<b>Expériences professionnelles</b>	<b>104</b>
<b>A.4</b>	<b>Animation scientifique</b>	<b>104</b>
A.4.1	Charges collectives .....	104
A.4.2	Jury de concours de recrutement .....	104
A.4.3	Jury de thèse .....	104
A.4.4	Comités de programme de conférences, relecture d'articles .....	104
A.4.5	Expertise .....	105
A.4.6	Participation à des actions de prospective .....	105
A.4.7	Groupes de travail .....	105

---

<b>A.5</b>	<b>Projets de recherche</b>	<b>105</b>
A.5.1	Coordinateur du projet .....	105
A.5.2	Responsable scientifique pour mon unité .....	106
A.5.3	Participant .....	106
<b>A.6</b>	<b>Activités d'encadrement</b>	<b>107</b>
A.6.1	Encadrement de thèses .....	107
A.6.2	Encadrement de Master .....	108
A.6.3	Encadrement de CDD .....	108
<b>A.7</b>	<b>Activités d'enseignement</b>	<b>108</b>
A.7.1	Comité d'organisation de formations et d'écoles thématiques .....	108
A.7.2	Cours .....	109
<b>A.8</b>	<b>Publications</b>	<b>110</b>
A.8.1	Revue à comité de lecture .....	110
A.8.2	Congrès avec comité de lecture .....	111
A.8.3	Contributions à des ouvrages collectifs .....	115
A.8.4	Rapports diplômants .....	115
A.8.5	Développements informatiques .....	115



# 1 — Introduction

## 1.1 Contexte

Si aujourd'hui l'interdisciplinarité est très recherchée, en particulier par les instituts de recherche publics, rares sont encore les chercheurs s'inscrivant réellement dans celle-ci. Mon parcours a justement pour originalité de m'avoir amené à travailler aux frontières de différentes disciplines, en particulier de l'informatique et de la géographie.

J'ai une formation initiale en informatique avec une spécialisation en Intelligence Artificielle (diplôme d'ingénieur de l'école CPE Lyon et Master de l'Université Lyon 1 en 2005) qui m'a permis d'obtenir un contrat doctoral entre 2005 et 2008 à l'IGN (laboratoire COGIT – Saint Mandé) touchant à des problématiques de gestion de données géographiques et de cartographie. Ce travail de thèse, dirigé par Alexis Drogoul (UMI UMMISCO, IRD) et encadré par Cécile Duchêne (COGIT, IGN), a porté sur la révision automatique des connaissances contenues dans les systèmes de généralisation cartographique. La généralisation de données géographiques, qui est l'une des étapes les plus difficiles à réaliser et à automatiser pour la conception de cartes, consiste à simplifier les données de manière à rendre la carte lisible tout en conservant les informations d'origine. Plusieurs approches d'automatisation de ce processus ont été proposées durant les vingt dernières années. L'une des approches qui a connu le plus de succès consiste à utiliser un système multi-agent : chaque objet géographique (bâtiment, tronçon de route, quartier, etc.) est représenté par une entité informatique (un agent) qui a la charge de s'appliquer des actions de généralisation (par exemple, grossir, diminuer ses détails...) afin de respecter au mieux ses contraintes cartographiques (ne pas être trop petit pour être lisible, ne pas avoir trop de détails, etc.). Pour choisir au mieux les actions à s'appliquer, les agents utilisent des connaissances intégrées au système (par exemple, des règles du type « si je suis trop petit et que j'ai beaucoup de détails, j'essaie d'abord de grossir »). Or, définir de telles connaissances, même pour des experts en généralisation, se révèle souvent délicat et fastidieux. De plus, il faut réadapter ces connaissances à chaque évolution du système (par exemple, lors de l'ajout d'une nouvelle action de généralisation ou lors de l'ajout de nouveaux thèmes sur les cartes). Mes travaux de thèse m'ont conduit à proposer des approches de diagnostic et de révision des connaissances des agents basées sur l'utilisation des traces de fonctionnement du système pour en tirer des enseignements. Ce travail,

qui a été valorisé par de nombreuses publications (Ruas *et al.*, 2006; Taillandier, 2007c,a,b; Steiniger *et al.*, 2007; Taillandier, 2008; Taillandier *et al.*, 2008; Taillandier et Gaffuri, 2009; Taillandier *et al.*, 2009a; Steiniger *et al.*, 2010; Taillandier *et al.*, 2010c, 2011a, 2012b), m'a permis de mettre en œuvre et d'acquérir un grand panel de techniques issues de l'Intelligence Artificielle (système multi-agent, apprentissage artificiel, optimisation), de la recherche opérationnelle (analyse multicritère) et de la géomatique (analyse spatiale, SIG).

A la suite de ce travail de thèse, j'ai effectué deux années de post-doctorat à Hanoï (Vietnam) dans le laboratoire MSI de l'IFI (Institut de la Francophonie pour l'Informatique) où j'ai travaillé avec Alexis Drogoul sur le développement de la plate-forme GAMA, dédiée à la conception et à la simulation de modèles à base d'agents. Ces deux années ont été particulièrement structurantes pour mes recherches. Elles m'ont en effet ouvert sur le domaine de la simulation à base d'agents et sur les enjeux liés à la démocratisation de cette approche, problématiques sur lesquelles je travaille encore aujourd'hui. Cette expérience fut également l'occasion de commencer à m'impliquer dans le développement et la diffusion de la plate-forme GAMA. Cette plate-forme de modélisation open-source a pris, depuis, une place très importante dans mes travaux de recherche : toutes les approches et méthodes que j'ai pu développer par la suite ont toutes été intégrées dans GAMA. J'ai également participé depuis 2008 à plus d'une vingtaine de formations en lien avec cette plate-forme

Après ces deux années à Hanoi, j'ai effectué un post-doctorat de 10 mois dans l'équipe SMAC de l'UMR IRIT à l'Université Toulouse 1 Capitole, où j'ai travaillé sur le projet MAELIA. Ce projet, qui a été depuis repris par l'INRA, avait pour objectif initial de développer un modèle à base d'agents destiné à évaluer l'impact de différentes politiques de gestion de l'eau (allocation de volumes prélevables pour l'agriculture) dans le bassin Adour-Garonne. Mon travail dans ce projet a porté sur la modélisation du comportement des agriculteurs : leur choix de rotation de cultures, mais également leur conduite d'exploitation au jour le jour (semis, récolte, irrigation, etc.). Ce projet, dans lequel a été développé le modèle le plus ambitieux jamais construit avec GAMA, a été pour moi extrêmement enrichissant : il m'a non seulement permis de découvrir les enjeux autour de l'agriculture, mais également d'être, pour la première fois, confronté aux difficultés inhérentes aux travaux pluridisciplinaires. Le projet regroupait en effet des chercheurs issus de différentes disciplines (informatique, agronomie, hydrologie, sociologie, géographie) qu'il a fallu réussir à faire travailler ensemble. Ce contexte pluridisciplinaire fertile m'a permis de perfectionner et de croiser mes compétences en Intelligence Artificielle et en Géomatique, ce qui m'a aidé à accéder avec succès en 2011 à un poste de Maître de Conférences en Géographie à l'Université de Rouen dans l'UMR IDEES avec une chaire d'excellence du CNRS.

J'ai passé 5 ans dans l'UMR IDEES dans un cadre nouveau pour moi (laboratoire et enseignement en Sciences Humaines et Sociales), mais très stimulant. Ces années, en plus de m'apporter de nombreux nouveaux questionnements par rapport à la dimension spatiale et des enjeux sociétaux, ont été particulièrement riches en projets financés (porteurs de plusieurs projets, dont un projet ANR, responsable scientifique pour l'UMR IDEES de nombreux autres). Elles m'ont également permis de mieux comprendre les enjeux autour de la facilitation de l'utilisation de la modélisation à base d'agents pour des chercheurs en SHS. En effet, cette expérience m'a montré que cette approche avait un intérêt majeur pour les chercheurs en SHS, mais que la difficulté à l'utiliser lorsqu'on ne pouvait avoir recours à des informaticiens était un frein à son adoption.

Enfin, après ces 5 années passées à Rouen, j'ai pu rejoindre fin 2016 l'unité MIAT (Mathématiques et



Informatique Appliquées de Toulouse) de l'INRA en tant que Chargé de Recherches, m'ouvrant ainsi à de nouvelles questions autour de l'agriculture et de l'alimentation.

Ces treize années de recherche m'ont permis de publier 16 articles dans des revues internationales, 4 articles dans des revues nationales, 4 chapitres dans des ouvrages collectifs et 36 articles dans des conférences internationales à comité de lecture et édition d'actes. Ce document ne présente qu'une partie de ces travaux et se focalise sur la thématique de l'aide au développement de modèles à base d'agents pour l'étude des systèmes complexes, thématique qui est introduite dans la partie suivante.

## 1.2 La modélisation à base d'agents de systèmes complexes

### 1.2.1 Concept de base

Ces dernières années ont été marquées par le succès des approches scientifiques liées aux systèmes complexes. Ces approches fournissent des concepts (émergence, non-linéarité...) et des outils (modélisation, simulation...) qui offrent de nombreuses perspectives tant pour la recherche fondamentale que pour la prospective territoriale. Elles ont ainsi été au coeur du développement de nombreux modèles qui permettent d'étudier l'évolution de systèmes à différentes échelles et sous différents angles.

Un modèle est une représentation simplifiée de la réalité visant à répondre à une ou plusieurs questions par rapport au système de référence étudié. En sciences sociales, ces modèles ont principalement trois types d'objectif : l'analyse de données, la prédiction et la recherche d'explication (Varenne, 2014). Il existe de nombreux types de modèles (systèmes dynamiques, automates cellulaires, modèles à base d'agents...) que l'on peut classifier en deux grandes catégories : les modèles statiques et les modèles dynamiques. Un modèle est statique quand son objectif est de fournir une représentation de la structure du système de référence photographié à un instant donné, sans considérer son évolution dans le temps (Treuil *et al.*, 2008). A l'inverse, un modèle est dynamique quand il inclut dans sa représentation des hypothèses ou des règles concernant l'évolution dans le temps du système de référence (Treuil *et al.*, 2008). Ce deuxième type de modèles permet d'entrer dans une démarche computationnelle qui consiste à utiliser l'informatique comme une machine à modéliser et à simuler au moyen de calculs (Varenne, 2014). La simulation informatique consiste à perturber, à l'aide d'un dispositif informatique, un modèle dynamique, par exemple en faisant évoluer la variable de temps, en vue de répondre à un objectif précis (Treuil *et al.*, 2008). Varenne (2014) cite trois principaux apports de la simulation informatique : elle permet de formaliser le caractère dynamique des phénomènes sociaux, de modéliser les effets des institutions et structurations sociales sur les entités individuelles et de représenter à la fois le temps et l'espace.

Les évolutions récentes en informatique ont permis de développer des approches de simulation intégrant des données riches et une description fine du comportement des entités composant le système de référence. Parmi ces approches, la simulation à base d'agents connaît depuis quelques années un grand essor, en particulier dans les sciences sociales et environnementales. Elle consiste à modéliser le système étudié sous la forme d'une collection d'entités autonomes interagissant appelées agents. Un modèle à base d'agents permet de générer les comportements macroscopiques du système à partir des interactions individuelles. Il peut également être utilisé comme un laboratoire virtuel pour mener des expériences in-silico qu'il ne serait pas possible de mener sur le système réel. Enfin, il peut servir de média de discussion dans le cadre de processus de modélisation et simulation participatifs (Le Page *et al.*, 2010). L'une des forces de cette approche est qu'elle est intégrative et versatile : elle ne donne aucune contrainte sur ce que représente un agent et sur la façon de décrire leur comportement. Ainsi,

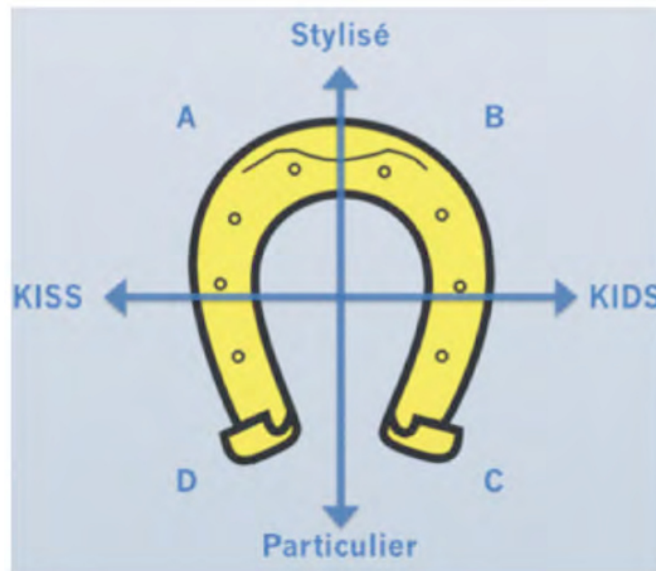


FIGURE 1.1 – Diagramme de catégorisation des modèles proposé par [Banos et Sanders \(2013\)](#)

un agent peut représenter un être humain, un animal, mais également un groupe de personnes, un bâtiment, une route ou même une ville. De même le comportement d'un agent peut être décrit avec des bases de règles (dans tel contexte, utiliser telle action et dans tel autre contexte, utiliser telle autre action), avec des équations, par de la décision multicritère ou par des approches venant de l'intelligence artificielle. La modélisation à base d'agents permet donc de réutiliser de nombreux modèles déjà existants et permet également de s'adapter à de nombreux cas d'application : le choix d'une méthode de description du comportement des agents dépendra de l'application, des données disponibles et des connaissances des modélisateurs.

Il existe aujourd'hui un grand nombre de modèles à base d'agents ayant des caractéristiques et des objectifs différents. Il est possible de catégoriser ces modèles en utilisant les 2 axes proposés par [Banos et Sanders \(2013\)](#) (Figure 1.1) :

- *simple - descriptif* : cet axe correspond au niveau de simplicité utilisé pour représenter le système étudié. On peut parler ici de deux grandes approches de modélisation originellement antagonistes, mais qui peuvent aujourd'hui être utilisées de façon complémentaires : l'approche KISS (Keep It Simple, Stupid) ([Axelrod, 1997](#)), qui cherche à utiliser les mécanismes les plus simples possibles pour représenter le système étudié, et l'approche KIDS (Keep It Descriptive, Stupid) ([Edmonds et Moss, 2004](#)) qui cherche elle à le représenter de façon plus descriptive. A noter qu'il s'agit d'un axe continu, on trouve par exemple au milieu de cet axe les modèles correspondants à l'approche KILT (Keep It a Learning Tool), proposée par [Le Page et Perrotton \(2017\)](#), qui décrit les modèles appropriés à la simulation participative : ces derniers ne doivent pas être trop simples afin que les participants puissent s'approprier le cas d'étude, mais en même temps ne pas être trop détaillés pour ne pas que les participants se focalisent sur des détails.
- *stylisé - particulier* : cet axe correspond au niveau d'abstraction du modèle par rapport au système étudié. Un modèle particulier sera lié à un cas d'étude réel et s'appuiera sur des données et connaissances précises sur celui-ci. Au contraire, un modèle stylisé cherchera à reproduire

des mécanismes plus généraux et ne sera pas lié à un cas d'étude précis.

Aujourd'hui où de plus en plus de données sont accessibles et où les ordinateurs permettent d'exécuter des simulations de plus en plus complexes, une tendance est le développement de modèles se trouvant dans la partie bas droite du diagramme de [Banos et Sanders \(2013\)](#), c'est-à-dire de modèles descriptifs (KIDS) basés sur de nombreuses données (particulier). Néanmoins, si développer des modèles simples avec peu de données est accessible à tous et ne nécessite pas de compétences poussées en informatique, il n'en est pas de même pour les modèles plus descriptifs et utilisant de façon intensive des données. Ainsi, la définition d'approches, méthodes et outils permettant le développement de tels modèles par le plus grand nombre est un vrai enjeu.

### 1.2.2 Validation, utilisation et diffusion des modèles

Une question couramment posée aux modélisateurs, en particulier lors de la présentation de modèles à base d'agents descriptifs, est celle de la validation du modèle. Effectivement, une critique souvent formulée à l'égard de ces modèles est la complexité et les quantités de données nécessaires à leur validation. Pour répondre à cette question, [Amblard \*et al.\* \(2006\)](#) invitent les modélisateurs à retourner la question : "Qu'entendez-vous par validation ?". Il n'existe en effet pas de "validation" objective et unique permettant d'affirmer, de manière catégorique, qu'un modèle est valide ou non. La notion de validation dépend fortement de l'objectif du modèle (prédiction, compréhension, ludique, etc.). Ainsi, en fonction de cet objectif et des données disponibles (quantitatives ou qualitatives, à différents niveaux), un ensemble de critères peut être défini pour évaluer le modèle. Certains de ces critères relèvent de la validation interne (conformité entre les spécifications et le programme implémenté), d'autres de la validation externe (adéquation entre le modèle et le phénomène réel). Plus de détails sur cette question complexe de la validation pourront être trouvés dans ([Amblard \*et al.\*, 2006](#)).

Quelles que soient les démarches de validation effectuées, considérant que par nature un modèle est toujours imparfait, il est important de garder un certain recul par rapport aux résultats des simulations. Cela est d'autant plus important lorsque l'on s'intéresse à la modélisation de systèmes socio-environnementaux. Par exemple, chercher à prédire comment vont évoluer les pratiques agricoles dans les cinquante prochaines années est extrêmement difficile, tant le nombre d'éléments susceptibles d'entraîner des modifications profondes (nouvelles technologies, évolutions sociétales, etc.) est grande. Ainsi, si je pense que la modélisation est un outil particulièrement pertinent dans les processus d'aide à la décision et de prospective, il est important de ne pas sur-interpréter les résultats des simulations. Les modèles doivent avant tout servir comme outil d'aide à la réflexion pour mettre en avant certains phénomènes ou processus qu'il aurait été difficile d'identifier sans eux et non comme des outils "boîtes noires" produisant directement des décisions qu'il faudrait suivre aveuglément.

Dans ce cadre de la réflexion autour de l'usage des modèles et de leur validation, un dernier point extrêmement important est la reproductibilité des expériences. Cette reproductibilité implique que le modèle soit décrit avec suffisamment de détails pour permettre sa réplique. Dans ce cadre, j'ai participé à un article collectif ([Müller \*et al.\*, 2014](#)) sur la communication des modèles à base d'agents. Il invite les modélisateurs à fournir, lors de la présentation des résultats des simulations, une description structurée en langage naturel du modèle (type ODD ([Grimm \*et al.\*, 2010](#))) ainsi que son code source. A titre personnel, j'ai toujours veillé, depuis la publication de cet article, à fournir, à chacune de mes publications sur des modèles, les code sources de ces derniers via des sites d'échange et de partage

(par exemple : GitHub ou openABM).

### 1.2.3 Construire et exploiter un modèle à base d'agents

Si la phase d'implémentation (codage informatique) d'un modèle à base d'agents est très demandeuse en temps, elle n'est pas la seule. En effet, la construction et l'exploitation d'un modèle passe par plusieurs étapes, qui peuvent elles-aussi être chronophages et que l'on peut décrire ainsi (Figure 1.2) :

- *Définition de la question de modélisation* : un modèle doit être construit par rapport à une (ou plusieurs) question à laquelle on souhaite que le modèle puisse nous aider à répondre. Définir cette ou ces questions est ainsi une étape clef du processus de modélisation qui permettra de guider tout le reste du processus.
- *Identifier les éléments à modéliser* : cette seconde étape consiste à identifier par analyse du système réel (directement ou indirectement au travers d'experts ou de la littérature) les éléments (entités, dynamiques...) que l'on va devoir intégrer dans le modèle si l'on souhaite pouvoir répondre à la question de modélisation.
- *Collecter des données* : une étape importante, en particulier lorsqu'on développe un modèle *particulier*, et qui va souvent demander des ressources importantes est la collecte des données. Ces données peuvent être de différentes natures aussi bien quantitatives que qualitatives et provenir de différentes sources (base de données géographiques ou statistiques, entretiens, dire d'expert...). Un travail de nettoyage, de croisement, voire de génération de données synthétiques sera souvent nécessaire à cette étape.
- *Définir les agents* : l'objectif de cette étape est d'aller plus loin dans la formalisation du modèle en concevant un modèle conceptuel complet. Ainsi, il s'agira ici de définir quels seront les types d'agents utilisés dans le modèle, quelles seront leurs caractéristiques et leurs dynamiques. Il est possible d'utiliser à cette étape, en particulier dans le cadre d'un projet de modélisation collectif, des langages de modélisation graphique comme l'UML pour faciliter la communication autour du modèle.
- *Implémenter le modèle* : cette étape consiste à transformer le modèle conceptuel conçu dans l'étape précédente en un modèle implémenté. S'il est courant qu'il y ait un écart entre le modèle conceptuel et le modèle implémenté (ajout ou suppression de certains types d'agents, modification de la nature de certaines variables ou ajout d'autres variables pour diminuer les temps de calcul), il est important de toujours documenter les choix qui ont été fait afin que même ceux qui n'ont pas participé à cette phase d'implémentation puissent toujours comprendre ce qu'il y a dans le modèle.
- *Calibrer le modèle* : cette étape optionnelle consiste à définir les valeurs de paramètres, en particulier des paramètres sur lesquels peu de données sont disponibles, permettant au modèle de reproduire au mieux le système réel. Elle nécessite de disposer de données ou de connaissances sur le système étudié. Si le modélisateur dispose d'une métrique pour caractériser numériquement la validité d'une simulation (par exemple, écart entre les données observées sur le système réel et les données simulées), il est possible de recourir à des méthodes de calibration automatique qui vont explorer les combinaisons possibles de valeurs de paramètres afin de trouver celles optimisant au mieux la métrique. On cherchera aussi dans cette étape à vérifier l'impact des différents paramètres du modèle sur les sorties de simulation.
- *Explorer le modèle* : cette étape consiste à utiliser le modèle pour répondre à la question de modélisation.

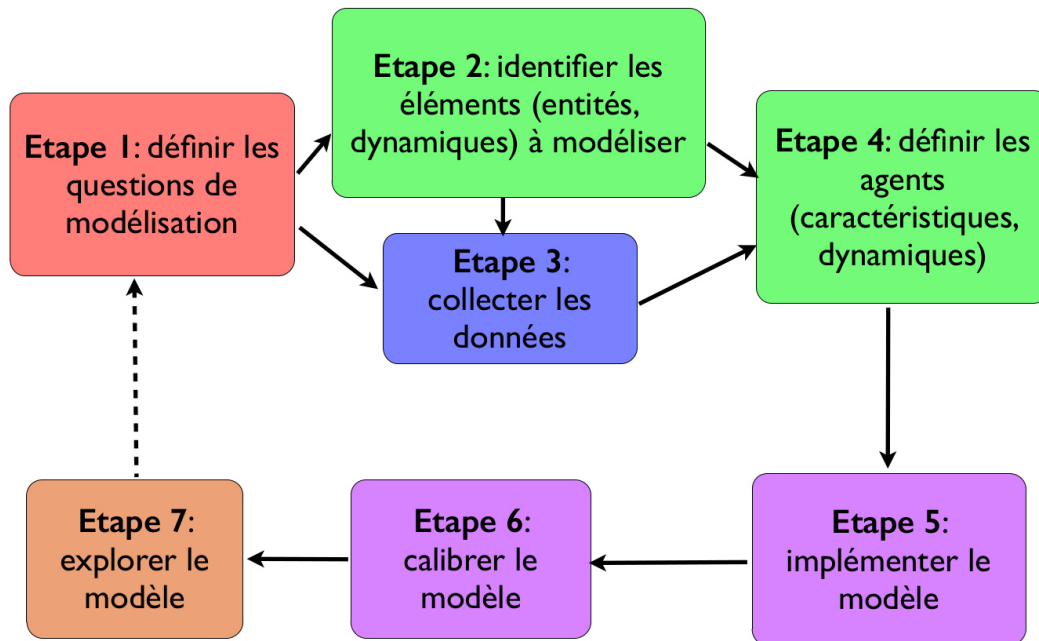


FIGURE 1.2 – cycle de modélisation

Lors d'un projet de modélisation, il est courant de réaliser ce cycle plusieurs fois. En effet, au fur et à mesure du processus de modélisation, la question de modélisation peut s'affiner, des nouvelles données peuvent être requises, tout comme le diagramme conceptuel peut évoluer. Le projet MAELIA est un bon exemple : le modèle tel qu'il existe aujourd'hui et les questions de modélisation auxquelles il cherche à répondre sont très différents de ce qu'ils étaient initialement. Il est d'ailleurs à noter que les étapes de calibration et d'exploration du modèle peuvent aussi avoir un impact très fort sur le processus de modélisation. Elles peuvent par exemple mettre en avant que certains processus intégrés dans le modèle sont inutiles pour reproduire correctement le système étudié ou au contraire qu'il manque des processus clés. Certaines approches de modélisation telles que celle présentée par [Cottineau et al. \(2015\)](#) proposent d'utiliser de façon intensive ces phases d'exploration et de calibration dans la conception même du modèle.

Comme cela a été mentionné dans une partie précédente, le développement d'un modèle à base d'agents, en particulier quand il s'agit de définir des modèles descriptifs s'appuyant sur des données, requiert un travail important de programmation informatique. En effet, développer un modèle nécessite de décrire sous la forme d'algorithmes l'ensemble des agents et leurs comportements. Afin de faciliter ce travail, des équipes de recherche ont été amenées à développer des plates-formes de modélisation et simulation. On compte aujourd'hui de nombreuses plates-formes (24 citées dans [Kravari et Bassiliades, 2015](#)) qui ont fortement contribué à l'appropriation et à la diffusion de la modélisation à base d'agents au sein de diverses communautés (géographie, sociologie, économie, écologie, etc.). L'un des critères possibles pour regrouper ces plates-formes est le type de langage utilisé pour implémenter les modèles. 3 catégories (non-exclusives) peuvent être ainsi définies :

- La première catégorie comprend les plates-formes qui permettent de définir les modèles à l'aide d'un langage de programmation générique (Java, C++, Smalltalk, Python...). Ces plates-formes sont en général dédiées aux informaticiens et sont souvent les plus adaptés au développement de modèles complexes comprenant de nombreux agents et processus. Dans cette catégorie, on trouve entre autres les plates-formes SWARM ([Minar et al., 1996](#)), Repast ([North et al., 2013b](#)), JADE ([Bellifemine et al., 2001](#)) et CORMAS ([Le Page et al., 2000](#)).
- La seconde catégorie comprend les plates-formes qui permettent de définir des modèles à l'aide d'un langage de modélisation dédié. Ces plates-formes sont généralement plus simples à utiliser que celles de la première catégorie et sont donc destinées à un plus large panel d'utilisateurs. Elles requièrent néanmoins des connaissances en algorithmique. Les plates-formes NetLogo ([Tisue et Wilensky, 2004](#)) et GAMA ([Taillandier et al., 2018](#)) font partie de cette catégorie.
- La dernière catégorie comprend les plates-formes qui permettent de définir des modèles à l'aide d'un langage de modélisation graphique. Ces plates-formes ne requièrent en général pas ou peu de connaissances en algorithmique mais sont assez limitées en termes de richesse des modèles qu'elles permettent de construire. On trouve dans cette catégorie les plates-formes Startlogo TNG ([Resnick, 1996](#)), AgentSheets ([Repenning, 2011](#)) et MAGéo ([Langlois et al., 2015](#))

La plupart de ces plates-formes de modélisation ont connu des évolutions majeures ces dix dernières années. En effet, beaucoup d'entre elles ne proposaient pas, quand j'ai commencé à travailler sur la plate-forme GAMA en 2008, autant de fonctionnalités. Afin de mettre en évidence ces limites, je propose dans la partie suivante un état de lieux des plates-formes de simulation en 2008.

#### 1.2.4 Les plates-formes de simulation en 2008

En 2008, quatre plates-formes étaient principalement utilisées : JADE, Repast, CORMAS et Netlogo.

JADE est une librairie Java dédiée au développement de systèmes multi-agents. L'une des forces, mais aussi faiblesses de JADE, est qu'elle est générique : son champ d'application dépasse le cadre de la simulation à base d'agents et couvre tous les domaines des systèmes multi-agents. Ainsi, elle ne dispose que de peu d'outils directement dédiés à la simulation et encore moins à la simulation spatialisée. Utiliser cette plate-forme pour le développement de modèles de simulation spatialisés requiert donc un effort important de développement en Java, ce qui nécessite de disposer de compétences élevées en informatique et programmation.

Une autre plate-forme open-source, ou plutôt une famille de plates-formes (il existe plusieurs plates-formes Repast), permettant de définir des modèles à l'aide de langage de programmation générique est Repast. Contrairement à JADE, ces plates-formes sont spécialement dédiées à la simulation à base d'agents et proposaient déjà en 2008 de nombreux outils pour la gestion de l'espace (intégration de données géographiques, gestion des grilles, etc.). Elles offraient de plus différentes façons de définir des modèles (Java, modélisation graphique...) afin de s'adapter aux plus grand nombre d'utilisateurs possibles. Si pour les modèles très simples, le modélisateur avait un vrai choix de langage de modélisation, le développement de modèles plus descriptifs nécessitait d'utiliser des langages génériques (exemple : Java), ce qui étaient hors de portée de beaucoup de modélisateurs. Enfin, contrairement à JADE, elles n'offraient aucun outil spécifique pour définir le comportement des agents.

Une autre plate-forme open-source basée sur un langage de programmation générique est CORMAS. En comparaison à JADE et Repast, cette plate-forme, qui permet de définir des modèles en Smalltalk,



est plus accessibles aux non-informaticiens. Si CORMAS était et reste assez limitée pour la définition de modèles descriptifs à large échelle (gestion très basique de l'espace, pas d'outil pour décrire des comportements complexes, etc.), elle offrait dès 2008 de nombreux outils dédiés à la modélisation et simulation participative, ce qui explique son succès pour les projets dans lesquels des acteurs humains sont impliqués.

Si toutes ces plates-formes offrent des outils puissants, elles restent, même CORMAS, complexes à utiliser par des modélisateurs avec des compétences limitées en programmation. C'est ainsi que la plate-forme Netlogo, en se positionnant sur ce créneau, a su forger son succès. Elle permet en effet aux modélisateurs de développer leurs modèles au travers d'un langage de modélisation dédié, simple d'accès. Cette plate-forme, qui est encore aujourd'hui l'une des plates-formes de modélisation les plus populaires, a grandement contribué à la diffusion de la modélisation à base d'agents dans les sciences sociales. Néanmoins, Netlogo souffrait de nombreuses limitations en 2008, qui perdurent aujourd'hui malgré ses nombreuses extensions : interfaces de modélisation et de simulation limitées, intégration basique des données et en particulier des données géographiques, pas d'outil pour décrire des comportements complexes, etc. En conséquence, la plupart des modèles développés avec Netlogo sont des modèles simples (KISS), et Netlogo a rarement été utilisée pour développer des modèles plus descriptifs.

C'est sur ce constat que la plate-forme open-source GAMA (Gis and Agent-based Modelling Architecture) a commencé à être développée en 2007 dans l'UMI UMMISCO sous la coordination d'Alexis Drogoul. L'objectif derrière son développement était de combiner les avantages de ces plates-formes : permettre de développer des modèles de façon aussi simple et rapide qu'avec Netlogo, tout en offrant des outils de modélisation allant au-delà de ce que proposent Repast ou CORMAS. La première version de GAMA reposait sur le moteur de simulation de Repast et proposait aux utilisateurs de définir leurs modèles en XML. Si le XML avait pour avantage d'être bien structuré et facilement interprété par les ordinateurs, il se montrait assez verbeux et lourd à l'utilisation. De même si GAMA permettait d'intégrer des données géographiques dans les modèles, la manipulation de celles-ci était très limitée : elles servaient avant tout à définir l'environnement des agents. Enfin, GAMA était assez limitée concernant la dimension comportementale des agents : la plate-forme ne proposait en effet que des architectures simples pour décrire le comportement des agents.

Pour conclure sur cet état des lieux des plates-formes de simulation en 2008, s'il existait déjà de nombreuses plates-formes qui sont encore aujourd'hui grandement utilisées, celles-ci étaient très limitées. Pour la dimension spatiale, même les plus avancées (GAMA et Repast) permettaient juste d'intégrer des données géographiques pour définir l'environnement et non de définir de vrais agents spatiaux. Concernant la dimension comportementale, seule JADE, qui reste très peu adaptée au contexte de la simulation, proposait des outils avancés. Enfin, sur la dimension participative, la plus avancée était CORMAS, même si elle disposait à l'époque de bien moins d'outils qu'actuellement.

La partie suivante décrit la plate-forme GAMA telle qu'elle existe aujourd'hui. Elle revient en particulier sur son langage de modélisation qui a beaucoup évolué depuis sa première version.

## 1.3 La plate-forme GAMA

### 1.3.1 Généralité sur la plate-forme GAMA

GAMA, dont la dernière version, GAMA 1.8, est disponible depuis 2018, fournit aujourd'hui aux modélisateurs un langage de modélisation orienté-agent simple d'accès, une gestion très avancée des données et en particulier des données géographiques, des outils de visualisation flexibles et la capacité d'exécuter des simulations comprenant des centaines de milliers d'agents.

Tout comme Netlogo, CORMAS et Repast, GAMA repose sur un moteur de simulation à temps discret : le temps est découpé en intervalles égaux (pas de simulation) et l'état du monde simulé est modifié à chaque pas de simulation (Taillandier, 2015).

Une évolution majeure de GAMA concerne son langage de modélisation qui s'est largement enrichi et qui a grandement gagné en facilité d'utilisation au fur et à mesure des versions. Une autre évolution marquante concerne la gestion de l'espace dans les modèles, point sur lequel je reviendrai au Chapitre 2, et qui a contribué à faire de GAMA une plate-forme de choix pour les géographes. Elle s'est de plus dotée d'architectures pour la définition des comportements des agents sur lesquels je reviendrai au Chapitre 3. Enfin, elle s'est enrichie de nombreuses fonctionnalités en lien avec la modélisation et la simulation participative (Chapitre 4).

### 1.3.2 Le langage de modélisation de GAMA

L'un des points forts de GAMA est sa facilité d'utilisation grâce à son langage de modélisation appelé GAML (GAMA Modeling Language). Le GAML est un langage orienté agent, ce qui signifie que tout ce qui est « actif » (entités d'un modèle, simulations, expérimentations...) est un agent.

Le GAML trouve ses racines dans les langages orientés-objet tels que le Java ou le Smalltalk, auquel il ajoute des concepts issus du paradigme agent. Il reprend également un certain nombre de concepts provenant du langage de modélisation de Netlogo permettant de faciliter la manipulation des agents.

Les concepts principaux derrière le GAML peuvent être décrits comme suit (Figure 1.3) : comme dans le paradigme objet, où la notion de classe est utilisée pour fournir une spécification pour des objets, les agents en GAML sont spécifiés par leur espèce (*species* en GAML), qui leur fournit un ensemble d'attributs (leur état), des actions (ce qu'ils peuvent faire), des comportements (ce qu'ils font réellement - appelés *reflex*) et spécifie également les propriétés globales liées à la population, par exemple leur organisation spatiale (continue, graphe ou grille) ou leur ordonnancement.

Le GAML est par nature multi-niveau : toute espèce d'agents peut être imbriquée dans une autre espèce. Une espèce peut également hériter des propriétés d'une autre espèce (appelée espèce parente), créant ainsi une relation similaire à la spécialisation dans la conception orientée objet. Par exemple, dans la Figure 1.3, l'espèce B est imbriquée dans l'espèce A et l'espèce C hérite (attributs et comportements) de l'espèce A.

Une particularité du GAML est que les modèles (*model*) et les plans d'expérience (*experiment*) sont considérées comme des espèces d'agents. Une expérimentation est ainsi un agent qui a la charge de gérer comment les simulations issues d'un modèle sont exécutées, en particulier quelles sont pour chacune d'elle les valeurs des paramètres d'entrée et comment sont affichées les sorties des simula-



tions.

L'écriture d'un modèle en GAML consiste ainsi à définir une espèce héritant de la classe *model*, dans laquelle d'autres espèces, héritant (directement ou non) de la classe *species* et représentant les entités qui peuplent ce modèle, seront imbriquées. Le modèle sera lui-même imbriqué dans une ou plusieurs *experiment* définissant les contextes d'exécution possibles du modèle.

La Figure 1.4 présente un exemple de modèle GAMA simple : dans ce modèle, une espèce d'agents appelée *simple\_agent* est définie. Nous avons attaché à cette espèce le *moving skill* de façon à fournir aux agents de cette espèce des attributs et des actions prédéfinis relatifs au déplacement. Nous définissons pour cette espèce un attribut appelé *color* qui est initialisé avec une valeur aléatoire. De plus, nous définissons pour cette espèce un comportement sous la forme d'un *reflex* appelé *behavior* qui sera activé à chaque pas de simulation et qui appelle l'action *wander* fournie par le *moving skill*. Cette action permet à un agent de se déplacer aléatoirement (*random walk*). Nous définissons aussi pour cette espèce un *aspect* appelé *circle* qui permet de représenter les agents de cette espèce sous la forme d'un cercle avec leur attribut *color*. En plus de l'espèce *simple\_agent* une section *global* est définie. Cette section permet de décrire la composante globale du modèle. Dans ce modèle, nous spécifions juste dans cette section que nous souhaitons à l'initialisation de la simulation créer 100 agents de l'espèce *simple\_agent* qui seront localisés aléatoirement dans l'environnement. Enfin, nous définissons un plan d'expérience appelé *main\_experiment*, qui n'exécutera qu'une seule simulation et qui définit pour celle-ci qu'une seule sortie du type *display* (fenêtre d'affichage) appelée *map* qui va afficher les agents de l'espèce *simple\_agent* avec leur aspect *circle*.

### 1.3.3 Utilisation de la plate-forme GAMA

La plate-forme GAMA a aujourd'hui une vraie reconnaissance et a de nombreux utilisateurs à travers le monde. Elle a ainsi été au cœur de nombreux projets touchant à des domaines aussi divers que les problématiques environnementales, les systèmes urbains, la mobilité, l'épidémiologie, l'adaptation aux changements climatiques ou la gestion des risques.

Si mon travail a essentiellement porté sur des aspects méthodologiques et l'ajout de nouvelles fonctionnalités dans GAMA, j'ai également directement participé depuis ma thèse au développement de plus de 14 modèles. Ces modèles portent sur des problématiques variées telles que l'évacuation de villes (Czura *et al.*, 2015; Chapuis *et al.*, 2018a; Daudé *et al.*, 2019), les évolutions des modalités de transports (Grignard *et al.*, 2018a), la gestion des risques pour les projets de construction (Taillandier *et al.*, 2015, 2016a), la reproduction de crises passées (Gasmi *et al.*, 2014), la propagation de maladies vectorielles (Cissé *et al.*, 2012; Philippon *et al.*, 2016), l'évacuation en cas de feu de forêt (Adam et Gaudou, 2017; Adam *et al.*, 2017), les évacuations de bâtiments (Le *et al.*, 2010; Valette *et al.*, 2018), la gestion des ressources naturelles (Therond *et al.*, 2014; Gaudou *et al.*, 2014; Taillandier *et al.*, 2012b), la gestion de l'énergie (Micolier *et al.*, 2018, 2019) et les évolutions d'occupation et d'utilisation du sol (Taillandier *et al.*, 2016b; Truong *et al.*, 2015).

A titre d'illustration, les parties suivantes détaillent 3 exemples mettant en avant les apports des travaux que j'ai menés sur les dimensions spatiales, comportementales et participatives des simulations.

#### Dimension spatiale : MOSAIIC et ESCAPE

Le modèle MOSAIIC (Czura *et al.*, 2015) (Figure 1.5) s'intéresse à la simulation du trafic routier dans les villes, en particulier en cas de situation de crise. Sa spécificité est qu'il permet de prendre en compte de façon fine le comportement des automobilistes (changement de voie, respects ou non des

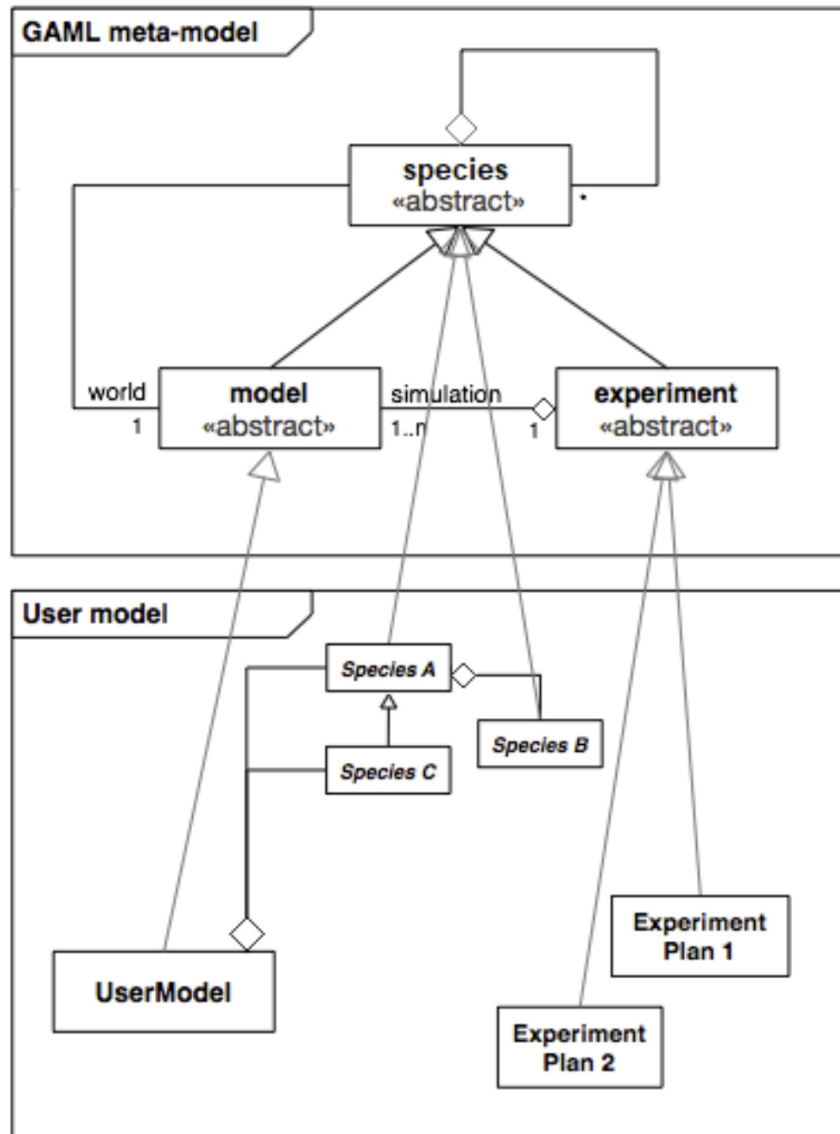


FIGURE 1.3 – Principaux concepts de la plate-forme GAMA (Taillandier *et al.*, 2018)

```
model SimpleModel

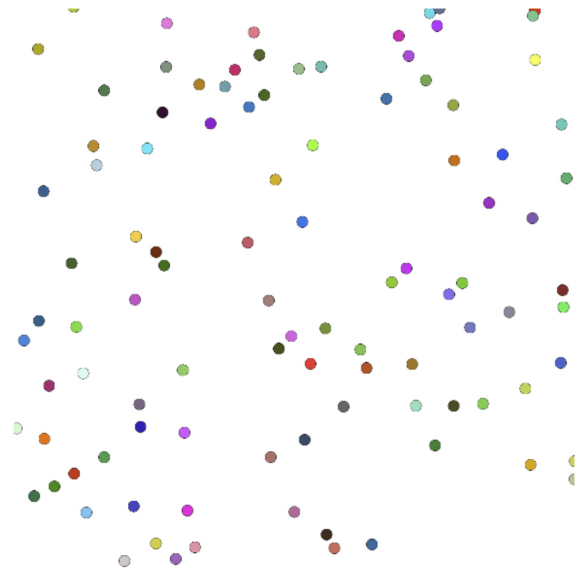
global {
  init {
    create simple_agent number: 100;
  }
}
species simple_agent skills: [moving] {
  rgb color <- rnd_color(255);

  reflex behavior {
    do wander;
  }

  aspect circle {
    draw circle(1) color: color border: #black;
  }
}

experiment main_experiment type: gui {
  output {
    display map {
      species simple_agent aspect: circle;
    }
  }
}
```

a)



b)

FIGURE 1.4 – a) code GAML du *simpleModel* : ce modèle crée 100 agents de l'espèce *simple\_agent* et les affiche par un cercle avec une couleur aléatoire ; à chaque pas de simulation, les agents se déplacent aléatoirement ; b) capture d'écran d'une simulation (Taillandier *et al.*, 2018).

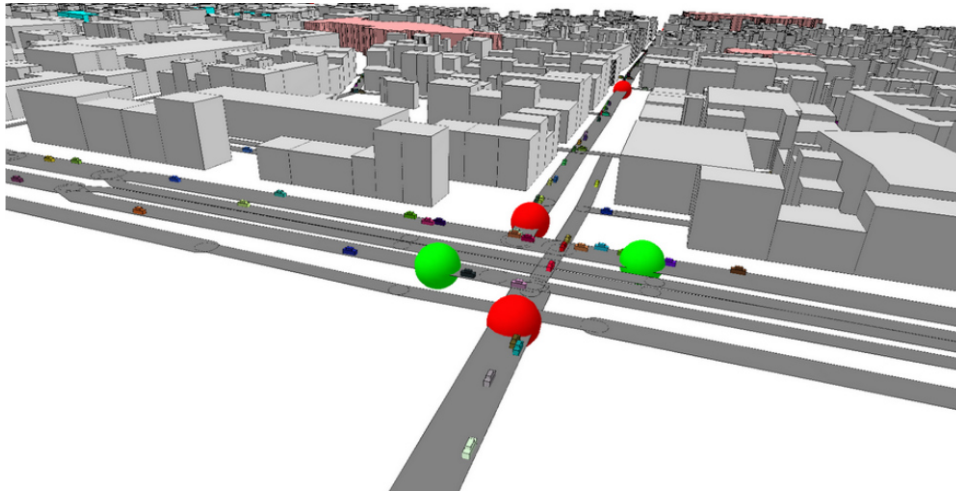


FIGURE 1.5 – Capture d’écran d’une simulation réalisée avec le modèle MOSAIIC pour la ville de Rouen (Czura *et al.*, 2015).

règles de circulation et des feux rouges...) à l’échelle de la ville. Il utilise pour cela les capacités de GAMA en termes de gestion des données géographiques ainsi que le plug-in que j’ai développé pour la simulation du trafic routier (Taillandier, 2014b) et que je présenterai au Chapitre 2. Le modèle a été utilisé pour simuler le trafic routier dans la ville de Rouen (plus de 70 000 automobilistes aux heures de pointes).

#### **Dimension comportementale : Li-BIM**

Le modèle Li-BIM (Micolier *et al.*, 2018, 2019) (Figure 1.6) s’intéresse à la consommation énergétique des occupants d’un bâtiment. Il intègre un sous-modèle de comportement des occupants, qui utilise les capacités de GAMA en termes de définition de comportements complexes, et plus particulièrement notre architecture d’agents cognitifs que je présenterai au Chapitre 3. Le modèle permet ainsi de prendre en compte finement le comportement des occupants, tant en termes de motivation personnelle que de concertation entre eux (par exemple, pour choisir la température du chauffage). Le modèle utilise également les capacités de GAMA pour gérer les formats de données BIM (Building Information Modeling) qui sont aujourd’hui une référence pour la modélisation des bâtiments dans le domaine du Génie Civil.

#### **Dimension participative : CityScope**

Le modèle produit dans le cadre du projet CityScope, coordonné par l’équipe City Science du Media Lab (MIT), porte sur les déplacements multi-modaux et le design urbain (Grignard *et al.*, 2018a) (Figure 1.7). Ce modèle utilise les capacités de GAMA en termes de développement de simulations interactives, dont que je parlerai au Chapitre 4, pour permettre à des utilisateurs d’interagir avec les simulations au travers d’une maquette physique (en LEGO).

## **1.4 Contributions et organisation du mémoire**

Les travaux que j’ai menés depuis la fin de ma thèse se sont ainsi intéressés à la modélisation à base d’agents, approche de modélisation que j’ai rapidement introduit dans ce chapitre. En particulier, ils

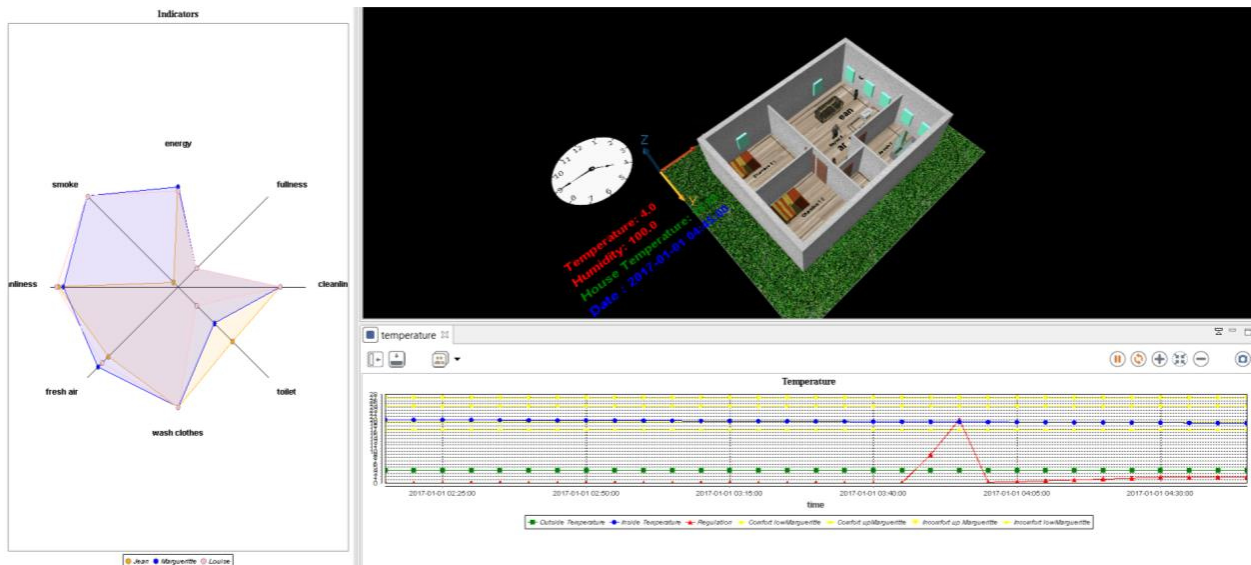


FIGURE 1.6 – Interface de simulation du modèle Li-BIM (Micolier *et al.*, 2018).

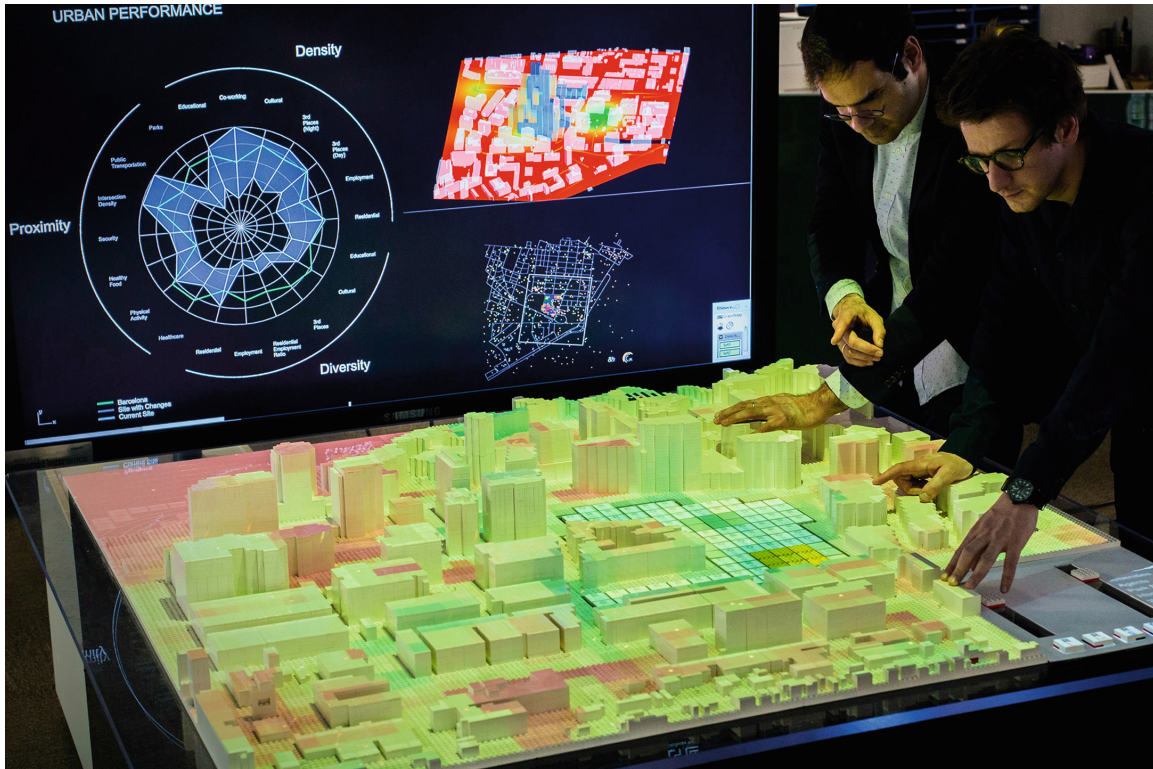


FIGURE 1.7 – CityScope : un outil de simulation interactif pour le design urbain (Grignard *et al.*, 2018a).

ont cherché à faciliter l'accès à cette approche de modélisation au plus grand nombre et se sont incarnés au travers du développement de la plate-forme GAMA. Ces travaux peuvent être regroupés en 3 axes qui forment les chapitres de ce mémoire.

Le Chapitre 2 sera consacré à la présentation des travaux que j'ai menés sur le développement de la composante spatiale dans les simulations. En effet, mes compétences en géomatique et en géographie m'ont naturellement guidé sur ces problématiques qui ont été, et restent, au cœur de mes recherches. Je détaillerai en particulier comment nous avons introduit cette composante spatiale dans GAMA pour en faire la plate-forme de modélisation agent de référence pour la simulation spatialisée. J'évoquerai ensuite mes travaux liés à la thématique de la mobilité. Enfin, j'évoquerai une problématique à laquelle les modélisateurs sont souvent confrontés durant les projets de modélisation : celui de l'accès aux données pour initialiser leur simulation et plus particulièrement comment initialiser la localisation de leurs agents. Nous reviendrons ainsi sur les travaux que nous avons menés dans le projet ANR GEN\*, coordonné par Alexis Drogoul entre 2013 et 2017, sur la spatialisation de populations synthétiques.

Le Chapitre 3 s'intéressera à une autre thématique forte de mes recherches : la modélisation du comportement des acteurs. En effet, un facteur important pour arriver à produire des modèles crédibles, en particulier en simulation sociale, est d'arriver à bien capturer les comportements des différents acteurs du système étudié. Malheureusement, modéliser le comportement d'acteurs ayant des capacités de raisonnement élevées et sujets à des émotions est une tâche particulièrement complexe. J'ai d'abord travaillé sur ce sujet en post-doctorat dans le cadre du projet RTRA MAELIA (Multi-Agents for Environmental norms Impact Assessment) coordonné par Pierre Mazzega, où j'ai proposé une architecture d'agents pour modéliser le comportement des agriculteurs. J'ai pu ensuite poursuivre sur ce sujet grâce à l'obtention en 2014 du projet ANR ACTEUR (Agents Cognitifs Territorialisés pour l'Etude des dynamiques Urbaines et des Risques). Ce chapitre sera ainsi consacré à la présentation de l'architecture proposée dans le projet MAELIA, puis de celle proposée dans le projet ACTEUR.

Le Chapitre 4 sera consacré à la présentation des travaux que j'ai menés sur la modélisation et la simulation participative. En effet, aujourd'hui, de plus en plus de projets impliquent directement des acteurs dans le processus de modélisation ou au travers de jeux sérieux. Cette participation des acteurs a de nombreux avantages dont une meilleure appropriation des modèles dans un contexte d'aide à la décision. Elle permet également de répondre à différents objectifs : former des acteurs, les amener à mieux communiquer entre eux ou simplement leur faire comprendre certaines problématiques. S'il existe de longue date des outils adaptés à la modélisation et la simulation participative pour des modèles simples (KISS), arriver à fournir des méthodes et outils permettant de faire de même pour des modèles plus complexes (KIDS) est un vrai challenge, surtout si l'on souhaite rendre ces outils facilement accessibles. Je présenterai dans ce chapitre les travaux que j'ai menés sur la modélisation graphique ainsi que sur le développement d'outils pour la conception de jeux sérieux. Enfin, je présenterai mes travaux sur l'apprentissage de comportements d'agents par la simulation participative.

Enfin, le Chapitre 5 permettra de conclure ce mémoire et, après y avoir exposé ma vision sur la modélisation à base d'agents à l'horizon 2030, je présenterai des pistes de recherche qui formeront mon projet de recherche dans les années à venir.





## 2 — L'espace dans la simulation à base d'agents

### 2.1 Prise en compte de l'espace dans la plate-forme GAMA

#### 2.1.1 Contexte

L'intégration de la dimension spatiale dans les simulations a depuis toujours fait partie des forces des modèles à base d'agents. Dans les premiers modèles, cet espace était souvent représenté sous la forme d'une grille régulière composée d'un ensemble de cellules uniformes, permettant de stocker dans chaque cellule un ensemble de variables et de donner un comportement à celles-ci. Si ce mode de représentation offre des atouts, en particulier sa simplicité d'utilisation, il a également des limites quand il s'agit de représenter des objets géographiques tels que des bâtiments ou des routes. Il pose également le problème du choix de la résolution : une résolution très fine impose un très grand nombre de cellules et peut donc s'avérer lourd en termes de mémoire et de temps de calcul ; une résolution plus grossière peut poser des problèmes lorsqu'il s'agit de représenter des entités de petites tailles ou ayant des comportements très localisés.

Ces dernières années ont donc été marquées par l'essor de modèles basés sur d'autres modes de représentation de l'environnement et en particulier par des modèles intégrant des données géographiques vectorielles. Cet essor a entre autres été permis par la révolution du Big Data et des bases de données libres qui ont grandement facilité l'accès aux données. L'utilisation de ces modes de représentation de l'espace permet de mieux ancrer les modèles dans une réalité territoriale (voir par exemple ([Crooks et Castle, 2012](#))).

Néanmoins, ces nouveaux modes de représentation des données posent de nombreuses questions, en particulier concernant l'implémentation des modèles. Ils nécessitent en effet de disposer d'outils permettant d'intégrer des données aux formats variés et de les manipuler. Si la plupart des plates-formes existantes permettent une gestion de plusieurs modes de représentation de l'espace, le niveau des outils qu'elles intègrent pour aider le modélisateur à les manipuler est assez disparate.

Ainsi, la plate-forme SWARM ([Minar et al., 1996](#)), qui a été la source d'inspiration de beaucoup d'autres plates-formes, n'intègre pas dans sa version de base de moyen d'intégrer des données géo-

graphiques vectorielles. De façon à dépasser cette limitation, une extension appelée KENGE (Box, 2002) a été développée. Cette extension permet de charger une couche de données géographiques vectorielles et de créer une grille à partir de cette couche. D'autres travaux tels que (Haklay *et al.*, 2001) ont montré au travers d'approches ad hoc qu'il était possible d'utiliser, pour des modèles spécifiques, des données géographiques avec SWARM. Malheureusement, SWARM ne fournit aucune primitive spatiale, ni la possibilité d'exporter les résultats sous forme de données spatiales.

Dans la plate-forme Netlogo, l'environnement est représenté par une grille composée de *patches*, qui sont des agents ayant des coordonnées définies dans un environnement continu et pouvant avoir leurs propres comportements. Des primitives de la plate-forme permettent de directement effectuer des opérations de diffusion, de calculs de distance et des requête spatiales simples (exemple : les voisins dans un certain rayon). Une extension de la plate-forme permet également d'intégrer dans les modèles des données géographiques vectorielles (Russell et Wilensky, 2008). Cette extension permet d'importer et d'exporter des données vectorielles et de charger leurs attributs dans des *patches*. Il est également possible d'accéder à leurs caractéristiques géométriques (centroïde, liste de points, etc.). Quelques opérations spatiales sont aussi fournies (calcul du rectangle englobant, union de polygones, etc.). Néanmoins, cette extension ne permet pas de manipuler directement les géométries et, de plus, les primitives spatiales offertes restent très basiques.

Autre plate-forme classique, CORMAS fournit elle aussi des outils pour gérer la dimension spatiale des modèles. Elle propose deux modes de représentation de l'environnement : vecteur et raster. Ces deux modes de représentation partagent la même organisation en 3 classes « entité spatiale », « agent » et « objet ». Cette organisation permet de disposer d'un protocole standard de déplacement des entités. Au sein de CORMAS, seuls les services spatiaux de base sont fournis (comme la localisation, le voisinage, les déplacements de base et la perception) et les services de plus haut niveau doivent être implémentés par le modélisateur. En 2008, Urbani (Urbani et Delhom, 2008) a proposé l'architecture SMAG permettant de faire le pont entre une plate-forme de modélisation agent et un SIG. L'auteur l'a implémenté sur CORMAS, l'appelant CORMGIS. L'intégration est relativement basique car l'accès aux données géoréférencées s'effectue via une connexion avec ArcGIS. De plus, aucune primitive SIG (union, intersection, etc.) n'est disponible.

Une dernière plate-forme, ou plutôt famille de plates-formes, très utilisée en modélisation à base d'agents est Repast (North *et al.*, 2013a). Repast fournit différents services pour la gestion des grilles, mais également pour la gestion des données géographiques vectorielles tels que l'importation/exportation de données vecteur (shapefile) et raster ou des opérations géométriques (requête spatiale, union, différence...). Repast permet également de construire un graphe à partir d'un ensemble de lignes, et de calculer des plus courts chemins à partir de ce graphe. Néanmoins, le nombre d'opérateurs spatiaux offerts par Repast est assez limité et les opérations plus complexes doivent être programmées à l'aide d'un langage générique par le modélisateur (exemple : Java), ce qui est hors de portée de beaucoup de modélisateurs.

Pour conclure, si les plates-formes de modélisation à base d'agents ont permis de populariser cette approche de modélisation, la plupart de celles-ci sont encore limitées lorsqu'il s'agit de manipuler des données spatiales (exemple : Swarm, Netlogo, Cormas) et peuvent requérir pour certaines un niveau de programmation élevé (exemple : Repast).

Cette problématique d'intégration de la dimension spatiale dans les modèles a depuis toujours été au



centre du développement de la plate-forme GAMA. En effet, dans sa première version datant de 2007, GAMA était une surcouche de Repast (pour être précis de Repast J) qui visait justement à faciliter l'utilisation de données géographiques dans les simulations (Amouroux *et al.*, 2007). Mon arrivée en post-doctorat à Hanoi en 2008 a contribué, grâce à mes compétences en Géomatique, à fortement faire progresser GAMA dans cette voie et à en faire aujourd'hui une plate-forme particulièrement puissante pour la gestion de l'espace.

La suite de cette partie propose une description synthétique des capacités actuelles de GAMA en termes de gestion de l'espace. Plus de détails pourront être trouvés dans les nombreuses publications auxquelles j'ai contribué sur le sujet : Taillandier *et al.* (2018); Grignard *et al.* (2013b); Taillandier *et al.* (2014); Taillandier et Drogoul (2011); Taillandier *et al.* (2010d); Drogoul *et al.* (2013b,a); Taillandier *et al.* (2010b)

### 2.1.2 Prise en compte de différentes topologies spatiales

Si la plupart des plates-formes permettent de gérer différents modes de représentation de l'environnement (continu, grilles, graphes), elles ne permettent pas toutes d'intégrer dans un modèle de multiples représentations de l'environnement. Il est en effet nécessaire pour certains modèles de pouvoir définir un environnement continu (pour intégrer des données vectorielles telles que des bâtiments ou des routes) et de représenter le même environnement sous la forme d'une ou plusieurs grilles (MNT, grille d'occupation des sols) et d'intégrer des graphes (graphe social, graphe des routes).

GAMA permet de définir et de gérer autant de représentation de l'environnements que nécessaire. L'ensemble de ces représentations est synchronisé par l'utilisation d'un environnement continu 3D de référence. Chaque représentation est munie d'une topologie qui fournit des services aux agents composant la topologie, en particulier concernant le calcul de voisinage et de distance. Il existe dans GAMA trois types de topologies : l'espace continu, la grille et le graphe (Figure 2.1). GAMA permet également de gérer l'aspect toroïdal des environnements. A titre d'illustration, la Figure 2.2 montre le résultat d'un calcul de distance entre deux agents selon différentes topologies.

Une autre spécificité de GAMA est que tout agent est situé dans l'environnement de référence et contient une géométrie – accessible au travers de sa variable *shape*. Basée sur une représentation vectorielle, cette géométrie peut être simple (un point, une polyligne, un polygone) ou complexe (composée de plusieurs sous-géométries). La géométrie d'un agent peut être modifiée à tout moment durant la simulation. Elle peut être définie par l'utilisateur (par une liste de points par exemple) ou directement chargée depuis un fichier de données. Par défaut, la géométrie d'un agent est un point. De manière à faciliter la manipulation des agents dans l'espace, les agents disposent également d'une variable de localisation (appelée *location*) de type point représentant le centroïde de la géométrie de l'agent. Modifier la géométrie d'un agent modifie automatiquement la localisation de l'agent. De la même manière, modifier la localisation d'un agent permet d'effectuer automatiquement une translation sur sa géométrie. Afin de faciliter la manipulation des géométries, GAMA, et plus particulièrement le langage GAML, intègre un type de variable appelé *geometry*. Ainsi, le modélisateur peut définir à tout moment une variable de type *geometry* et effectuer sur celle-ci toutes les opérations qu'il souhaite. GAMA fournit pour cela de nombreux opérateurs spatiaux (présentés en partie suivante).

Concernant les grilles, chaque cellule de la grille (cellule rectangulaire ou cellule hexagonale) est un agent. L'ensemble de ces agents forme une topologie de type grille. Comme tout agent, un agent cel-

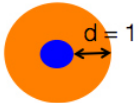


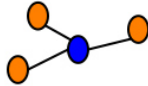
Topologie	Continue	Grille	Graphe
Voisinage (distance 1)		 Moore  Von Neumann	

FIGURE 2.1 – Les différentes topologies dans GAMA : en orange, le voisinage à une distance de 1 pour chacune de ces topologies (Grignard *et al.*, 2013b).

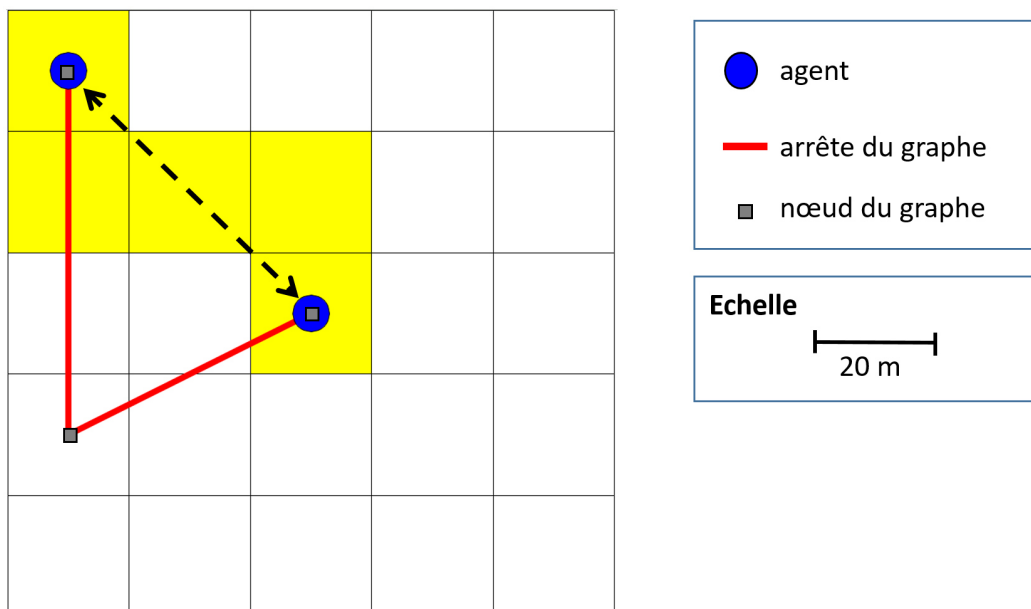


FIGURE 2.2 – Calcul des distance entre deux agents (cercles bleus) en fonction de la topologie spatiale : distance dans un espace continu (Euclidienne) : 56.6 m ; distance à partir de la grille - voisinage de Von Neumann (cellules jaunes) : 4 cellules, distance à partir du graphe (lignes rouges) : 104.7 m (Taillandier *et al.*, 2018).

lule a une géométrie (polygone), peut avoir des variables, des comportements (réflexes et actions) et des aspects.

GAMA fournit également de nombreux outils pour définir et travailler sur les graphes. Les sommets et les arrêtes d'un graphe peuvent être de différentes natures, par exemple des agents ou des géométries. Afin de faciliter la manipulation des graphes, GAMA permet de définir des variables de type *graph*. Nous reviendrons sur la création et les opérateurs de graphe dans la partie suivante.

### 2.1.3 Intégration des données géographiques dans les simulations

Aujourd'hui, de plus en plus de modèles utilisent des données géographiques issues d'un SIG pour améliorer leur réalisme. Si la plupart des plates-formes de simulation permettent de lire et d'écrire des données géographiques, la liberté offerte par ces dernières pour manipuler ces données est plus inégale. GAMA offre de nombreux outils pour intégrer des données géographiques. En particulier, GAMA permet, au travers d'une simple instruction GAML, d'utiliser un fichier géographique (shapefile, OSM, GML) pour créer des agents d'une espèce donnée : chaque objet contenu dans les données géographiques sera utilisé pour instancier un agent (la géométrie de l'agent sera celle lue dans le fichier), GAMA se chargeant de gérer le système de coordonnées spatiales des données (possibilité de projeter à la volée les données) et si nécessaire de lire les valeurs des attributs dans le fichier.

De la même manière, GAMA permet d'utiliser un fichier raster (asc, geotiff ou image) pour créer une grille aux dimensions voulues. La valeur contenue dans le fichier raster sera automatiquement utilisée pour initialiser la valeur des variables *grid\_value* et/ou *bands* (cas de données multi-bandes) de chaque agent cellule. GAMA se charge également de construire la géométrie (ici un rectangle) correspondant au fichier pour chaque agent cellule.

GAMA permet également, à l'aide d'une simple instruction GAML, d'exporter les géométries d'une liste d'agents (ou directement une liste de géométries) sous la forme d'un shapefile ou d'un fichier KML pour les données vectorielles, ou pour une grille sous la forme d'un fichier asc ou geotiff. Pour les shapefiles, il est également possible d'exporter la valeur des variables souhaitées des agents sous la forme de tableaux d'attributs.

De façon à faciliter la manipulation de la dimension spatiale dans les modèles, GAMA fournit plus de 100 opérateurs spatiaux accessibles en GAML. Ces derniers sont directement inspirés de ceux généralement proposés par les SIG :

- *Créations* : création de géométries basiques telles que des cercles, triangles, rectangles, hexagones, courbes de Bézier, cubes ou sphères, ainsi que de géométries plus complexes à l'aide de liste de points.
- *Transformations* : buffer, translation, rotation, grossissement, simplification, enveloppe convexe, décomposition en sous géométries telles que des carrés ou des triangles (Delaunay)...
- *Projections* : application d'opérations de projection spatiale sur des géométries en fonction d'un système de coordonnées de référence cible.
- *Relations* : calcul d'angle, distance et chemin entre deux géométries.
- *Propriétés* : détermine si deux géométries s'intersectent, se croisent, se touchent, si l'une couvre l'autre, etc.
- *Opérations* : union, différence, intersection, etc.

- *Statistiques* : clustering spatial, interpolation spatiale, indice de Moran, etc.
- *Ponctuels* : point aléatoire dans une géométrie, point le plus proche d'une géométrie par rapport à une autre géométrie, point le plus éloigné d'une géométrie, etc.
- *Requêtes* : Requête spatiale concernant d'autres géométries ou agents, par exemple les agents se trouvant à une certaine distance, qui intersectent une géométrie, à l'intérieur d'une géométrie, les plus proches ou les plus éloignées d'une géométrie, etc.

De façon à optimiser les calculs, et en particulier les requêtes spatiales, GAMA utilise une structure de type QuadTree dynamique qui est mise à jour en fonction du déplacement, de la création et de l'élimination des agents.

Des outils permettant de gérer et d'interroger des bases de données sont aussi intégrés dans la plateforme (Truong *et al.*, 2013). Il est ainsi possible d'effectuer au travers du langage GAML des requêtes et des opérations sur des bases de données spatiales (par exemple PostGIS).

Comme mentionné précédemment, GAMA permet également de créer et de manipuler des graphes. En particulier, GAMA fournit différents opérateurs pour construire un graphe spatial à partir de géométries :

- *as\_edge\_graph* : construction d'un graphe à partir d'un ensemble de polygones (ou d'agents ayant pour géométrie des polygones). Dans ce cas, un arc sera créé par polygone et les nœuds représenteront les extrémités des polygones. Cet opérateur permet par exemple de créer un graphe à partir d'un réseau de routes.
- *as\_distance\_graph* : construction d'un graphe à partir d'un ensemble de géométries (ou d'agents). Dans ce cas, les nœuds représenteront les géométries (ou agents) et un arc sera créé si deux géométries se trouvent à une distance inférieure à un seuil donné.
- *as\_intersection\_graph* : construction d'un graphe à partir d'un ensemble de géométries (ou d'agents). Dans ce cas, les nœuds représenteront les géométries (ou agents) et un arc sera créé si deux géométries s'intersectent.

Concernant le calcul de distances ou de plus courts chemins dans les graphes, GAMA offre différents algorithmes tels que Dijkstra, Floyd Warshall, A\* ou NBA\* (Pijls et Post, 2009). Des algorithmes sont également disponibles pour le calcul de distances et de plus courts chemins sur les grilles (avec la possibilité de considérer des obstacles) tels que Dijkstra, A\* ou JPS (Harabor *et al.*, 2014).

#### 2.1.4 Déplacement des agents dans l'espace : le *moving skill*

GAMA fournit pour les agents de nombreuses primitives de déplacement permettant à un agent de se déplacer de façon transparente dans un environnement continu (borné ou non par une géométrie), dans une grille ou sur un graphe.

Ainsi, s'il est possible de déplacer un agent en modifiant juste son attribut *location* ou sa géométrie, il est possible d'attacher à une espèce d'agents un *skill* de façon à fournir aux agents de cette espèce des attributs et des actions prédéfinis relatifs au déplacement. Le *skill* le plus utilisé pour les déplacements est le *moving skill*.

Ce *skill* fournit aux agents différents attributs liés au déplacement. Certains de ces attributs sont di-

rectement modifiables par le modélisateur (vitesse de déplacement souhaitée, direction de l'agent), d'autres ont pour objectif de donner des informations sur le déplacement de l'agent (vitesse réelle de déplacement, chemin courant, etc.).

Le *moving skill* fournit également quatre actions de déplacement pour les agents :

- *wander* : déplacement de l'agent dans une direction aléatoire. Il est possible de spécifier l'angle maximal de déplacement (autour de la direction actuelle de l'agent) et une géométrie qui contraint le déplacement de l'agent.
- *move* : déplacement de l'agent dans sa direction courante. De la même manière que pour l'action *wander*, il est possible de définir une géométrie restreignant les déplacements de l'agent.
- *goto* : déplacement de l'agent dans la direction d'une cible (agent, géométrie ou point). Cette action permet de spécifier l'espace dans laquelle doit se déplacer l'agent : il peut s'agir d'un déplacement dans l'environnement continu, d'un graphe ou d'une grille (en prenant en compte des cellules interdites). Dans ces deux derniers cas, GAMA va faire appel à ses opérateurs de calcul de plus courts chemins pour calculer un chemin à suivre. De nombreuses options sont disponibles pour cette action : possibilité de différencier les poids des arrêtes utilisés pour le calcul du plus court chemin de ceux utilisés pour le déplacement, de recalculer ou non le chemin suivi si le graphe est modifié, etc.
- *follow* : déplacement de l'agent en suivant un chemin donné en argument.

Pour ces quatre actions, la distance maximale de déplacement est calculée en fonction de la vitesse souhaitée de l'agent et de la durée du pas de simulation. Il est à noter que GAMA permet de gérer des unités de temps (seconde, minute, heure, année, etc.) et de distance (mètre, kilomètre, pied, etc.).

Un travail important a été réalisé pour optimiser ces actions, permettant par exemple de réaliser des simulations reposant sur le *moving skill* comprenant des centaines de milliers d'agents (par exemple (Fosset *et al.*, 2016)).

## 2.2 Modélisation du trafic routier avec GAMA

### 2.2.1 Contexte

La modélisation à base d'agents est utilisée depuis plusieurs décennies pour modéliser le trafic routier. Ce domaine de recherche bien établi a ainsi vu le développement de nombreux outils dédiés à l'étude du trafic tels que MATSim (Balmer *et al.*, 2009), SUMO (Krajzewicz *et al.*, 2012) ou SimMobility (Adnan *et al.*, 2016) pour ne citer que les plus connus. La plupart de ces outils se concentrent sur la demande, la gestion et le contrôle du système de transport (Bazzan et Klügl, 2014). Ainsi, si leur utilisation permet de couvrir une large part des préoccupations des urbanistes pour la conception, l'entretien et l'expansion du réseau de transport, ils ne conviennent pas pour étudier les conséquences d'événements rares et/ou extrêmes sur la mobilité des personnes (Hawe *et al.*, 2012). En effet, la mobilité dans un contexte de crise est radicalement différente de la mobilité dans une situation normale : les individus peuvent changer d'objectif de destination très rapidement ; ils peuvent adopter une attitude différente de celle qui les caractérise habituellement, par exemple beaucoup de nervosité, de peur, d'impatience ; enfin, les réseaux de transport peuvent eux-mêmes être affectés directement par le danger (exemple : inondation ou tremblement de terre) ou indirectement (exemple : interruption des feux de signalisation). Malheureusement, adapter ces outils à un autre cadre que celui prévu à l'origine requiert d'écrire des lignes de code dans un langage de programmation générique tel que

Java ou C++, ce qui est hors de portée de la plupart des modélisateurs.

Pour faire face à ce problème, une solution consiste à utiliser une plate-forme de modélisation à base d'agents générique comme GAMA, qui est particulièrement bien adaptée à ce type d'application. En effet, comme montré dans la partie précédente, GAMA fournit aux modélisateurs de nombreuses fonctionnalités permettant de gérer l'espace et le déplacement des agents. Elle a ainsi déjà été utilisée pour plusieurs modèles de trafic tels que ceux proposés par [Fosset et al. \(2016\)](#) et [Grignard et al. \(2018a\)](#). Néanmoins, même avec le *moving skill*, développer un modèle de trafic détaillé prenant en compte des phénomènes tels que les congestions, le changement de voie ou la prise en compte des règles de circulation (feux rouges, priorité à droite) à une échelle temporelle fine (la seconde) est complexe et nécessite de nombreuses lignes de code.

Or, pouvoir définir un tel modèle était au coeur du projet GRR MOSAIIC (Accidents Industriels Majeurs et Vulnérabilités Humaine en Haute-Normandie : Modélisation et Simulation en Milieu Urbain). C'est dans le cadre de ce projet financé par la région Haute-Normandie et coordonné par Eric Daudé (CR CNRS, UMR IDEES) que j'ai développé un nouvel outil intégré dans GAMA permettant de simplifier la construction de modèles de trafic routier dans GAMA. L'idée était de proposer un outil générique, s'inspirant d'un premier modèle développé dans le cadre du projet MOSAIIC ([Tranouez et al., 2012](#)), qui propose une description fine du comportement des automobilistes. L'outil avait également pour vocation de rester facilement adaptable à divers contextes d'application. La suite de cette partie décrit de façon synthétique cet outil. Une description plus complète de celui-ci est disponible dans ([Taillandier, 2014b](#)).

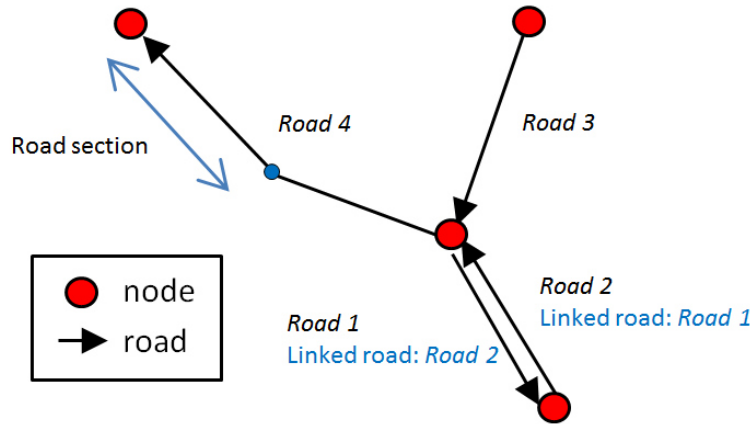
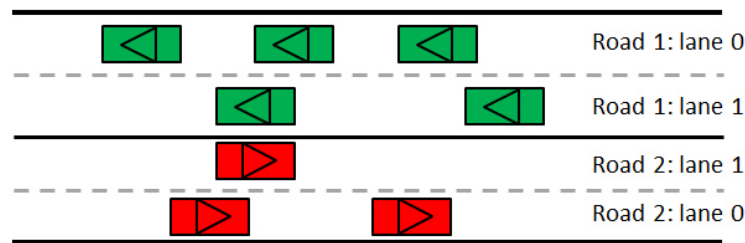
### 2.2.2 Le *advanced driving skill* de GAMA

Un premier choix dans ce travail a été de représenter les infrastructures routières (routes, feux rouges, etc.) sous la forme d'agents. L'objectif est de permettre aux modélisateurs de pouvoir simplement ajouter des dynamiques à ces infrastructures, par exemple une dynamique de détérioration des routes. J'ai ainsi développé 3 *skills* :

- *Advanced driving skill* : dédié à la définition des agents automobilistes. Il fournit à ces agents des variables et actions leur permettant de se déplacer sur un réseau de routes.
- *Road skill* : dédié à la définition des agents route. Il fournit à ces agents des variables et des actions leur permettant d'enregistrer les agents automobilistes les traversant.
- *Road Node Skill* : dédié à la définition des intersections entre routes. Il fournit à ces agents des variables permettant aux automobilistes de prendre en compte les règles de circulation sur ces intersections (priorité, feu rouge, etc.).

Un enjeu important dans la définition de cet outil a été de le rendre utilisable avec la plupart des données SIG classiques, en particulier avec les données OpenStreetMap (OSM). Dans cette perspective, j'ai fait le choix de représenter les routes et intersections sous une forme très classique (Figure 2.3) : chaque route est représentée par une polyligne composée de tronçons (segments) et a une intersection (noeud) source et destination. Chaque noeud connaît toutes les routes entrantes et sortantes. Une route est considérée comme unidirectionnelle. Pour une route bidirectionnelle, deux routes sont définies correspondant aux deux directions, chaque route étant la *linked\_road* de l'autre.

Plus spécifiquement, une route peut être composée de plusieurs voies (Figure 2.4) et les véhicules

FIGURE 2.3 – Structure des routes et des intersections (Czura *et al.*, 2015).FIGURE 2.4 – Structures des routes et des voies (Czura *et al.*, 2015).

peuvent changer de voie à tout moment. Une autre propriété des routes qui est prise en compte est la vitesse maximale autorisée. Les valeurs de ces propriétés peuvent provenir de données ou être définies par l'utilisateur. A noter qu'il est également possible de redéfinir à tout moment ces valeurs, ce qui permet par exemple de simuler l'impact d'un accident qui bloquerait une voie ou la diminution de la vitesse sur une route par les autorités. En plus de ces propriétés, les routes stockent la liste des agents automobilistes qui se trouvent sur chacune de leurs voies.

Les intersections conservent pour chacune des routes des informations sur les règles de circulation quant à leur accès (exemple : routes bloquées par un feu rouge) que l'agent automobiliste pourra choisir d'ignorer s'il le souhaite (conduite non-normative). Les intersections conservent également des listes de routes bloquées par des automobilistes. Cette liste permet de prendre en compte, par exemple, un automobiliste choisissant de bloquer une intersection plutôt que de laisser passer les automobilistes venant des routes perpendiculaires.

Les agents automobilistes reçoivent de l'*advanced driving skill* un certain nombre de variables qui définissent les propriétés de leur véhicule et la personnalité du conducteur : longueur du véhicule, son accélération maximale, sa vitesse maximale, sa propension à rouler en-dessous ou au-dessus des limites de vitesse, à garder une distance de sécurité importante ou non, à changer de voie, à respecter les règles de priorité et les feux rouges, à accepter de laisser passer les automobilistes venant des autres routes aux intersections, etc. L'*advanced driving skill* fournit également deux actions : la première, appelée *compute\_path*, permet à l'agent de calculer un chemin en direction de sa destination, la se-



conde appelée *drive* permet à l'agent de conduire en direction de sa destination en suivant le chemin calculé.

L'action *drive* suit la procédure suivante (Figure 2.5) : l'agent commence par définir la vitesse à laquelle il souhaite se déplacer sur la route sur laquelle il se trouve. Cette vitesse correspond à la vitesse idéale que l'agent souhaite atteindre s'il n'y avait pas d'autres véhicules sur la route. Elle dépend de différents facteurs comme la vitesse maximale autorisée sur la route, les capacités du véhicule (accélération et vitesse maximales) et la personnalité du conducteur (propension à rouler au-dessus ou au-dessous des limites de vitesse). L'agent se déplace ensuite sur la route jusqu'à ce qu'il arrive à la fin de la route ou que son temps de déplacement soit fini (un pas de simulation). Durant son déplacement l'agent peut changer de voie, et va adapter sa vitesse en fonction des autres véhicules. Si l'agent atteint la fin de la route et qu'il a toujours du temps pour se déplacer, il vérifie s'il peut ou non traverser l'intersection pour atteindre la prochaine route. Si c'est le cas, il continue son déplacement sur la nouvelle route.

La fonction permettant de savoir si l'agent peut franchir une intersection ou non suit la procédure suivante (Figure 2.6) : d'abord, l'agent vérifie si l'intersection n'est pas bloquée par un autre conducteur. Il vérifie ensuite s'il n'y a pas de signalisation (feu rouge, stop, etc.) bloquant son passage ou s'il souhaite ou non respecter cette signalisation. Il vérifie ensuite s'il n'y a pas de véhicules à qui il doit laisser la priorité (priorité à droite par exemple) ou s'il souhaite respecter ou non cette priorité. Il regarde enfin s'il a assez d'espace pour s'insérer sur la nouvelle route sans bloquer l'intersection : si ce n'est pas le cas, il peut choisir de quand même s'insérer sur la nouvelle route, auquel cas les véhicules venant des routes perpendiculaires seront alors bloqués par lui.

Pour leur déplacement sur une route (Figure 2.6), les agents se déplacent de tronçon en tronçon en fonction du temps de déplacement qu'il leur reste : pour chaque tronçon, l'agent calcule d'abord la distance maximale qu'il peut parcourir en fonction du temps restant et de sa vitesse idéale. Il calcule ensuite la distance de sécurité qu'il souhaite conserver en fonction de sa vitesse et de sa propension à garder sa distance par rapport aux autres véhicules. Tant que la distance qu'il peut parcourir n'est pas nulle et qu'il n'est pas arrivé à la fin de la route, l'agent calcule pour chacune des voies la distance maximale qu'il peut parcourir en fonction de sa propension à changer de voie et des autres véhicules et choisit celle sur laquelle il peut parcourir le plus de distance.

### 2.2.3 Validation

Afin de vérifier la capacité de notre outil à reproduire des comportements macroscopiques classiques, nous avons testé sa capacité à se rapprocher du diagramme fondamental (Geroliminis et Daganzo, 2008). Le diagramme fondamental est basé sur les observations suivantes :

- Quand le nombre de véhicules sur le réseau est petit, les interactions entre ces véhicules sont proches de zéro. Chaque véhicule peut donc rouler à sa vitesse maximale sans être gêné par les autres.
- Quand le nombre de véhicules augmente, les interactions entre véhicules augmentent aussi, ce qui a pour effet de réduire la vitesse des véhicules.
- Il y a un cas limite où un embouteillage se forme et la vitesse des véhicules tend vers 0.

Nous avons testé la capacité de notre outil à reproduire ce comportement au travers d'une expérimentation.



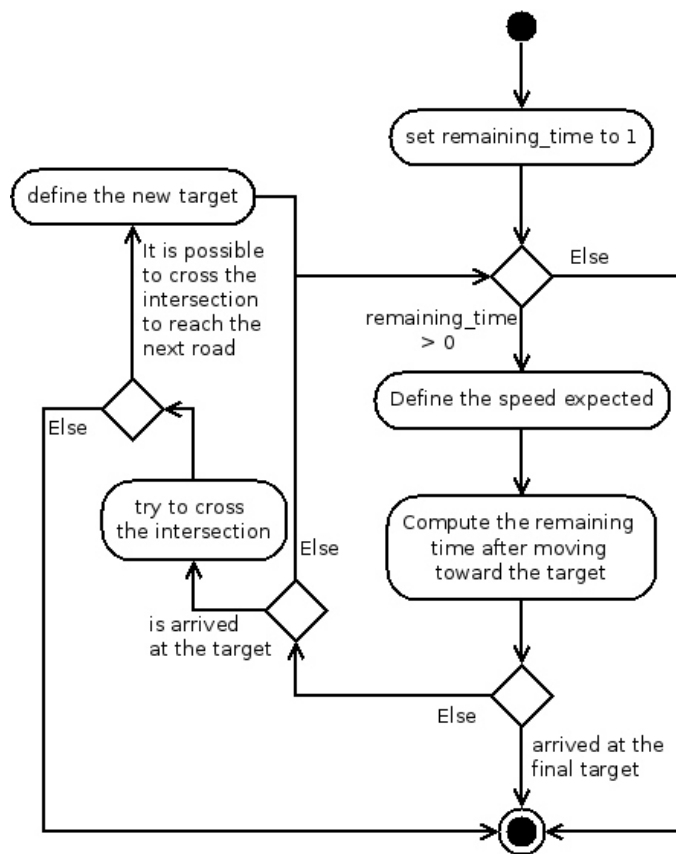


FIGURE 2.5 – Action drive (Czura et al., 2015).

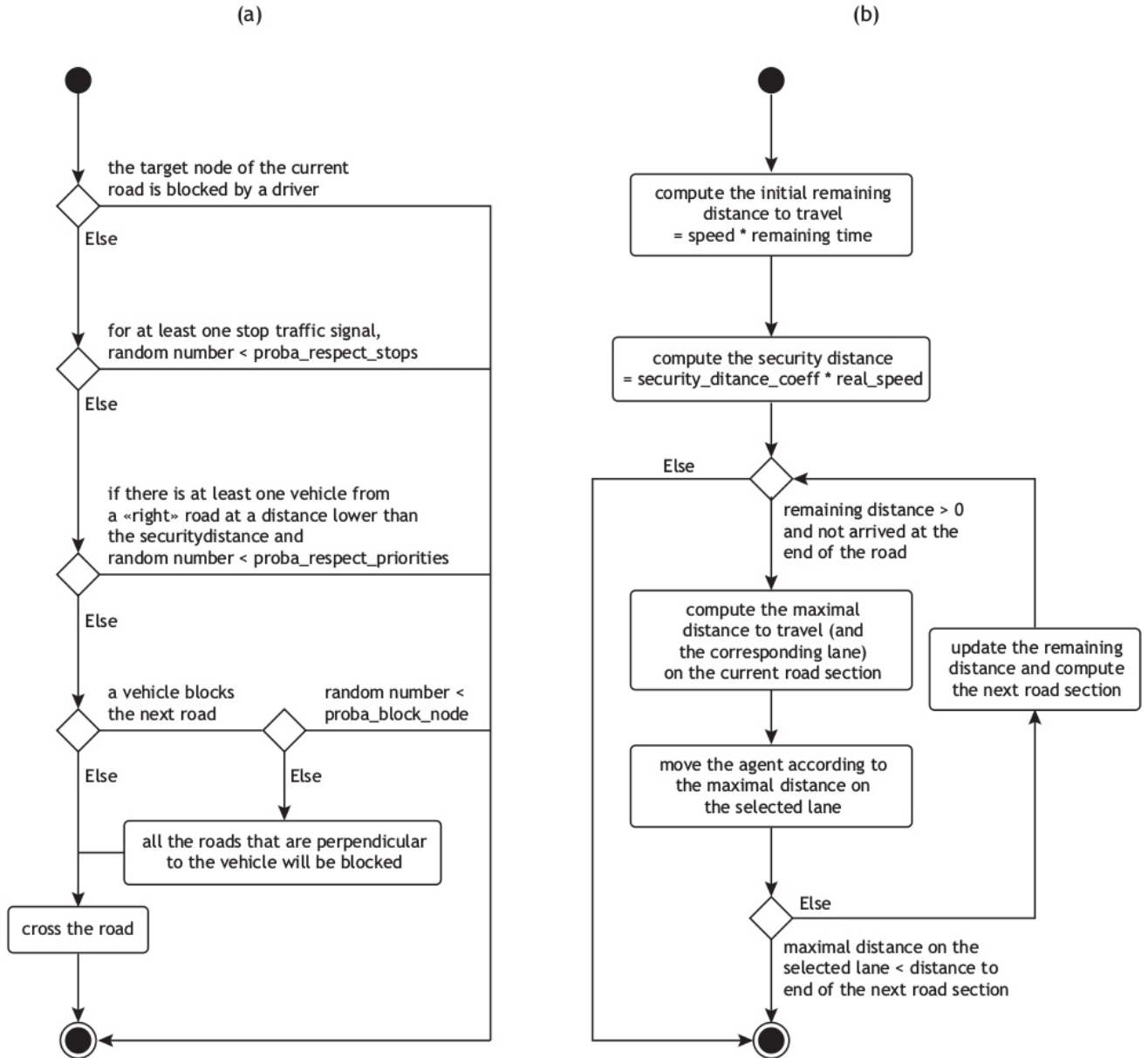


FIGURE 2.6 – (a) Franchissement d'une intersection (cas où le conducteur conduit à droite) et (b) déplacement sur une route (Czura *et al.*, 2015).

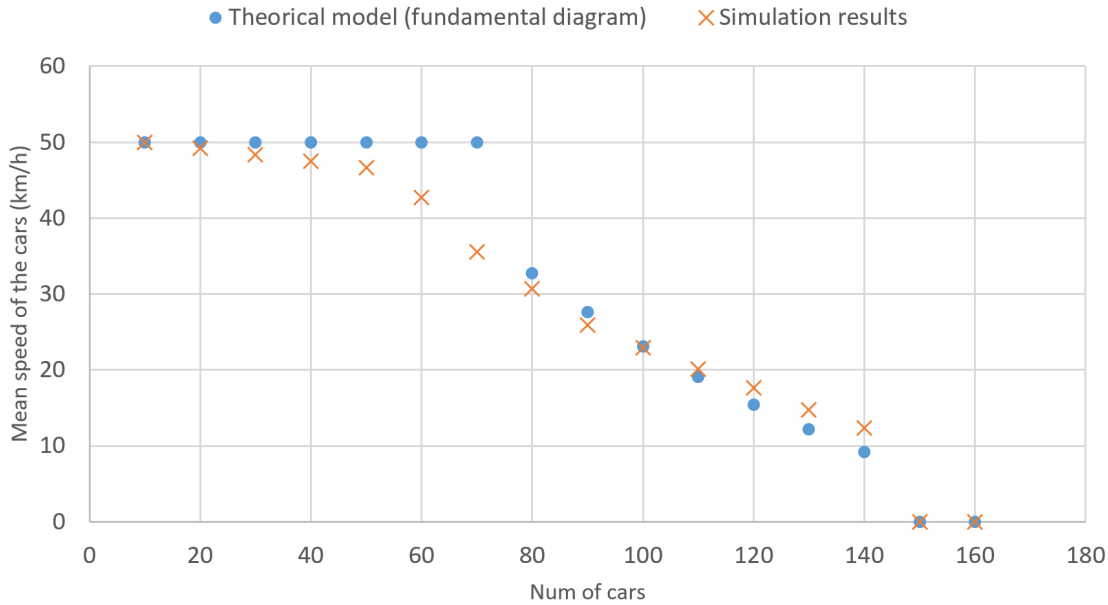


FIGURE 2.7 – Vitesse moyenne en fonction du nombre de véhicules pour le modèle théorique (diagramme fondamental) et pour le modèle de conducteur proposé (simulation)

tation sur un réseau routier simple composé de 2 routes de 500m connectées en un circuit : nous avons simulé le trafic en ajoutant progressivement de plus en plus de véhicules avec une vitesse souhaitée de conduite de 50 km/h. Après 2 minutes de temps simulé utilisées pour l'initialisation de façon à éviter les biais dû à l'ajout de véhicules (ajoutés de façon homogène sur les routes), nous avons calculé la vitesse moyenne des véhicules sur le circuit pendant 1 minute, puis ajouté 10 nouveaux véhicules sur le circuit et ainsi de suite jusqu'à arriver à saturation du réseau routier.

La Figure 2.7 présente un graphique décrivant les vitesses obtenues en fonction du nombre de véhicules pour le modèle théorique (diagramme fondamental) et pour le modèle de conducteur proposé (simulation). Le graphique montre que notre modèle est bien capable de reproduire les différentes phases du diagramme fondamental et reste relativement proche des résultats obtenus avec le modèle théorique, ce qui tend à le valider.

### 2.2.4 Application

L'outil que j'ai développé est directement intégré et distribué avec la plate-forme GAMA depuis sa version 1.6. Il offre des fonctionnalités complémentaires au *moving skill* pour la simulation du trafic routier.

Il a depuis été utilisé dans différents projets : le premier est bien sûr le projet MOSAIIC (Czura *et al.*, 2015) pour lequel il a été conçu. Il est également utilisé dans le projet ESCAPE (Chapuis *et al.*, 2018a; Daudé *et al.*, 2019). Enfin, il a été utilisé dans le cadre d'une expertise menée par le Boston Consulting Group sur l'impact des véhicules autonomes sur le trafic (BCG, 2017).

## 2.3 Spatialisation de populations synthétiques

### 2.3.1 Contexte

Une question importante dans la modélisation à base d'agents, et plus particulièrement dans le cas de développement de modèles descriptifs utilisant intensivement des données, est le problème de l'initialisation des agents. En effet, dans ces modèles, les comportements des agents sociaux (par exemple, des individus, des ménages ou des institutions) sont fortement déterminés par leurs attributs, ainsi que par leur localisation dans les mondes artificiels qu'ils peuplent. Dans ce contexte, pouvoir générer des populations synthétiques d'agents localisés conformes aux données disponibles sur les populations réelles est devenu une nécessité et une préoccupation pour la plupart des modélisateurs.

Certain travaux récents ont déjà proposé des méthodes pour générer ce type de population ([Harland et al., 2012](#); [Cornelis et al., 2013](#); [Swarup et Marathe, 2016](#)). Cependant, ces méthodes restent sous-utilisées en raison de la difficulté à les mettre en oeuvre avec des données mal formatées : par exemple, le générateur Web simSALUD ([Kosar et Tomintz, 2015](#)) requiert que les données d'entrée soient formatées dans un standard précis. Dans le même ordre d'idées, le générateur SPEW ([Gallagher et al., 2017](#)) en R requiert que les données d'entrée soient sous la forme de tableaux de données au format IPUMS. Une autre limite de ces méthodes est que les attributs spatiaux des agents sont traités de façon identique aux autres attributs, typiquement au travers d'un attribut unique spécifiant la zone (une ville, un quartier, etc.) dans laquelle l'agent est localisé. En effet, les informations spatiales proviennent généralement de la même source de données : des recensements ou des enquêtes. Cependant, cette information spatiale n'est généralement pas assez précise pour de nombreux cas de simulation. Par exemple, beaucoup de modèles de mobilité résidentielle fonctionnent à l'échelle du bâtiment ([Antoni et al., 2016](#)) et exigent donc l'attribution d'un bâtiment spécifique à chaque ménage et non pas seulement à une ville ou à un quartier. Il est donc souvent nécessaire de spécifier ou d'extrapoler une localisation à partir de données spatiales. Dans ce cas, le problème ne consiste pas simplement à déduire une distribution interne d'attributs démographiques, mais à estimer ce qui détermine réellement la localisation spatiale des personnes. Comme l'a souligné [Anderson et al. \(2014\)](#), obtenir une estimation fiable de la population à une résolution donnée constitue un défi de taille.

L'un des rares outils génériques permettant à la fois de générer et de spatialiser des populations est SPEW ([Gallagher et al., 2017](#)). Cependant, même avec cette bibliothèque open-source, un échantillon de données d'une partie de la population est nécessaire pour générer une population synthétique fiable. En ce qui concerne la localisation de la population, SPEW propose des méthodes simples pour localiser uniformément des entités dans une région ou le long des routes, mais ne permet pas de tirer parti des multiples couches d'informations dont le modélisateur peut disposer.

C'est pour remédier à ce manque de flexibilité que le projet ANR Gen\* (2013-2017) a été lancé. Ce projet, qui était coordonné par Alexis Drogoul (DR IRD, UMI UMMISCO), avait justement pour objectif de développer une bibliothèque open-source en Java permettant aux modélisateurs de générer une population synthétique réaliste, même avec des données au format non classique, de la localiser à l'échelle requise à partir d'une ou plusieurs couches de données géographiques et de pouvoir créer des liens entre les individus la composant (réseaux sociaux). Un autre enjeu était de rendre cette bibliothèque facilement utilisable par les modélisateurs, en particulier au travers d'une extension pour la plate-forme GAMA permettant d'utiliser les fonctions de Gen\* dans des modèles.

Étant responsable scientifique de ce projet pour l'UMR IDEES, mon rôle était plus particulièrement

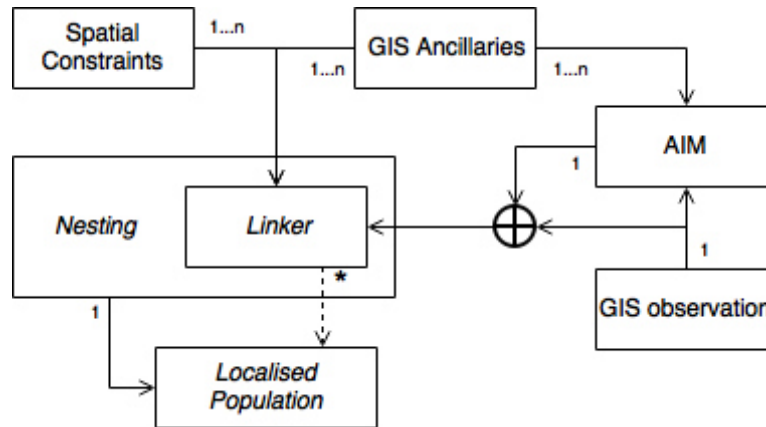


FIGURE 2.8 – Diagramme conceptuel de *SpII* (Chapuis *et al.*, 2018b).

de travailler sur la spatialisation des populations Ce travail a été réalisé en collaboration avec Kevin Chapuis, en post-doctorat sur Gen\* à l'UMR IDEES entre 2016 et 2017 et dont j'assurais l'encadrement. La suite de cette partie propose une présentation synthétique de ce travail qui a fait l'objet d'une publication en 2018 dans le journal IJGIS (Chapuis *et al.*, 2018b).

### 2.3.2 Le module *SpII* de la bibliothèque Gen\*

La bibliothèque Gen\* est composée de quatre modules : *Gospl* permet de générer la population, *SpII* permet de localiser les entités de la population générée et enfin, *Spin* permet de connecter des entités en un réseau social. Bien que les modules soient indépendants, ils reposent tous sur le module *Core*, qui définit les abstractions d'ordre supérieur pour les populations, les entités, leurs attributs et les valeurs associées.

*SpII* (Synthetic population localization library) est le module sur lequel j'ai été le plus impliqué ; il permet de spatialiser une population synthétique définie par *Gospl* (générée à l'aide des algorithmes fournis par *Gospl* ou chargée directement à partir d'un fichier csv/xlsx). La configuration du processus de localisation de la population est définie via un fichier de configuration qui harmonise les données d'entrée. Ces données doivent au moins contenir un fichier géographique spécifiant les objets géographiques sur lesquels *SpII* localisera les entités. Il peut également contenir d'autres données, telles que des données de "mapping", afin de créer un lien entre des entités de la population et des objets géographiques, par exemple le nombre de personnes par région administrative. L'utilisateur peut aussi fournir d'autres données utiles au processus de localisation, telles que des fichiers raster d'utilisation des sols ou des fichiers de routes. En plus de la localisation des entités, *SpII* permet de relier chaque entité à un ensemble d'objets géographiques (lieu de travail, écoles, etc.).

La figure 2.8 décrit le processus conceptuel sous-jacent à *SpII*. Le principal processus est celui de *nesting* qui consiste à relier chaque entité de la population avec un *nest*, c'est-à-dire un objet géographique dans lequel l'entité sera située, et à définir une localisation dans ce *nest*. Nous avons fait le choix de localiser explicitement les entités dans des objets spatiaux afin que chaque utilisateur puisse définir l'emplacement le mieux adapté aux besoins du modèle : les *nest* peuvent être des bâtiments, des cellules d'un fichier raster ou même simplement un polygone représentant les limites du système étudié.

Afin de faciliter le processus de localisation, *Spill* permet de générer une carte de contingence ou une carte de densité, à partir de données d'entrée brutes (*GIS observation*) ou en utilisant une méthode d'interpolation spatiale (*AIM*), par exemple une régression linéaire. *Spill* permet également de définir des contraintes spatiales sur la localisation des entités : par exemple, une distance maximale par rapport aux routes ou à un point d'intérêt (POI). Une autre possibilité offerte par *Spill* est de pouvoir choisir la fonction de distribution spatiale à utiliser pour le choix d'un *nest* pour chaque entité. *Spill* permet d'ailleurs d'utiliser ces contraintes et distributions spatiales pour lier une entité à des objets spatiaux, par exemple pour choisir un lieu de travail ou une école pour chaque entité. Enfin, *Spill* permet d'exporter la population spatialisée dans un fichier SIG (shapefile).

Notre approche de localisation est conçue pour être aussi adaptable que possible à tous les cas d'études. Ainsi, la seule étape obligatoire est la spécification d'un fichier de données géographiques (vecteur ou raster) permettant de définir les *nests* possibles pour localiser les entités. Les paragraphes suivants décrivent les différentes étapes du processus de localisation.

**Appariement entre la population et l'espace :** cette étape facultative permet de répartir chaque entité dans des zones spatiales prédéfinies. Un exemple typique est la répartition d'entités dans des zones administratives à partir d'un identifiant associé aux entités (qui a pu être généré par *Gospl* par exemple). Pour être utilisé, ce processus requiert d'avoir un fichier d'entrée qui décrit des zones spatiales et un moyen de connecter les entités de la population à ces zones.

**Interpolation spatiale :** *Spill* permet d'affiner la localisation des entités en utilisant des méthodes d'interpolation spatiale telles que la régression. L'idée centrale est d'estimer une distribution spatiale plus précise en utilisant deux entrées : d'une part, une observation à expliquer (par exemple, des données sur le nombre d'habitants par zone administrative) et d'autre part un ensemble de données explicatives (par exemple des cartes d'occupation du sol). Ce processus permet de calculer une densité ou une contingence d'entités pour chaque objet spatial explicatif.

**Processus de nesting :** comme information minimale, le processus de *nesting* a besoin d'un fichier d'entrée qui décrit les objets géographiques dans lesquels localiser les entités. Si ce fichier est la seule contrainte imposée à l'algorithme, chaque individu de la population choisira au hasard un *nest* parmi tous ceux possibles en utilisant la fonction de distribution. Afin d'obtenir une spatialisation plus réaliste des entités, l'utilisateur peut définir des contraintes spatiales pour filtrer les *nests* possibles. *Spill* intègre trois types de base de contraintes spatiales : géométriques, de contingence et de densité. Si les contraintes ne permettent pas de trouver un *nest* approprié (sur-contraint), elles peuvent être relâchées. Plus précisément, pour chaque contrainte, un processus de relaxation peut être défini ainsi qu'une fonction de relaxation maximale qui définit jusqu'à quel point la contrainte peut être relâchée.

- *Contrainte géométrique :* elle spécifie que les entités doivent choisir un *nest* se trouvant dans une géométrie donnée. Par exemple, si nous avons des informations sur la zone administrative dans lequel vit chaque entité (via un attribut de zone administrative défini par *Gospl* par exemple), cette contrainte peut être utilisée pour limiter la liste des *nests* possibles à ceux situés à l'intérieur de la zone administrative. S'il n'y a pas de *nests* possibles à l'intérieur de la géométrie de référence, la contrainte pourra être relâchée afin d'inclure également les *nests* proches de la géométrie de référence. La notion de proximité dépend d'une variable de distance qui augmentera à mesure que la contrainte sera relâchée jusqu'à atteindre un seuil maximal défini

par l'utilisateur. Une contrainte géométrique peut également être utilisée pour spécifier que les entités doivent être localisées proche des routes ou d'un point d'intérêt (POI).

- *Contrainte de contingence* : elle spécifie le nombre maximum d'entités pour une ou plusieurs géométries. Généralement, cette contrainte se présente sous la forme d'un décompte démographique par zone géographique. Cette contrainte filtrera les *nests* possibles pour ne conserver que ceux pour lesquels la contrainte de contingence maximale n'est pas encore atteinte. Si tous les *nests* sont pleins, la relaxation augmentera petit à petit la capacité des *nests* jusqu'à une valeur maximale définie par l'utilisateur.
- *Contrainte de densité* : elle spécifie la densité maximale d'entités pour une ou plusieurs géométries. Cette contrainte fonctionne comme la contrainte de contingence avec le même type de processus de relaxation.

En fonction de l'application, il peut s'avérer plus intéressant d'essayer de relâcher d'abord la contrainte géométrique avant la contrainte de contingence et inversement. Pour spécifier cette notion de priorité de relaxation, l'utilisateur peut définir une priorité pour chaque contrainte. Un processus d'ordonnement est également effectué lors de la recherche d'un *nest* approprié, pour lequel chaque contrainte filtrera successivement la liste des *nests* possibles. Lorsque plusieurs *nests* satisfont à toutes les contraintes pour une entité, *Spill* utilise une fonction de distribution pour en choisir un. *Spill* fournit à l'utilisateur plusieurs fonctions de distribution prédéfinies :

- *Distribution uniforme* : (fonction de distribution par défaut) spécifie que tous les *nests* ont la même probabilité d'être choisis.
- *Distribution selon l'aire* : spécifie que la probabilité qu'un *nest* soit choisi est proportionnelle à sa surface.
- *Distribution selon les capacités* : spécifie que la probabilité qu'un *nest* soit choisi est une fonction linéaire de sa capacité, c'est-à-dire le nombre d'entités que le *nest* peut encore accueillir (plus sa capacité est élevée, plus il a de chance d'être choisi). La capacité du *nest* est mise à jour (décrémentée) chaque fois qu'il est choisi.

Lorsque chaque entité a été associée à un *nest*, la procédure suivante consiste à lui attribuer une localisation précise (coordonnées  $x, y$ ) dans ce *nest*. *Spill* propose deux fonctions de base pour le faire : centroïde et aléatoire. La fonction *centroïde* récupère le centroïde de l'objet. La fonction *aléatoire* récupère, quant à elle, un point aléatoire dans l'objet (ou dans une zone proche de l'objet).

À la fin du processus de localisation, chaque entité de la population synthétique a un *nest* - c'est-à-dire un objet géographique - et des coordonnées  $(x, y)$  précises à l'intérieur de celui-ci.

L'architecture de Gen\* a été pensée de sorte à être modulaire : si la liste des contraintes, des fonctions de distribution ou des processus de localisation prédéfinis ne couvre pas les besoins de l'utilisateur, il est simple pour celui-ci de définir ses propres types de contraintes, de fonctions de distribution ou de processus de localisation en utilisant le langage Java.

**Lien entre entités et objets géographiques** : comme SPEW, Gen\* permet de relier un ensemble d'objets géographiques à chaque entité. Ce processus de liaison est très proche du processus de localisation : le principe est de choisir pour chaque type de lien (par exemple, lieu de travail, école) un objet géographique parmi un ensemble de lieux possibles. Pour ce faire, *Spill* utilise une combinaison de contraintes spatiales et une fonction de distribution spatiale. Cependant, comme il est souvent né-



cessaire de lier des entités avec des objets géographiques proches de leur lieu d'habitation, *Spill* offre la possibilité de définir une distribution en fonction de la distance entre la localisation de l'entité et les objets géographiques. Gen\* fournit plusieurs fonctions de distribution prédéfinies supplémentaires spécifiquement dédiées à ce processus :

- *Distribution basée sur la distance* : spécifie que la probabilité qu'un objet géographique soit choisi par une entité est une fonction linéaire négative par rapport à la distance qui les sépare.
- *Modèle gravitaire* : spécifie que la probabilité qu'un objet géographique soit choisi par une entité dépend de la masse de l'objet géographique divisée par la distance (à une puissance donnée) qui les sépare. Par défaut, la distance correspond à la distance euclidienne entre l'entité et l'objet géographique. Deux fonctions de base sont proposées pour la masse : la somme des distances entre l'objet géographique et toutes les entités et le nombre d'entités à une distance donnée de l'objet géographique.

### Application pour la ville de Rouen

Pour illustrer l'utilisation de Gen\*, nous proposons un exemple concernant la ville de Rouen. Dans cet exemple, nous avons généré une population avec Gen\* à partir des données disponibles auprès de l'INSEE. Dans ces données la localisation des individus est donnée sous la forme d'un nombre d'habitants par code IRIS, l'unité statistique spatiale la plus précise que l'on puisse obtenir en France. Nous considérons trois scénarios correspondant à différents types de données disponibles :

- localisation à l'intérieur des géométries des IRIS (polygones)
- localisation le long des routes (polylignes)
- localisation à l'intérieur des bâtiments, avec des informations sur la géométrie des IRIS, des images satellites et des contraintes spatiales.

**Scénario 1 : localisation à l'intérieur des géométries des IRIS.** Dans le cadre de ce scénario, le processus de localisation se base exclusivement sur les données démographiques spatiales concernant le nombre d'habitants de chaque IRIS, illustré à la figure 2.9. Sur la base de ces informations, Gen\* effectue une mise en correspondance entre l'attribut donnant le code IRIS de chaque individu et les objets géographiques IRIS. Une fois cette correspondance effectuée, Gen\* donne pour chaque individu une localisation précise choisie aléatoirement dans l'IRIS à laquelle il est lié. La figure 2.10 montre un exemple de résultat obtenu. En termes de temps de calcul, ce processus de localisation pour les 110 688 habitants de la ville a pris moins de 3 secondes sur un ordinateur avec un processeur i7.

**Scénario 2 : localisation le long des routes.** Dans ce scénario, les individus doivent être localisés le long des routes. Nous utilisons donc, dans un premier temps, une fonction intégrée de Gen\* qui permet de calculer pour chaque objet géographique la zone se trouvant entre une distance minimale et une distance maximale par rapport à l'objet. Pour cet exemple, les individus sont situés à une distance comprise entre 2,0 et 10,0 mètres des routes. Une fois que ces géométries sont calculées pour toutes les routes, le processus de localisation normal de Gen\* est appliqué avec la fonction de distribution spatiale basée sur la surface. Cela nous permet de privilégier les routes les plus longues pour la sélection du *nest*. La figure 2.11 présente un exemple de résultat obtenu. Le temps de calcul était de 99 secondes, avec 87 secondes dédiées au calcul des zones entourant les 4700 routes.

**Scénario 3 : localisation dans les bâtiments avec une interpolation spatiale.** Dans ce dernier scéna-

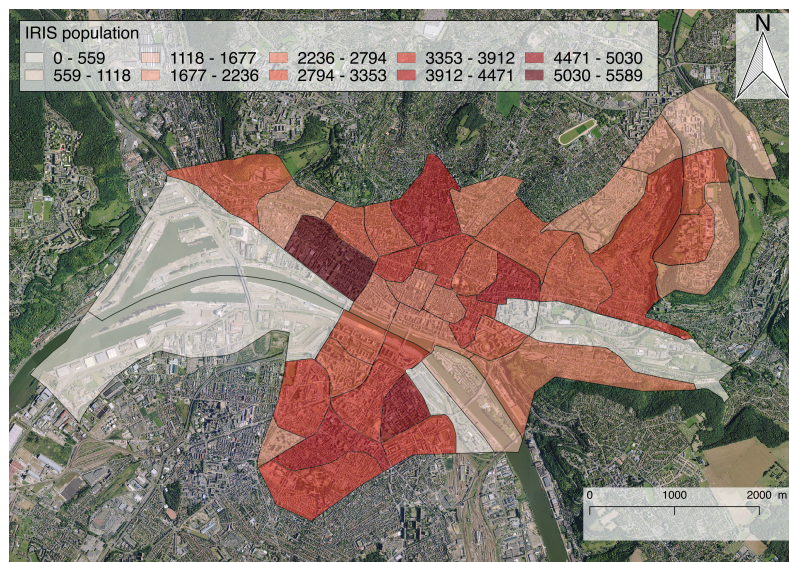


FIGURE 2.9 – Géométrie de chaque IRIS de la ville de Rouen (Chapuis *et al.*, 2018b).

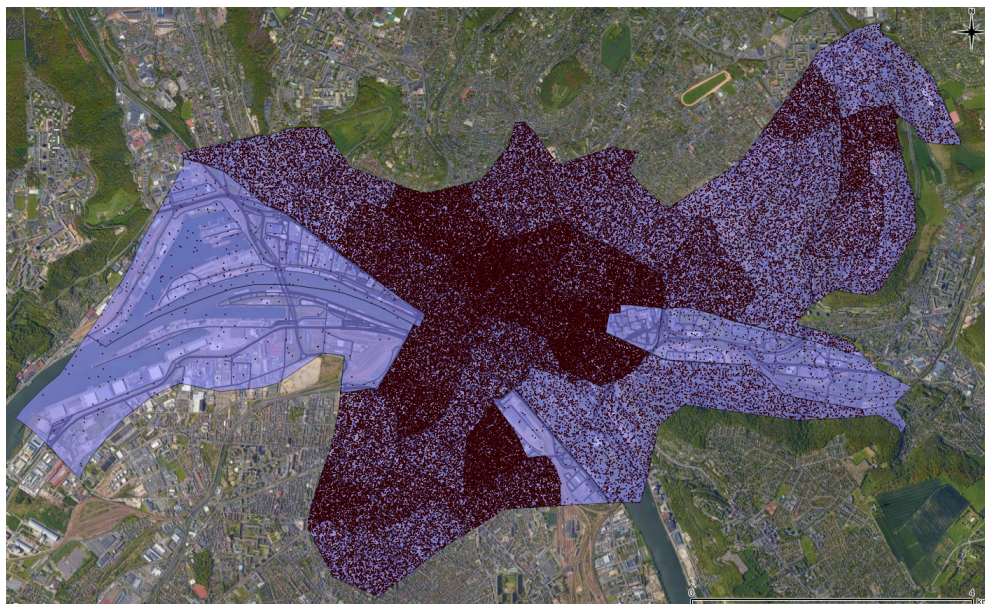


FIGURE 2.10 – Scénario 1 : localisation des individus dans la population de Rouen (Chapuis *et al.*, 2018b).



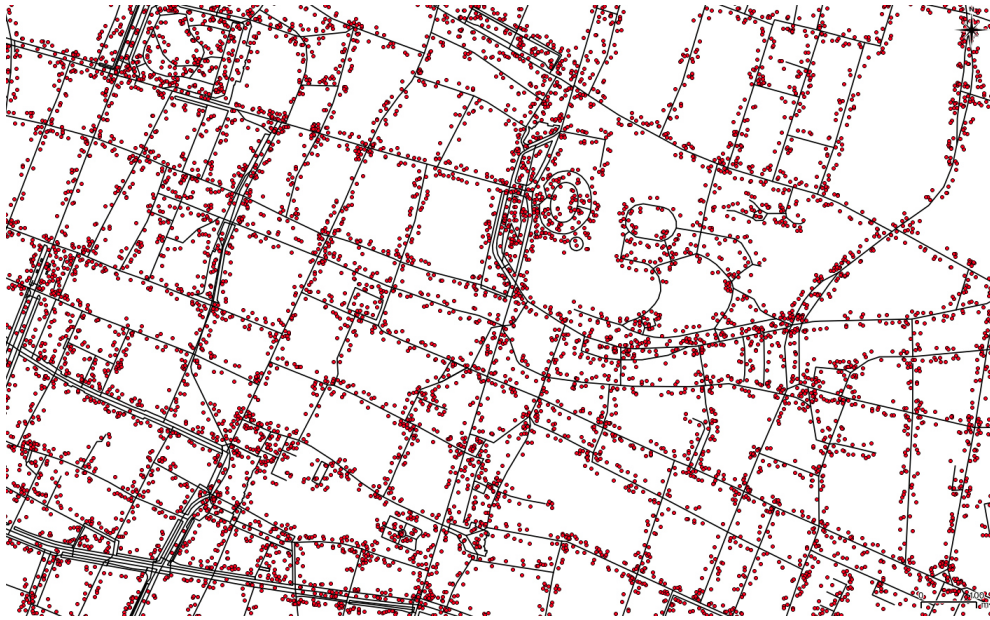


FIGURE 2.11 – Scénario 2 : localisation des individus dans une petite partie de la ville de Rouen (Chapuis *et al.*, 2018b).

rio, trois sources de données géographiques sont utilisées : des images satellite prétraitées d'utilisation des sols, le fichier des géométries des IRIS et le fichier des bâtiments. Comme pour le premier scénario, le processus de localisation débute en utilisant le fichier des IRIS pour déterminer la liste des individus résidant dans chaque IRIS. Ensuite, cette distribution spatiale est utilisée comme objectif d'un processus d'interpolation et les données d'utilisation des sols comme variables explicatives. L'interpolation spatiale se termine par le calcul d'une fonction qui permet à Gen\* de déterminer le nombre de personnes à affecter à chaque cellule du raster utilisé comme variable explicative. La carte 2.12 donne un aperçu des contingences résultantes pour les cellules. Pour chaque IRIS, les individus habitants dans celui-ci sont ensuite distribués de manière aléatoire dans les cellules appartenant à l'IRIS en utilisant une contrainte de contingence. A partir de cette distribution, Gen\* associe à chaque individu un bâtiment qui sera son *nest* en prenant en compte un nombre maximal de personnes par bâtiments et en relâchant peu à peu les contraintes si besoin. En résumé, le processus de localisation se compose de trois étapes : l'appariement entre entités et IRIS, l'interpolation spatiale pour calculer la densité dans les cellules et, enfin, la localisation de chaque entité dans les bâtiments. L'ensemble du processus a pris moins de 25 secondes pour être accompli.

### 2.3.3 Bilan sur *SpII*

*SpII* propose un outil flexible pour localiser une population. Pour comparer avec SPEW, *SpII* peut faire tout ce que fait SPEW, mais permet d'aller plus loin. En effet, la possibilité de définir des contraintes spatiales permet d'intégrer de nouvelles connaissances dans le processus de spatialisation. De plus, *SpII* permet d'utiliser des méthodes d'interpolation spatiale pour utiliser des informations provenant de différentes sources afin d'améliorer la spatialisation. Enfin, la généralité de l'approche de spatialisation proposée (indépendante du type d'objet géographique concerné) permet de l'étendre facilement par la définition de nouvelles contraintes spatiales et de nouvelles fonctions de distribution. Ainsi, Gen\* peut facilement être adapté aux besoins spécifiques des modélisateurs.



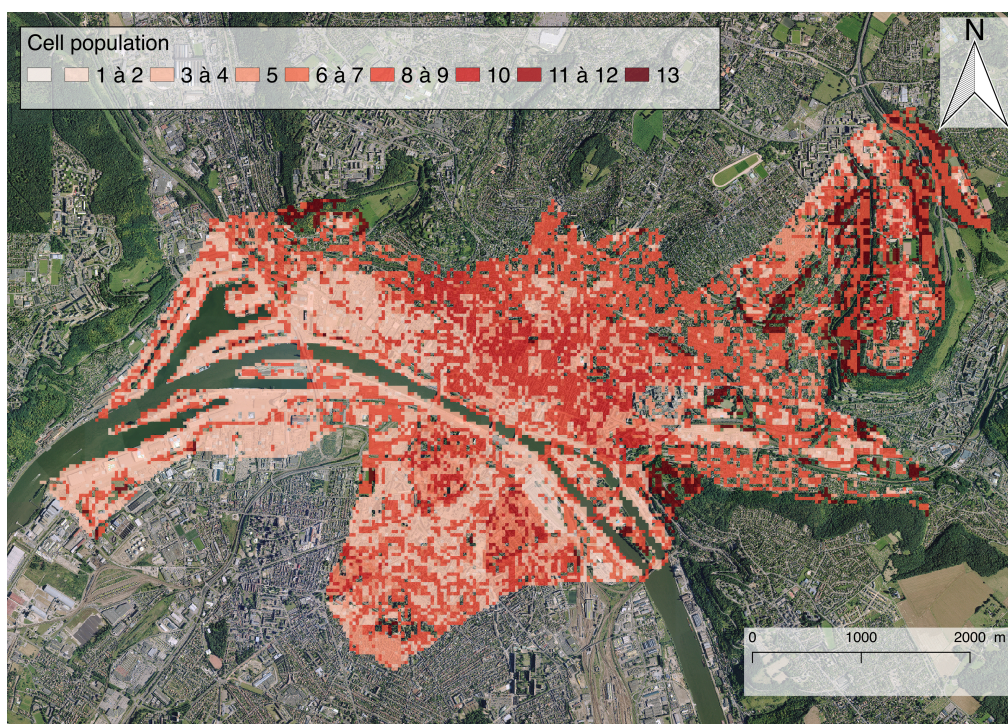


FIGURE 2.12 – Contingences résultantes du processus d’interpolation spatiale (Chapuis *et al.*, 2018b).



FIGURE 2.13 – Scénario 3 : localisation des individus dans une petite partie de la ville de Rouen (Chapuis *et al.*, 2018b).

Pour finir, nous avons développé un plug-in pour la plate-forme GAMA permettant d'utiliser Gen\* directement à partir de commandes en GAML. Cette extension a déjà pu être testée dans le cadre d'une formation donnée à Marrakech en février 2017.

## 2.4 Conclusion

Les travaux que j'ai effectués sur la dimension spatiale des simulations ont toujours occupé une place importante dans mes recherches. Ils ont en effet été particulièrement structurants et ont fortement guidé mes recherches depuis mon post-doctorat à Hanoi. Ils m'ont également permis de publier de nombreux articles et de me donner une place particulière au sein de la communauté des développeurs de GAMA.

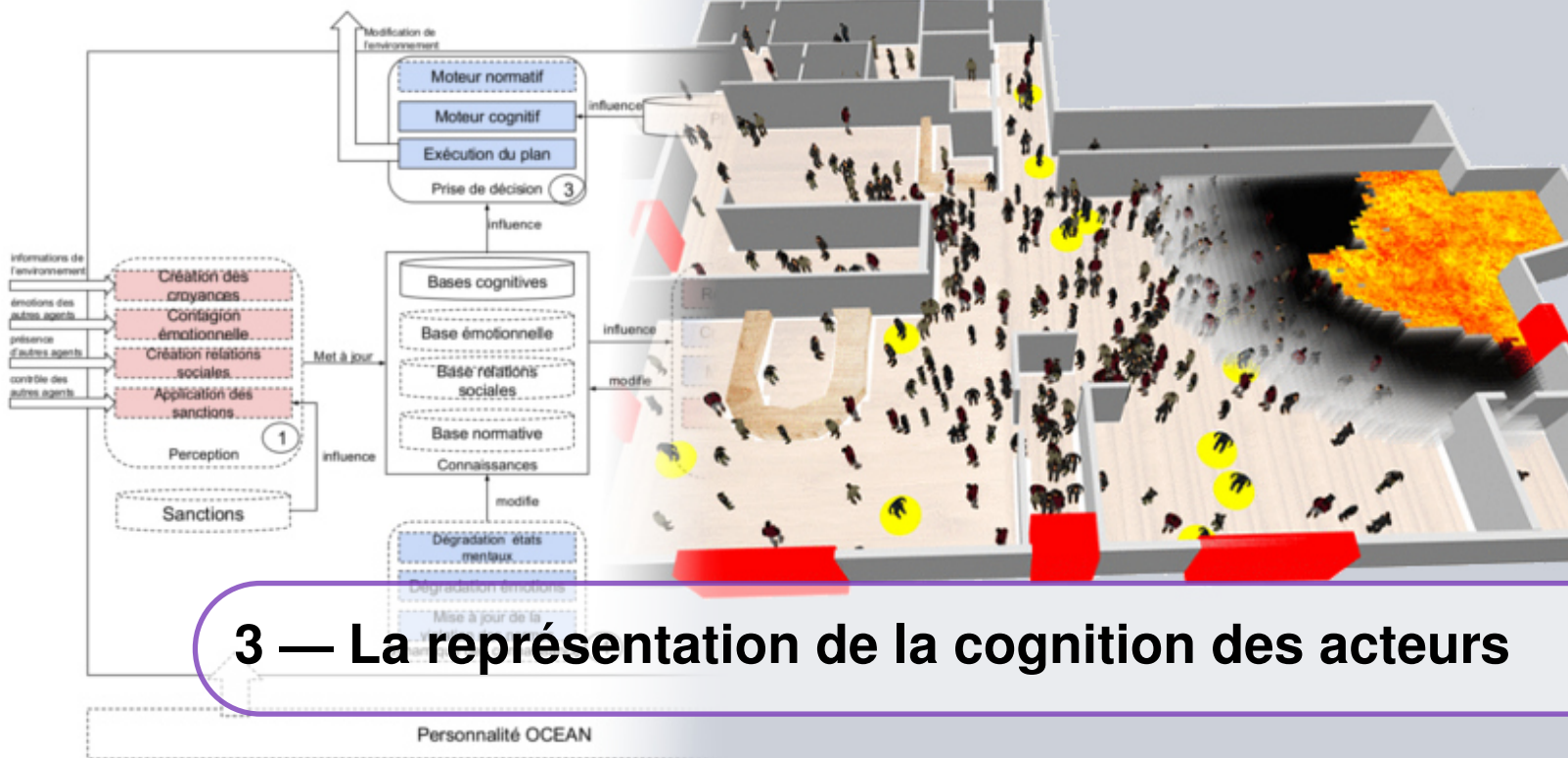
Si ces dernières années, à l'exception du projet Gen\*, mes recherches se sont plutôt orientées sur d'autres sujets (voir en particulier le chapitre suivant sur la modélisation de la cognition des acteurs), je continue activement et par petites touches à améliorer l'intégration de la composante spatiale dans GAMA : ajout de nouveaux formats de fichier pour l'import de données (e.g. le GML ajouté en 2018) ou l'export (j'ai intégré en septembre 2018 un export en KML), de nouveaux opérateurs (e.g. opérateurs de nettoyage de données et de calcul des N plus proches voisins ajoutés fin 2018) ou de nouveaux algorithmes (par exemple, algorithmes de plus court chemin dans des graphes ou des grilles). Je continue également à répondre régulièrement à des questions sur le forum de GAMA touchant à des problématiques spatiales.

Dans les prochaines années je compte poursuivre mes travaux sur cette question de façon épisodique en continuant à alimenter GAMA avec de nouvelles fonctionnalités et à optimiser celles existantes.

Un travail particulier, qui a déjà commencé depuis quelques temps et sur lequel je souhaite m'attarder un peu plus dans l'année à venir, est le développement du plug-in Gen\* pour GAMA. Ce plug-in est d'ores et déjà utilisable, même s'il ne permet pas de profiter de toute la richesse de Gen\*. Son enrichissement va également requérir un travail de réflexion sur la question de la description d'un processus de génération de populations spatialement structurées en GAML.

Un autre axe, qui va je pense prendre de l'importance dans le futur, concerne la 3D. Pour le moment, si la plate-forme GAMA gère bien la plupart des géométries 3D, elle ne permet pas d'effectuer des opérations spatiales sur ces dernières. Cette limitation est en partie due à la librairie que nous utilisons pour les calculs géométriques (JTS) qui ne gère pas la 3D. Passer à de la vraie 3D permettrait d'ouvrir GAMA à de nouveaux questionnements, mais nécessiterait, si l'on souhaite atteindre des temps de calcul acceptables, un important travail de réflexion sur la gestion de l'espace dans GAMA.





### 3.1 Contexte général

Une étape clef du processus de modélisation à base d'agents réside dans la définition du comportement des agents. Cette étape peut être particulièrement complexe, notamment lorsqu'il s'agit de modéliser la prise de décision d'entités douées de capacités cognitives et sociales élevées comme des êtres humains. Aujourd'hui encore, la plupart des travaux cherchant à modéliser cette prise de décision utilisent des modèles de comportement très simples (Sun, 2007). Si l'utilisation de tels modèles peut être tout à fait justifiée pour répondre à certaines questions, elle peut s'avérer limitée lorsqu'il s'agit de reproduire de façon crédible certains phénomènes. Ainsi, des travaux prônent l'utilisation d'agents plus "cognitifs", c'est-à-dire possédant des capacités de délibération et sociales élevées (Sun, 2006). Ce point de vue est également partagé par les tenants du principe EROS (Enhancing Realism Of Simulation) (Jager, 2017), qui promeut l'intégration de théories psychologiques dans la définition du comportement des agents de façon à favoriser la crédibilité des résultats des simulations.

Pour aborder ces questions, plusieurs architectures d'agents, couvrant chacune différents aspects du comportement humain, ont été proposées (voir (Balke et Gilbert, 2014) pour une description des architectures majeures). Parmi celles-ci, une famille d'architectures a connu une grande popularité ces dernières années : il s'agit des architectures fondées sur le modèle de cognition BDI (Belief Desire Intention) (Bratman, 1987). Le principe de base du modèle de décision BDI est de considérer qu'un agent possède trois bases d'états mentaux sur lesquels il va s'appuyer pour prendre une décision :

- *Croyances* : représentent ce que l'agent sait du monde.
- *Désirs* : représentent les états du monde que l'agent souhaite voir se réaliser.
- *Intentions* : représentent les états du monde que l'agent s'emploie à voir se réaliser.

Un moteur de raisonnement va se charger de définir, en fonction des croyances et désirs de l'agent, les intentions de l'agent ainsi que le plan d'action mis en œuvre.

Ce modèle théorique se trouve au coeur de différentes architectures (e.g. (Howden et al., 2001; Pokahr

*et al.*, 2005)), souvent appelées architectures BDI. Si des travaux ont montré l'intérêt de ces architectures BDI pour la simulation (Adam et Gaudou, 2016), ces dernières sont encore rarement utilisées par les modélisateurs. En effet, la plupart des architectures BDI ont été implémentées dans des plateformes spécifiques difficilement appréhendables par les modélisateurs.

Pour faire face à cette difficulté, une approche consiste à directement intégrer une architecture d'agents cognitifs dans les plateformes utilisées par les modélisateurs. Ainsi, Sakellariou *et al.* (2008) ont proposé une architecture d'agents cognitifs intégrée dans Netlogo. Celle-ci, qui a avant tout un objectif éducatif, se révèle assez limitée. Par exemple, elle ne permet pas de prendre en compte les croyances de l'agent dans le moteur de raisonnement.

C'est donc dans la perspective de répondre à ce besoin des modélisateurs que j'ai développé ces dernières années plusieurs architectures pour la plate-forme GAMA. La suite de ce chapitre présente les 2 principales architectures développées. La première, qui est basée sur la théorie des fonctions de croyance, a été développée dans le cadre du projet MAELIA pour modéliser les processus de décision des agriculteurs. La seconde, développée dans le cadre du projet ANR ACTEUR, se base sur le modèle de cognition BDI et intègre des dimensions à la fois cognitives, émotionnelles et sociales.

## 3.2 Utilisation de la théorie des fonctions de croyance

### 3.2.1 Contexte

Je suis arrivé en 2010 en post-doctorat à l'IRIT (Université Toulouse 1) pour travailler sur le projet MAELIA qui était coordonné par Pierre Mazzega (DR IRD, GET) portant sur l'évaluation de l'impact de différentes réglementations sur les ressources en eau. Mon travail dans ce projet a porté sur la modélisation du comportement des agriculteurs, en particulier sur la façon de représenter leurs décisions sur les choix de rotation de cultures et de conduite d'exploitation (semis, récolte, irrigation, etc.). Ce travail a été réalisé en étroite collaboration avec des chercheurs de différentes disciplines (informatique, agronomie et hydrologie). Dans le but de proposer un modèle de décision facilement partageable au sein de l'équipe, j'ai fait le choix de proposer une architecture reposant sur une approche de décision multicritère. En effet, les différents partenaires du projet avaient tous l'habitude d'utiliser ce type d'approche. De plus, cette approche se prêtait très bien à MAELIA : des travaux préalables avait montré l'importance de certains facteurs facilement traduisibles sous la forme de critères pour expliquer les décisions des agriculteurs en termes de choix de rotation de cultures (Dury *et al.*, 2012).

L'utilisation de ce type d'approche pour représenter la prise de décision d'un agent est très classique. Néanmoins, la plupart des modélisateurs tendent à utiliser des méthodes de décision multicritère très simples comme des moyennes pondérées, alors que la littérature dans ce domaine est riche en méthodes alternatives qui peuvent être plus adaptées pour représenter des processus de décision.

Les approches de décision multicritère peuvent être classifiées selon deux grandes catégories.

La première catégorie d'approches est connue sous le terme d'agrégation complète. Ces approches consistent à agréger tous les critères en un unique critère par le biais d'une fonction d'utilité. La décision prise est celle maximisant la fonction d'utilité. Un système de poids est souvent utilisé pour prendre en compte l'importance relative des différents critères dans la prise de décision. L'exemple le plus simple et le plus connu de ces méthodes est la moyenne pondérée. Il existe de nombreuses autres



méthodes, plus complexes, et fournissant des fonctions d'utilité plus fines (e.g. (Siskos *et al.*, 1984)). Cette catégorie d'approches pose le problème de la commensurabilité (Mena, 2000). Ainsi, chaque critère doit pouvoir être comparé directement aux autres, ce qui implique qu'ils soient définis sur une échelle comparable.

La seconde catégorie d'approches est connue sous le terme d'agrégation partielle. Ces approches consistent à comparer deux à deux chaque décision possible. La comparaison est basée sur l'analyse de chaque critère afin d'établir si l'une des deux décisions ne surclasse pas l'autre (e.g. (Behzadian *et al.*, 2010)). Contrairement aux approches d'agrégation complète, ces approches n'agrègent pas directement les critères entre eux mais uniquement le résultat de leur comparaison. Ainsi, ces approches permettent de comparer deux décisions en comparant chaque critère indépendamment des autres. Ces approches permettent donc de faire face au problème de l'incomparabilité des différents critères mais fournissent des résultats moins facilement interprétables que les approches d'agrégation complète (Mena, 2000).

J'avais eu l'occasion de travailler durant ma thèse sur l'utilisation de plusieurs méthodes de décision multicritère pour évaluer les connaissances d'un système de généralisation géographique. Cette expérience m'a amené à proposer d'utiliser pour MAELIA une approche basée sur la théorie des fonctions de croyance de Shafer (1976). Cette approche de décision, qui fait partie de la catégorie des approches d'agrégation complète, avait déjà été appliquée avec succès pour différentes applications (e.g. (Raymond et Mustière, 2008; Taillandier *et al.*, 2009b)). Elle avait pour avantage de pouvoir gérer les connaissances partielles ainsi que les conflits entre critères. La partie suivante détaille comment nous avons utilisé cette approche de décision multicritère pour modéliser le comportement d'un agent.

### 3.2.2 Formalisme proposée

#### Décision par la théorie des fonctions de croyance

Nous avons défini dans ce travail une architecture, ou plutôt un modèle de décision, permettant à un agent de choisir une décision parmi un ensemble fini d'alternatives possibles en utilisant la théorie des fonctions de croyance.

L'utilisation de cette théorie implique de définir un univers de référence  $\Theta$ , appelé cadre de discernement, qui définit l'ensemble des hypothèses pouvant potentiellement répondre au problème considéré. Pour notre application, où l'on cherche quelle est la décision la plus adaptée, le cadre de discernement est constitué de l'ensemble des décisions applicables. Nous notons  $A^i$ , une décision donnée :

$$\Theta = \{A^1, A^2, \dots, A^N\}$$

Ce cadre de discernement permet de définir un référentiel de définition, noté  $2^\Theta$ , contenant l'ensemble des combinaisons possibles d'hypothèses :

$$2^\Theta = \{\emptyset, \{A^1\}, \{A^2\}, \dots, \{A^1, A^2\}, \dots, \Theta\}$$

Chaque ensemble  $\{A^i, \dots, A^j\}$  est appelé "proposition" et représente le fait que la solution du problème est l'une ou l'autre des hypothèses de cet ensemble.

Cette approche de décision est basée sur l'utilisation de fonctions de croyance. Ces fonctions associent, à une proposition  $P \in 2^\Theta$ , une masse de croyance, notée  $m(P)$ , comprise entre 0 et 1. Cette masse de croyance représente le degré de croyance envers cette proposition. Les fonctions de croyance sont définies telles que :

$$\sum_{P \in 2^\Theta} m(P) = 1$$

Le processus de décision est composé de 3 étapes : (i) initialisation des masses de croyances, (ii) combinaison des critères et (iii) sélection de la décision qui semble la plus adaptée vis-à-vis des critères de choix et des connaissances de l'agent.

#### *Étape 1 - Initialisation des masses de croyances*

La première étape consiste à initialiser les masses de croyances pour chaque critère. Nous utilisons dans ce cadre les travaux de [Appriou \(1991\)](#), qui proposent de "spécialiser" les critères sur une hypothèse du cadre de discernement (une décision pour nous) afin que ceux-ci se prononcent uniquement, soit en faveur de l'hypothèse, soit en sa défaveur ou alors ne se prononcent pas par manque de connaissances sur l'hypothèse. Cette proposition permet de définir pour chaque hypothèse, c'est-à-dire chaque décision  $A^i$ , un sous-ensemble  $S^i$  de  $2^\Theta$  tel que :

$$S^i = \{\{A^i\}, \{\neg A^i\}, \Theta, \emptyset\}$$

- $\{A^i\}$  : cette proposition signifie que la décision  $A^i$  est la meilleure.
- $\{\neg A^i\}$  : cette proposition signifie que la décision  $A^i$  n'est pas la meilleure (i.e. la meilleure décision est l'une des autres décisions).
- $\Theta = \{A^1, A^2, \dots, A^N\}$  : cette proposition représente l'ignorance. Le ou les critères ne peuvent donner d'indication sur la meilleure décision.
- $\emptyset$  : cette proposition représente le conflit entre deux critères (les deux critères donnent des indications contraires concernant la décision).

L'initialisation des masses de croyances revient donc à calculer, pour chaque critère, les valeurs des masses de croyances pour les propositions  $\{A^i\}$ ,  $\{\neg A^i\}$  et  $\Theta$ . La proposition  $\emptyset$  a toujours une masse de croyances égale à 0 à l'initialisation des masses de croyances. De façon à calculer ces masses de croyances, des fonctions de croyance doivent être définies. Une fonction de croyance est une fonction associée à un critère qui renvoie une valeur réelle comprise entre 0 et 1 en fonction de la valeur du critère pour une hypothèse donnée.

*Étape 2 - Combinaison des critères*

Cette étape consiste à combiner les différents critères entre eux pour chaque décision  $A^i$  afin d'obtenir des masses de croyances synthétisant les connaissances issues de tous les critères. Nous nous intéressons pour cela aux intersections de propositions formulées par les différents critères.

Nous utilisons l'opérateur de fusion proposé par [Smets et Kennes \(1994\)](#) pour calculer les masses de croyances résultant de la combinaison de deux critères :

$$\forall P \in 2^\Theta, m_{C_1, C_2}^{A^i}(P) = \sum_{P' \cap P'' = P} m_{C_1}^{A^i}(P') \times m_{C_2}^{A^i}(P'')$$

Cet opérateur permet d'obtenir les quatre masses de croyances résultant de la fusion des deux critères  $C_1$  et  $C_2$  pour une décision  $A^i$  :  $m_{C_1, C_2}^{A^i}(A^i)$ ,  $m_{C_1, C_2}^{A^i}(\neg A^i)$ ,  $m_{C_1, C_2}^{A^i}(\Theta)$  et  $m_{C_1, C_2}^{A^i}(\emptyset)$

Il est associatif et commutatif et permet donc de recombinaison des masses de croyance déjà combinées à un troisième critère  $C_3$ .

Nous calculons ainsi les masses de croyance associées aux différentes propositions pour chacune des décisions. Nous obtenons, à la fin de cette étape, les masses de croyance combinées sur l'ensemble des critères pour chaque décision  $A^i$ .

*Étape 3 - Sélection de la meilleure décision*

Cette dernière étape consiste à déterminer parmi les propositions celle qui a le plus de chances d'être vraie (c'est-à-dire la meilleure décision).

Pour déterminer quelle est la meilleure proposition, nous proposons d'utiliser la probabilité pignistique ([Smets, 1989](#)), qui permet de répartir uniformément la masse de croyance associée à un sous-ensemble à chacun des éléments composant celui-ci. Cette probabilité est très largement utilisée pour les problèmes où l'on souhaite sélectionner une hypothèse simple et non un ensemble d'hypothèses ; cela correspond bien à notre cas d'étude où nous souhaitons sélectionner une décision unique.

La probabilité pignistique d'une proposition  $P$  est donnée par la formule suivante :

$$Proba(P) = \sum_{P' \subseteq P} m(P') \times \frac{|P' \subseteq P|}{|P|}$$

La probabilité pignistique représente l'utilité d'une décision : la décision choisie est donc celle qui a la probabilité pignistique la plus élevée.

**Application pour le processus de prise de décision d'un agent**

Comme présenté dans la section précédente, la théorie des fonctions de croyances permet de choisir une décision parmi un ensemble possible en fonction d'un ensemble de critères. Pour utiliser cette

théorie dans le cadre du processus de prise de décision d'un agent, le modélisateur doit définir plusieurs éléments :

- Un ensemble de critères permettant d'évaluer les différentes décisions possibles.
- Pour chaque critère : une fonction de croyance pour les hypothèses "cette décision est la meilleure", "cette décision n'est pas la meilleure", "l'ignorance".

De façon à réduire la complexité du calcul de la meilleure décision, il est possible de filtrer l'ensemble des décisions possibles pour ne conserver que celles qui sont sur le front de Pareto. Le front de Pareto correspond à l'ensemble des décisions qui ne sont pas dominées. Une décision est dominée par une autre si aucun critère n'a une valeur préférable (supérieure ou inférieure en fonction du type de critères) pour cette décision par rapport à l'autre décision.

Pour certains agents, il est également possible de diviser le processus de prise de décision en plusieurs sous-processus. Cette division peut être utilisée pour réduire la complexité du processus de décision ou pour utiliser différents ensembles de critères correspondant à différentes étapes du raisonnement. En effet, il est par exemple possible de diviser le processus de prise de décision en deux étapes : la première consiste à choisir un objectif général pour l'agent (exemple : manger, dormir) et la seconde consiste à choisir le meilleur endroit pour réaliser cet objectif.

### 3.2.3 Application pour le modèle MAELIA

Nous avons utilisé le modèle de décision décrit ci-avant pour représenter le comportement des agriculteurs dans la plate-forme MAELIA. L'enjeu était d'être en mesure de simuler les interactions entre utilisation agricole du sol et ressources en eau à une résolution spatiale et temporelle fine : la parcelle ou le groupe de parcelles et à un pas de temps journalier. Ce modèle se devait pour cela de représenter le choix d'assolement et les pratiques agricoles qui ont un effet majeur sur les prélèvements en eau. Il se devait également de simuler les propriétés émergentes des multiples interactions spatialisées entre agriculture et ressources en eau.

Je présente ici le modèle MAELIA tel qu'il était en 2012 ; il a connu depuis de nombreuses évolutions qui dépassent largement le cadre du travail que j'ai effectué sur le comportement des agriculteurs et que je ne présenterai donc pas dans ce document.

Le modèle était composé de treize types d'entités (Figure 3.1). Parmi celles-ci, sept étaient statiques et six étaient dynamiques.

Liste des entités statiques :

- *Agent exploitation* : un agent exploitation représente une exploitation agricole d'un agriculteur. A chaque exploitation est attribué un nombre d'UTH (Unité de Travail Humain) travaillant sur l'exploitation.
- *Agent îlot* : un agent îlot représente un ensemble de parcelles délimitées par des éléments naturels (e.g. haies, fossés) ou artificiels du paysage (e.g. route). Un îlot est considéré homogène en termes de météo et de sol.
- *Agent parcelle* : un agent parcelle représente une parcelle agricole au sein d'un îlot agricole. Les agriculteurs exploitent des cultures sur les parcelles.

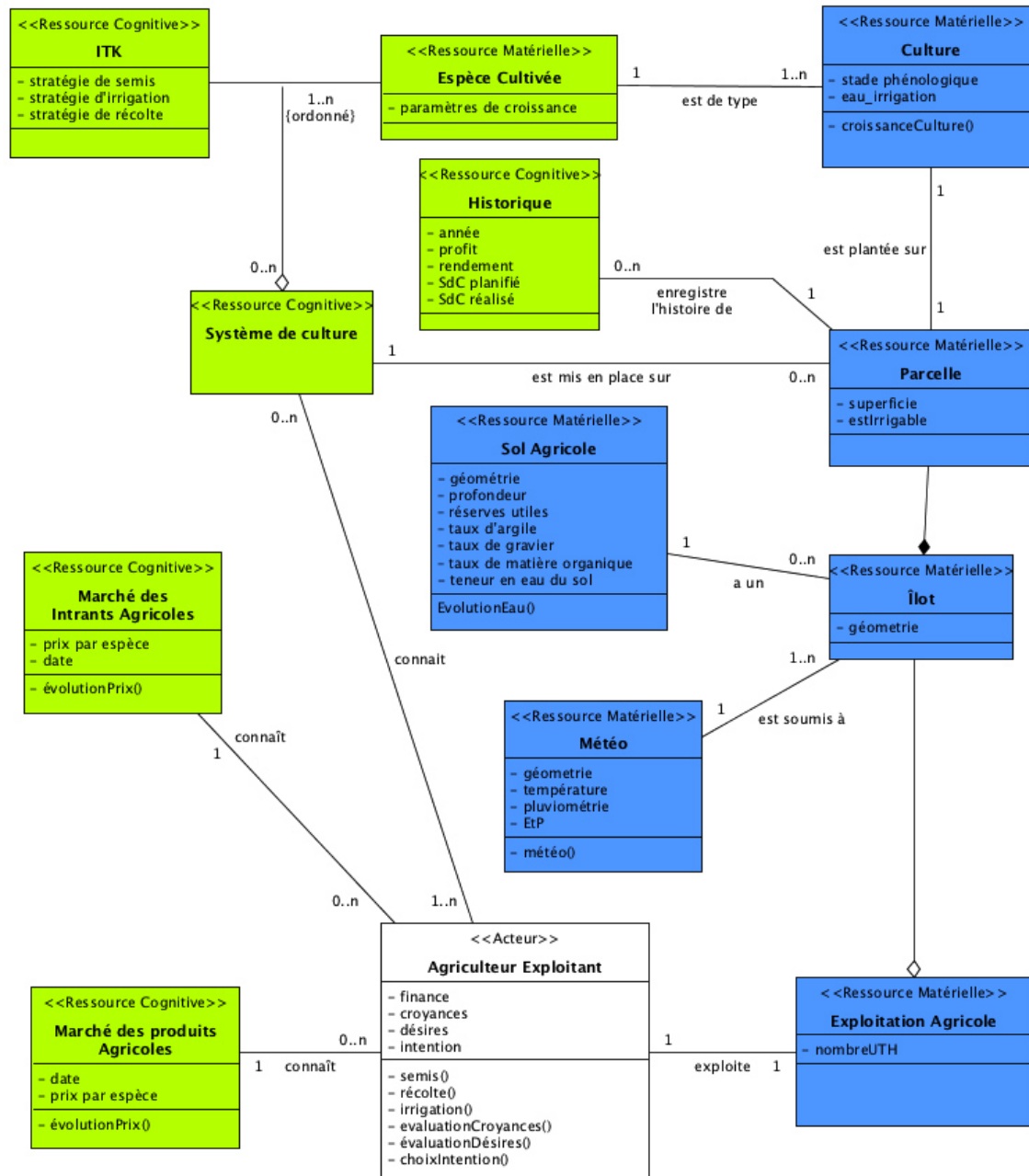


FIGURE 3.1 – Diagramme UML du modèle, en blanc l'agent agriculteur, en bleu les ressources matérielles et en vert les ressources cognitives (Therond *et al.*, 2014).

- *Agent espèce cultivée* : un agent espèce cultivée représente un type de culture possible (maïs, blé dur, etc.).
- *Agent ITK* : un agent ITK (Itinéraire Technique) correspond à un itinéraire technique de culture. Il définit en particulier l'espèce qui sera cultivée et les stratégies de semis, d'irrigation et de récoltes mises en place.
- *Agent système de culture* : un agent système de culture correspond à une séquence d'ITK : un ITK par année.
- *Agent historique* : un agent historique représente la mémoire d'un agent agriculteur. Il lui permet de conserver l'historique des événements qui se sont déroulés sur une parcelle durant 1 an.

Liste des entités dynamiques :

- *Agent agriculteur* : un agent agriculteur est un agent cognitif capable de planifier sur plusieurs années ses séquences de cultures et de les remettre en cause à tout moment si nécessaire. Il dispose également d'un comportement journalier (semier, irriguer, récolter) guidé par ses choix d'ITK.
- *Agent sol* : un agent sol représente une unité cartographique de sol. Ses caractéristiques sont utilisées par le processus de croissance des plantes. Les agents sol ont la charge de calculer la teneur en eau du sol.
- *Agent culture* : un agent culture représente une culture semée sur une parcelle. Ces agents gèrent le processus de croissance des plantes (calcul du rendement).
- *Agent météo* : un agent météo représente une zone homogène en termes de météo (pluie, température, évapo-transpiration). La météo évolue chaque jour.
- *Agent marché des produits agricoles* : un agent marché des produits agricoles représente les acheteurs des productions agricoles. Il définit pour chaque année les prix à la tonne de chaque espèce cultivée.
- *Agent marché des intrants agricoles* : un agent marché des intrants agricoles représente les vendeurs des intrants agricoles. Il définit pour chaque année les prix à l'hectare des intrants pour chaque espèce cultivée.

Dans ce modèle, un pas de simulation correspond à une journée. Nous détaillons par la suite les dynamiques des agents.

#### *Agent culture*

Les agents culture ont un seul comportement : calculer la production. La plate-forme MAELIA propose plusieurs modèles, plus ou moins simples et empiriques, pour réaliser ce calcul.

#### *Agent agriculteur*

Pour conduire leurs processus de décisions stratégiques et opérationnels, les agents agriculteurs utilisent un ensemble de 4 bases de données qui s'inspirent du paradigme BDI [Bratman \(1987\)](#) :

- *Croyances* : les croyances de l'agent représentent les connaissances que l'agriculteur a sur le monde. Ces informations peuvent être partielles, voire erronées. Les croyances sont utilisées pour calculer les valeurs des critères lors des choix d'assolement. Elles sont également utilisées dans le cadre du comportement opérationnel de l'agent. Dans notre modèle, ces croyances cor-

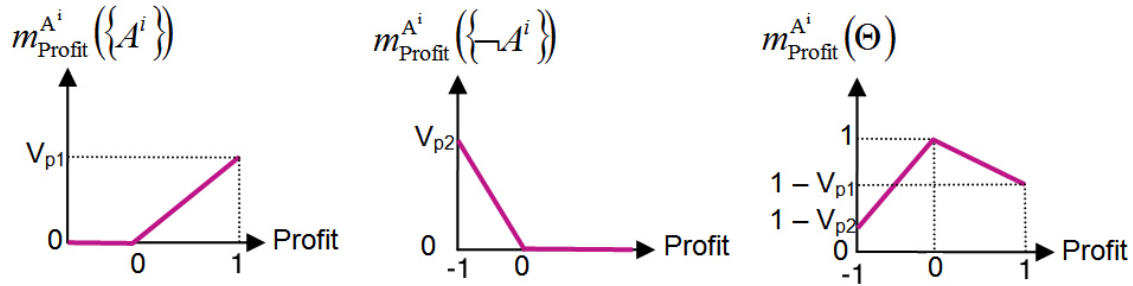


FIGURE 3.2 – Forme générale des fonctions de croyance pour le désir de maximisation du profit (Taillandier *et al.*, 2012b).

respondent aux variables suivantes : historiques des dernières productions, la météo des derniers jours, le prix des cultures, le coût des cultures et l'eau disponible.

- *Désirs* : les désirs représentent les critères qui vont guider le choix d'un assolement par l'agriculteur. Dans notre modèle, nous avons défini 4 critères tirés des travaux de Dury *et al.* (2012) : maximiser le profit, minimiser les risques de chute importante du revenu, minimiser la charge de travail et maximiser les similarités avec le dernier plan choisi.
- *Plans* : ensemble des assolements que l'agent agriculteur peut potentiellement réaliser. Un plan contient également des actions définissant le comportement opérationnel de l'agent (semier, irriguer et récolter) ainsi que des bases de règles permettant de définir sous quelles conditions ces actions seront exécutées (la conduite de culture). Enfin, un plan contient des règles permettant de le remettre en cause ou de le mettre à jour. En effet, un choix d'assolement peut être remis en cause à tout moment si des événements imprévus arrivent (par exemple l'impossibilité de semer sur une parcelle pour cause de trop fortes pluies).
- *Intention* : correspond à l'assolement choisi après exécution de la procédure de choix multicritère.

Le premier désir des agents agriculteurs est ainsi de choisir un plan qui leur permet de maximiser leur profit. Les fonctions de croyance que nous proposons pour ce critère sont présentées Figure 3.2. Ce critère est basé sur l'évaluation, en euros, du profit espéré pour l'ensemble des années concernées par les rotations. Ce profit tient compte du prix de vente des cultures (prix annuel du marché agricole), des charges opérationnelles liées à la conduite de chaque culture (semence, engrais, eau, etc.) et de l'expérience de l'agriculteur concernant les rendements qu'il a pu obtenir sur chacune des parcelles. Cette valeur est normalisée entre -1 et 1 par rapport au profit/déficit maximal obtenu sur l'ensemble des plans.

Le deuxième désir des agents agriculteurs est d'éviter les plans qui peuvent entraîner une forte diminution du profit espéré. Les fonctions de croyance que nous proposons pour ce critère sont présentées Figure 3.3. Ces fonctions dépendent de l'écart type du profit espéré. Cette valeur est également normalisée par rapport à l'écart type maximal obtenu sur l'ensemble des plans.

Un troisième désir concerne la quantité de travail. Les agents agriculteurs tendent à choisir un plan qui minimise le nombre de jours de travail. Les fonctions de croyance que nous proposons pour ce critère sont présentées Figure 3.4. Ces fonctions dépendent du nombre de jours de travail.



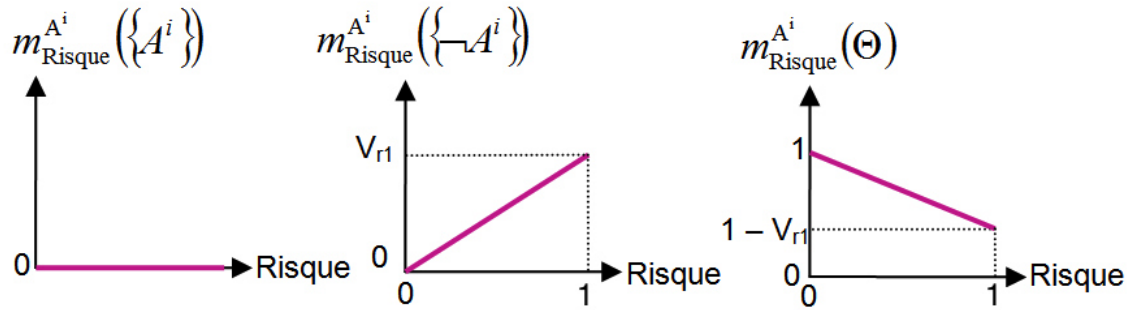


FIGURE 3.3 – Forme générale des fonctions de croyance pour le désir de minimisation du risque (Taillandier *et al.*, 2012b).

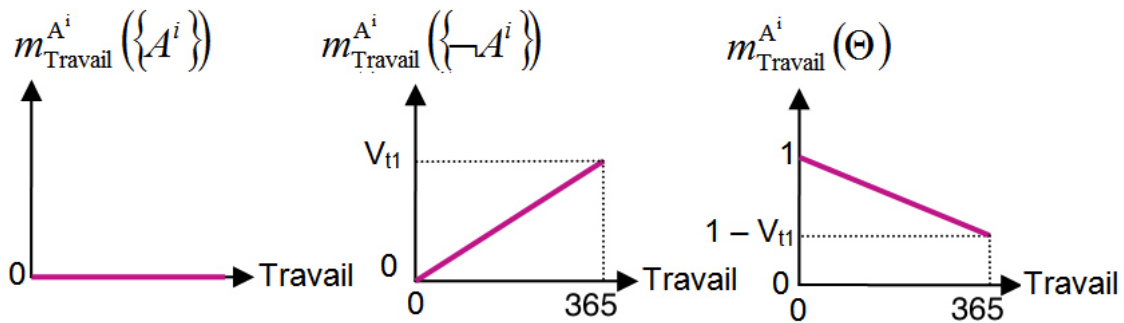


FIGURE 3.4 – Forme générale des fonctions de croyance pour le désir de minimisation de la charge de travail (Taillandier *et al.*, 2012b).

Le dernier désir est d'essayer de conserver ses habitudes. Les agents agriculteurs tendent ainsi à choisir le même plan (ou au moins des plans proches) d'année en année puisque changer d'assolement a un coût cognitif important. De plus, l'introduction d'un nouveau système de culture représente une prise de risque liée au manque de connaissances procédurales et empiriques sur celui-ci. Les fonctions de croyance que nous proposons pour ce critère sont présentées Figure 3.5. Ces fonctions dépendent du taux de similarité par rapport au dernier plan choisi ( $S$ ).

Comme mentionné préalablement, pour nos agents agriculteurs, un plan correspond à un choix d'assolement, c'est-à-dire à une prévision d'affectation à chaque parcelle cultivée d'un système de culture et, pour une année donnée, d'une culture de la séquence de culture correspondante. Un plan est aussi caractérisé par des actions et des règles de conduite de culture :

- *Semer* : met à jour la culture sur une parcelle (état semé) et les finances de l'agent agriculteur.
- *Irriguer* : met à jour le niveau d'eau apportée sur une parcelle et l'eau disponible pour l'agent agriculteur. Du fait de leurs contraintes en termes d'équipement d'irrigation et de débit instantané, les agriculteurs ne peuvent irriguer qu'une certaine surface par jour.
- *Récolter* : met à jour la culture sur une parcelle (état récolté) et les finances de l'agent agriculteur.

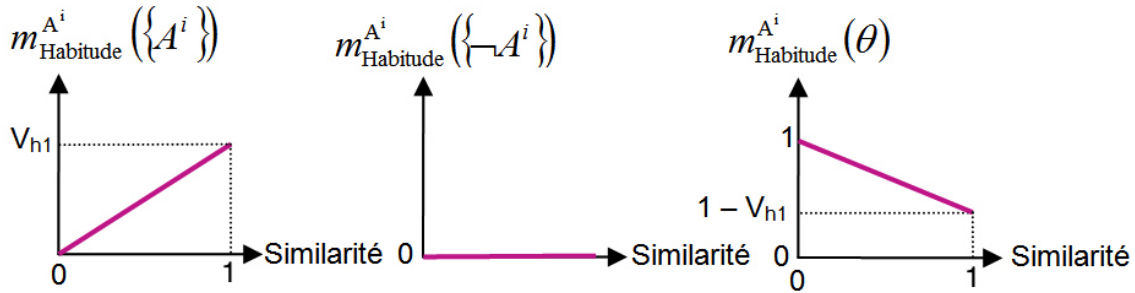


FIGURE 3.5 – Forme générale des fonctions de croyance pour le désir de maximisation de la similarité avec le dernier plan choisi (Taillandier *et al.*, 2012b).

D'une manière générale, l'exécution de chacune des opérations culturales au sein des parcelles est conditionnée par un ensemble de contraintes : fenêtre temporelle, état de l'environnement biophysique, gestion des ressources de l'exploitation, temps d'exécution des opérations culturales et topologie des parcelles.

Le plan de conduite des parcelles est remis à jour à chaque fois qu'une séquence de cultures mise en œuvre dans un lot de parcelles arrive à terme. Dans ce cas, le plan est supprimé de la base d'intentions et un nouveau processus de choix de plan est déclenché. Ce nouveau choix de plan prend en compte les séquences toujours en cours d'exécution. Le plan est également revu si le semis programmé d'une culture ne peut pas être réalisé i.e. si la période de semis pour une parcelle est passée et que l'agent agriculteur n'a pas pu semer. Dans ce cas, l'agent relance un choix de plan afin de remplacer la culture prévue sur la parcelle par une culture de substitution, en considérant ; là encore, les séquences de culture en cours d'exécution. En conclusion, le choix d'assolement n'est pas réalisé une bonne fois pour toute mais il est adapté en fonction des contraintes liées à son exécution.

Ce modèle de comportement a été testé pour la simulation du comportement des agriculteurs dans 4 bassins versants (taille de l'environnement : 125x100 km) sur une période de 4 ans (2006 à 2009). Cette expérimentation a montré que notre modèle permettait d'obtenir des résultats proches du système réel en termes de proportion observée des différentes espèces cultivées. Les résultats complets ainsi que leur analyse peuvent être trouvés dans (Taillandier *et al.*, 2012b).

### 3.2.4 Discussion

Ce travail sur la représentation du processus de décision des agents par la théorie des fonctions de croyance et son application pour modéliser le comportement des agriculteurs dans le projet MAELIA ont fait l'objet de différentes publications (Taillandier *et al.*, 2010a, 2011b, 2012b,a).

La plate-forme MAELIA a depuis connu de nombreuses évolutions (nouveaux processus et acteurs), mais intègre toujours ce modèle de comportement pour les agriculteurs. A l'usage, le choix d'utiliser la théorie des fonctions de croyance pour représenter le processus de décision des acteurs s'est révélé intéressant : les participants au projet, en particulier les agronomes, ont bien adhéré à ce formalisme et il a permis d'obtenir des résultats de simulation crédibles. Néanmoins, certaines difficultés de mise en œuvre et limites ont pu être identifiées. La première concerne la définition des fonctions de croyance pour chacun des critères. Si les experts du domaine ont bien compris la signification des différentes

fonctions à définir, donner des valeurs précises pour celles-ci s'est révélé plus fastidieux. Il a fallu pour cela procéder à des phases de calibration qui ont nécessité de nombreuses données et qui restent encore aujourd'hui perfectibles. Une autre limite est qu'elle nécessite de retranscrire tous les facteurs intervenants dans le processus de décision sous la forme de critères, ce qui peut s'avérer d'une part peu adapté pour certains facteurs (en particulier les facteurs sociaux ou émotionnels) et qui de plus restreint le pouvoir explicatif des décisions prises par l'agent lorsque trop de critères sont en jeu : il est en effet difficile de comprendre pourquoi un agent a pris telle ou telle décision dans un contexte donné si ce choix résulte de l'évaluation d'une dizaine de critères potentiellement interdépendants.

Cette expérience m'a en tout cas fortement motivé à continuer de travailler sur la formalisation de comportements humains. A la suite de mon arrivée à l'Université de Rouen, je me suis ainsi attelé à l'écriture d'un projet ANR, appelé ACTEUR, centré sur cette problématique. C'est finalement en 2014 que ce projet a été accepté et financé par l'ANR. La partie suivante détaille l'architecture d'agents développée dans le cadre de ce projet.

### 3.3 Architecture BEN

#### 3.3.1 Contexte

En parallèle de mon travail en post-doctorat sur le projet MAELIA, j'ai co-encadré avec Benoit Gaudou (MCF, Université de Toulouse 1) le stage de Master 2 de Le Van Minh sur la définition d'une architecture pour les agents cognitifs dans GAMA. Ce travail a permis le développement d'une architecture basée sur le paradigme BDI codée en GAML et appliquée pour des modèles jouets ([Le et al., 2013](#)). Cette expérience fut pour moi l'occasion d'utiliser pour la première fois une architecture de type BDI dans un contexte de simulation. Si l'architecture développée avait de nombreuses limites (complexe à utiliser et problème de passage à l'échelle), elle avait le mérite de constituer une base de réflexion intéressante pour une future architecture directement intégrée dans GAMA et de montrer les potentialités des architectures BDI en termes d'expressivité.

Une seconde expérience sur les architectures BDI est arrivée en 2012, lors d'un Coding Camp GAMA, où j'ai participé, sous l'impulsion de Philippe Caillou (MCF, Université d'Orsay) et toujours avec Benoit Gaudou, au développement d'une extension pour GAMA visant à proposer aux modélisateurs une architecture BDI simple à utiliser, appelée *simple BDI* ([Caillou et al., 2017](#)).

Cette première architecture BDI pour GAMA fut pour moi une grande source d'inspiration pour l'écriture du projet ANR ACTEUR (Agents Cognitifs Territorialisés pour l'étude des dynamiques Urbaines et des Risques) dont l'objectif était de développer des outils visant à faciliter le travail des modélisateurs pour la définition d'agents cognitifs. Un élément clef dans ce projet était de définir une nouvelle architecture d'agents qui étende le *simple BDI* et permettent d'intégrer à la fois des éléments cognitifs, sociaux, normatifs et émotionnels. Concernant ce dernier point, j'avais déjà, par le passé, co-encadré en 2010 avec Benoit Gaudou le stage de Master 1 de Le Van Minh qui portait sur la simulation d'une situation d'évacuation d'un bâtiment en cas d'incendie en prenant en compte les émotions de peur et leur diffusion ([Le et al., 2010](#)).

Le travail sur la nouvelle architecture a été en grande partie porté par la thèse de Mathieu Bourgeois (2015-2018), dont j'ai assuré le co-encadrement et qui a été dirigée par Laurent Vercoüter (Pr INSA Rouen, LITIS). Ce travail a ainsi résulté dans le développement de l'architecture BEN (Behavior with Emotions and Norms). Cette architecture permet aux modélisateurs de créer des agents simulant des

êtres humains en intégrant différentes dimensions sociales telles que la cognition, les émotions, la contagion émotionnelle, la personnalité, les relations sociales et les normes.

La suite de cette partie décrit les grands principes derrière l'architecture BEN. La description complète de l'architecture pourra être retrouvée dans le mémoire de thèse de Mathieu Bourgaïs ([Bourgaïs, 2018](#)).

### 3.3.2 Formalisation

L'architecture BEN intègre différentes dimensions affectives et sociales en plus d'un processus cognitif de raisonnement. Plus précisément, l'architecture est basée sur 6 concepts (Figure 3.6) : la cognition, les émotions, la contagion émotionnelle, les relations sociales, les normes et la personnalité.

Pour la cognition, comme mentionné dans les parties précédentes, nous avons fait le choix de baser notre architecture sur le paradigme BDI. En effet, de nombreux travaux ont déjà montré l'intérêt d'utiliser ce type d'architecture pour la simulation à base d'agents ([Adam et Gaudou, 2016](#)). Il offre des avantages en termes d'adaptabilité, de robustesse et de capacité à expliquer les comportements des agents.

Pour ce qui est des émotions, nous avons fait le choix d'utiliser la théorie de [Ortony et al. \(1990\)](#). Cette théorie, connue sous le nom de théorie OCC, se concentre sur les conditions de déclenchement des émotions et repose sur l'idée que les émotions sont des réponses à l'évaluation de trois types de stimulus : les événements, les actions réalisées par les agents et les objets. Elle définit ainsi 22 émotions. Cette théorie est aujourd'hui la plus utilisée en Intelligence Artificielle ([Bourgaïs et al., 2018](#)). Elle se prête très bien à une implémentation et peut facilement être couplée à une architecture BDI. C'est donc logiquement que nous avons choisi d'utiliser celle-ci pour BEN et en particulier la formalisation logique de cette théorie proposée par [Adam \(2007\)](#).

Un autre aspect que nous avons souhaité intégrer dans cette architecture est la contagion émotionnelle. En effet, différents travaux ont montré que les personnes tendent à adapter leurs émotions en fonction des émotions de ceux qu'ils perçoivent, que ce soit physiquement ou via un réseau social ([Hatfield et al., 1993](#); [Kramer et al., 2014](#); [Barsade, 2002](#)). Ce processus s'appuie sur une analyse cognitive rapide pour, soit copier les émotions perçues, soit créer une émotion en rupture avec le reste du groupe. Cette vision de la contagion émotionnelle a été formalisée par le modèle ASCRIBE (Agent-based Social Contagion Regarding Intentions Beliefs and Emotions) ([Bosse et al., 2009](#)) qui propose un modèle du processus de contagion émotionnelle entre deux agents. C'est cette formalisation bien établie et compatible avec les autres composants que nous avons choisi d'intégrer dans BEN.

Nous nous sommes également intéressés à l'impact des relations sociales sur les comportements. Une façon de concevoir les relations sociales est de les considérer comme des relations interpersonnelles ([Rousseau et Hayes-Roth, 1998](#); [Gratch, 2002](#)). Ces travaux proposent de représenter les relations sociales entre les agents par un nombre fini de variables numériques qui correspondent aux variables minimales permettant de décrire une relation entre deux personnes. Nous avons fait le choix dans ce cadre d'utiliser le modèle dimensionnel des relations interpersonnelles proposé par [Svennevig \(2000\)](#) qui se base sur 4 variables : l'appréciation, la dominance, la solidarité et la familiarité.

Un autre aspect important qui entre également dans la dimension sociale est le rapport aux normes et

obligations. Les normes permettent en effet de décrire le comportement attendu d'un système, qu'il soit social ou non. Dans ce cadre, Tuomela (1995) propose de distinguer les normes sociales (imposées implicitement par une population) et les obligations (imposés explicitement par une institution); ces deux types de processus étant soumis à des sanctions. Plusieurs travaux en Intelligence Artificielle (IA) reposant sur ce type de classification ont cherché à proposer des systèmes normatifs (Dignum, 1999; Stratulat, 2002) pour décrire les comportements des agents. Un travail particulièrement intéressant dans ce contexte est celui de (y López *et al.*, 2006), qui propose une représentation unifiée des concepts nécessaires à la définition d'un système normatif. C'est cette formalisation, classique en IA, que nous avons décidé d'intégrer dans BEN.

Enfin, différentes études ont montré l'importance de la personnalité dans les prises de décision (Eysenck *et al.*, 1987; Watson et Clark, 1992; Ortony, 2002). Il est ainsi intéressant d'utiliser la notion de personnalité pour différencier la façon dont les agents vont réagir par rapport à une situation. Il existe dans la littérature différents modèles de personnalité tels que MBTI (Myers *et al.*, 1985) et OCEAN (McCrae et John, 1992). Ce dernier, que nous avons choisi d'intégrer dans BEN et qui le plus utilisé en IA, propose de décrire la personnalité d'une personne en utilisant 5 facteurs numériques : Ouverture, Conscience, Extraversion, Agréabilité et Neurotisme. Nous avons également fait le choix d'utiliser cette personnalité comme levier pour paramétrer les agents : ainsi, pour paramétrer un agent BEN, le modélisateur pourra se limiter à donner des valeurs à ces 5 facteurs.

### 3.3.3 Architecture

L'architecture BEN (Figure 3.7) se compose de 4 modules qui sont exécutés de façon séquentielle lorsque l'agent est activé. Le premier module se charge de la gestion des perceptions de l'agent. A partir des éléments perçus, un deuxième module permet de gérer les connaissances de l'agent. Ces connaissances sont ensuite prises en compte par un troisième module qui permet à l'agent de prendre des décisions. Enfin, un dernier module se charge de gérer la dynamique des connaissances, en particulier leur dégradation. Chaque module propose des processus gérés automatiquement par l'architecture (en bleu sur la figure) ou devant être définis manuellement par le modélisateur (en rose sur la figure). Certains de ces processus sont obligatoires dans le fonctionnement de l'architecture (en lignes pleines sur la figure), les autres sont optionnels (en pointillés sur la figure). En effet, une idée forte derrière BEN est d'avoir une architecture fortement modulaire : c'est au modélisateur de choisir les éléments qui sont intéressants à utiliser pour son cas d'étude et il n'est en aucun cas obligé d'utiliser tous les éléments inclus de BEN. Ainsi, au minimum, un modélisateur devra définir un désir pour l'agent et un plan permettant de le réaliser, tout le reste est optionnel. C'est également dans cette optique de faciliter l'utilisation de BEN que nous avons utilisé la notion de personnalité pour paramétrer un agent BEN : tous les paramètres des agents sont contenus dans les 5 valeurs du modèle de personnalité OCEAN.

La suite de cette partie décrit plus en détails les différents composants de l'architecture BEN.

#### Bases de connaissances

Les bases de connaissances de l'agent utilisent comme élément unificateur le concept de *prédicat* qui représente une information sur le monde, que ce soit un événement, une situation ou une action. Ainsi, les états mentaux des agents seront construits à partir de ces prédicats. Plus précisément, un état mental est défini par une modalité (exemple : croyance) et par un objet sur lequel il porte. Cet objet peut être un autre état mental, un prédicat ou une émotion. Un état mental a enfin une durée de

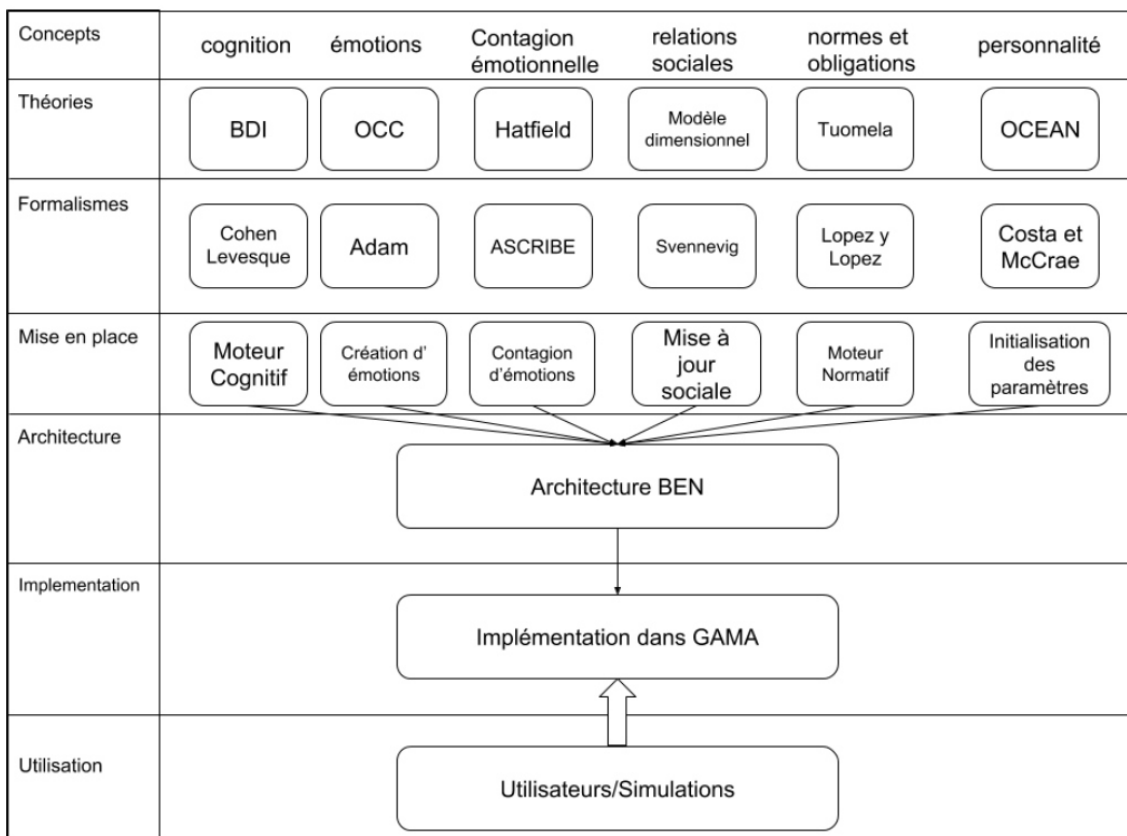


FIGURE 3.6 – Schéma méthodologique (Bourgais, 2018).

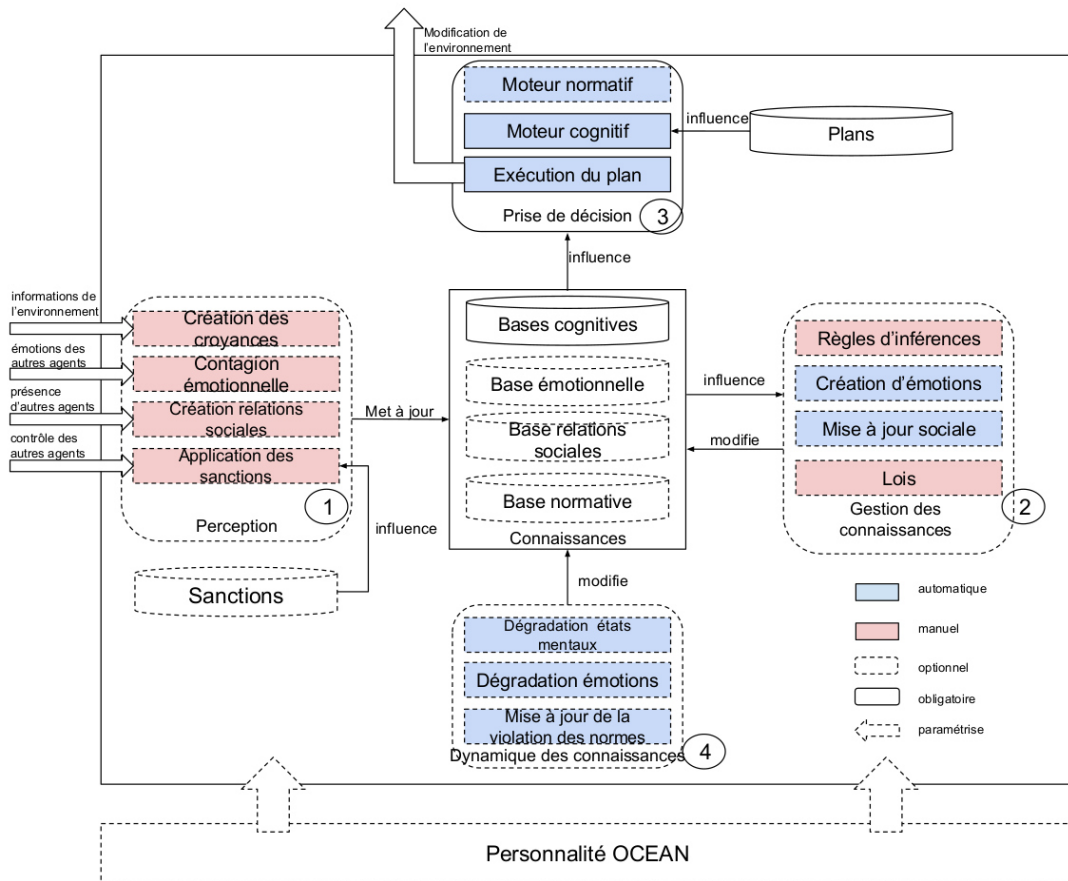


FIGURE 3.7 – Architecture BEN (Bourgeois, 2018).



vie indiquant le temps que mettra l'agent pour l'oublier.

BEN comprend 6 modalités qui correspondent chacune à une base de connaissances de l'agent :

- *Croyance* : une information que l'agent croit vraie sur le monde.
- *Incertitude* : une information incertaine sur le monde.
- *Désir* : un état du monde que l'agent souhaite atteindre.
- *Intention* : un état du monde que l'agent s'engage à atteindre.
- *Idéal* : une information sur laquelle l'agent porte un jugement social (louable ou blâmable).
- *Obligation* : un état du monde que l'agent doit atteindre.

Les croyances de l'agent représentent des informations que l'agent a sur le monde. Ces informations peuvent donc se révéler fausses même si l'agent est lui-même sûr de leur vérité contrairement aux incertitudes, qui représentent des informations dont l'agent a connaissance mais sur lesquelles il ne possède pas de certitudes.

Les désirs représentent les états du monde que l'agent souhaite lui-même voir atteint alors que les obligations sont imposées par une institution. Le concept d'intention permet d'engager l'agent dans la tâche d'atteindre un état du monde, que cette tâche vienne des désirs ou des obligations.

L'agent dispose également d'une base d'émotions. Les émotions sont définies selon le modèle OCC (Ortony *et al.*, 1990) et peuvent intégrer un prédicat à propos duquel l'émotion est ressentie, un agent responsable de l'émotion et une intensité.

Concernant les relations sociales, elles sont définies en s'appuyant sur les 4 variables de Svennevig (2000) (l'appréciation, la dominance, la solidarité et la familiarité) auxquelles nous avons ajouté une cinquième variable représentant la confiance. Cette dernière variable permet de faire le lien avec le système normatif : la valeur de confiance va évoluer en fonction des sanctions et des récompenses sociales. Une relation sociale n'est pas nécessairement symétrique : un agent peut en apprécier un autre qui en retour ne l'apprécie pas.

En plus de ces bases de connaissances qui vont évoluer au cours de la simulation, l'architecture BEN définit 2 autres bases qui sont, elles, statiques : la base des plans, qui contient des plans d'actions permettant à l'agent d'agir sur le monde, et la base des sanctions qui contient l'ensemble des sanctions, c'est-à-dire des actions prises par l'agent si un autre agent enfreint une norme.

### **Module de perception**

Une fois un agent BEN activé, le premier module entrant en jeu est celui des perceptions. Ce module est utilisé pour lier l'environnement et les connaissances de l'agent, en transformant les informations venant du monde en états mentaux, en émotions ou en liens sociaux. C'est aussi lors de cette étape que les normes des autres agents sont contrôlés et que des sanctions sont appliquées. Tous ces processus sont effectués à l'intérieur d'une *perception*. Pour définir une perception, le modélisateur peut spécifier l'ensemble des agents à observer soit directement par une liste d'agents ou soit à travers l'utilisation de requêtes spatiales (par exemple, les agents dans un cône de perception).

Le premier processus que le modélisateur peut définir à l'intérieur d'une perception est l'ajout de connaissances sur le monde. Ce processus permet de transformer les informations venant de l'envi-

ronnement en croyances ou en incertitudes dans les bases cognitives de l'agent.

Un second processus est la contagion émotionnelle qui permet de mettre à jour les émotions de l'agent effectuant la perception en fonction des émotions des autres agents perçus.

Un troisième processus concerne la création de relations sociales entre l'agent réalisant la perception et les agents perçus.

Finalement, le dernier processus consiste en l'application des sanctions envers les agents perçus. Lors de la perception d'un autre agent soumis au système normatif, l'agent effectuant la perception vérifie si la norme, l'obligation ou la loi contrôlée a été violée et s'il peut appliquer une sanction le cas échéant.

### **Module de gestion des connaissances**

Le deuxième module de BEN permet de gérer les connaissances de l'agent avant la prise de décision. Plus précisément, il va créer de nouveaux états mentaux et émotions à partir de l'état de l'agent.

Le premier processus de ce module va appliquer les règles d'inférence définies par le modélisateur. Une règle d'inférences est une règle de type « si . . . alors . . . » qui permet de modifier les états mentaux cognitifs de l'agent en fonction de l'état courant de celui-ci (i.e. ses états mentaux, émotions ou autre).

Le deuxième processus, qui lui ne nécessite aucune intervention du modélisateur, va consister à générer des émotions en fonction de l'état de l'agent. BEN utilise pour cela les règles définies par [Adam \(2007\)](#). Par exemple, un agent va automatiquement ajouter dans sa base une émotion de joie s'il désire qu'un événement soit vrai et qu'il croit que cet événement est vrai. BEN permet ainsi de créer automatiquement 20 émotions dont les règles de déclenchement vont dépendre des croyances, des incertitudes, des autres émotions, des idéaux et des relations avec les autres agents. Si une émotion créée automatiquement par le moteur émotionnel est déjà présente dans la base des émotions de l'agent, celle-ci sera renforcée.

Un autre processus automatique est l'évolution des relations sociales. Lorsqu'un agent déjà connu est de nouveau perçu, la relation avec cet agent est mise à jour automatiquement. Plus précisément, BEN va calculer les nouvelles valeurs pour les 5 facteurs représentant les relations sociales à partir des états mentaux cognitifs et des émotions de l'agent. Cette mise à jour s'appuie sur le travail de [Ochs et al. \(2009\)](#).

Un dernier processus est l'application de lois. Ce processus fonctionne exactement comme le processus d'activation des règles d'inférence. De façon similaire aux règles d'inférence, le modélisateur peut définir des lois qui vont créer des obligations en fonction du contexte qui va dépendre du seuil d'obéissance relatif à la loi et de la valeur d'obéissance de l'agent.

### **Module de prise de décision**

Le troisième module de l'architecture est le seul obligatoire dans BEN. C'est ce module qui permet à l'agent de prendre une décision (i.e. choisir une intention à suivre et un plan d'actions ou une norme pour l'atteindre) et exécute le plan/norme correspondant.

Le moteur cognitif et normatif est présenté Figure 3.8 :

- *Étape 1* : le moteur vérifie si l'intention courante est toujours valide et si l'agent souhaite ou non garder la même intention (utilisation d'une probabilité de remise en cause de l'intention).
- *Étape 2* : le moteur vérifie si le plan ou la norme en cours est toujours utilisable ou non et si l'agent souhaite le conserver (utilisation d'une probabilité de remise en cause du plan ou de la norme).
- *Étape 3* : l'agent choisit une nouvelle intention parmi ses désirs réalisables en fonction de leur priorité (choix déterministe ou probabiliste).
- *Étape 4* : le plan de plus haute priorité répondant à l'intention courante et dont le contexte est valide est sélectionné comme nouveau plan courant.
- *Étape 5* : le plan courant est exécuté.
- *Étape 6* : le moteur vérifie si l'agent est soumis à une obligation et s'il souhaite s'y plier.
- *Étape 7* : l'agent choisit comme intention l'une de ses obligations réalisables en fonction de leur priorité (choix déterministe ou probabiliste).

### Module de dynamique des connaissances

Le dernier module permet de créer une dynamique temporelle dans le comportement de l'agent. Ainsi, le module va automatiquement dégrader les états mentaux cognitifs et les émotions de l'agent (processus d'oubli), puis met à jour le statut des normes (i.e. est-ce que la norme est activable et si oui, est-ce qu'elle a été violée ou non).

#### 3.3.4 Discussion

L'architecture BEN, qui a fait l'objet de plusieurs publications (Caillou *et al.*, 2017; Taillandier *et al.*, 2016c; Bourgaïs *et al.*, 2017b, 2016, 2017a, 2019), a été implémentée dans la plate-forme GAMA et est directement utilisable au travers du langage GAML. J'ai également développé et intégré dans GAMA une version distribuée de l'architecture (Taillandier *et al.*, 2017) permettant de réduire les temps de calcul.

L'un des objectifs avec BEN était de disposer d'une architecture qui soit à la fois expressive, mais en même temps simple d'utilisation. Nous avons ainsi mené plusieurs travaux visant à évaluer cette expressivité et cette facilité d'utilisation.

Un premier travail qui a été mené, a consisté à comparer plusieurs architectures, dont BEN, pour modéliser le comportement de petits producteurs (riz, fruit, crevette, etc.) dans les villages du Delta de Mékong (Truong *et al.*, 2015). Ce travail, qui a été au cœur de la thèse de Truong Chi Quang, dont j'ai participé au co-encadrement, a permis de mettre en avant les qualités de BEN pour définir des comportements d'agents crédibles.

Concernant la facilité d'utilisation, j'ai participé à deux études sur la question qui se sont concentrées sur la composante cognitive de BEN. La première (Taillandier *et al.*, 2016c) a consisté à demander à des utilisateurs ayant des compétences diverses en développement de modèles avec GAMA (novice à expert) et en IA (aucune connaissance à expert) de réaliser plusieurs exercices en une heure après avoir reçu un cours d'une trentaine de minutes sur l'utilisation de BEN dans GAMA. Tous les participants à l'expérimentation ont montré une bonne compréhension du fonctionnement de l'architecture, même si tous n'ont pas réussi à finir les exercices dans le temps imparti. La deuxième étude avait pour

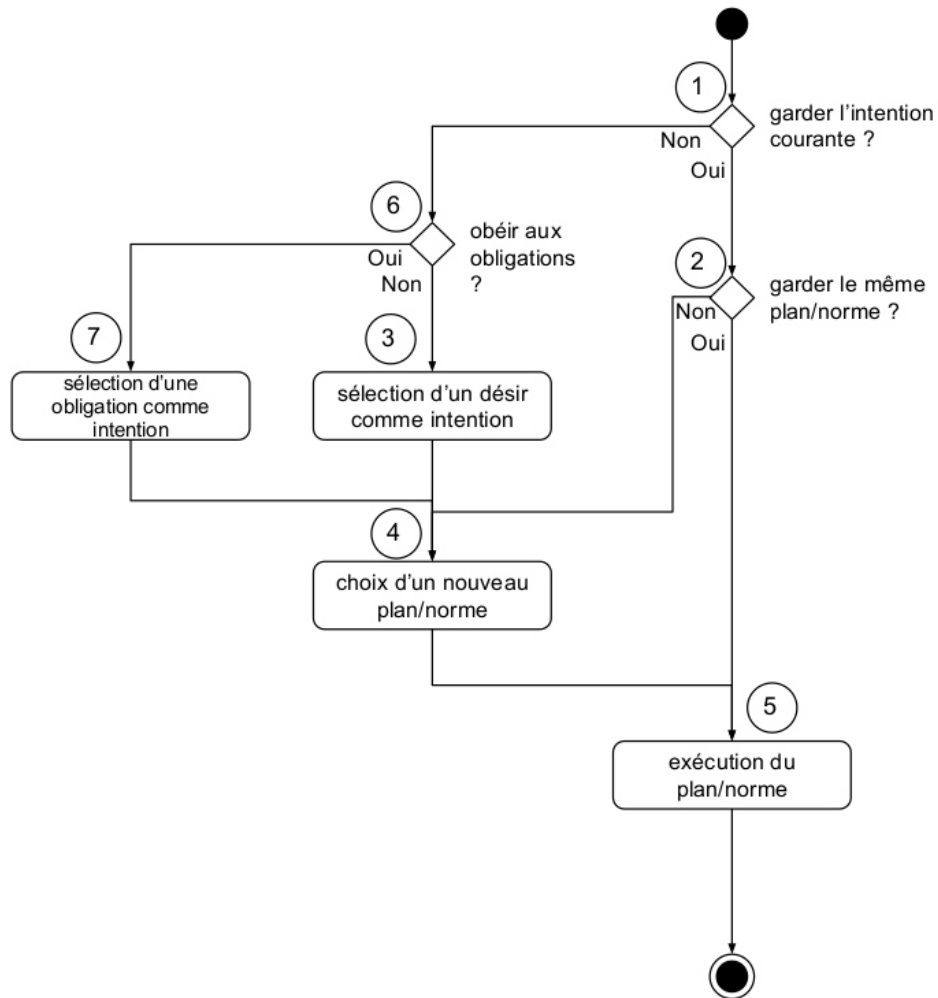


FIGURE 3.8 – Diagramme d’activité du moteur de prise de décision de BEN (Bourgais, 2018).

objectif de comparer BEN à une architecture basée sur un automate à états finis (Adam et Dugdale, 2016; Adam *et al.*, 2017). Nous avons posé des questions sur les deux architectures et leur utilisation dans GAMA, à des étudiants en Master Informatique. Les résultats ont montré que le modèle construit avec l'architecture BEN, bien que pouvant être perçu comme plus complexe à programmer, se révélait plus modulaire, plus flexible et plus facile à comprendre.

Une autre étude à citer est le modèle développé par Marion Valette, dont j'ai encadré le stage de niveau M1 (2<sup>ième</sup> année à ISAE-SUPAERO) avec Benoit Gaudou et Dominique Longin (CR CNRS, UMR IRIT) en 2018, sur l'évacuation de bâtiments. L'un des intérêts de ce travail était de voir si une personne n'ayant jamais eu de cours sur la simulation à base d'agents et les architecture BDI était capable en quelques mois de s'approprier cet outil et de développer un modèle crédible. Le résultat a dépassé nos espérances. En effet, en quelques mois, Marion Valette a réussi à produire un modèle utilisant la plupart des concepts de BEN (moteur cognitifs, émotions, relations sociales) permettant de reproduire de façon crédible un cas réel d'évacuation (incendie du Station Nightclub aux Etats Unis) et a publié un article sur son modèle dans la conférence PRIMA (Valette *et al.*, 2018).

Pour finir, un dernier point tendant à accréditer l'utilisabilité de BEN est son utilisation pour deux projets en cours. Adam et Dugdale (2018) ont développé un modèle sur l'évacuation d'une population lors d'incendies de bush en Australie qui utilise fortement la composante cognitive de BEN. Un autre exemple d'utilisation, et cette fois menée par des chercheurs non informaticiens, qui n'avaient pas de connaissances préalables en IA, est LI-BIM (Micolier *et al.*, 2018, 2019). Ce modèle qui porte sur les consommations énergétiques et la qualité de l'air intérieur des bâtiments, intègre un modèle très détaillé de l'occupant utilisant BEN. En plus du moteur cognitif, ce modèle utilise également les relations sociales de BEN pour décrire les relations entre occupants et les capacités de chacun à convaincre les autres (par exemple pour augmenter la température de chauffage). Ce même modèle d'occupant sera utilisé dans le cadre d'une thèse portant sur l'intelligence des bâtiments et qui vient de démarrer à l'Université de Nice-Sophia Antipolis.

### 3.4 Conclusion

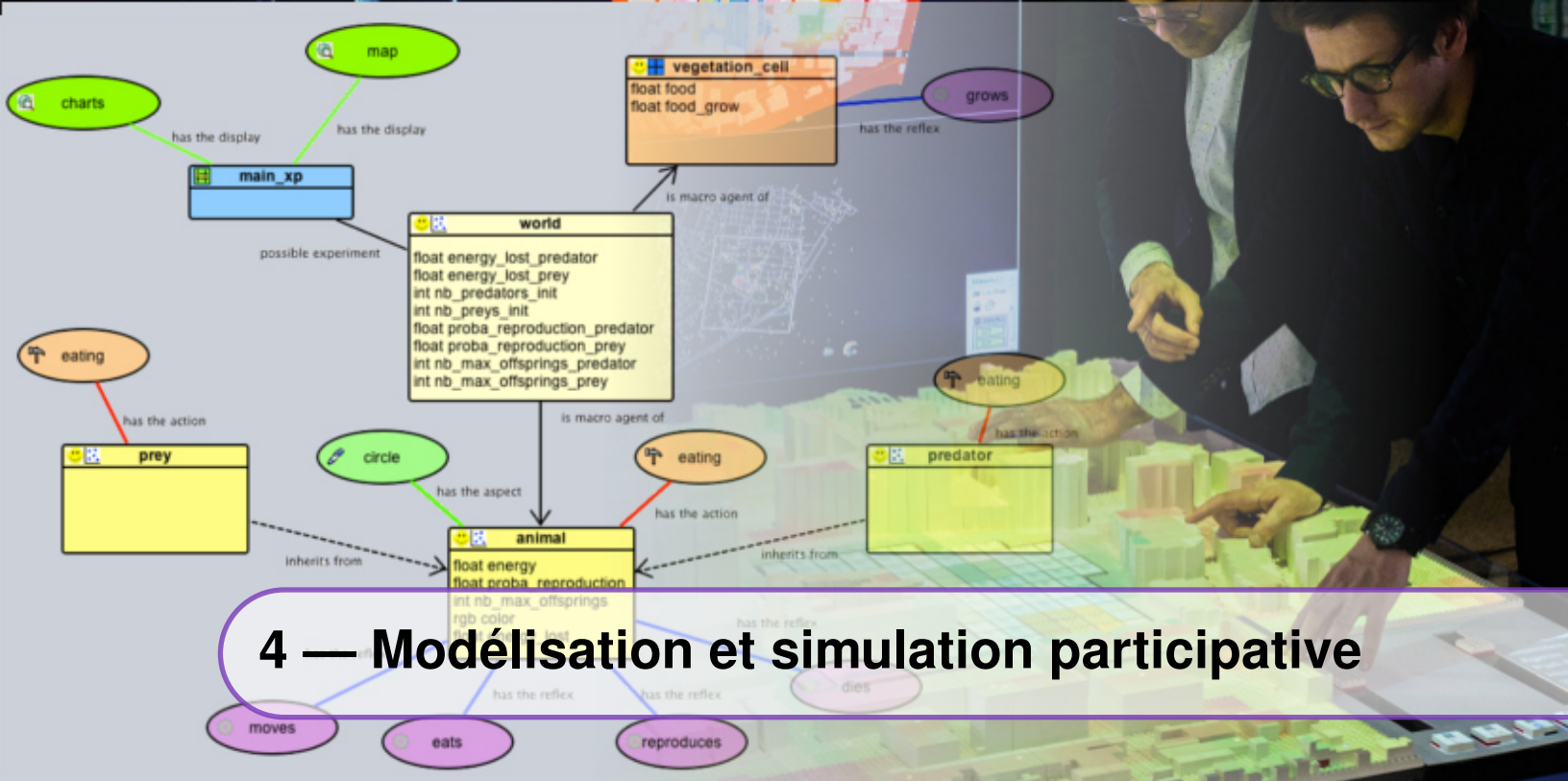
Ces dernières années, la problématique du développement d'architectures d'agents pour modéliser les processus de décision chez les acteurs humains a occupé une place très importante dans mes travaux, en particulier grâce à l'obtention en 2014 du projet ANR ACTEUR.

La question de savoir comment mieux représenter les acteurs et leur décision dans les simulations est aujourd'hui un sujet prioritaire pour moi. Je compte ainsi continuer à fortement m'impliquer sur ce sujet en enrichissant BEN avec l'objectif de continuer à développer cette architecture dans la même logique que pour le projet ACTEUR : proposer une architecture fortement modulaire avec un grand nombre de modules facultatifs.

Il y a plusieurs sujets de recherche qu'il m'intéresserait d'explorer pour enrichir BEN. Un premier sujet concerne la formalisation des plans d'action. En effet, pour le moment, les plans d'action dans BEN sont peu formalisés : ils sont juste composé d'une série d'instructions. Or, de nombreux travaux existent sur la formalisation de plans. Je souhaiterai en particulier utiliser les formalismes développés par Roger Martin-Clouaire (DR INRA, MIAT) (Martin-Clouaire et Rellier, 2009, 2002) qui permettrait d'intégrer dans BEN toute une grammaire pour construire des plans flexibles et réellement adap-

tables au contexte.

Une autre voie de recherche concerne le couplage entre BEN et les systèmes argumentatifs. Cette problématique est apparue au travers du projet DID'IT VITAMIN débuté en 2018 avec Rallou Thomopoulos (DR INRA, IATE) et Nicolas Saillou (Post-doc, ETH Zurich) sur le végétarisme. L'idée derrière ce projet est d'étudier les réseaux d'arguments utilisés par les personnes concernant le végétarisme et de les intégrer dans un modèle de simulation pour étudier la diffusion de ce type de diète alimentaire. Les premiers résultats du projet sont très encourageants et me motivent fortement à continuer à travailler sur ce sujet.



## 4 — Modélisation et simulation participative

### 4.1 Introduction

Certains projets de modélisation peuvent être menés par un unique modélisateur maîtrisant à la fois les aspects techniques et thématiques. Mais, pour beaucoup d'autres, il est nécessaire de fonctionner en équipe, où chaque individu, qui peut être un acteur du système étudié, va apporter ses propres connaissances. Ainsi, dans les dernières années, il y a eu un essor important des approches de modélisation participative qui consistent à impliquer directement les acteurs du système dans l'une ou plusieurs étapes du processus de modélisation : définition de la question de modélisation, étape de formalisation, de calibration ou de définition de scénarios (Barreteau *et al.*, 2014). L'implication d'acteurs du système peut également intervenir de façon postérieure au processus de modélisation en permettant à des participants d'interagir avec un environnement simulé ; on parlera alors de simulation participative.

Si je n'ai personnellement jamais été amené à animer une démarche de modélisation ou de simulation participative, cette problématique m'a intéressé depuis longtemps, en particulier au travers de la question des outils pouvant être fournis aux modélisateurs pour que ces derniers puissent mener à bien une telle démarche. En effet, ma première expérience dans le domaine de la recherche, en 2005 lors de mon stage de M2, a porté sur le passage d'un jeu sérieux développé sous CORMAS par Christophe Le Page (CR CIRAD, GREEN) en version distribuée. Je me suis ensuite intéressé, en 2009 lors de mon post-doctorat à Hanoi, à la question de l'apprentissage de comportements d'agents par la simulation participative. Concernant le processus de modélisation, j'ai commencé à m'intéresser aux outils graphiques de conception de modèles à mon arrivée en 2011 à l'UMR IDEES en tant que Maître de Conférences. J'ai pu observer avec beaucoup d'intérêt les travaux menés par Eric Daudé, Patrice Langlois et Baptiste Blanpain sur la plate-forme MAGéo, qui permettait à des utilisateurs néophytes en programmation de développer un modèle à base d'agents à l'aide d'un éditeur graphique. Ces travaux m'ont motivé à intégrer un outil de ce type dans GAMA afin de faciliter les échanges autour des modèles. La suite de ce chapitre revient sur ces différents travaux que j'ai pu mener sur le sujet.



## 4.2 Modélisation participative

### 4.2.1 Contexte

La modélisation participative consiste à impliquer des acteurs du système étudié dans le processus de conception du modèle. Une approche classique pour atteindre cet objectif est d'organiser des ateliers durant lesquels les modélisateurs et les acteurs construisent ensemble un modèle en utilisant des outils de modélisation graphique. La plupart des modélisateurs qui mènent ce type d'atelier tendent à utiliser des outils de modélisation graphique indépendants, qui ne sont pas directement intégrés dans des plates-formes de modélisation.

En effet, s'il existe des plates-formes de modélisation à base d'agents permettant de développer des modèles à travers une interface graphique, elles sont souvent trop complexes à comprendre pour des non-informaticiens et donc non adaptées à la modélisation participative ou trop limitées pour développer des modèles descriptifs.

Pour être plus précis, certaines des plates-formes comme StarLogo TNG ([Resnick, 1996](#)) et Modelling4All ([Kahn et Noble, 2009](#)) sont principalement des outils pédagogiques qui sont limités au développement de modèles simples.

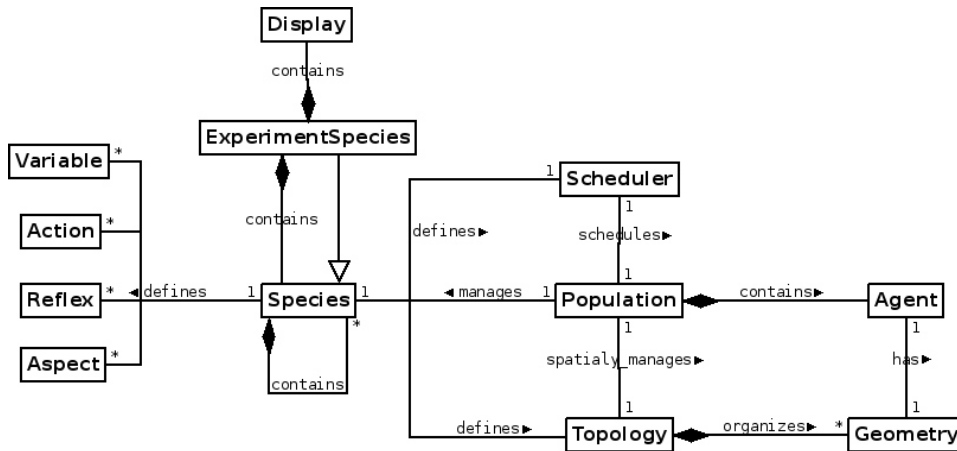
Repast Symphony ([North et al., 2013a](#)) est une autre plate-forme offrant un outil de modélisation graphique. S'il est possible de définir des modèles complexes avec Repast en utilisant Java, son outil de modélisation graphique est lui assez limité et reste complexe à utiliser.

Beaucoup plus intéressante dans un contexte de modélisation participative, la plate-forme CORMAS ([Le Page et al., 2012](#)) offre des outils de modélisation graphique permettant de définir des diagrammes d'activités directement exploitables dans les modèles. Néanmoins, cette plate-forme, qui est plus adaptée au développement de modèle simple (KISS), n'offre pas la même richesse que la plate-forme GAMA en termes de développement de modèles, en particulier pour les modèles basés sur un environnement spatial complexe.

Pour finir, la plate-forme MAGéo ([Langlois et al., 2015](#)) permet de simplement définir des modèles à partir d'une interface graphique dédiée. Le comportement des agents est formalisé sous la forme d'une agrégation de comportements basiques définis à l'aide d'une grammaire très simple. Cette grammaire qui est adaptée aux modèles simples, ne permet pas de définir des modèles descriptifs. De plus, le nombre d'opérateurs disponibles pour le modélisateur est très limité, en particulier si on le compare au nombre d'opérateurs de GAMA.

Le langage de modélisation graphique le plus utilisé est UML. Dans le contexte de la modélisation à base d'agents, des travaux ont montré l'intérêt d'utiliser ce langage pour la communication de modèles ([Bersini, 2012](#)). Néanmoins, certains auteurs ont mis en évidence que l'utilisation d'UML comme langage de modélisation orienté-agent pouvait être inappropriée ([Beydoun et al., 2009](#)).

D'autres langages de modélisation basés sur UML et dédiés aux systèmes multi-agents ont été proposés : les plus connus sont AUML ([Bauer et al., 2001](#)) et AML ([Cervenka et al., 2005](#)). Ces langages permettent d'introduire des notions liées au paradigme agent. Néanmoins, leur portée dépasse celui de la modélisation à base d'agents et couvre tout le champ des systèmes multi-agents, ce qui rend ces langages difficiles à appréhender pour des non-informaticiens et non adaptés au contexte de la

FIGURE 4.1 – Méta-modèle de GAMA (Grignard *et al.*, 2013b).

modélisation de systèmes complexes.

Un dernier langage de modélisation graphique intéressant est celui proposé par la plate-forme MA-Géo. Ce langage est basé sur le méta-modèle AOC (Acteur - Organisation - Comportement) (Daudé *et al.*, 2010). Il est proche de UML et respecte la plupart des propriétés des langages orientés objet. De plus, il permet de définir de façon native des modèles multi-niveaux dans lesquels des agents sont composés d'autres agents. Malheureusement, ce langage impose de nombreuses contraintes en raison des limitations de la plate-forme MAGéo.

Ainsi, il m'est apparu intéressant d'intégrer dans GAMA un langage de modélisation graphique simple (avec un faible nombre de concepts), qui permette de définir un modèle descriptif en réutilisant la plupart des concepts présents dans la plate-forme.

#### 4.2.2 Modélisation graphique avec GAMA

L'outil de modélisation graphique que j'ai développé permet à des modélisateurs de définir leur modèle GAMA et, si nécessaire, de les traduire en GAML. De plus, mon outil permet de transformer un modèle GAML en modèle graphique. Cette fonctionnalité a pour objectif de faciliter la discussion autour d'un modèle et la communication de modèle.

Le processus de modélisation avec cet outil consiste à d'abord définir le diagramme conceptuel représenté par un diagramme entité-relation, puis à spécifier les paramètres et comportements de tous les types d'entités à l'aide de boîtes de dialogue.

Nous avons choisi de baser le langage de modélisation graphique sur le méta-modèle de GAMA. En effet, si beaucoup de méta-modèles orientés agent ont été proposés dans la littérature (voir (Beydoun *et al.*, 2009) pour une présentation des plus utilisés), la plupart d'entre eux ne sont pas adaptés au cadre de la simulation à base d'agents et/ou sont complexes à comprendre pour les non-informaticiens. Un autre avantage d'utiliser le méta-modèle de GAMA est de limiter l'écart entre le modèle conceptuel et l'implémentation finale du modèle.

La Figure 4.1 présente le méta-modèle de GAMA. Le composant principal de ce méta-modèle est l'espèce (**Species**). Comme nous l'avons déjà évoqué au Chapitre 1, une espèce définit, comme une classe dans les langages orienté-objet, les caractéristiques communes des agents d'une population. En particulier, une espèce définit ses variables, ses actions, ses réflexes et ses aspects. Une **Action** est une capacité que les agents d'une population ont, c'est-à-dire quelque chose que les agents savent faire. Un **Reflex** est un comportement, c'est-à-dire quelque chose que les agents de la population vont faire (si certaines conditions sont respectées). Un **Aspect** représente un affichage possible pour les agents. Une espèce d'agents peut spécifier plusieurs actions, réflexes et aspects. Concernant l'aspect spatial, il est possible de définir pour une espèce d'agents une topologie spatiale associée à cette population. GAMA est par nature multi-niveau : une relation de composition entre espèces permet de décrire les différents niveaux hiérarchiques d'agents. Enfin, une relation de spécialisation entre espèces d'agents permet de définir des liens d'héritage entre elles.

Un plan d'expérience (**ExperimentSpecies**) représente un contexte d'exécution du modèle et permet de définir des fenêtres d'affichage pour les simulations.

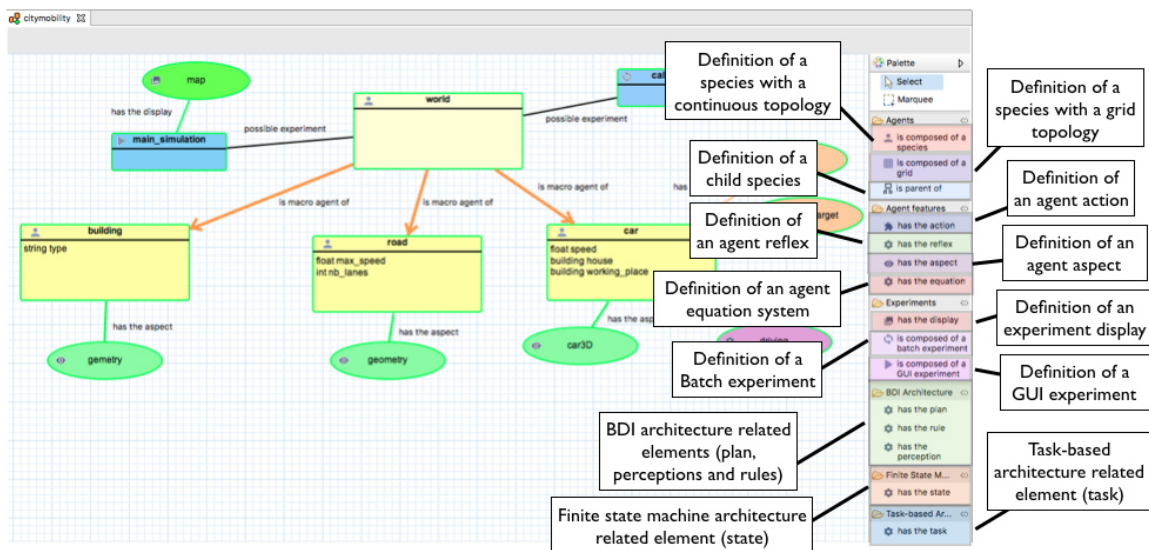


FIGURE 4.2 – Interface de l'outil de modélisation graphique (Taillandier, 2014a).

La Figure 4.2 présente l'outil de modélisation graphique. La palette de droite permet de sélectionner le type d'éléments à ajouter au diagramme. Il est à noter que l'outil intègre toutes les fonctionnalités classiques des éditeurs graphiques (undo, drag and drop, etc).

La Table 4.1 présente tous les éléments qui peuvent être ajoutés dans le diagramme conceptuel.

Quand un modèle graphique est créé, une première espèce d'agents est automatiquement créée : l'espèce correspondant à l'agent **World**. Cette espèce correspond au premier niveau dans la hiérarchie des agents. Elle décrit la topologie spatiale du modèle, son ordonnancement, les paramètres et les comportements globaux. Elle est également l'hôte des populations d'agents définies par le modélisateur.

Ainsi, définir le modèle conceptuel consiste à définir toutes les espèces (avec leur topologie spatiale :

TABLE 4.1 – Les composants du langage de modélisation graphique

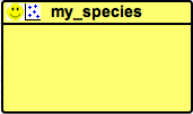
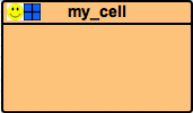
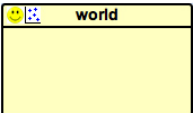
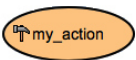
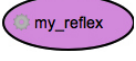
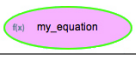
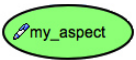
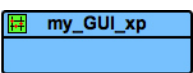
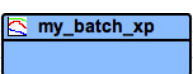

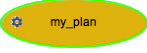

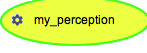
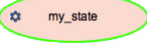
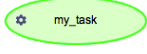
Symbole	Source	Description
	Species	<b>Species</b> : Une espèce d'agents avec une topologie spatiale de type continu.
	Species	<b>Grid</b> : Une espèce d'agents avec une topologie spatiale de type grille.
	-	<b>World</b> : Le premier niveau d'agents. Il contient toutes les autres espèces d'agents.
	Species	<b>Action</b> : Une capacité que les agents ont.
	Species	<b>Reflex</b> : Un comportement (séquence d'instructions) qui sera activé à chaque pas de simulation en fonction d'une certaine condition.
	Species	<b>Equation</b> : un système d'équations différentielles.
	Species	<b>Aspect</b> : un affichage possible pour les agents.
	World	<b>GUI Experiment</b> : exécution d'une simulation avec l'interface graphique classique.
	World	<b>Batch Experiment</b> : exécution d'une série de simulations avec une interface graphique dédiée.
	GUI Experiment	<b>Display</b> : fenêtre d'affichage.

TABLE 4.2 – Les composants du langage de modélisation graphique (suite)

Symbole	Source	Description
	Species avec architecture BEN	<b>Plan</b> : séquence d'instructions qui peuvent être exécutées pour répondre à une intention particulière.
	Species avec architecture BEN	<b>Rule</b> : fonction exécutée à chaque pas de temps pour inférer de nouveaux états mentaux en fonction de l'état courant de l'agent.
	Species avec architecture BEN	<b>Perception</b> : fonction exécutée à chaque pas de temps et qui met à jour la base de croyances de l'agent en fonction de ses perceptions.
	Species avec architecture machine à état fini	<b>State</b> : séquence d'instructions qui sera exécutée si l'agent est dans cet état.
	Species avec architecture basée sur les tâches	<b>Task</b> : séquence d'instructions qui peut être exécutée par l'agent. Si plusieurs tâches peuvent être exécutées, l'agent choisira d'exécuter la tâche avec la plus forte priorité.

The screenshot shows a 'Species definition' dialog box. At the top, the 'Name' field is filled with 'people'. Below this, the 'Skills' section is divided into 'Available Skills' and 'Selected Skills'. 'Available Skills' lists 'GAMASQL', 'MDXSKILL', and 'SQLSKILL'. The 'Selected Skills' field is currently empty. Below the skills section, there are two input fields: 'schedules (container)' and 'frequency (int)', both of which are empty. A 'control' dropdown menu is set to 'reflex'. The 'Variables' section contains a table with the following columns: Name, Type, init value, update, function, min, and max. Below the table are 'Add variable' and 'Delete variable' buttons. The 'Init' section is a large empty text area. The 'Reflex order' section is another empty text area with up and down arrow buttons at the bottom.

FIGURE 4.3 – Boîte de dialogue pour la définition d’une espèce (Taillandier, 2014a).

continue ou grille) vivant dans le monde, leurs capacités (actions), leurs comportements (réflexes, tâches, états ou éléments BEN) et leurs possibles affichages (aspects). Il est possible de définir des relations d’héritage entre 2 espèces. Il est également possible de définir des contextes d’exécution des simulations (experiments) et pour chacune d’entre elles les sorties correspondantes (displays).

A chaque fois que le modélisateur modifie le diagramme, celui-ci est validé automatiquement par GAMA : s’il n’y a pas d’erreurs dans le diagramme, tous les composants apparaissent avec des bordures vertes et les boutons correspondant aux expériences apparaissent en haut de l’éditeur. En cliquant sur l’une des experiments, l’utilisateur peut la lancer (et exécuter les simulations correspondantes). S’il y a des erreurs, les composants avec des erreurs apparaissent avec des bordures rouges.

Une fois le diagramme conceptuel défini, l’étape suivante consiste à décrire les propriétés de chaque composant défini. Quand le modélisateur clique sur un composant, une nouvelle boîte de dialogue s’ouvre permettant de paramétrer celui-ci. C’est par ces boîtes de dialogue que le modélisateur peut transformer son modèle conceptuel en un modèle de simulation. Pour certains types de composants, le paramétrage va juste consister à faire le choix entre différentes options (exemple : choix d’une couleur d’affichage), pour d’autres, il va consister à écrire des instructions en GAML (par exemple pour les actions).

Le plus important type de composants à paramétrer est l'espèce. La boîte de dialogue pour une espèce permet de définir certaines propriétés avec un minimum de code (voir Figure 4.3). En particulier, cette boîte de dialogue permet de définir les variables (attributs) de l'espèce. Pour chaque variable, le modélisateur doit définir son nom et son type (à choisir parmi les types existants comme entier, réel, chaîne de caractères, liste, matrice, dictionnaire, point, géométrie, graphe, chemin...). En plus, le modélisateur peut définir des facettes optionnelles pour chaque variable telles que leurs valeurs initiales, un minimum ou un maximum.

Le modélisateur peut également donner des *Skills* à son espèce et choisir d'ajouter à son espèce une architecture particulière (architecture BEN, machine à état fini ou architecture basée sur les tâches). Le modélisateur peut aussi définir l'ordonnement des agents de l'espèce et leur fréquence d'activation. Enfin, il peut définir le bloc *init* qui représente le constructeur de l'espèce, c'est-à-dire une série d'instructions qui sera exécutée à la création des agents.

Les boîtes de dialogue pour les autres types de composants fonctionnent de la même manière : elles proposent des choix entre plusieurs possibilités quand cela s'avère adapté, ou demandent si besoin à l'utilisateur d'écrire du code GAML dans le cas contraire.

### 4.2.3 Discussion

L'outil que j'ai développé est intégré et distribué dans un plug-in externe qui peut être directement téléchargé et installé depuis l'interface de GAMA. Il est aujourd'hui utilisé dans le cadre de différentes formations en France et au Vietnam où il offre un véritable intérêt pour discuter autour d'un modèle. Il a également fait l'objet de trois publications (Taillandier, 2014a, 2013; Taillandier *et al.*, 2019).

Si un travail important a été fourni pour le rendre fonctionnel et robuste, de nombreuses améliorations sont envisagées. La plus prioritaire est d'améliorer le design de l'interface graphique (design des boîtes et des icônes, choix des couleurs, etc.) afin de rendre l'outil plus attrayant. Une autre amélioration porte sur la gestion des erreurs : l'enjeu est de faciliter encore plus le travail des modélisateurs en leur donnant plus d'information sur les erreurs de leur modèle et sur la façon de les corriger. Enfin, une dernière amélioration envisagée concerne les fonctionnalités de mise en forme automatique des diagrammes (*automatic layout*) qui restent aujourd'hui assez pauvres (un seul type de *layout* proposé).

## 4.3 Simulation participative

### 4.3.1 Contexte

Il est courant aujourd'hui d'inclure dans les projets de modélisation une tâche en lien avec la simulation participative, souvent au travers de la conception d'un jeu sérieux. La simulation participative est une simulation dans laquelle des participants humains sont invités à interagir avec un environnement simulé. Ce type de simulation peut être utilisé pour transmettre un message d'une manière efficace (Klabbers, 2009), pour servir de média de communication entre participants ou encore pour obtenir des informations sur le comportement d'acteurs (Guyot *et al.*, 2006). Un exemple historique est Fish Banks (Meadows *et al.*, 1986), qui permet à des participants de jouer le rôle d'un gestionnaire d'une entreprise de pêche et d'interagir avec une simulation informatique de réserve halieutique.



Plusieurs plates-formes proposent des outils facilitant l'implémentation de jeux sérieux.

La plate-forme Netlogo permet d'interagir directement avec les agents d'une simulation au travers de l'inspecteur d'agents et de l'observateur. De plus, cette plate-forme permet d'ajouter à l'interface graphique des composants (boutons, champs à remplir...) permettant à l'utilisateur d'interagir avec la simulation. Elle offre également la possibilité de créer des boîtes de dialogue pour demander à l'utilisateur de donner une valeur pour une variable spécifique. Enfin, avec son extension Hubnet (Blikstein *et al.*, 2005), Netlogo permet à plusieurs instances d'un même modèle de communiquer ensemble à travers un réseau local. Cette dernière fonctionnalité est particulièrement intéressante pour implémenter des jeux sérieux distribués. Néanmoins, Netlogo, de par ses nombreuses contraintes, est plus adaptée au développement de modèles simples (KISS) que de modèles descriptifs (KIDS). En plus, le fait qu'une seule fenêtre d'affichage puisse être définie est un défaut important pour la définition de jeux sérieux.

Une autre plate-forme qui propose des outils dédiés à la simulation participative est CORMAS (Becu *et al.*, 2016). En particulier, comme Netlogo, CORMAS permet d'interagir directement avec les différents agents (ou groupes d'agents) au travers de l'inspecteur d'agents, mais elle permet aussi de manipuler directement, au travers d'une interface dédiée, un ou plusieurs agents (par exemple, pour exécuter certaines de leurs actions). CORMAS permet également de distribuer le contrôle d'une simulation à travers un réseau d'ordinateurs (Becu *et al.*, 2015). Plus précisément, elle permet de donner le contrôle sur une même simulation (tournant sur un serveur) à différents clients qui peuvent chacun avoir une vue différente de la simulation et interagir avec elle selon leurs propres modalités. Enfin, CORMAS permet de revenir dans le temps d'une simulation : en utilisant le bouton "backward", l'utilisateur peut revenir à l'état de la simulation correspondant au pas de temps précédent ou simplement rejouer une simulation. Si la plate-forme CORMAS est particulièrement adaptée à une utilisation dans un contexte participatif, le développement de modèles, qui demande d'utiliser le langage Smalltalk et l'environnement VisualWorks (Brauer, 2015), peut être complexe pour des non-informaticiens. De plus, comme Netlogo, CORMAS n'est pas adapté au développement de modèles descriptifs (pas de réelle intégration des données SIG, représentation simple de l'environnement, etc.).

Pour conclure, Netlogo et CORMAS proposent de nombreux outils qui peuvent être utilisés dans le cadre de la simulation participative, mais elles sont toutes les deux limitées pour le développement de modèles descriptifs.

En conséquence de ces limitations, plusieurs projets ont choisi d'utiliser GAMA pour développer des modèles interactifs ambitieux.

Un premier exemple est SPRITE (Adam *et al.*, 2016; Taillandier et Adam, 2018), qui permet au participant de jouer le rôle d'un maire en charge de la gestion de l'île d'Oléron pendant plusieurs années, avec pour mission de trouver un compromis entre popularité, économie, attractivité, sécurité et écologie. SPRITE propose une représentation précise de l'île au travers de données géographiques et intègre un modèle avancé de submersion marine. L'intégration des données SIG aurait été très difficile à réaliser avec CORMAS, alors que les multiples vues du territoire offertes par le modèle n'auraient pas été possibles avec Netlogo.

Un autre exemple, qui concerne aussi la gestion des risques de submersion, est le jeu Littosim (Becu *et al.*, 2017). Dans Littosim, les participants jouent le rôle de gestionnaires de la planification urbaine

d'une municipalité côtière sujette à des risques de submersion marine. Ce jeu intègre plusieurs sous-modèles dont un modèle de montée des eaux et un modèle d'évolution socio-économique. L'une des particularités de Littosim est de permettre l'utilisation de contrôles distribués (à travers des tablettes) et d'un projecteur pour offrir une vue globale partagée aux participants concernant les conséquences des inondations. Comme pour SPRITE, développer ce modèle avec CORMAS ou Netlogo aurait été difficile en raison de l'utilisation intensive de données géographiques. De plus, l'architecture utilisée dans GAMA pour assurer le contrôle distribué en plusieurs simulations est beaucoup plus robuste face aux déconnexions réseau que celle proposée par Netlogo.

CityScope est un dernier exemple intéressant à citer ([Grignard et al., 2018a](#); [Alonso et al., 2018](#); [Grignard et al., 2018b](#)). Dans ce modèle, qui simule la vie quotidienne des habitants d'une ville, en particulier leur mobilité, une interface tangible (une représentation de la ville en LEGO) est utilisée pour permettre aux utilisateurs d'interagir directement avec la simulation. Les utilisateurs peuvent ainsi modifier le plan de la ville et obtenir un retour direct de l'impact de ces modifications sur la simulation. Deux types d'affichage sont utilisés pour montrer ces impacts : un premier, projeté sur la maquette en LEGO, qui fournit des informations sur les bâtiments et la mobilité des habitants, et un second, projeté sur un écran, présentant différents sous-affichages liés aux performances urbaines. Le modèle illustre plusieurs fonctionnalités de GAMA en plus de la gestion des données géographiques : la connexion d'une simulation GAMA avec un autre logiciel (ici, celui utilisé pour analyser la maquette en LEGO), la définition de plusieurs affichages riches composés de sous-affichages et la projection d'un affichage sur une surface physique spécifique (mur, table, etc.).

Ces trois exemples illustrent certaines des fonctionnalités intégrées dans la plate-forme GAMA (visualisation avancée, définition d'interaction utilisateur, simulations distribuées) que je présenterai dans la suite de cette partie. Ces travaux s'inscrivent dans un travail conjoint de l'ensemble de l'équipe de développement de GAMA et constitue donc un travail collaboratif. Je présenterai également dans cette partie un travail effectué sur l'utilisation des données recueillies au travers des jeux sérieux pour apprendre des comportements d'agents.

### 4.3.2 Fonctionnalités pour la conception de jeux sérieux avec GAMA

#### Visualisation avec GAMA

La visualisation est l'un des composants principaux des simulations à base d'agents, en particulier dans le contexte de la simulation participative ([Dorin et Geard, 2014](#)). Pour de nombreux modélisateurs, la visualisation constitue non seulement leur premier point d'entrée lors de la construction d'un modèle, mais également une méthode de plus en plus répandue de conception, de vérification et de validation des modèles. Le va-et-vient entre l'écriture du modèle et la visualisation des simulations est une démarche très courante en modélisation. La visualisation permet en effet de vérifier intuitivement les états des agents et d'affiner leurs comportements ([Grignard et al., 2013a](#); [Grignard, 2015](#)). La visualisation est également un élément clé dans les approches de simulation participative, où la visualisation du modèle sert de médiation entre les acteurs et de support à leur prise de décision.

GAMA permet aux modélisateurs de différencier facilement un modèle de sa visualisation en offrant la possibilité de définir différentes fenêtres d'affichage composées de plusieurs couches. L'utilisateur, en fonction de son rôle et de son niveau d'expertise, visualisera alors la simulation de manière spécifique et interagira avec sa représentation sans modifier le modèle initial. La figure 4.4 montre un modèle de type *boids* observé à partir de 3 perspectives différentes (3 positions différentes de la caméra). Pour

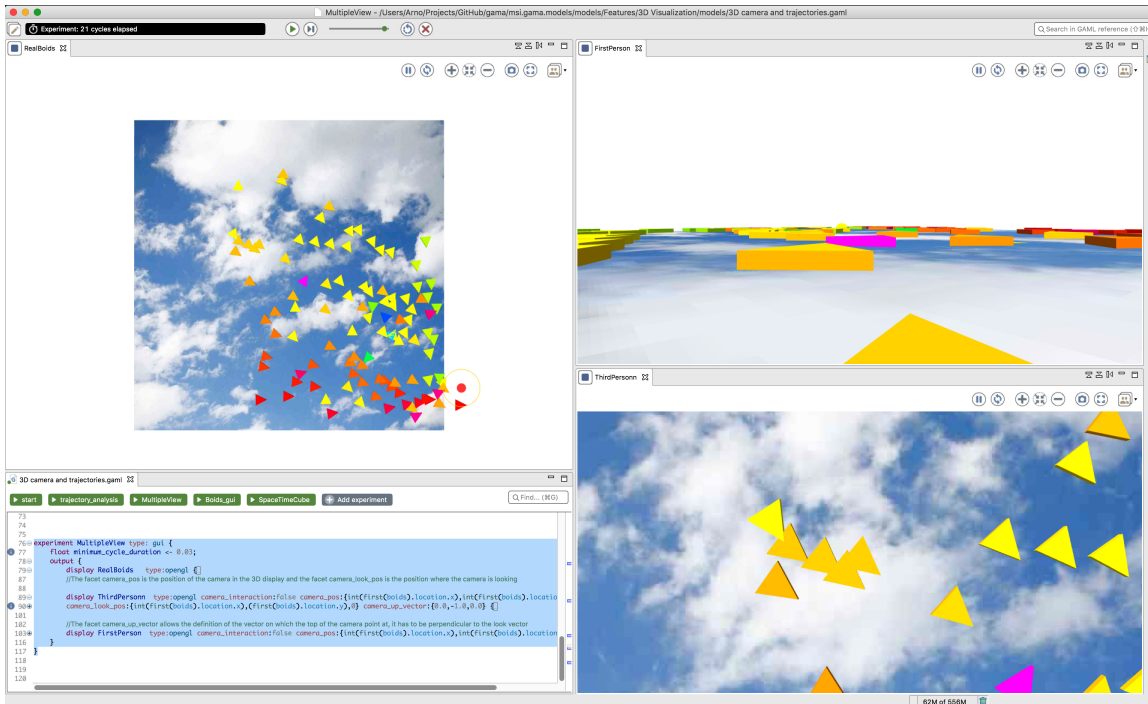


FIGURE 4.4 – Le même modèle observé selon 3 perspectives différentes.

aller plus loin, la figure 4.5 montre une simulation de trafic selon 2 points de vue : le premier affichage montre les différents agents représentant les véhicules qui se déplacent sur le réseau et génèrent du trafic (embouteillages) et un second affichage représente les embouteillages eux-mêmes. GAMA permet en effet de définir pour une expérimentation (*experiment*) autant de fenêtres d’affichage que le modélisateur le souhaite. Chaque fenêtre d’affichage est composée de différentes *couches*, pour lesquelles le modélisateur peut configurer la visibilité, la transparence, la position et la taille.

Comme indiqué par Allan (2009) et par Railsback *et al.* (2006), l’utilisation de la 3D est encore rare dans le monde de la simulation à base d’agents, alors que l’utilisation de la troisième dimension peut augmenter considérablement la propriété immersive d’une simulation. GAMA offre des fonctionnalités de visualisation avancées relatives à la visualisation 3D avec des textures, des éclairages complexes et des caméras dynamiques personnalisables au moyen de simples instructions GAML. Par exemple, le modèle CityScope utilise ces fonctionnalités 3D pour certains sous-affichages.

Enfin, comme cela a été évoqué dans la section précédente, GAMA propose des outils permettant de gérer la projection d’une fenêtre d’affichage sur une surface physique spécifique (mur, table, etc.) en permettant d’éviter toute déformation de l’image. Comme l’ont montré les projets Littosim et CityScope, cette fonctionnalité offre un fort potentiel en termes de communication des résultats d’un modèle. La figure 4.6 montre un exemple d’utilisation de cette fonctionnalité pour le projet MarrakAir (Emery *et al.*, 2017) : dans ce projet, qui vise à alerter le grand public sur la pollution de l’air de la ville de Marrakech, une simulation GAMA est affichée sur une maquette physique de Marrakech (construite avec une imprimante 3D). Cette interface concrète facilite la prise de conscience et la compréhension par un large public des effets de la pollution dans les zones urbaines.

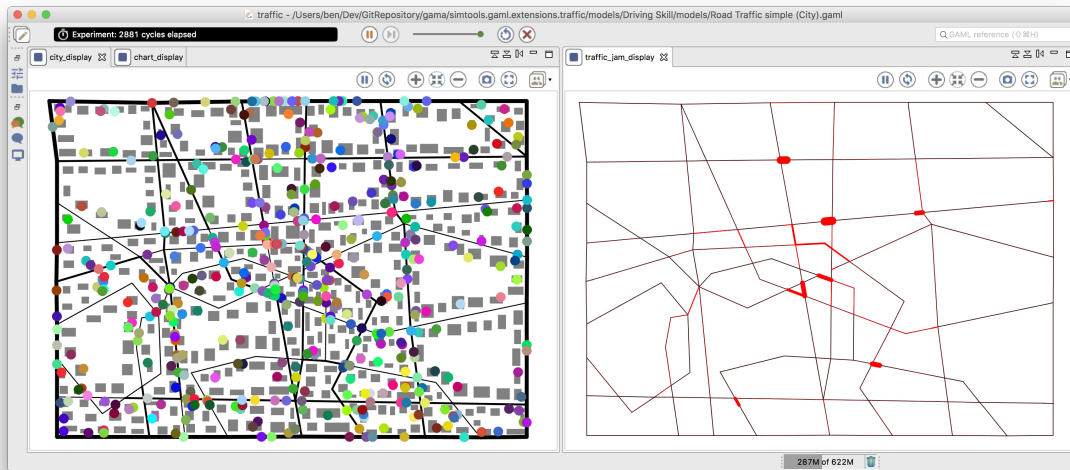


FIGURE 4.5 – Un modèle de trafic observé selon 2 points de vue.

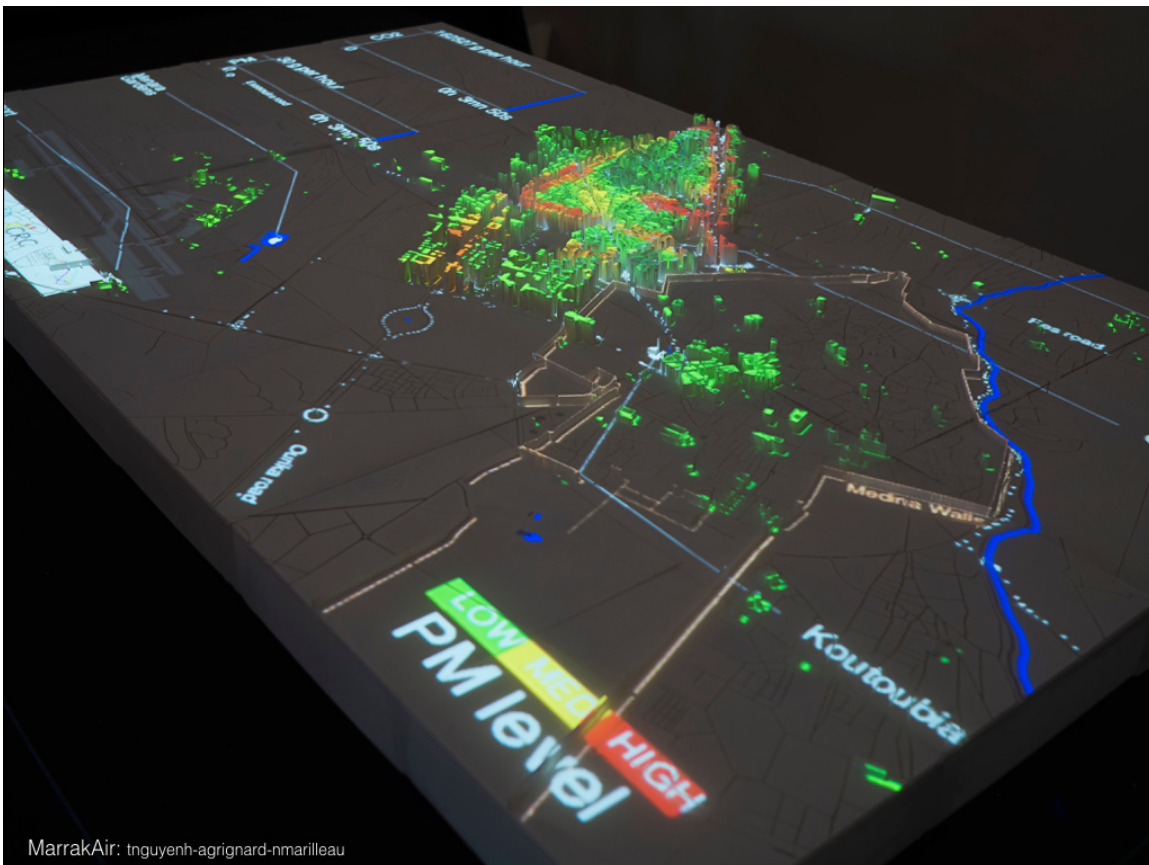


FIGURE 4.6 – Photo de l'installation MarrakAir (Emery *et al.*, 2017).

### Interaction utilisateur

Comme Netlogo et CORMAS, GAMA fournit de nombreuses fonctionnalités pour définir des interactions utilisateur, c'est-à-dire la possibilité pour les utilisateurs d'interagir avec une simulation. Si certaines de ces fonctionnalités, telles que la possibilité de modifier la valeur des variables des agents via l'inspecteur d'agent, sont directement disponibles pour tous les modèles, d'autres peuvent être spécifiées par le modélisateur à l'aide du langage GAML. Les paragraphes suivants décrivent ces fonctionnalités.

Contrairement à Netlogo, GAMA permet de définir autant de fenêtres d'affichage que nécessaire. En outre, un modélisateur peut spécifier, dans une fenêtre d'affichage, une couche spécifique, appelée couche d'événements, permettant à une action d'être déclenchée lorsqu'un événement se produit. Des exemples d'événements possibles sont le clic de la souris, son déplacement, son entrée ou sa sortie d'une zone, ou un événement du clavier. Une fois que l'événement est déclenché, l'action liée à cet événement est appelée et peut modifier l'état du monde. Le modélisateur peut accéder directement à la position de la souris dans un affichage (en termes de coordonnées dans le monde) et ainsi obtenir la liste des agents situés dans le voisinage ou chevauchant l'emplacement de la souris. Par exemple, les modèles SPRITE et Littosim utilisent ce type d'événements pour détecter les actions des joueurs à l'écran (par exemple, pour la construction des digues).

Une deuxième façon d'intégrer des interactions utilisateur dans un modèle est de définir des actions que l'utilisateur pourra déclencher directement depuis l'interface de simulation. Un exemple d'utilisation de cette fonctionnalité peut être trouvé dans l'une des versions de CityScope où il est possible de modifier le type de mobilité accepté (piéton, voiture, vélo) en cliquant sur une route et en choisissant l'action appropriée.

Comme dans Netlogo, GAMA permet au modélisateur de définir une boîte de dialogue invitant l'utilisateur à attribuer une valeur à une ou plusieurs variables via un opérateur spécifique. Le modèle SPRITE utilise cette fonctionnalité pour demander des informations au joueur (par exemple : modifier le taux d'imposition).

Une dernière façon de définir des interactions utilisateur consiste à associer à une espèce d'agents l'une des architectures de contrôle de type *user*. Ces architectures permettent aux utilisateurs de prendre le contrôle des agents de cette espèce au cours de la simulation via des panneaux de contrôle : ces panneaux permettent de définir les actions que l'utilisateur peut déclencher à chaque pas de temps en fonction de l'état de l'agent.

### Interconnexion de simulations

Une dernière fonctionnalité offerte par GAMA est la possibilité d'interconnecter des simulations. Comme l'illustrent les travaux présentés dans (Becu *et al.*, 2017) et (Grignard *et al.*, 2018a), GAMA permet à une simulation de communiquer avec d'autres programmes via différents protocoles (TCP, UDP, MQTT). Cette fonctionnalité permet, par exemple, à une simulation GAMA de communiquer avec une autre simulation GAMA ou avec divers terminaux tels que des smartphones, des tablettes ou des capteurs pour exporter des sorties ou collecter des données. L'idée de base est que tout agent du modèle peut obtenir la capacité, via un *skill* GAML dédié, de se connecter à un serveur particulier et d'envoyer des messages à tout autre agent qui se trouve dans une simulation connectée à ce serveur. Les données envoyées sont sérialisées automatiquement avant d'être envoyées en tant que contenu de messages : cela permet à un agent d'envoyer des données de n'importe quel type via le réseau.



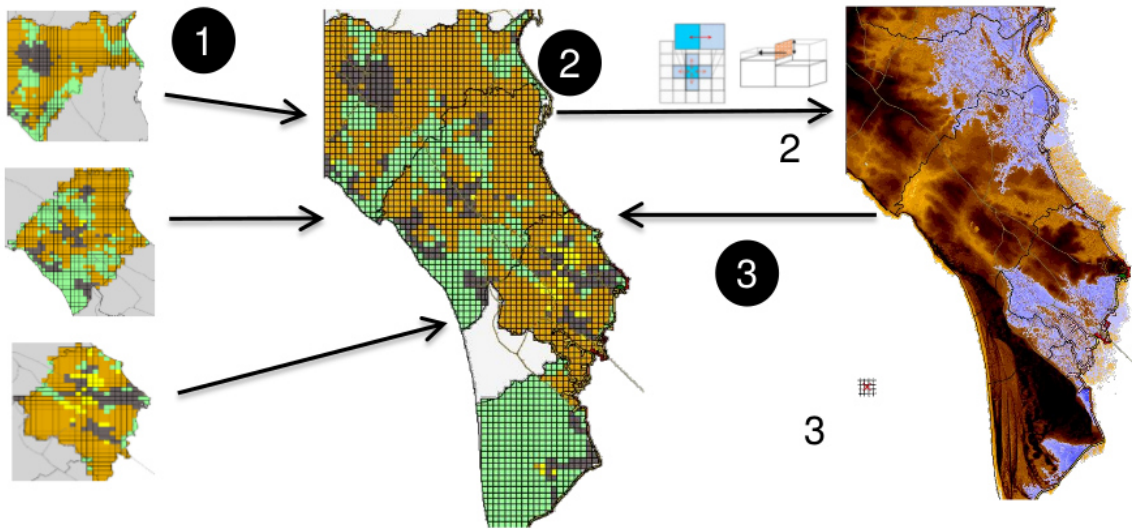


FIGURE 4.7 – Processus de Littosim : 1) les données sont envoyées par les clients (joueurs); 2) le serveur envoie les données à un modèle d'inondation; 3) le modèle d'inondation renvoie les résultats au serveur (Becu *et al.*, 2017).

Les fonctionnalités d'interconnexions de simulations de GAMA offrent l'avantage d'être résilientes par rapport aux problèmes de connexion. De plus, elles permettent d'accepter des connexions avec tout type de terminaux. Par exemple, il est possible de connecter des simulations GAMA exécutées sur différents ordinateurs avec une interface Android déployée sur un téléphone mobile contrôlant le jeu. Le projet Littosim utilise ce type d'architecture (Figure 4.7) :

1. chaque client est contrôlé par un joueur, les données sont envoyées à la simulation faisant office de serveur,
2. une fois toutes les données recueillies, le serveur renvoie toutes ces données à un modèle d'inondation (programme externe qui n'est pas codé avec GAMA),
3. les résultats de la simulation d'inondation sont renvoyés d'abord au serveur qui va les renvoyer à chaque joueur qui pourra ensuite évaluer les dommages sur son territoire.

### 4.3.3 Apprentissage de comportements par des jeux sérieux

#### Introduction

Un problème récurrent en modélisation à base d'agents concerne la définition des comportements des agents. En effet, il est parfois difficile de formaliser la façon dont des entités prennent des décisions, en particulier lorsqu'il s'agit de modéliser le comportement d'acteurs humains. Une approche pour faire face à cette difficulté consiste à utiliser des simulations participatives pour recueillir des données sur les comportements attendus. L'idée est en effet de laisser des participants humains contrôler des agents dans la simulation et d'enregistrer leurs choix d'actions afin d'apprendre à partir des données collectées des comportements pour les agents. Mettre en place une telle approche requiert de disposer d'une architecture adaptée à un tel processus d'apprentissage et d'une méthode d'apprentissage qui permette de "configurer" cette architecture pour reproduire au mieux les comportements des partici-



pants.

L'utilisation de la simulation participative pour apprendre le comportement d'acteurs a fait l'objet de différents travaux. Pour n'en citer que quelques-uns, (Sempé *et al.*, 2005) a proposé une approche dans laquelle les acteurs humains jouent leur propre rôle dans une simulation et interagissent avec des agents artificiels qui remettent en question leurs comportements. L'objectif est d'utiliser ces interactions pour stimuler les réactions des acteurs et ainsi révéler des connaissances cachées. Un autre travail intéressant est celui de Chu *et al.* (2008), qui propose une approche visant à apprendre le comportement de l'expert au moyen d'interactions entre l'expert et le système. Ainsi, l'expert observe le comportement d'un agent et a la possibilité de le corriger si celui-ci n'est pas pertinent. L'agent prend en compte cette intervention pour affiner son comportement. Dans ce travail, l'architecture d'agents utilisée repose sur une méthode d'aide à la décision multicritère. Plus précisément, elle utilise une moyenne pondérée de différents critères pour choisir l'action à appliquer : la révision du comportement de l'agent consiste ainsi à modifier le poids des différents critères.

S'inscrivant dans la continuité de cette dernière approche, j'ai proposé avec Chu Thanh Quang (Taillandier *et Chu*, 2009) une approche d'apprentissage de comportement d'agents basée sur le recueil par la simulation participative. Par rapport aux travaux présentés dans (Chu *et al.*, 2008), nous avons fait le choix de complexifier l'architecture d'agents utilisée de façon à permettre la prise en compte de changements de mode de fonctionnement en fonction du contexte : par exemple, une personne se trouvant dans un contexte de crise ne va pas forcément prioriser les actions qu'il va entreprendre de la même façon que dans un contexte normal. A noter que ce travail a fortement bénéficié des travaux que j'ai pu mener durant mon doctorat sur la révision des connaissances.

### Architecture

Nous avons ainsi proposé une nouvelle architecture d'agents basée sur l'utilisation d'une fonction d'utilité : l'agent choisit à chaque pas de temps l'action qui maximise l'utilité parmi l'ensemble  $A_s$  représentant les actions que l'agent peut appliquer quand il se trouve dans l'état  $s$ . Plus précisément, la fonction d'utilité sera composée d'un ensemble de règles de régression prenant la forme suivante :

$$\text{Si } condition \text{ alors } utility(a) = \frac{1}{\sum_{i \in C} w_i} \times \sum_{i \in C} w_i \times Val_i(a)$$

Avec  $C$  l'ensemble des critères caractérisant les actions possibles ;  $w_i$  le poids associé au critère  $i$  et  $Val_i(a)$ , la valeur du critère  $i$  pour l'action  $a$  appartenant à l'ensemble  $A_s$ .

Cette représentation permet de définir des fonctions d'utilité expressives qui restent facilement interprétables par les experts du domaine, ce qui facilite ainsi leur validation.

### Apprentissage des règles de régression

L'apprentissage des règles de régression se base sur l'ensemble des données recueillies par la simulation participative : un acteur humain interagit directement dans la simulation en jouant le rôle d'un agent. À chaque pas de temps de la simulation, l'action qu'il choisit est enregistrée.

Une fois les données recueillies, une deuxième étape de notre approche consiste à analyser ces données pour apprendre un comportement pour l'agent, c'est-à-dire, un ensemble de règles de régression

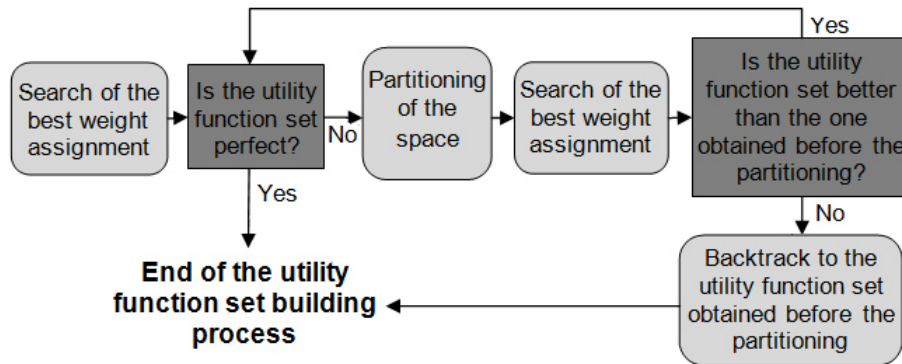


FIGURE 4.8 – Apprentissage des règles de régression (Taillandier et Chu, 2009).

permettant de se rapprocher le plus possible des choix d’actions de l’acteur humain. Nous avons proposé pour cela une méthode basée sur 2 grandes étapes (Figure 4.8) : la recherche des meilleurs poids de critères pour chaque règle de régression et le partitionnement de l’espace caractérisant l’état de l’agent (ce qui correspond à l’ajout de nouvelles règles de régression).

A l’état initial, une seule règle de régression est définie : l’espace d’états de l’agent est composé d’une seule partition. Lors de la première étape, le système recherche une affectation de poids qui optimise l’adéquation entre la fonction d’utilité et le comportement de l’acteur humain. Si la combinaison de poids permet à l’agent d’être en totale adéquation avec le comportement de l’acteur humain, le processus se termine. Sinon, de nouvelles règles de régression sont introduites : pour chaque partition de l’espace posant problème, le système calcule des sous-partitions afin de détecter des états de l’agent incompatibles avec les autres. Ensuite, une nouvelle affectation de poids est à nouveau recherchée pour l’ensemble des règles de régression. Si l’affectation de poids obtenue après le partitionnement permet d’obtenir un meilleur résultat que le précédent, elle est conservée. Sinon, le système revient aux règles de régression précédentes et met fin au processus d’apprentissage. Cette procédure de partitionnement est répétée jusqu’à ce que les règles de régression apprises permettent d’obtenir un comportement en adéquation totale avec le comportement de l’acteur humain ou jusqu’à ce qu’il n’y ait plus d’amélioration des règles de régression.

Nous avons proposé de formuler le problème de la meilleure affectation de poids sous la forme d’un problème de minimisation de l’inadéquation entre le jeu de règles de régression et le comportement de l’acteur humain (les exemples recueillis). La taille de la zone de recherche étant la plupart du temps trop importante pour permettre une recherche complète, nous avons donc proposé d’utiliser des algorithmes génétiques [Holland et al. \(1992\)](#) pour trouver la meilleure affectation de poids des critères ; ce type d’algorithmes est particulièrement efficace lorsque l’espace de recherche est bien structuré comme dans notre problème de recherche.

Concernant le partitionnement de l’espace d’états de l’agent, nous avons proposé d’utiliser des techniques d’apprentissage supervisé pour le réaliser. L’objectif est de rechercher les parties de l’espace d’états de l’agent ayant un comportement différent en termes de fonctions d’utilité. Ainsi, nous cherchons à détecter, pour chaque règle de régression, les parties de l’espace qui contiennent des exemples incompatibles. Pour chaque règle de régression, nous construisons à partir des exemples recueillis par la simulation participative un ensemble d’apprentissage composé de couples (*état de l’agent, compa-*

tible ?). Ensuite, un algorithme d'apprentissage supervisé est utilisé pour partitionner l'espace d'états de l'agent lié à la règle de régression considérée. L'enjeu étant d'exprimer les partitions de l'espace d'états de l'agent sous forme de règles, nous avons choisi dans ce travail d'utiliser l'algorithme d'apprentissage RIPPER [Cohen \(1995\)](#).

### Discussion

L'approche que nous avons proposée dans ce travail a été appliquée pour une simulation de gestion des secours en cas de séisme à Hanoi (Vietnam). L'idée était d'apprendre le comportement des ambulanciers, en particulier par rapport à leur stratégie de recherche des blessés. L'expérimentation complète, ainsi que les résultats sont présentés dans [\(Taillandier et Chu, 2009\)](#).

J'ai par la suite poursuivi ce travail en proposant une approche similaire, mais se basant cette fois sur une autre méthode de décision multicritère que des règles de régression. J'ai en effet dans [\(Taillandier et Buard, 2009\)](#) proposé d'utiliser la méthode ELECTRE [\(Roy, 1990\)](#) pour formaliser la prise de décision des acteurs. Ce travail a été appliqué pour un modèle de déplacement des éléphants dans le parc de Hwange au Zimbabwe.

Si je n'ai malheureusement pas eu l'occasion de retravailler sur cette thématique d'apprentissage de comportements d'agents depuis, je souhaite néanmoins m'y replonger dans le cadre du projet ANR ACTEUR. L'un des objectifs du projet est en effet de proposer des méthodes pour aider les modélisateurs à définir les comportements de leur agent en utilisant la simulation participative. Plus précisément, l'enjeu est de laisser un acteur humain jouer le rôle d'un agent dans la simulation pour aider à paramétrer un modèle utilisant l'architecture BEN (voir Chapitre 3) : l'idée n'est pas d'apprendre complètement un modèle de comportement, mais plutôt d'aider le modélisateur à définir certains paramètres du modèle tels que les priorités des plans ou des désirs. Une première étape de ce travail a déjà été réalisé en recueillant des données issues du jeu SPRITE [\(Taillandier et Adam, 2018\)](#). L'étape suivante est maintenant de mettre en place une procédure pour paramétrer automatiquement le comportement d'un agent artificiel à partir de ces données.

## 4.4 Conclusion

Dans ce chapitre, j'ai présenté les travaux que j'ai pu mener sur le développement d'outils visant à faciliter la mise en place de démarches participatives. Bien que ces travaux aient été assez espacés dans le temps, cette thématique reste pour moi un domaine de recherche important. En effet, l'expérience que j'ai pu accumuler au cours de mes différents projets de recherche me conforte dans l'idée que l'implication directe des acteurs des territoires dans les processus de modélisation et de simulation est extrêmement importante pour l'appropriation des enseignements que l'on peut tirer des modèles.

Je souhaite ainsi continuer à promouvoir, dans mes futurs projets, le recours aux approches participatives et travailler sur des outils permettant de faciliter leur utilisation.

Un premier enjeu est ainsi de continuer à proposer des façons de co-construire des modèles qui soient à la fois intuitives et peu contraintes, mais qui garantissent une grande expressivité des modèles. Dans ce cadre, le travail que j'ai effectué sur l'outil de modélisation graphique de GAMA est un premier pas dans cette direction, même s'il est loin d'être totalement abouti. Je désire donc continuer à faire progresser GAMA sur ce point.

Pour ce qui est de la simulation participative, une piste de recherche qui me paraît intéressante porte sur le travail sur les modalités d'interaction entre un ou plusieurs utilisateurs et une simulation. Un exemple qui montre bien le potentiel et l'intérêt d'intégrer à ce niveau des interfaces tangibles est CityScope ; dans ce projet mené au Media Lab du MIT, des utilisateurs peuvent interagir avec une simulation au travers d'une maquette en LEGO. J'aimerais ainsi pouvoir intégrer dans GAMA des outils permettant à des modélisateurs de pouvoir facilement reproduire ce type de dispositif en utilisant le langage GAML. Une autre voie de recherche intéressante dans ce domaine serait de coupler GAMA à un moteur de jeu tel que Unity afin de pouvoir proposer des affichages 3D de haute qualité et des interactions riches en temps réel avec un environnement 3D. Le défi est ici de conserver la simplicité de la modélisation avec GAMA, même dans des environnements 3D complexes.



## 5 — Conclusion et perspectives

### 5.1 Résumé de mes travaux et projets de recherche

#### 5.1.1 Bilan de mes travaux de recherche

Ce document a présenté les principaux travaux de recherche que j'ai conduits depuis la fin de ma thèse. Ces travaux ont porté sur l'aide au développement de modèles à base d'agents pour l'étude des systèmes complexes. J'ai ainsi introduit, dans un premier chapitre, le domaine de la modélisation à base d'agents pour l'étude des systèmes complexes en détaillant les étapes nécessaires à la mise en œuvre d'une telle approche et le rôle des plates-formes de modélisation. Cette introduction a permis de faire un état des lieux des plates-formes de modélisation en 2008 afin d'en souligner les limitations concernant l'intégration des dimensions spatiales, comportementales et participatives. J'ai enfin présenté la plate-forme GAMA qui a cherché, depuis sa création, à combler ces limitations et dans laquelle se sont inscrits tous mes travaux de recherche.

J'ai ensuite abordé dans un deuxième chapitre la problématique de la gestion de l'espace dans les simulations à base d'agents. Cette problématique, qui m'a particulièrement occupée depuis 2008, m'a amené à introduire dans GAMA de nombreuses fonctionnalités permettant à cette plate-forme d'être aujourd'hui une référence pour la gestion de l'espace et des données spatiales. Elle m'a également amené à travailler sur des questions liées à la modélisation du trafic routier et à la spatialisation de populations synthétiques.

Le chapitre suivant a abordé une autre thématique sur laquelle je me suis fortement impliquée ces dernières années : la modélisation du comportement des acteurs. J'ai, dans ce cadre, proposé différents formalismes et architectures pour modéliser la prise de décision des acteurs. Le travail le plus abouti dans ce contexte est l'architecture BEN qui introduit des composantes cognitives, sociales, émotionnelles et normatives dans le processus de décision.

Enfin, j'ai présenté dans le chapitre 4 mes travaux sur le développement d'outils d'aide à la mise en place de démarches participatives. Ces travaux m'ont conduit à intégrer dans GAMA un outil de modélisation graphique ainsi qu'à travailler sur l'intégration dans la plate-forme de fonctionnalités

pour le développement de simulations interactives et de jeux sérieux. Enfin, j'ai proposé différentes approches d'apprentissage automatique de comportements d'agents par la simulation participative.

La plupart de ces travaux se sont inscrits dans le cadre de projets financés. En effet, j'ai toujours cherché à rendre opérationnel mes travaux, ce qui demande à la fois des ressources matérielles (ordinateurs) et humaines (ingénieurs, post-doctorants, etc.). Ainsi, les travaux présentés au chapitre 2 sur l'espace, ont été financés par : (i) le projet ANR 3-World, sur lequel j'ai été post-doctorant entre 2009 et 2010, (ii) le projet GRR (Régional) MOSAIIC et (iii) le projet ANR GEN\*, dont j'étais responsable scientifique pour l'UMR IDEES. Les travaux présentés dans le chapitre 3 sur la cognition des agents ont été financés par le projet RTRA STAE MAELIA, sur lequel j'ai été post-doctorant entre 2010 et 2011 et surtout par le projet ANR ACTEUR que je coordonne depuis 2014. Enfin, les travaux présentés dans le chapitre 4 ont également été en partie financés par l'ANR ACTEUR.

Pour conclure sur ce bilan, mes travaux ont contribué, en collaboration avec d'autres chercheurs, à enrichir les plates-formes de modélisation et ainsi à offrir à la modélisation à base d'agents de nouveaux terrains d'application. En effet, les plates-formes actuelles, qui ont gagné en facilité d'utilisation, permettent aujourd'hui de développer des modèles à large échelle, intégrant de nombreuses données et une représentation fine de l'espace, ainsi que des interactions riches avec les utilisateurs. Les plates-formes de modélisation à base d'agents ont ainsi beaucoup évolué depuis 2008 et devraient encore connaître des évolutions dans les années à venir. Je propose donc, dans la partie suivante, ma vision sur les évolutions possibles de ces plates-formes dans les onze prochaines années.

### 5.1.2 La modélisation à base d'agents à l'horizon 2030

Ces dernières années ont connu des progrès importants en termes de puissance de calcul, d'accès aux données, mais également d'algorithmes. Ces progrès, qui ont déjà commencé à impacter la façon de construire des modèles à base d'agents, vont dans les années à venir accélérer ce phénomène. Si le développement de modèles simples et abstraits garde et gardera toujours un réel intérêt pour répondre à certaines questions, les plates-formes de demain permettront d'aller plus loin en termes de réalisme des simulations, mais aussi de proposer des modèles à de plus larges échelles. Elles permettront également de mettre en place de nouvelles façons de construire des modèles et d'interagir avec les simulations.

Ainsi, pour moi, ces évolutions futures seront de différents ordres :

- amélioration des capacités de calculs pour des simulations à plus large échelle,
- environnement 3D et réalité virtuelle,
- nouvelles façons d'interagir avec les simulations,
- nouvelles façons de construire des modèles,
- des agents au comportement plus crédible,
- une meilleure intégration dans les processus de prospective.

Une première évolution sera la généralisation du calcul distribué. Si certaines plates-formes comme GAMA permettent déjà aujourd'hui de distribuer des calculs sur plusieurs cœurs du CPU, ces dernières sont encore loin d'utiliser au mieux toute la puissance disponible, en particulier si l'on prend en compte également les cartes graphiques. D'autres plates-formes se sont, elles, spécialisées dans le recours au calcul intensif sur GPU telles que FLAME-GPU ([Richmond et Chimeh, 2017](#)). Mal-



heureusement, ces plates-formes sont extrêmement complexes à utiliser en particulier par des non-informaticiens. Une évolution à venir est donc l'intégration, dans les plates-formes classiques, d'outils pour faciliter l'utilisation des calculs distribués sur carte graphique et de les rendre, si possible, transparents.

Un domaine qui a su tirer toute la puissance des cartes graphiques est celui des jeux vidéo. Ainsi, une solution pour mieux bénéficier des puissances de calcul de demain est de coupler les plates-formes de simulation à des moteurs de jeux, comme par exemple Unity. Utiliser ce type de moteur apporterait plusieurs bénéfices : permettre de créer des simulations dans de vrais environnements 3D, créer des simulations visuellement proches de la réalité, permettre à des utilisateurs de s'immerger, voire d'interagir avec les simulations à l'aide de la réalité virtuelle. Néanmoins, il existe de nombreux verrous pour ce couplage. Le plus important concerne la facilité de développement des modèles. En effet, l'intérêt des plates-formes de simulation telles que Netlogo, GAMA ou CORMAS est qu'elles sont simples à utiliser et qu'elles permettent de produire un modèle simulable en quelques dizaines de lignes de codes, ce qui est loin d'être le cas avec les moteurs de jeux ; Unity, le moteur le plus accessible, nécessite un travail long et fastidieux avant d'arriver à recréer un environnement virtuel. Il reste donc un travail important qui va, je l'espère, avancer dans les années à venir, pour permettre un véritable couplage entre ces deux types de plates-formes. Cela offrirait la possibilité aux modélisateurs de décrire simplement, à l'aide d'un langage de modélisation de type GAML, un modèle 3D complexe utilisant la puissance d'un moteur de jeux.

Le couplage entre plates-formes de simulation et moteurs de jeux permettrait de créer de nouvelles façons d'interagir avec les simulations au travers, entre autres, de la réalité virtuelle ou de la réalité augmentée. Néanmoins, je pense que les années à venir nous réservent également de nouvelles façons d'interagir avec les simulations. Un bon exemple de ce qui peut se faire aujourd'hui et qui va, à mon avis, se généraliser est l'utilisation d'interfaces physiques (tangibles) pour interagir avec les simulations. C'est par exemple ce qui a été fait dans le projet CityScope où une maquette en légo est utilisée pour interagir avec la simulation. D'autres exemples sont donnés par l'équipe de développement de CORMAS dans la vidéo qu'ils ont produit sur la thématique "CORMAS in 10 years" ([Cormas, 2018](#)), en particulier dans le cadre de jeux sérieux. Ainsi, ils imaginent par exemple un cas où chaque joueur pourra entrer les actions qu'il souhaite réaliser directement depuis son smartphone. S'il est d'ores et déjà possible de mettre en place de tels dispositifs avec les technologies actuelles, cette mise en place demande beaucoup de travail de développement qui doit être réalisé pour chaque nouvelle application. Ainsi une évolution qui devrait, je pense, arriver dans les prochaines années serait l'intégration dans les plates-formes de modélisation des outils permettant de mettre en place simplement ce type de dispositif.

Dans une même perspective, afin de faciliter la participation d'acteurs dans le processus de modélisation ou de simulation, de nouvelles évolutions devraient concerner la façon de construire des modèles. En effet, si je reste persuadé que l'écriture de lignes de codes est la meilleure façon de définir un modèle lorsque l'on travaille seul (ou en petit nombre où chacun est expert de la plate-forme utilisée), ce type d'approche n'est que rarement approprié pour co-construire un modèle. GAMA, mais surtout CORMAS, proposent aujourd'hui des outils graphiques pour faciliter la discussion autour d'un modèle et sa co-construction. Néanmoins, des progrès seront sans doute réalisés dans les années à venir pour améliorer ces outils et les rendre plus ergonomiques. On pourrait également imaginer des outils proches de ce qui se fait actuellement avec Scratch ([Resnick et al., 2009](#)) où la construction de modèles reposerait sur la composition de « pièces de puzzle » représentant des sous-modèles. Je pense égale-

ment que d'autres approches pourraient émerger, par exemple, en liant plus fortement modélisation et simulation. Ainsi, il serait possible de créer un modèle directement depuis une interface de simulation : après avoir créé un environnement virtuel, le modélisateur pourrait directement, par des clics de souris, peupler ce monde avec un agent en définissant son comportement à partir d'un ensemble de primitives simples (là encore, une source d'inspiration pourrait être Scratch), puis éventuellement dupliquer cet agent pour créer toute une population d'agents ou modifier à la volée son comportement.

Un point crucial dans la modélisation à base d'agents est la définition du comportement des agents. Beaucoup de choses ont été proposées ces dernières années pour offrir aux modélisateurs des outils pour les aider à mieux décrire leur comportement, notamment au travers de l'intégration d'architectures BDI dans les plates-formes de simulation. Je pense clairement que cette tendance visant à proposer des architectures d'agents de plus en plus complètes va perdurer et qu'elles permettront demain de gérer de nouvelles dimensions sociales ou affectives en fonction des besoins des modélisateurs. Un autre aspect qui devrait s'amplifier est la possibilité d'avoir des modèles composés d'agents mixtes dont certains auront leur comportement décrit par des bases de règles ou des équations, d'autres au travers d'une architecture de type BDI, et enfin certains au travers de modèles construits par apprentissage artificiel. En effet, nous disposons maintenant de plus en plus de données et les méthodes d'apprentissage artificiel (Deep learning en tête) ont connu ces dernières années un véritable essor. Si, bien évidemment, il n'est pas possible d'apprendre le comportement de tous les agents, ne serait-ce que pour des raisons de prise en compte d'éléments disruptifs dans les modèles (par exemple si l'on souhaite tester l'impact de nouvelles pratiques agricoles qui n'existent pas encore aujourd'hui), compléter les modèles par des agents dont le comportement a été appris directement à partir de données peut s'avérer intéressant.

Un dernier point que je souhaite aborder est l'utilisation des modèles et des simulations. La modélisation à base d'agents a maintenant acquis un certain niveau de maturité qui lui permettra à l'avenir d'être plus utilisée dans un contexte d'aide à la réflexion et de prospective. S'il est important d'être toujours prudent quant à l'utilisation des modèles, et cela quel que soit le type de modèles utilisés (à base d'agents ou non), je pense que les modèles à base d'agents, et plus particulièrement les modèles descriptifs, offrent de nombreux atouts pour une utilisation en prospective. Pour cela des réflexions méthodologiques sur la construction des modèles et l'analyse des résultats des simulations (en liens avec les scénarios) seront nécessaires.

Pour conclure, je compte dans les années à venir contribuer à ces évolutions, même s'il ne sera pas possible pour moi de participer à tous ces chantiers. La partie suivante, qui constitue mon projet de recherche, propose donc 3 axes de recherche issus de ces réflexions, dont deux sont en lien direct avec mes travaux précédents sur les dimensions comportementales et participatives :

- *modélisation de la décision et argumentation* : qui porte dans un premier temps sur l'enrichissement de BEN sur la formalisation des plans d'actions et des dynamiques de croyances, et dans un second temps sur la représentation des processus d'échange d'arguments.
- *simulation interactive par interfaces tangibles et physicalisation des données* : qui porte sur le développement d'outils permettant de spécifier de nouvelles façons d'interagir et d'explorer des modèles de simulation en utilisant l'environnement physique.
- *prospective par la modélisation à base d'agents* : qui porte sur l'intégration de modèles à base d'agents dans un contexte de prospective pour l'évaluation de scénarios et son couplage à des approches d'analyse de cycle de vie (ACV).

De même que pour mes travaux précédents, je compte déposer des projets (ANR, ERC, etc.) pour financer ces travaux de recherche.

### 5.1.3 Projet de recherche

#### Modélisation de la décision et argumentation

L'architecture BEN fournit aujourd'hui un cadre complet pour définir des agents au comportement riche. Si un travail très important a été réalisé sur l'architecture pour intégrer différentes composantes émotionnelles et sociales, le moteur cognitif de l'architecture reste encore largement perfectible. En effet, BEN ne propose pas de formalisme particulier pour les plans, qui sont juste constitués d'une série d'instructions. De même, cette architecture ne propose pas non plus d'outils pour gérer les dynamiques des croyances les unes par rapport aux autres, par exemple sur la façon dont un agent doit agir lors de l'acquisition d'une nouvelle croyance en contradiction avec une précédente croyance.

Ainsi, une première perspective de recherche est d'enrichir BEN sur la formalisation des plans et la fusion des croyances.

J'aimerais tout d'abord travailler sur une meilleure formalisation des plans de façon à permettre aux modélisateurs de définir plus simplement des plans flexibles composés d'un ensemble d'activités. L'idée est d'offrir des primitives GAML permettant au modélisateur de spécifier comment est composé un plan : par exemple, spécifier des règles de précedence et de parallélisation entre activités. Dans ce cadre, J'ai commencé une collaboration avec Roger Martin-Clouaire (DR INRA, MIAT), qui a déjà beaucoup travaillé sur cette question, en particulier au travers de la plate-forme DIESE ([Martin-Clouaire et Rellier, 2009, 2002](#)) qui fournit un ensemble de primitives pour composer des plans. Mon objectif est ainsi de réfléchir à comment intégrer ces aspects dans BEN.

Un deuxième aspect qu'il m'intéresserait d'explorer est la gestion de la fusion des croyances : comment un agent va gérer le fait de disposer de plusieurs croyances se renforçant ou au contraire contradictoires. Prenons l'exemple d'un modèle portant sur le comportement d'agriculteurs, une question qui peut se poser est de savoir comment ces derniers vont gérer des informations potentiellement contradictoires venant de leur expérience, d'autres agriculteurs ou de conseillers agricoles. Une piste intéressante pour répondre à ce problème est d'utiliser la théorie des fonctions de croyance. Ainsi, [Thiriot \(2016\)](#) propose d'utiliser cette approche pour fusionner les croyances d'agents sur les conséquences de leurs actions dans le contexte d'un système normatif. Je souhaite m'inspirer de ces travaux pour intégrer dans BEN un système de fusion des croyances.

Également en lien avec la question de la modélisation des comportements, je souhaiterais m'intéresser à la question des processus d'échange d'arguments, en particulier dans un contexte de diffusion d'opinions et d'innovations. Cet intérêt découle des travaux entrepris depuis 2018 avec Rallou Thomopoulos (DR INRA, CEPIA) sur l'étude de la diffusion des diètes végétariennes.

Les processus de diffusion d'opinions et d'innovations ont fait l'objet de nombreux travaux en simulation à base d'agents ([Kiesling et al., 2012](#)). Néanmoins, la plupart de ces travaux se limitent à une représentation très simplifiée et abstraite du processus de diffusion : l'espace est souvent ignoré ou représenté sous la forme d'une grille, le choix d'adopter une opinion ou innovation est traduite sous la forme d'une simple valeur numérique, etc. Cette représentation très abstraite peut poser des questions

en termes de réalisme et surtout d'explicabilité des changements d'opinion des acteurs.

Dans ce contexte, nous proposons d'adopter une approche contraire en représentant explicitement les échanges d'arguments entre acteurs. Plus précisément, nous souhaitons coupler une formalisation des arguments utilisés par les acteurs sous la forme de graphes d'argumentation (Dung, 1995; Karanikolas *et al.*, 2018) avec l'architecture BEN. Les arguments seront alors représentés sous la forme de croyances pour les agents, et chaque agent aura la capacité de construire son propre réseau d'arguments. Afin de prendre en compte la différence de perception des arguments par les différents acteurs, nous proposons d'attacher des "valeurs" à chaque argument (Bench-Capon, 2003) ainsi que des préférences pour les agents vis-à-vis de chacune de ces valeurs.

En plus de la question de la diffusion des diètes végétariennes, je souhaite appliquer cette approche pour une autre problématique : celle de la diffusion et de l'appropriation des technologies numériques en agriculture. En effet, aujourd'hui, alors que les problématiques environnementales prennent de plus en plus de place dans le débat public, la question de produire mieux, c'est à dire de permettre de garder un fort rendement tout en étant plus respectueux de l'environnement, est un enjeu fort. Les technologies numériques pourraient participer à relever ce défi majeur. Néanmoins, si les agriculteurs sont conscients de l'apport des nouvelles technologies, leur utilisation de ces dernières reste encore modeste. Effectivement, s'il existe aujourd'hui une grande variété d'outils permettant aux agriculteurs de mieux contrôler leur exploitation (capteurs, drone, images satellites), de faciliter les opérations techniques (véhicule autonome, nouveau système d'irrigation) ou encore de trouver plus facilement des conseils (internet, réseaux sociaux), ceux-ci restent encore d'usage confidentiel par rapport à leur potentiel. Cet état de fait peut s'expliquer par l'existence de nombreux freins à l'utilisation de ces outils : manque de connaissance des outils disponibles, mauvaise image de certains outils, manque de technicité ou encore prix élevé de certains équipements. La question est ainsi de savoir comment participer à lever ces freins et plus particulièrement s'il est possible de mettre en place des politiques publiques à l'échelle territoriale telles que l'organisation de formations ou des aides à l'équipement de façon à permettre une meilleure appropriation de ces outils tout en évitant une montée des inégalités.

### **Simulation interactive par interfaces tangibles et physicalisation**

Un deuxième axe de recherche qu'il m'intéresserait d'explorer est celui des simulations interactives et plus précisément, l'utilisation d'interfaces tangibles et de la physicalisation des données pour explorer les modèles de simulation.

En effet, de nombreux travaux ont déjà montré l'intérêt d'ajouter de l'interactivité aux simulations à des fins d'apprentissage et de compréhension (par exemple, (Chittaro et Sioni, 2015)). Dans ce domaine, une tendance récente consiste à utiliser des formes innovantes d'interfaces utilisateur, telles que les interfaces tangibles qui permettent aux utilisateurs d'interagir avec une simulation par le moyen de l'environnement physique. Faciles à appréhender par les nouveaux arrivants (Shaer *et al.*, 2010), de telles interfaces ont déjà été utilisées dans de nombreux domaines. Un exemple d'interface tangible est celle proposée par le projet CityScope (Alonso *et al.*, 2018), qui permet à des utilisateurs d'interagir avec une simulation au travers d'une maquette en LEGO représentant le plan d'une ville : la modification de la maquette va permettre aux utilisateurs de directement voir les impacts de ces modifications sur au travers de la simulation. Récemment, des travaux ont également montré que l'intégration de données numériques dans le monde physique facilite l'exploration et la compréhension des données (Jansen *et al.*, 2015) : la physicalisation des données est un nouveau domaine de recherche qui vise à comprendre comment le rendu physique de données, par des maquettes par exemples, peut contri-

buer à la cognition de l'utilisateur, la résolution de problèmes et la prise de décision. La combinaison d'interfaces tangibles et de la physicalisation des données pourrait ainsi permettre l'émergence de nouvelles façons d'interagir et d'explorer des modèles de simulation et de faciliter grandement l'appropriation des résultats de simulation par les utilisateurs des modèles.

Cependant, mettre en place une telle approche nécessite de répondre à de nombreuses questions concernant la spécification de ces interactions et la façon de physicaliser les résultats de simulation. En effet, l'ambition pour moi, toujours en lien avec mes travaux sur la plate-forme GAMA, est de proposer des outils accessibles au plus grand nombre permettant de faciliter la conception et le développement de tels dispositifs. Pour mener à bien ce travail, je pourrais m'appuyer sur différentes collaborations : déjà avec Arnaud Grignard (CR MIT, Media-lab) au travers de son travail sur le projet CityScope ; ensuite, avec les membres de l'équipe Eclipse de l'UMR IRIT qui ont une forte expérience dans le développement d'interfaces tangibles pour l'exploration de données (Perelman *et al.*, 2015).

### **La prospective par la modélisation à base d'agents**

Ces dernières années m'ont permis de participer à différentes actions et formations autour de la prospective et m'ont amené à me poser la question de l'utilisation de modèles à base d'agents descriptifs dans ces dispositifs. La prospective est une démarche qui vise à élaborer des scénarios sur les futurs possibles. Les modèles peuvent intervenir à différents niveaux de cette démarche, que cela soit pour apporter des connaissances sur le futur, faciliter la médiation entre acteurs ou aider à l'action (Houet, 2015). Ils peuvent ainsi fortement contribuer à répondre aux objectifs de la prospective. L'intérêt de disposer d'un modèle descriptif dans ce cadre est de permettre de mieux représenter la complexité du système étudié dans ses dimensions spatiale, temporelle et sociale. Il peut alors participer à construire des trajectoires possibles plus crédibles, plus diversifiées et mieux acceptées par les acteurs.

L'idée de cet axe de recherche est donc de développer une méthodologie complète qui permettrait de mener à bien une démarche de prospective en utilisant un modèle à base d'agents descriptif pour construire et évaluer des trajectoires possibles. Cet axe de recherche s'intègre dans une collaboration que je viens de débiter avec des chercheurs de l'IRSTEA et plus particulièrement avec Eléonore Loiseau (CR IRSTEA, ITAP), qui travaille sur des approches d'Analyse de Cycle de Vie (ACV) territoriale (Loiseau *et al.*, 2013).

L'enjeu applicatif de cette collaboration est de répondre aux préoccupations des professionnels quant à la viabilité à long terme des filières vin et volaille. Effectivement, la viabilité à long terme de ces filières est compromise par les changements globaux, tels que le changement climatique, la pression sur les terres, les nouvelles législations, les nouveaux modes de consommation et les attentes de la société. Afin d'étudier cette viabilité et les stratégies d'adaptation, la méthodologie envisagée se base sur 5 étapes. La première consiste à sélectionner les facteurs clefs (changements globaux) pouvant avoir un impact sur l'avenir des filières. Une fois cette sélection effectuée, une deuxième étape consiste à spécifier les impacts possibles de ces facteurs sur les comportements des acteurs territoriaux. Ces spécifications seront utilisées dans une troisième étape pour développer un modèle à base d'agents qui permettra de générer, par la simulation, un ensemble de trajectoires d'évolution des filières à différents horizons temporels. L'étape suivante consiste à évaluer les impacts environnementaux pour chacune des trajectoires simulées selon une perspective ACV à la fois territoriale mais aussi sociale (Macombe *et al.*, 2013). Enfin, la dernière étape consiste à identifier les principaux facteurs favorisant l'émergence de trajectoires durables. Cette méthodologie se veut itérative et participative : les différentes étapes ont vocation à être menées en concertation avec les acteurs des filières et avec les principaux

acteurs territoriaux.

### **Conclusion de mon projet de recherche**

J'ai ébauché dans cette partie trois axes de recherche portant sur des dimensions comportementales (modélisation de la décision et argumentation), participatives (simulation interactive par interfaces tangibles et physicalisation) et prospectives (la prospective par la modélisation à base d'agents), qu'il m'intéresserait d'explorer. Si j'ai déjà commencé à engager des travaux sur ces axes, je reste bien sûr ouvert à de nouveaux questionnements. En effet, mes expériences passées m'ont maintes fois démontré que les rencontres avaient une influence majeure sur le travail du chercheur. C'est en effet grâce à ma rencontre avec Alexis Drogoul en stage de Master 2 que j'ai commencé à m'intéresser à la modélisation à base d'agents. C'est ensuite ma rencontre avec Cécile Duchêne qui m'a permis de me plonger dans le domaine du traitement des données géographiques. Enfin, mon intérêt pour la modélisation des systèmes agricoles a été fortement nourri par ma rencontre avec Olivier Therond. C'est ainsi que j'espère pour l'avenir continuer à faire de belles rencontres pour enrichir mon travail de chercheur.



# Bibliographie

- Carole ADAM : *Emotions : from psychological theories to logical formalization and implementation in a BDI agent*. Phd thesis, INP Toulouse, 2007.
- Carole ADAM et Julie DUGDALE : Comparing agent architectures in social simulation : Bdi agents versus finite-state machines. *In HICSS*, 2016.
- Carole ADAM et Julie DUGDALE : Agent-based analysis of the spread of awareness in the population in the prodromal phase of bushfires. *In Proceedings of the 51st Hawaii International Conference on System Sciences*, 2018.
- Carole ADAM et Benoit GAUDOU : Bdi agents in social simulations : a survey. *The Knowledge Engineering Review*, 2016.
- Carole ADAM et Benoit GAUDOU : Modelling human behaviours in disasters from interviews : Application to melbourne bushfires. *Journal of Artificial Societies and Social Simulation*, 20(3):12, 2017. ISSN 1460-7425. URL <http://jasss.soc.surrey.ac.uk/20/3/12.html>.
- Carole ADAM, Franck TAILLANDIER, Etienne DELAY, Odile PLATTARD et Mira TOUMI : Sprite-participatory simulation for raising awareness about coastal flood risk on the oleron island. *In International Conference on Information Systems for Crisis Response and Management in Mediterranean Countries*, pages 33–46. Springer, 2016.
- Carole ADAM, Patrick TAILLANDIER et Julie DUGDALE : Comparing agent architectures in social simulation : Bdi agents versus finite-state machines. *In Hawaii International Conference on System Sciences (HICSS)*, 2017.
- Muhammad ADNAN, Francisco C PEREIRA, Carlos Miguel Lima AZEVEDO, Kakali BASAK, Milan LOVRIC, Sebastián RAVEAU, Yi ZHU, Joseph FERREIRA, Christopher ZEGRAS et Moshe BEN-AKIVA : Simmobility : A multi-scale integrated agent-based simulation platform. *In 95th Annual Meeting of the Transportation Research Board Forthcoming in Transportation Research Record*, 2016.
- Robert John ALLAN : Survey of agent based modelling and simulation tools. Rapport technique, 2009.
- Luis ALONSO, Yan Ryan ZHANG, Arnaud GRIGNARD, Ariel NOYMAN, Yasushi SAKAI, Markus ELKATSHA, Ronan DOORLEY et Kent LARSON : Cityscope : A data-driven interactive simulation tool for urban design. use case volpe. *In Alfredo J. MORALES, Carlos GERSHENSON, Dan BRAHA, Ali A. MINAI et Yaneer BAR-YAM, éditeurs : Unifying Themes in Complex Systems IX*, pages 253–261, Cham, 2018. Springer International Publishing.

- Frédéric AMBLARD, Juliette ROUCHIER, Pierre BOMMEL, Franck VARENNE et Denis PHAN : Evaluation et validation de modèles multi-agents. 2006.
- Edouard AMOUROUX, Thanh-Quang CHU, Alain BOUCHER et Alexis DROGOUL : Gama : an environment for implementing and running spatially explicit multi-agent simulations. *In Pacific Rim International Conference on Multi-Agents*, pages 359–371. Springer, 2007.
- Weston ANDERSON, Seth GUIKEMA, Ben ZAITCHIK et William PAN : Methods for Estimating Population Density in Data-Limited Areas : Evaluating Regression and Tree-Based Models in Peru. *PLoS ONE*, 9(7), 2014.
- Jean-Philippe ANTONI, Nicolas LUNARDI et Gilles VUIDEL : Simuler les mobilités individuelles - Les enjeux de l'information géographique. *Revue Internationale de Géomatique*, 26(2):237–262, 2016. ISSN 1260-5875, 2116-7060. URL <http://dx.doi.org/10.3166/RIG.26.237-262>.
- Alain APPRIOU : Probabilities and unknowns in multisensor data fusion(probabilites et incertitude en fusion de donnees multi-senseurs). *Revue Scientifique et Technique de la Defense*, 1 st Quarter, 1991,, pages 27–40, 1991.
- Robert M AXELROD : *The complexity of cooperation : Agent-based models of competition and collaboration*. Princeton University Press, 1997.
- Tina BALKE et Nigel GILBERT : How do agents make decisions? a survey. *Journal of Artificial Societies and Social Simulation*, 17(4):13, 2014. ISSN 1460-7425. URL <http://jasss.soc.surrey.ac.uk/17/4/13.html>.
- Michael BALMER, Marcel RIESER, Konrad MEISTER, David CHARYPAR, Nicolas LEFEBVRE, Kai NAGEL et K AXHAUSEN : Matsim-t : Architecture and simulation times. *Multi-agent systems for traffic and transportation engineering*, 2009.
- Arnaud BANOS et Lena SANDERS : Modéliser et simuler les systèmes spatiaux en géographie. *Modéliser & simuler. Sous la dir. de F. Varenne et M. Silberstein. Edition Matériologiques*, 2:2, 2013.
- Olivier BARRETEAU, François BOUSQUET, Michel ÉTIENNE, Véronique SOUCHÈRE et Patrick D'AQUINO : Companion modelling : a method of adaptive and participatory research. *In Companion Modelling*, pages 13–40. Springer, 2014.
- Sigal G BARSADÉ : The ripple effect : Emotional contagion and its influence on group behavior. *Administrative Science Quarterly*, 47(4):644–675, 2002.
- Bernhard BAUER, Jörg P MÜLLER, James ODELL *et al.* : Agent uml : A formalism for specifying multiagent interaction. *In Agent-oriented software engineering*, volume 1957, pages 91–103. Springer, Berlin, 2001.
- Ana LC BAZZAN et Franziska KLÜGL : A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 29(3):375–403, 2014.
- BCG : Mobility revolution in the city of boston, 2017. URL <https://www.youtube.com/watch?v=1kmE1oYe0-s>.

- Nicolas BECU, Marion AMALRIC, Brice ANSELME, Elise BECK, Xavier BERTIN, Etienne DELAY, Nathalie LONG, Nicolas MARILLEAU, Cécilia PIGNON-MUSSAUD et Frederic ROUSSEAU : Participatory simulation to foster social learning on coastal flooding prevention. *Environmental Modelling & Software*, 98:1–11, 2017.
- Nicolas BECU, Pierre BOMMEL, Christophe LE PAGE et François BOUSQUET : Cormas, une plateforme multi-agent pour concevoir collectivement des modèles et interagir avec les simulations. In *Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*. Cépaduès, 2016.
- Nicolas BECU, Nathalie FRASCARIA-LACOSTE et Julie LATUNE : Experiential learning based on the newdistrict asymmetric simulation game : results of a dozen gameplay sessions. In *Hybrid Simulation & Gaming in the Networked Society : The 46th ISAGA Annual Conference 2015*, 2015.
- Majid BEHZADIAN, Reza Baradaran KAZEMZADEH, Amir ALBADVI et Mohammad AGHDASI : Promethee : A comprehensive literature review on methodologies and applications. *European journal of Operational research*, 200(1):198–215, 2010.
- Fabio BELLIFEMINE, Agostino POGGI et Giovanni RIMASSA : Jade : a fipa2000 compliant agent development environment. In *Proceedings of the fifth international conference on Autonomous agents*, pages 216–217. ACM, 2001.
- Trevor JM BENCH-CAPON : Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.
- Hugues BERSINI : Uml for abm. *JASSS*, 15(1):9, 2012.
- Ghassan BEYDOUN, Graham LOW, Brian HENDERSON-SELLERS, Haralambos MOURATIDIS, Jorge J GOMEZ-SANZ, Juan PAVON et Cesar GONZALEZ-PEREZ : Faml : a generic metamodel for mas development. *Software Engineering, IEEE Transactions on*, 35(6):841–863, 2009.
- Paulo BLIKSTEIN, Dor ABRAHAMSON et Uri WILENSKY : Netlogo : Where we are, where we're going. In *Proceedings of the annual meeting of Interaction Design and Children, Press*. Citeseer, 2005.
- Tibor BOSSE, Rob DUELL, Zulfiqar Ali MEMON, Jan TREUR et C Natalie VAN DER WAL : Multi-agent model for mutual absorption of emotions. *ECMS*, 2009:212–218, 2009.
- Mathieu BOURGAIS : *Vers des agents cognitifs, affectifs et sociaux dans la simulation*. Thèse de doctorat, Normandie Université, 2018.
- Mathieu BOURGAIS, Patrick TAILLANDIER et Laurent VERCOUTER : An agent architecture coupling cognition and emotions for simulation of complex systems. In *Social Simulation Conference*, 2016.
- Mathieu BOURGAIS, Patrick TAILLANDIER et Laurent VERCOUTER : Cognition, émotions et relations sociales pour la simulation multi-agent. In *JFSMA 2017*, 2017a.
- Mathieu BOURGAIS, Patrick TAILLANDIER et Laurent VERCOUTER : Enhancing the behavior of agents in social simulations with emotions and social relations. In *The 18th Workshop on Multi-agent-based Simulation-MABS 2017*, 2017b.
- Mathieu BOURGAIS, Patrick TAILLANDIER et Laurent VERCOUTER : Ben : An agent architecture for explainable and expressive behavior in social simulation. In *International Workshop on EXplainable TRansparent Autonomous Agents and Multi-Agent Systems*, 2019.

- Mathieu BOURGAIS, Patrick TAILLANDIER, Laurent VERCOUTER et Carole ADAM : Emotion modeling in social simulation : A survey. *Journal of Artificial Societies and Social Simulation*, 2018.
- Paul BOX : Spatial units as agents : Making the landscape an equal player in agent-based simulations. *Integrating geographic information systems and agent-based modeling techniques for simulating social and ecological processes*, page 59, 2002.
- Michael BRATMAN : Intentions, plans, and practical reason. 1987.
- Johannes BRAUER : The visualworks development environment. *In Programming smalltalk–Object-orientation from the beginning*, pages 77–96. Springer, 2015.
- Philippe CAILLOU, Benoit GAUDOU, Arnaud GRIGNARD, Chi Quang TRUONG et Patrick TAILLANDIER : A simple-to-use bdi architecture for agent-based modeling and simulation. *In Advances in Social Simulation 2015*, pages 15–28. Springer, 2017.
- Radovan CERVENKA, Ivan TRENCANSKY et Monique CALISTI : Modeling social aspects of multi-agent systems : The aml approach. *In International Workshop on Agent-Oriented Software Engineering*, pages 28–39. Springer, 2005.
- Kevin CHAPUIS, Patrick TAILLANDIER, Benoit GAUDOU, Drogoul DROGOUL et Eric DAUDÉ : A multi-modal urban traffic agent-based framework to study individual response to catastrophic events. *In The 21st International Conference on Principles and Practice of Multi-Agent Systems (PRIMA 2018)*, 2018a.
- Kevin CHAPUIS, Patrick TAILLANDIER, Renaud MISSLIN et Alexis DROGOUL : Gen\* : a generic toolkit to generate spatially explicit synthetic populations. *International Journal of Geographical Information Science*, pages 1–17, 2018b.
- Luca CHITTARO et Riccardo SIONI : Serious games for emergency preparedness : Evaluation of an interactive vs. a non-interactive simulation of a terror attack. *Computers in Human Behavior*, 50:508–519, 2015.
- Thanh-Quang CHU, Alain BOUCHER, Alexis DROGOUL, Duc-An VO, Hong-Phuong NGUYEN et Jean-Daniel ZUCKER : Interactive learning of expert criteria for rescue simulations. *In Pacific Rim International Conference on Multi-Agents*, pages 127–138. Springer, 2008.
- Aboubacar CISSÉ, Alassane BAH, Alexis DROGOUL, Ahmed Tidjane CISSÉ, Jacques André NDIONE, Cheikh MF KÉBÉ et Patrick TAILLANDIER : Un modèle à base d’agents sur la transmission et la diffusion de la fièvre de la vallée du rift à barkédji (ferlo, sénégal). *Stud. Inform. Univ.*, 10(1):77–97, 2012.
- William W COHEN : Fast effective rule induction. *In Machine Learning Proceedings 1995*, pages 115–123. Elsevier, 1995.
- CORMAS : Cormas in 10 years!, 2018. URL <https://forum.comses.net/t/cormas-in-10-years/7014>.
- Eric CORNELIS, Laurie HOLLART, Johan BARTHELEMY et Philippe L. TOINT : An original synthetic population tool applied to Belgian case : VirtualBelgium. 2013. URL <http://ro.uow.edu.au/smartpapers/125/>.

- Clémentine COTTINEAU, Paul CHAPRON et Romain REUILLON : Growing models from the bottom up. an evaluation-based incremental modelling method (ebimm) applied to the simulation of systems of cities. *Journal of Artificial Societies and Social Simulation*, 18(4):9, 2015. ISSN 1460-7425. URL <http://jasss.soc.surrey.ac.uk/18/4/9.html>.
- Andrew T CROOKS et Christian JE CASTLE : The integration of agent-based modelling and geographical information for geospatial simulation. *In Agent-based models of geographical systems*, pages 219–251. Springer, 2012.
- Guillaume CZURA, Patrick TAILLANDIER, Pierrick TRANOUEZ et Éric DAUDÉ : Mosaiic : City-level agent-based traffic simulation adapted to emergency situations. *In Proceedings of the International Conference on Social Modeling and Simulation, plus Econophysics Colloquium 2014*, pages 265–274. Springer International Publishing, 2015.
- Eric DAUDÉ, Patrice LANGLOIS, Baptiste BLANPAIN, Emmanuel SAPIN *et al.* : Aoc, une ontologie formelle pour la modélisation de systèmes complexes en géographie. *In Outils, méthodes et modèles en géomatique pour la production de connaissances sur les territoires et le paysage*, 2010.
- Eric DAUDÉ, Kevin CHAPUIS, Clément CARON, Alexis DROGOUL, Benoit GAUDOU, Sebastien REY-COYREHOURQ, Arnaud SAVAL, Patrick TAILLANDIER, Pierrick TRANOUEZ et Jean-Daniel ZUCKER : Escape : Exploring by simulation cities awareness on population evacuation. *In International Conference on Information Systems for Crisis Response and Management*, 2019.
- Frank DIGNUM : Autonomous agents with norms. *Artificial intelligence and law*, 7(1):69–79, 1999.
- Alan DORIN et Nicholas GEARD : The practice of agent-based model visualization. *Artificial life*, 20(2):271–289, 2014.
- Alexis DROGOUL, Edouard AMOUROUX, Philippe CAILLOU, Benoit GAUDOU, Arnaud GRIGNARD, Nicolas MARILLEAU, Patrick TAILLANDIER, Maroussia VAVASSEUR, Duc-An VO et Jean-Daniel ZUCKER : Gama : A spatially explicit, multi-level, agent-based modeling and simulation platform. *In Advances on Practical Applications of Agents and Multi-Agent Systems*, pages 271–274. Springer Berlin Heidelberg, 2013a.
- Alexis DROGOUL, Edouard AMOUROUX, Philippe CAILLOU, Benoit GAUDOU, Arnaud GRIGNARD, Nicolas MARILLEAU, Patrick TAILLANDIER, Maroussia VAVASSEUR, Duc-An VO et Jean-Daniel ZUCKER : 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*, pages 1361–1362. International Foundation for Autonomous Agents and Multiagent Systems, 2013b.
- Phan Minh DUNG : On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.
- Jérôme DURY, Noémie SCHALLER, Frédéric GARCIA, Arnaud REYNAUD et Jacques Eric BERGEZ : Models to support cropping plan and crop rotation decisions. a review. *Agronomy for sustainable development*, 32(2):567–580, 2012.
- Bruce EDMONDS et Scott MOSS : From kiss to kids—an ‘anti-simplistic’ modelling approach. *In International workshop on multi-agent systems and agent-based simulation*, pages 130–144. Springer, 2004.

- Justin EMERY, Nicolas MARILLEAU, Nadège MARTINY, Thomas THÉVENIN, Tri NGUYEN-HUU, Mohamed BADRAM, Arnaud GRIGNARD, Hassan HBDID, Ah-Med LAATABI et Saad TOUBHI : Marrakair : une simulation participative pour observer les émissions atmosphériques du trafic routier en milieu urbain. *In Treizièmes Rencontres de Théo Quant*, 2017.
- Hans J EYSENCK, Michael W EYSENCK *et al.* : *Personality and individual differences*. Plenum New York, NY, 1987.
- Pierre FOSSET, Arnaud BANOS, Elise BECK, Sonia CHARDONNEL, Christophe LANG, Nicolas MARILLEAU, Arnaud PIOMBINI, Thomas LEYSENS, Alexis CONESA, Isabelle ANDRE-POYAUD *et al.* : Exploring intra-urban accessibility and impacts of pollution policies with an agent-based simulation platform : Gamirod. *Systems*, 4(1):5, 2016.
- Shannon GALLAGHER, Lee RICHARDSON, Samuel L. VENTURA et William F. EDDY : SPEW : Synthetic Populations and Ecosystems of the World. *arXiv :1701.02383 [physics, q-bio, stat]*, janvier 2017. URL <http://arxiv.org/abs/1701.02383>. arXiv : 1701.02383.
- Nasser GASMI, Arnaud GRIGNARD, Alexis DROGOUL, Benoit GAUDOU, Patrick TAILLANDIER, Olivier TESSIER et Vo Duc AN : Reproducing and exploring past events using agent-based geo-historical models. *In International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 151–163. Springer, 2014.
- Benoit GAUDOU, Christophe SIBERTIN-BLANC, Olivier THEROND, Frédéric AMBLARD, Yves AUDA, Jean-Paul ARCANGELI, Maud BALESTRAT, Marie-Hélène CHARRON-MOIREZ, Etienne GONDET, Yi HONG, Romain LARDY, Thomas LOUAIL, Eunata MAYOR, David PANZOLI, Sabine SAUVAGE, José-Miguel SÁNCHEZ-PÉREZ, Patrick TAILLANDIER, Nguyen VAN BAI, Maroussia VAVASSEUR et Pierre MAZZEGA : The maelia multi-agent platform for integrated analysis of interactions between agricultural land-use and low-water management strategies. *In Shah Jamal ALAM et H. Van Dyke PARUNAK, éditeurs : Multi-Agent-Based Simulation XIV*, pages 85–100, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. ISBN 978-3-642-54783-6.
- Nikolas GEROLIMINIS et Carlos F. DAGANZO : Existence of urban-scale macroscopic fundamental diagrams : Some experimental findings. *Transportation Research Part B : Methodological*, 42(9): 759 – 770, 2008. ISSN 0191-2615. URL <http://www.sciencedirect.com/science/article/pii/S0191261508000180>.
- Jonathan GRATCH : Socially situated planning. *In Socially Intelligent Agents*. Springer, 2002.
- Arnaud GRIGNARD : *Modèles de visualisation à base d'agents*. Thèse de doctorat, Paris 6, 2015.
- Arnaud GRIGNARD, Luis ALONSO, Patrick TAILLANDIER, Benoit GAUDOU, Tri NGUYEN-HUU, Wolfgang GRUEL et Kent LARSON : The impact of new mobility modes on a city : A generic approach using abm. *In International Conference on Complex Systems*, pages 272–280. Springer, 2018a.
- Arnaud GRIGNARD, Alexis DROGOUL et Jean-Daniel ZUCKER : Online analysis and visualization of agent based models. *In International Conference on Computational Science and Its Applications*, pages 662–672. Springer, 2013a.
- Arnaud GRIGNARD, Núria MACIÀ, Luis ALONSO PASTOR, Ariel NOYMAN, Yan ZHANG et Kent LARSON : Cityscope andorra : a multi-level interactive and tangible agent-based visualization.



- In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1939–1940. International Foundation for Autonomous Agents and Multiagent Systems, 2018b.
- Arnaud GRIGNARD, Patrick TAILLANDIER, Benoit GAUDOU, Duc An VO, Nghi Quang HUYNH et Alexis DROGOUL : Gama 1.6 : Advancing the art of complex agent-based modeling and simulation. *In International Conference on Principles and Practice of Multi-Agent Systems*, pages 117–131. Springer, 2013b.
- Volker GRIMM, Uta BERGER, Donald L DEANGELIS, J Gary POLHILL, Jarl GISKE et Steven F RAILSBACK : The odd protocol : a review and first update. *Ecological modelling*, 221(23):2760–2768, 2010.
- Paul GUYOT, Alexis DROGOUL et Shinichi HONIDEN : Power and negotiation : lessons from agent-based participatory simulations. *In Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 27–33. ACM, 2006.
- Mordechai HAKLAY, David O’ SULLIVAN, Mark THURSTAIN-GOODWIN et Thorsten SCHELHORN : “so go downtown” : Simulating pedestrian movement in town centres. *Environment and planning B : Planning and design*, 28(3):343–359, 2001.
- Daniel Damir HARABOR, Alban GRASTIEN *et al.* : Improving jump point search. *In ICAPS*, 2014.
- Kirk HARLAND, Alison HEPPENSTALL, Dianna SMITH et Mark BIRKIN : Creating realistic synthetic populations at varying spatial scales : a comparative critique of population synthesis techniques. *Journal of Artificial Societies and Social Simulation*, 15(1):1–15, 2012.
- Elaine HATFIELD, John T CACIOPPO et Richard L RAPSON : Emotional contagion. *Current directions in psychological science*, 2(3):96–100, 1993.
- Glenn I HAWE, Graham COATES, Duncan T WILSON et Roger S CROUCH : Agent-based simulation for large-scale emergency response : A survey of usage and implementation. *ACM Computing Surveys (CSUR)*, 45(1):8, 2012.
- John Henry HOLLAND *et al.* : *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
- Thomas HOUET : Usages des modèles spatiaux pour la prospective. *Revue internationale de géomatique*, 25(1):123–143, 2015.
- Nick HOWDEN, Ralph RÖNNQUIST, Andrew HODGSON et Andrew LUCAS : Jack intelligent agents-summary of an agent infrastructure. *In 5th International conference on autonomous agents*, 2001.
- Wander JAGER : Enhancing the realism of simulation (eros) : On implementing and developing psychological theory in social simulation. *Journal of Artificial Societies and Social Simulation*, 20(3):14, 2017. ISSN 1460-7425. URL <http://jasss.soc.surrey.ac.uk/20/3/14.html>.
- Yvonne JANSEN, Pierre DRAGICEVIC, Petra ISENBERG, Jason ALEXANDER, Abhijit KARNIK, Johan KILDAL, Sriram SUBRAMANIAN et Kasper HORNBEK : Opportunities and challenges for data physicalization. *In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3227–3236. ACM, 2015.

- Ken KAHN et Howard NOBLE : The modelling4all project a web-based modelling tool embedded in web 2.0. *In Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, page 50. ICST (Institute for Computer Sciences, Social-Informatics and . . . , 2009.
- Nikos KARANIKOLAS, Pierre BISQUERT, Patrice BUCHE, Christos KAKLAMANIS et Rallou THOMOPOULOS : A decision support tool for agricultural applications based on computational social choice and argumentation. *International Journal of Agricultural and Environmental Information Systems (IJAEIS)*, 9(3):54–73, 2018.
- Elmar KIESLING, Markus GÜNTHER, Christian STUMMER et Lea M WAKOLBINGER : Agent-based simulation of innovation diffusion : a review. *Central European Journal of Operations Research*, 20(2):183–230, 2012.
- Jan HG KLABBERS : *The magic circle : Principles of gaming & simulation*. Sense Publishers Rotterdam, 2009.
- Bernhard KOSAR et Melanie TOMINTZ : simSALUD : A Web-based Spatial Microsimulation to Model the Health Status for Small Areas Using the Example of Smokers in Austria. pages 207–216. Austrian Academy of Sciences Press, 2015. ISBN 978-3-87907-545-4. URL <http://www.oeaw.ac.at?arp=0x0030d5ae>.
- Daniel KRAJZEWICZ, Jakob ERDMANN, Michael BEHRISCH et Laura BIEKER : Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, 2012.
- Adam DI KRAMER, Jamie E GUILLORY et Jeffrey T HANCOCK : Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, 111(24):8788–8790, 2014.
- Kalliopi KRAVARI et Nick BASSILIADES : A survey of agent platforms. *Journal of Artificial Societies and Social Simulation*, 18(1):11, 2015.
- Patrice LANGLOIS, Baptiste BLANPAIN et Eric DAUDÉ : Magéo, une plateforme de modélisation et de simulation multi-agent pour les sciences humaines. *Cybergeo : European Journal of Geography*, 2015.
- Van Minh LE, Carole ADAM, Richard CANAL, Benoit GAUDOU, Tuong Vinh HO et Patrick TAILLANDIER : Simulation of the emotion dynamics in a group of agents in an evacuation situation. *In Pacific Rim international Conference on Multi-Agents (PRIMA)*, Lecture Notes in Computer Science (LNCS), pages 604–619. Springer, 2010.
- Van Minh LE, Benoit GAUDOU, Patrick TAILLANDIER et Duc An VO : A new bdi architecture to formalize cognitive agent behaviors into simulations. *In Advanced Methods and Technologies for Agent and Multi-Agent Systems (KES-AMSTA)*, volume 252 de *Frontiers in Artificial Intelligence and Applications*, pages 395–403. IOS Press, 2013.
- Christophe LE PAGE, Géraldine ABRAMI, Olivier BARRETEAU, Nicolas BECU, Pierre BOMMEL, Aurélie BOTTA, Anne DRAY, Claude MONTEIL et Véronique SOUCHÈRE : Chapitre 3-des modèles pour partager des représentations. *In Modélisation d'accompagnement*, pages 71–101. Editions Quae, 2010.

- Christophe LE PAGE, Nicolas BECU, Pierre BOMMEL et François BOUSQUET : 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):10, 2012.
- Christophe LE PAGE, François BOUSQUET, Innocent BAKAM, Alassane BAH et Christian BARON : Cormas : A multiagent simulation toolkit to model natural and social dynamics at multiple scales. *In Proceedings of Workshop " The ecology of scales", Wageningen (The Netherlands)*, 2000.
- Christophe LE PAGE et Arthur PERROTON : Kilt : a modelling approach based on participatory agent-based simulation of stylized socio-ecosystems to stimulate social learning with local stakeholders. *In International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 156–169. Springer, 2017.
- Eléonore LOISEAU, Philippe ROUX, Guillaume JUNQUA, Pierre MAUREL et Véronique BELLON-MAUREL : Adapting the lca framework to environmental assessment in land planning. *The International Journal of Life Cycle Assessment*, 18(8):1533–1548, 2013.
- Catherine MACOMBE, Pekka LESKINEN, Pauline FESCHET et Riina ANTIKAINEN : Social life cycle assessment of biodiesel production at three levels : a literature review and development needs. *Journal of Cleaner Production*, 52:205–216, 2013.
- Roger MARTIN-CLOUAIRE et Jean-Pierre RELIER : Fondements ontologiques des systèmes pilotés. *Rapport interne, BIA-INRA.[cité p. 84, 97]*, 2002.
- Roger MARTIN-CLOUAIRE et Jean-Pierre RELIER : Modelling and simulating work practices in agriculture. *International Journal of Metadata, Semantics and Ontologies*, 4(1-2):42–53, 2009.
- Robert R MCCRAE et Oliver P JOHN : An introduction to the five-factor model and its applications. *Journal of personality*, 60(2):175–215, 1992.
- DLua MEADOWS, Thomas FIDDAMAN et Diana SHANNON : Fish banks, ltd. *Laboratory for Interactive Learning*, 1986.
- S Ben MENA : Introduction aux méthodes multicritères d'aide à la décision. *Biotechnologie, Agromonie, Société et Environnement*, 4(2):83–93, 2000.
- Alice MICOLIER, Franck TAILLANDIER, Patrick TAILLANDIER et Frédéric BOS : Li-bim : un modèle cognitif avancé simulant le comportement de l'occupant et son confort intérieur à partir d'une maquette numérique afin de guider les choix de conception de bâtiments durables bim. *In Conférence Francophone de l'International Building Performance Simulation Association*, 2018.
- Alice MICOLIER, Franck TAILLANDIER, Patrick TAILLANDIER et Frédéric BOS : Li-bim : an advanced cognitive model to simulate the occupant behavior and its indoor comfort from building information modeling. *Engineering Applications of Artificial Intelligence*, 2019.
- Nelson MINAR, Roger BURKHART, Chris LANGTON, Manor ASKENAZI *et al.* : The swarm simulation system : A toolkit for building multi-agent simulations. 1996.
- Birgit MÜLLER, Stefano BALBI, Carsten M BUCHMANN, Luís DE SOUSA, Gunnar DRESSLER, Jürgen GROENEVELD, Christian J KLASSERT, Quang Bao LE, James DA MILLINGTON, Henning

- NOLZEN *et al.* : Standardised and transparent model descriptions for agent-based models : Current status and prospects. *Environmental Modelling & Software*, 55:156–163, 2014.
- Isabel Briggs MYERS, Mary H MCCAULLEY et Robert MOST : *Manual, a guide to the development and use of the Myers-Briggs type indicator*. Consulting Psychologists Press, 1985.
- Michael NORTH, Nicholson COLLIER, Jonathan OZIK, Eric TATAR, Charles MACAL, Mark BRAGEN et Pam SYDELKO : Complex adaptive systems modeling with repast simphony. *Complex Adaptive Systems Modeling*, 1(1):3, 2013a. ISSN 2194-3206. URL <http://www.casmodeling.com/content/1/1/3>.
- Michael J NORTH, Nicholson T COLLIER, Jonathan OZIK, Eric R TATAR, Charles M MACAL, Mark BRAGEN et Pam SYDELKO : Complex adaptive systems modeling with repast simphony. *Complex adaptive systems modeling*, 1(1):3, 2013b.
- Magalie OCHS, Nicolas SABOURET et Vincent CORRUBLE : Simulation of the dynamics of non-player characters' emotions and social relations in games. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(4):281–297, 2009.
- Andrew ORTONY : On making believable emotional agents believable. *Trappl et al.(Eds.)(2002)*, pages 189–211, 2002.
- Andrew ORTONY, Gerald L CLORE et Allan COLLINS : *The cognitive structure of emotions*. Cambridge university press, 1990.
- Gary PERELMAN, Marcos SERRANO, Mathieu RAYNAL, Celia PICARD, Mustapha DERRAS et Emmanuel DUBOIS : The roly-poly mouse : Designing a rolling input device unifying 2d and 3d interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 327–336. ACM, 2015.
- Damien PHILIPPON, Marc CHOISY, Alexis DROGOUL, Benoit GAUDOU, Nicolas MARILLEAU, Patrick TAILLANDIER et Chi Quang TRUONG : Exploring trade and health policies influence on dengue spread with an agent-based model. In *International Workshop on Multi-Agent-Based Simulation (MABS), Singapore*, 2016.
- Wim PIJLS et Henk POST : Yet another bidirectional algorithm for shortest paths. Rapport technique, Econometric Institute Research Papers, 2009.
- Alexander POKAHR, Lars BRAUBACH et Winfried LAMERSDORF : Jadex : A bdi reasoning engine. In *Multi-agent programming*. Springer, 2005.
- Steven F RAILSBACK, Steven L LYTINEN et Stephen K JACKSON : Agent-based simulation platforms : Review and development recommendations. *Simulation*, 82(9):609–623, 2006.
- Ana-Maria Olteanu RAIMOND et Sébastien MUSTIÈRE : Data matching—a matter of belief. In *Headway in spatial data handling*, pages 501–519. Springer, 2008.
- Alexander REPENNING : Making programming more conversational. In *Visual Languages and Human-Centric Computing (VL/HCC), 2011 IEEE Symposium on*, pages 191–194. IEEE, 2011.
- Mitchel RESNICK : Starlogo : An environment for decentralized modeling and decentralized thinking. In *Conference companion on Human factors in computing systems*, pages 11–12. ACM, 1996.

- Mitchel RESNICK, John MALONEY, Andrés MONROY-HERNÁNDEZ, Natalie RUSK, Evelyn EASTMOND, Karen BRENNAN, Amon MILLNER, Eric ROSENBAUM, Jay SILVER, Brian SILVERMAN *et al.* : Scratch : programming for all. *Communications of the ACM*, 52(11):60–67, 2009.
- Paul RICHMOND et Mozghan K CHIMEH : Flame gpu : Complex system simulation framework. *In High Performance Computing & Simulation (HPCS), 2017 International Conference on*, pages 11–17. IEEE, 2017.
- Daniel ROUSSEAU et Barbara HAYES-ROTH : A social-psychological model for synthetic actors. *In Proceedings of the second international conference on Autonomous agents*. ACM, 1998.
- Bernard ROY : The outranking approach and the foundations of electre methods. *In Readings in multiple criteria decision aid*, pages 155–183. Springer, 1990.
- Anne RUAS, Aurélie DYEUVRE, Cécile DUCHÊNE et Patrick TAILLANDIER : Methods for improving and updating the knowledge of a generalization system. *In Proceedings of AUTOCARTO 2006*, 2006.
- Eric RUSSELL et Uri WILENSKY : Consuming spatial data in netlogo using the gis extension. *In The annual meeting of the Swarm Development Group*, 2008.
- Ilias SAKELLARIOU, Petros KEFALAS et Ioanna STAMATOPOULOU : Enhancing netlogo to simulate bdi communicating agents. *In Hellenic Conference on Artificial Intelligence*. Springer, 2008.
- François SEMPÉ, Minh NGUYEN-DUC, Stanislas BOISSAU, Alain BOUCHER et Alexis DROGOU : An artificial maieutic approach for eliciting experts' knowledge in multi-agent simulations. *In International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 75–87. Springer, 2005.
- Orit SHAER, Eva HORNECKER *et al.* : Tangible user interfaces : past, present, and future directions. *Foundations and Trends in Human-Computer Interaction*, 3(1–2):4–137, 2010.
- Glenn SHAFER : *A mathematical theory of evidence*, volume 42. Princeton university press, 1976.
- Jean SISKOS, Gerhard WÄSCHER et Heinz MICHAEL WINKELS : A bibliography on outranking approaches (1988-1982). *Mathematische Operationsforschung und Statistik. Series Optimization*, 15(4):573–582, 1984.
- Philippe SMETS : Constructing the pignistic probability function in a context of uncertainty. *In UAI*, volume 89, pages 29–40, 1989.
- Philippe SMETS et Robert KENNES : The transferable belief model. *Artificial intelligence*, 66(2):191–234, 1994.
- Stefan STEINIGER, Patrick TAILLANDIER *et al.* : Improving map generalisation of buildings by introduction of urban context rules. *In Geocomputation*, 2007.
- Stefan STEINIGER, Patrick TAILLANDIER et Robert WEIBEL : Utilising urban context recognition and machine learning to improve the generalisation of buildings. *International Journal of Geographical Information Science*, 24(2):253–282, 2010.
- Tiberiu STRATULAT : *Systèmes d'agents normatifs : concepts et outils logiques*. Thèse de doctorat, Université de Caen, 2002.

- Ron SUN : *Cognition and multi-agent interaction : From cognitive modeling to social simulation*. Cambridge University Press, 2006.
- Ron SUN : The importance of cognitive architectures : an analysis based on clarion. *Journal of Experimental & Theoretical Artificial Intelligence*, 19(2):159–193, 2007.
- Jan SVENNEVIG : *Getting acquainted in conversation : a study of initial interactions*. John Benjamins Publishing, 2000.
- Samarth SWARUP et Madhav V. MARATHE : Generating Synthetic Populations for Social Modeling : Tutorial at the Autonomous Agents and Multi-Agents Systems (AAMAS) Conference. Singapore, mai 2016.
- Franck TAILLANDIER et Carole ADAM : Games ready to use : A serious game for teaching natural risk management. *Simulation & Gaming*, page 1046878118770217, 2018.
- Franck TAILLANDIER, Patrick TAILLANDIER, Fethi HAMZAOUI et Denys BREYSSE : A new agent-based model to manage construction project risks—application to the crossroad of bab el karmadine at tlemcen. *European Journal of Environmental and Civil Engineering*, 20(10):1197–1213, 2016a.
- Franck TAILLANDIER, Patrick TAILLANDIER, Esra TEPELI, Denys BREYSSE, Rasool MEHDIZADEH et Fadi KHARTABIL : A multi-agent model to manage risks in construction project (smacc). *Automation in Construction*, 58:1–18, 2015.
- Patrick TAILLANDIER : Acquisition automatique de connaissances de guidage d’un processus de généralisation de données géographiques. *In Rencontres des jeunes chercheurs en intelligence artificielle*, pages 213–229, 2007a.
- Patrick TAILLANDIER : Automatic knowledge revision of a generalisation system. *In The 10th ICA Workshop on Generalisation and Multiple Representation*, 2007b.
- Patrick TAILLANDIER : Révision à base d’agents des connaissances de guidage d’un processus de généralisation de données géographiques. *In Journées Francophones sur les systèmes Multi-Agents*, pages 243–252, 2007c.
- Patrick TAILLANDIER : *Révision automatique des connaissances guidant l’exploration informée d’arbres d’états : application au contexte de la généralisation de données géographiques*. Thèse de doctorat, Université Paris Est, 2008.
- Patrick TAILLANDIER : Gamagram : Modélisation graphique sous gama. *In Masyco 2013*, 2013.
- Patrick TAILLANDIER : Gamagram : Graphical modeling with the gama platform. *In Proceedings of ICCSA 2014*, 2014a.
- Patrick TAILLANDIER : Traffic simulation with the gama platform. *In Eighth International Workshop on Agents in Traffic and Transportation (ATT)*, 2014b.
- Patrick TAILLANDIER : La modélisation du temps dans la simulation à base d’agents. *L’information géographique*, 2(2):65–78, 2015.



- Patrick TAILLANDIER, Edouard AMOUROUX, Duc An VO et Ana-Maria OLTEANU-RAIMOND : Using belief theory to formalize the agent behavior : application to the simulation of avian flu propagation. *In International Conference on Principles and Practice of Multi-Agent Systems*, pages 575–587. Springer, 2010a.
- Patrick TAILLANDIER, Arnaud BANOS, Alexis DROGOUL, Benoit GAUDOU, Nicolas MARILLEAU et Quang Chi TRUONG : Simulating urban growth with raster and vector models : A case study for the city of can tho, vietnam. *In International Conference on Autonomous Agents and Multiagent Systems*, pages 154–171. Springer, 2016b.
- Patrick TAILLANDIER, Gaudou BENOIT, Arnaud GRIGNARD, Quang Nghi HUYNH, Nicolas MARILLEAU, Damien PHILIPPON, Philippe CAILLOU et Alexis DROGOUL : Participatory modeling and simulation with the gama platform. *Journal of Artificial Societies and Social Simulation*, 2019.
- Patrick TAILLANDIER, Mathieu BOURGAIS, Philippe CAILLOU, Carole ADAM et Benoit GAUDOU : A bdi agent architecture for the gama modeling and simulation platform. *In MABS 2016 Multi-Agent-Based Simulation*, 2016c.
- Patrick TAILLANDIER, Mathieu BOURGAIS, Alexis DROGOUL et Laurent VERCOUTER : Using parallel computing to improve the scalability of models with bdi agents. *In Social Simulation Conference*, 2017.
- Patrick TAILLANDIER et Elodie BUARD : Designing agent behaviour in agent-based simulation through participatory method. *In Principles of Practice in Multi-Agent Systems*, pages 571–578. Springer, 2009.
- Patrick TAILLANDIER et Thanh-Quang CHU : Using participatory paradigm to learn human behaviour. *In International Conference on Knowledge and Systems Engineering, 2009*, pages 55–60. IEEE, 2009.
- Patrick TAILLANDIER, A DROGOUL, DA VO et E AMOUROUX : Gama : a simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. *In Proceedings of PRIMA 2010*, 2010b.
- Patrick TAILLANDIER et Alexis DROGOUL : From grid environment to geographic vector agents, modeling with the gama simulation platform. *In Conference of the International Cartographic Association*, 2011.
- Patrick TAILLANDIER, Cécile DUCHÊNE et Alexis DROGOUL : Knowledge revision in systems based on an informed tree search strategy : application to cartographic generalisation. *In Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology*, pages 273–278. ACM, 2008.
- Patrick TAILLANDIER, Cécile DUCHÊNE et Alexis DROGOUL : Using belief theory to diagnose control knowledge quality : Application to cartographic generalisation. *In Computing and Communication Technologies, 2009. RIVF'09. International Conference on*, pages 1–8. IEEE, 2009a.
- Patrick TAILLANDIER, Cecile DUCHENE et Alexis DROGOUL : Using belief theory to diagnose control knowledge quality : Application to cartographic generalisation. *In International Conference on Computing and Communication Technologies*, pages 1–8. IEEE, 2009b.

- Patrick TAILLANDIER, Cécile DUCHÊNE et Alexis DROGOUL : Automatic revision of the control knowledge used by trial and error methods : Application to cartographic generalisation. *Applied Soft Computing*, 11, 2010c.
- Patrick TAILLANDIER, Cécile DUCHÊNE et Alexis DROGOUL : Automatic revision of rules used to guide the generalisation process in systems based on a trial and error strategy. *International Journal of Geographical Information Science*, 25:1971–1999, 2011a.
- Patrick TAILLANDIER et Julien GAFFURI : Objective function designing led by user preferences acquisition. *In International Conference on Information Technology and Applications*, 2009.
- Patrick TAILLANDIER, Benoit GAUDOU, Arnaud GRIGNARD, Quang Nghi HUYNH, Nicolas MARRILLEAU, Philippe CAILLOU, Damien PHILIPPON et Alexis DROGOUL : Building, composing and experimenting complex spatial models with the gama platform. *Geoinformatica*, 2018.
- Patrick TAILLANDIER, Arnaud GRIGNARD, Benoit GAUDOU et Alexis DROGOUL : Des données géographiques à la simulation à base d'agents : application de la plate-forme gama. *Cybergeo : European Journal of Geography*, 2014.
- Patrick TAILLANDIER, Olivier THEROND *et al.* : Use of the belief theory to formalize agent decision making processes : Application to cropping plan decision making. *In European Simulation and Modelling Conference*, pages 138–142, 2011b.
- Patrick TAILLANDIER, Olivier THEROND et Benoit GAUDOU : Une architecture d'agent bdi basée sur la théorie des fonctions de croyance : application à la simulation du comportement des agriculteurs. *In Journées Francophones sur les Systèmes Multi-Agents 2012*, pages 107–116, 2012a.
- Patrick TAILLANDIER, Olivier THEROND et Benoit GAUDOU : A new bdi agent architecture based on the belief theory. application to the modelling of cropping plan decision-making. *In International Environmental Modelling and Software Society (iEMSs)*, 2012b.
- Patrick TAILLANDIER, Duc-An VO, Edouard AMOUROUX et Alexis DROGOUL : Gama : a simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. *In International Conference on Principles and Practice of Multi-Agent Systems*, pages 242–258. Springer, 2010d.
- Olivier THEROND, Christophe SIBERTIN-BLANC, Romain LARDY, Benoit GAUDOU, Maud BALES-TRAT, Yi HONG, Thomas LOUAIL, David PANZOLI, Jose-Miguel SANCHEZ-PEREZ, Sabine SAUVAGE *et al.* : Integrated modelling of social-ecological systems : The maelia high-resolution multi-agent platform to deal with water scarcity problems. *In 7th International Environmental Modelling and Software Society (iEMSs 2014)*, 2014.
- Samuel THIRIOT : Décrire l'impact de normes institutionnelles sur un système social : unification par les fonctions de croyance. *In JFSMA*, pages 233–242, 2016.
- Seth TISUE et Uri WILENSKY : Netlogo : A simple environment for modeling complexity. *In International conference on complex systems*, volume 21, pages 16–21. Boston, MA, 2004.
- Pierrick TRANOUEZ, Eric DAUDÉ et Patrice LANGLOIS : A multiagent urban traffic simulation. *Journal of Nonlinear Systems and Applications*, 3(2):98–106, 2012.

- Jean-Pierre TREUIL, Alexis DROGOUL et Jean-Daniel ZUCKER : Modélisation et simulation à base d'agents. *Dunod Editions*, pages 1–13, 2008.
- Minh Thai TRUONG, Frédéric AMBLARD et Benoit GAUDOU : Combination framework of bi solution & multi-agent platform (cfbm) for multi-agent based simulations. *In 13eme Conference francophone sur le Gestion et l'Extraction de Connaissances : Journée Atelier aide à la Décision à tous les Etages (AIDE@ EGC 2013)*, pages 35–42, 2013.
- Quang Chi TRUONG, Patrick TAILLANDIER, Benoit GAUDOU, Minh Quang VO, Trung Hieu NGUYEN et Alexis DROGOUL : Exploring agent architectures for farmer behavior in land-use change. a case study in coastal area of the vietnamese mekong delta. *In International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 146–158. Springer, 2015.
- Raimo TUOMELA : The importance of us : A philosophical study of basic social notions. 1995.
- Dominique URBANI et Marielle DELHOM : Hybrid mas gis coastal systems modeling methodology. *In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 523–530. Springer, 2008.
- Marion VALETTE, Benoit GAUDOU, Dominique LONGIN et Taillandier PATRICK : Modeling a real-case situation of egress using bdi agents with emotions and social skills. *In The 21st International Conference on Principles and Practice of Multi-Agent Systems (PRIMA 2018)*, 2018.
- Franck VARENNE : Epistémologie des modèles et des simulations : tour d'horizon et tendances. *In Les modèles, possibilités et limites. Jusqu'où va le réel ?, J.M. Levy (dir.)*, pages 13–46. Matériologiques, 2014.
- David WATSON et Lee Anna CLARK : On traits and temperament : General and specific factors of emotional experience and their relation to the five-factor model. *Journal of personality*, 60(2):441–476, 1992.
- Fabiola López y LÓPEZ, Michael LUCK et Mark D'INVERNO : A normative framework for agent-based systems. *Computational & Mathematical Organization Theory*, 12(2-3):227–250, 2006.





## A — Travaux et publications

### A.1 Etat civil

Nom : Patrick Taillandier  
Date de naissance : 17 décembre 1979  
Adresse : INRA Toulouse - MIAT  
24 chemin de Borde Rouge,  
31326 Castanet Tolosan  
Téléphone : +33 (0) 561 285 715  
E-mail : patrick.taillandier@inra.fr  
Page web : [https://mia.toulouse.inra.fr/Patrick\\_TAILLANDIER](https://mia.toulouse.inra.fr/Patrick_TAILLANDIER)

### A.2 Formation

- 2008 Doctorat en informatique, obtenu le 2 décembre 2008,  
Université Paris-Est, France,  
Directeur de thèse : Alexis Drogoul, co-encadrante : Cécile Duchêne
- 2005 Master Recherche en informatique à l'INSA Lyon,  
filière Connaissances et Raisonnement
- 2005 Diplôme d'ingénieur en informatique de l'Ecole CPE Lyon,  
section "Electronique, Télécoms, Informatique"

### A.3 Expériences professionnelles

Depuis 2016	Chargé de Recherche INRA, MIAT
2011 - 2016	Maître de Conférences en Géographie Université de Rouen, UMR IDEES, Chaire d'Excellence du CNRS
2010 - 2011	Post-doctorat, IRIT, Université Toulouse 1, projet MAELIA
2008 - 2010	Post-doctorat, IRD/IFI, Hanoi, Vietnam
2005 - 2008	Doctorat, Laboratoire COGIT, IGN, Saint Mandé, France
Mars - Août 2005	Stage M2, LIP 6, Paris, France
Juillet - Déc 2003	Stage, Université de South Australia

### A.4 Animation scientifique

#### A.4.1 Charges collectives

- Co-responsable des séminaires de l'unité MIAT depuis 2018.
- Membre nommé du comité national de la section 39 du CNRS (2016-2021).

#### A.4.2 Jury de concours de recrutement

- Concours de recrutement DR, CNRS, Section 39, 2017-2021.
- Concours de recrutement CR, CNRS, Section 39, 2016-2021.
- Concours de recrutement MCF, UMR IDEES, Université de Rouen, 2018.
- Concours de recrutement MCF, UMR IDEES, Université de Rouen, 2015.
- Concours de recrutement IE, UMI UMMISCO IRD, 2017.

#### A.4.3 Jury de thèse

- Touhbi Saad, Université Cadi Ayyad, Marrakech, 2018.
- Mathieu Bourgeois, INSA Rouen, 2018.
- Hugo Fernandez-Mena, INRA de Bordeaux Aquitaine, 2017.

#### A.4.4 Comités de programme de conférences, relecture d'articles

##### Membre du comité d'organisation de conférences et d'ateliers

- *PFIA* (1-5 juillet 2019, Toulouse, France) : Plate-Forme Intelligence Artificielle.
- *SAGEO* (6-9 novembre 2017, Rouen, France) : conférence internationale francophone sur l'Analyse Spatiale et des Sciences de l'Information Géographique.
- *JFSMA* (5-7 octobre 2016, Rouen, France) : Journées Francophones sur les Systèmes Multi-Agents.
- *PICS IMEA Mini-workshop* (26 mai 2015, VIASM, Hanoi, Vietnam) : Multi-Paradigm Modeling and Simulation of Complex Systems.
- *Workshop Modeling and Simulation Platforms* (23-26 juin 2014, workshop satellite de ICCSA 2014, Le Havre, France).
- *Workshop ABSEA* (4-6 juin 2014, workshop satellite de PAAMS 2014, Salamanca, Spain) : Agent-based modeling and simulation of complex systems : engineering and applications.
- *Workshop MASyCo* (2 juillet 2013, workshop satellite des JFSMA 2013, Lille, France) : Modélisation Agents pour les Systèmes Complexes.
- *Workshop Simtools* (18 mars 2013, Paris, France) : Des outils libres pour la conception et l'analyse de simulations à base d'agents.



- *Workshop PRACSYS* (12 novembre 2010, workshop satellite de PRIMA 2010, Kolkata, India) : The first Pacific Rim workshop on Agent-based modeling and simulation of Complex Systems.

#### **Relecture pour des conférences (comité de programme)**

- International Conference on Autonomous Agents and Multiagent Systems.
- Conférence internationale francophone sur l'Analyse Spatiale et des Sciences de l'Information Géographique.
- International Conference on Practical Applications of Agents and Multi-Agent Systems,
- International Congress on Environmental Modelling and Software,
- International Conference on Behavioral, Economic, and Socio-Cultural Computing
- etc.

#### **Relecture pour des revues**

- PLOS ONE,
- Cybergéo,
- Revue d'Intelligence Artificielle,
- International Journal of Geographical Information Science,
- Modeling and Simulation Techniques for Improved Business Processes,
- etc.

#### **A.4.5 Expertise**

- Projets du "Austrian Science Fund", 2015.
- Projets ANR, 2017.

#### **A.4.6 Participation à des actions de prospective**

- Prospective "Transition numérique et pratiques de recherche et d'enseignement supérieur en agronomie, environnement, alimentation et sciences vétérinaires à l'horizon 2040". 22 octobre 2018.
- Prospective "Approches prédictives en biologie et écologie". GT 7 "Construction de modèles multi-échelle". 2018.

#### **A.4.7 Groupes de travail**

- Réseau *RNSC MAPS* (membre du comité de pilotage depuis 2011) : Réseau dédié à l'échange entre modélisateurs et à la diffusion des approches de modélisation à base d'agents de phénomènes spatialisés.
- Réseau *RNSC simtools* (membre du comité de pilotage - participation au montage du réseau, 2011-2016) : Réseau dédié à l'échange entre développeurs de logiciels libres pour la modélisation et la simulation de systèmes complexes.
- Réseau *CNRS Normandev* (membre du comité de pilotage, 2013-2016) : Réseau dédié au développement logiciel en Normandie.

### **A.5 Projets de recherche**

#### **A.5.1 Coordinateur du projet**

##### **Projet ANR ACTEUR**

- **Intitulé** : *Agents Cognitifs Territorialisés pour l'Etude des dynamiques Urbaines et des Risques*.
- **Période** : 2014-2019.

- **Responsable** : Patrick Taillandier, CR INRA, MIAT.
- **Financement** : ANR, Systèmes urbains durables

#### Projet PICS IMEA

- **Intitulé** : *Intégration de modèles à base d'équations dans des modèles à base d'agents.*
- **Période** : 2014-2017.
- **Responsable** : Patrick Taillandier, CR INRA, MIAT.
- **Financement** : CNRS, PICS

#### Projet GRR Bibliothèque de modèles

- **Intitulé** : *Bibliothèque de modèles pour l'étude de la Normandie.*
- **Période** : 2013-2014.
- **Responsable** : Patrick Taillandier, CR INRA, MIAT.
- **Financement** : Région Haute-Normandie, GRR

### A.5.2 Responsable scientifique pour mon unité

#### Projet PSDR TIP TOP

- **Intitulé** : *Transferts diffus agricoles et Infrastructures Paysagères : modélisation participative et Optimisation agro-écologique.*
- **Période** : 2016-2020.
- **Responsable** : Dominique Trévisan, IR INRA, UMR CARRETEL.
- **Financement** : Région Rhône Alpes, PSDR

#### Projet ANR Gen\*

- **Intitulé** : *Génération de populations synthétiques, localisées et socialement structurées pour la simulation sociale.*
- **Période** : 2013-2017.
- **Responsable** : Alexis Drogoul, DR IRD, UMI UMMISCO.
- **Financement** : ANR, Modèles Numériques

#### Projet ARCHIVES

- **Intitulé** : *Analysis and Reconstruction of Catastrophes in History within Interactive Virtual Environments and Simulations.*
- **Période** : 2015-2018.
- **Responsable** : Alexis Drogoul, DR IRD, UMI UMMISCO.
- **Financement** : USTH, Hanoi

### A.5.3 Participant

#### Projet VITAMIN

- **Intitulé** : *Vegetarian Transition Argument ModellIng.*
- **Période** : 2018-2019.
- **Responsable** : Rallou Thomopoulos, CR, UMR IATE, INRA.
- **Financement** : INRA, Meta-programme DID'IT

#### Projet ESCAPE

- **Intitulé** : *Exploring by Simulation Cities Awareness on Population Evacuation.*
- **Période** : 2016-2021.
- **Responsable** : Eric Daudé, CR, UMR IDEES, CNRS.

- **Financement** : ANR, Sécurité Globale & Cybersécurité

#### Projet ANR GeObs

- **Intitulé** : *Géographie des obsidiennes d'Anatolie orientale : sources, caractérisation et diffusion aux époques préhistoriques.*
- **Période** : 2015-2018.
- **Responsable** : Damase Mouralis, MCF, UMR IDEES, Université de Rouen.
- **Financement** : ANR, Programme Blanc

#### Projet ANR AEDESS

- **Intitulé** : *Analyse de l'Emergence de la Dengue et Simulation Spatiale.*
- **Période** : 2010-2014.
- **Responsable** : Richard Paul, Institut Pasteur
- **Financement** : ANR, Changements Environnementaux Planétaires & Sociétés

#### Projet ANR 3-Worlds

- **Intitulé** : *A multiscale simulation platform for ecology and environmental science.*
- **Période** : 2010-2014.
- **Responsable** : Jacques Gignoux, CR CNRS, UMR BIOEMCO
- **Financement** : ANR, Changements Environnementaux Planétaires & Sociétés

#### Projet GRR MOSAIC

- **Intitulé** : *Accidents Industriels Majeurs et Vulnérabilités Humaine en Haute-Normandie : Modélisation et Simulation en Milieu Urbain.*
- **Période** : 2009-2015.
- **Responsable** : Eric Daudé, CR CNRS, UMR IDEES
- **Financement** : Région Haute-Normandie, GRR

#### Projet RTRA MAELIA

- **Intitulé** : *Multi-Agent for Environmental Norms Impact Assessment .*
- **Période** : 2009-2012.
- **Responsable** : Pierre Mazzega, DR IRD, UMR LMTG
- **Financement** : RTRA STAE

## A.6 Activités d'encadrement

### A.6.1 Encadrement de thèses

- **Mathieu Bourgeois**. 2015-2018. INSA Rouen, UMR IDEES, UMR LITIS. *Vers des agents cognitifs, affectifs et sociaux dans la simulation.*  
Encadrement : L. Vercouter (INSA Rouen - IRIT, Rouen, *directeur de thèse*), P. Taillandier (INRA - MIAT, Toulouse, *co-encadrant*).
- **Chi Quang Truong**. 2012-2016. Université Paris 6, Université de Can Tho, IRD, UMI UMMISCO, *Integrating cognitive models of Human decision-making in agent-based models : an application to land use planning under climate change in the Makong river delta.*  
Encadrement : A. Drogoul (IRD - UMI UMMISCO, UMR IRIT, *directeur de thèse*), M.Q. Vo (Université de Can Tho, Vietnam, *co-directeur de thèse*), P. Taillandier (INRA - MIAT, Toulouse, *co-encadrant*), B. Gaudou (Université Toulouse 1 - UMR IRIT, Toulouse, *co-encadrant*).

### A.6.2 Encadrement de Master

- M. Valette (SupAero). Vers la construction de sémantiques argumentatives à base de classement. Mars-Juillet 2018. Encadrement assuré en collaboration de Dominique Longin (CNRS, UMR IRIT) et Benoit Gaudou (Université Toulouse 1, UMR IRIT).
- O. Rousselle (AgroParisTech). Complex agent-based modeling for cumulated impact assessment on ressources of a coral reef social-ecological system. Mai-Novembre 2017. Encadrement assuré en collaboration de Joachim Claudet et Mélodie Dubois (CNRS, UMR CRIOBE).
- M. Bourgais (INSA Rouen). Définition d'une architecture d'agent cognitif modulaire pour la simulation à base d'agents. Avril-Septembre 2015.
- V.M. Le (IFI, Hanoi, Vietnam). Intégration d'agents cognitifs dans la plateforme de simulation GAMA. Mars-août 2011. Encadrement assuré en collaboration de Benoit Gaudou (Université Toulouse 1, UMR IRIT).

### A.6.3 Encadrement de CDD

- N. Ayari, IE, Structure et Organisation des Paysages Agricoles - Incidence sur les transferts d'eau et les fonctions écosystémiques du paysage. CDD INRA sur le projet PSDR TIP-TOP (2018 - 12 mois).
- A. Grignard, IR, Développement d'une architecture d'agents cognitifs adaptée à la simulation. CDD CNRS sur le projet ANR ACTEUR (2015 - 9 mois).
- K. Chapuis, Post-doc, Génération et Spatialisation de populations synthétiques. CDD CNRS sur le projet ANR GEN\* (2016-2017 - 13 mois).

## A.7 Activités d'enseignement

### A.7.1 Comité d'organisation de formations et d'écoles thématiques

- *MISS-ABMS 2018* (10 au 21 septembre 2018, Montpellier, France) : Ecole d'été internationale Designing, Implementing and Running Agent-Based Models for Renewable Resources Management.
- *Asian Water Platform 2018* (5 au 14 juillet 2018, Can Tho, Vietnam).
- *SCEMSITE 2018* (4 au 8 juin 2018, Mérignac, France) : École Chercheurs sur la Prospective, participation et modélisation spatiale pour la gestion des ressources dans les territoires.
- *Formation GAMA 2018* (3 au 5 avril 2018, Toulouse, France) : Introduction à la modélisation à base d'agents et à l'utilisation de la plate-forme GAMA.
- *MISS-ABMS 2017* (4 au 15 septembre 2017, Montpellier, France) : Ecole d'été internationale Designing, Implementing and Running Agent-Based Models for Renewable Resources Management.
- *Journées de Tam Dao 2017* (7 au 15 juillet 2017, Can Tho, Vietnam) : Fleuves et deltas en Asie de du Su-Est : Méthodes et pluridisciplinarité d'analyse. Atelier 3 : Approche pratique de la modélisation à base d'agents.
- *MAPS 10* (18-23 juin 2017, Oléron, France) : Ecole thématique CNRS Modélisation multi-agents appliquée aux phénomènes spatialisés.
- *Gen\* Training Session* (1-2 février 2017, Marrakech, Maroc) : École thématique d'introduction à la génération de population synthétique et à la librairie Gen\*.
- *MISS-ABMS 2016* (12 au 23 septembre 2016, Montpellier, France) : Ecole d'été internationale Designing, Implementing and Running Agent-Based Models for Renewable Resources Management.

- *Journées de Tam Dao 2016* (8-11 juillet 2016, Danang, Vietnam) : « Les enjeux de la transition énergétique au Vietnam et en Asie du Sud-Est. Atelier 3 : Approche pratique de la modélisation à base d'agents.
- *MAPS 8* (21-26 juin 2015, Saint Paul de Vence, France) : Ecole thématique CNRS Modélisation multi-agents appliquée aux phénomènes spatialisés.
- *NeX Days* (28-30 avril 2015, Agadir, Maroc) : Journées réseaux et systèmes complexes.
- *GAMA Training Session* (12-16 janvier 2015, Manille, Philippines) : Ecole thématique de formation à la plate-forme GAMA.
- *MISS-ABMS 2014* (8 au 19 septembre 2014, Montpellier, France) : Ecole d'été internationale Designing, Implementing and Running Agent-Based Models for Renewable Resources Management.
- *Journées de Tam Dao 2014* (21 au 29 juillet 2014, Da Lat, Vietnam) : Regards sur le développement urbain durable : approches méthodologiques, transversales et opérationnelles. Atelier 4 : Outils et modèles pour la compréhension et l'exploration des dynamiques spatiales urbaines.
- *MISS-ABMS 2013* (23 septembre au 4 octobre 2013, Montpellier, France) : Ecole d'été internationale Designing, Implementing and Running Agent-Based Models for Renewable Resources Management.
- *Journées de Tam Dao 2013* (22 au 27 juillet 2013, Tam Dao, Vietnam) : Perception et gestion des risques : Approches méthodologiques appliquées au développement : Méthodes et pluridisciplinarité d'analyse. Atelier 4 : Comprendre les crises passées pour mieux gérer le présent : initiation à la modélisation géo-historique des risques (la crue du fleuve Rouge de 1926).
- *MAPS 5* (23-28 juin 2013, Oléron, France) : Ecole thématique CNRS Modélisation multi-agents appliquée aux phénomènes spatialisés.
- *EMISCOE* (10 - 14 juin 2013, Bondy, France) : Ecole de Modélisation Informatique des Systèmes Complexes Ecologiques.
- *GAMA Summer School* (12-16 novembre 2012, Can Tho, Vietnam) : Ecole thématique de formation à la plate-forme GAMA.
- *Journées de Tam Dao 2012* (16 au 21 juin 2012, Tam Dao, Vietnam) : L'eau dans tous ses états : Méthodes et pluridisciplinarité d'analyse. Atelier 4 : Approche pratique de la modélisation à base d'agents.
- *Formation IRD à l'épidémiologie* (24 juillet 2010, Saint Louis, Sénégal) : Ecole thématique de formation à la modélisation pour l'épidémiologie.
- *Formation GAMA* (26-27 octobre 2009, Hanoi, Vietnam) : Ecole thématique de formation à la plate-forme GAMA.

## A.7.2 Cours

### Université de Rouen

- Risques, modélisation et approches systémiques : M2 Recherche Géographie (2013-2016)
- Modélisation agent : M2 TRIAD (2012-2016)
- Algorithmique : M2 TRIAD (2011-2016)
- Géomatique : M1 TRIAD (2011-2013)
- Statistiques spatiales : L3 Géographie (2012-2013)
- Statistiques multi-variées : L2 Géographie (2012-2013)
- Modélisation appliquée aux territoires locaux : M2 Recherche Géographie (2011-2012)

### Institut de la Francophonie pour l'Informatique, Hanoi, Vietnam

- Apprentissage Artificiel : M2 Informatique (2012-2013)

### Université Paris Est

- Programmation : L1 Informatique (2005-2008)
- Programmation : IMAC 1ère année (2007-2008)
- Introduction à l'apprentissage artificiel : ENSG 2ième année (2007-2008)
- Introduction à l'environnement système et réseau : L1 Informatique (2005-2007)
- Structure de données : L2 Informatique (2005-2006)

## A.8 Publications

### A.8.1 Revues à comité de lecture

#### Revues internationales

- [1] Taillandier, P., Gaudou B., Grignard, A., Huynh, Q.N., Marilleau, N., Philippon, D., Caillou, P., Drogoul, A. Participatory Modeling and Simulation with the GAMA platform. *Journal of Artificial Societies and Social Simulation*. A paraître.
- [2] Micolier, A., Taillandier F., Taillandier, P., Bos, F. Li-BIM : an advanced cognitive model to simulate the occupant behavior and its indoor comfort from Building Information Modeling. *Engineering Applications of Artificial Intelligence*. A paraître.
- [3] Taillandier, P., Gaudou B., Grignard, A., Huynh, Q.N., Marilleau, N., Caillou, P., Philippon, D., Drogoul, A. Building, composing and experimenting complex spatial models with the GAMA platform. *Geoinformatica*, 2018.
- [4] Chapuis, K., Taillandier, P., Misslin, R., Drogoul, A. Gen\* : a generic toolkit to generate spatially explicit synthetic populations. *International Journal of Geographical Information Science*, 2018, pp. 1–17.
- [5] Bourgais, M., Taillandier, P., Vercoouter, L., Adam, C. Emotion Modeling in Social Simulation : A Survey. *Journal of Artificial Societies and Social Simulation*, vol. 20, 2018.
- [6] Adam, C., Taillandier, P., Dugdale, J., Gaudou, B. BDI vs FSM Agents in Social Simulations for Raising Awareness in Disasters : A Case Study in Melbourne Bushfires. *International Journal of Information Systems for Crisis Response and Management (IJISCRAM)*, vol. 9, 2017, pp. 27–44.
- [7] Taillandier, F., Taillandier, P., Hamzaoui, F., Breyse, D. A new agent-based model to manage construction project risks – application to the crossroad of Bab El Karmadine at Tlemcen. *European Journal of Environmental and Civil Engineering*, vol. 20, 2016, pp. 1197–1213.
- [8] Taillandier, F., Taillandier, P., Tepeli, E., Breyse, D., Mehdizadeh, R., Khartabil, F. A multi-agent model to manage risks in construction project (SMACC). *Automation in Construction*, vol. 55, 2015, pp. 156–163.
- [9] Taillandier, F., Abi-Zeid, I., Taillandier, P., Sauce, G., Bonetto, R. An interactive decision support method for real estate management in a multi-criteria framework – REMIND. *International Journal of Strategic Property Management*, vol. 18, 2014, pp. 265–278.
- [10] Müller, Balbi, Buchmann, de Sousa, Dressler, Groeneveld, Klassert, Le, Millington, Nolzen, Parker, Polhill, Schlüter, Schulze, Schwarz, Sun, Taillandier, Weise Standardised and transparent model descriptions for agent-based models : Current status and prospects. *Environmental Modelling & Software*, vol. 55, 2014, pp. 156–163.
- [11] Taillandier, P., Taillandier, F. Multi-criteria diagnosis of control knowledge for cartographic generalisation. *European Journal of Operational Research*, vol. 2017, 2012, pp. 633–642.



- [12] Taillandier, P., Gaffuri, J. Improving map generalisation with new pruning heuristics. *International Journal of Geographical Information Science*, vol. 26, 2012, pp. 1309–1323.
- [13] Taillandier, P., Duchêne, C., Drogoul, A. Automatic revision of the control knowledge used by trial and error methods : Application to cartographic generalisation. *Applied Soft Computing*, vol. 11, 2011, pp. 2818–2832.
- [14] Taillandier, P., Duchêne, C., Drogoul, A. Automatic revision of rules used to guide the generalisation process in systems based on a trial and error strategy. *International Journal of Geographical Information Science*, vol. 25, 2011, pp. 1971-1999.
- [15] Amouroux, E., Taillandier, P., Drogoul, A. Complex environment representation in epidemiology ABM : application on H5N1 propagation. *Journal of Science and Technology*, vol. 48, 2010, pp. 13–25.
- [16] Steiniger, S., Taillandier, P., Weibel, R. Utilising urban context recognition and machine learning to improve the generalisation of buildings. *International Journal of Geographical Information Science*, vol. 24, 2010, pp. 253–282.

#### Revues nationales

- [17] Taillandier, P. La modélisation du temps dans la simulation à base d'agents. *Information géographique*, vol. 79, 2015.
- [18] Taillandier, P., Grignard, A., Gaudou, B., Drogoul, A. Des données géographiques à la simulation à base d'agents : application de la plate-forme GAMA. *Cybergéo*, 2014.
- [19] Cisse, A., Bah, A., Drogoul, A., Cissé, A. T., Ndione, J. A., Kébé, C. M. F., Taillandier, P. Un modèle à base d'agents sur la transmission et la diffusion de la fièvre de la Vallée du Rift à Barkédji (Ferlo, Sénégal). *Studia Informatica Universalis*, vol. 10, 2015, pp. 77–97.
- [20] Banos, A., Moreno, D., Pivano, C., Taillandier, P. Christaller, still alive ! *Cybergéo*, 2011.

#### Articles en cours de soumission/révision

- [21] Taillandier, P., Czura, G., Tranouez, P., Daudé, E. Agent-based Simulation of Traffic with the GAMA Platform : Application to the Evacuation of a City. *Journal of Simulation*
- [22] Binois M., Picheny, V., Taillandier, P., Habbal, A. The Kalai-Smorodinsky solution for many-objective Bayesian optimization. *Journal of Machine Learning Research*

### A.8.2 Communications dans des congrès avec sélection par comité de lecture sur texte intégral

- [23] Daudé, E., Chapuis, K., Caron, C., Drogoul, A., Gaudou, B., Rey-Coyrehourq, S., Saval, A., Taillandier, P., Tranouez, P., Zucker, J.D. – ESCAPE : Exploring by Simulation Cities Awareness on Population Evacuation. *International Conference on Information Systems for Crisis Response and Management* – Valencia, Spain, May 2019.
- [24] Bourgeois, M., Taillandier, P., Vercouter, L. – BEN : An Agent Architecture for Explainable and Expressive Behavior in Social Simulation. *International Workshop on EXplainable TRansparent Autonomous Agents and Multi-Agent Systems* – Montreal, Canada, May 2019.
- [25] Valette, M., Gaudou, B., Longin, D., Taillandier, P. – Modeling a real-case situation of egress using BDI agents with emotions and social skills. *The 21st International Conference on Principles and Practice of Multi-Agent Systems* – Tokyo, Japan, Oct 2018.

- [26] Chapuis K., Taillandier, P., Gaudou B., Drogoul A., Daudé E. – A multi-modal urban traffic agent-based framework to study individual response to catastrophic events. *The 21st International Conference on Principles and Practice of Multi-Agent Systems* – Tokyo, Japan, Oct 2018.
- [27] Grignard, A., Alonso, L., Taillandier, P., Gaudou, B., Nguyen-Huu, T., Gruel, W., Larson, K. – The Impact of New Mobility Modes on a City : A Generic Approach Using ABM. *International Conference on Complex Systems* – Cambridge, USA, July 2018.
- [28] Bourgeois, M., Taillandier, P., Vercouter, L. – Enhancing the Behavior of Agents in Social Simulations with Emotions and Social Relations. *MABS 2017 Multi-Agent Based Simulation* – Rio, Brasil, May 2017.
- [29] Taillandier, P., Bourgeois, M., Drogoul, A., Vercouter, L. – Using parallel computing to improve the scalability of models with BDI agents. *Social Simulation Conference* – Dublin, Ireland, Sep 2017.
- [30] Bourgeois, M., Taillandier, P., Vercouter, L. – Cognition, émotions et relations sociales pour la simulation multi-agent. *vingt-cinquième édition des Journées Francophones sur les Systèmes Multi-Agents (JFSMA)* – Caen, France, 2017.
- [31] Adam, C., Taillandier, P., Dugdale, J. – Comparing Agent Architectures in Social Simulation : BDI Agents versus Finite-state Machines. *50th Hawaii International Conference on System Sciences* – Waikoloa, USA, 2017.
- [32] Taillandier, P., Banos, A., Drogoul, A., Gaudou, B., Marilleau, N., Truong, QC. – Simulating urban growth with raster and vector models : A case study for the city of Can Tho, Vietnam. *International Workshop on Agent Based Modelling of Urban Systems*, pp. 21–38. – Singapour, Singapour, 2016.
- [33] Taillandier, P., Bourgeois, M., Caillou, P., Adam, C., Gaudou, B. – A BDI agent architecture for the GAMA modeling and simulation platform. *17th International Workshop on Multi-Agent-Based Simulation* – Singapour, Singapour, 2016.
- [34] Philippon, D., Choissy, M., Drogoul, A., Gaudou, B., Marilleau, N., Taillandier, P., Truong, QC, – Exploring trade and health policies influence on Dengue spread with an Agent-Based Model *17th International Workshop on Multi-Agent-Based Simulation* – Singapour, Singapour, 2016.
- [35] Caillou P., Gaudou B., Grignard A., Truong C.Q., Taillandier P. – A Simple-to-use BDI architecture for Agent-based Modeling and Simulation. *Advances in Social Simulation 2015*, pp. 15–28. – Groningen, Netherlands, Sep 2015.
- [36] Truong, Q. C., Taillandier, P., Gaudou, B., Vo, M. Q., Nguyen, Drogoul, A. – Exploring agent architectures for farmer behavior in land-use change. a case study in coastal area of the Vietnamese Mekong Delta *The Sixteenth International Workshop on Multi-Agent-Based Simulation*, p. 146–158 – Istanbul, Turquie, 2015.
- [37] Czura, G, Taillandier, P., Tranouez, P., Daudé, É. – MOSAIIC : City-level agent-based traffic simulation adapted to emergency situations. *International Conference on Social Modeling and Simulation, plus Econophysics Colloquium*, p. 265–274 – Kobe, Japan, 2015.
- [38] Gasmi, N., Grignard, A., Drogoul, A., Gaudou, B., Taillandier, P., Tessier, O., An, V. D. – Reproducing and exploring past events using agent-based geo-historical models. *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, p. 151–163– Paris, France, 2014.
- [39] Taillandier, P. – GAMAGRAM : graphical modeling with the GAMA platform *4th International Conference on Complex Systems and Applications* – Le Havre, France, 2014.
- [40] Taillandier, P. – Traffic simulation with the GAMA platform *Eighth International Workshop on Agents in Traffic and Transportation* – Paris, France, 2014.

- [41] Gaudou, B., Sibertin-Blanc, Therond, O., Amblard, Auda, Y., Arcangeli, Balestrat, M., Moirez-Charron, M.-H., Gondet, E., Hong, Y., Lardy, R., Louail, Mayor, Panzoli, D., Sauvage, S., Sánchez-Pérez, Taillandier, P., Van Bai, N., Vavasseur, M., Mazzega, P. – The MAELIA multi-agent platform for integrated analysis of interactions between agricultural land-use and low-water management strategies. *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, p. 1361–1362–Saint-Paul, USA, 2013.
- [42] Drogoul, A., Amouroux, E., Caillou, P., Gaudou, B., Grignard, A., Marilleau, N., Taillandier, P., Vavasseur, M., Vo, D.-A., Zucker, J.-D. – GAMA : multi-level and complex environment for agent-based models and simulations *12th International Conference on Autonomous agents and multi-agent systems* – Sain-Paul, USA, 2013.
- [43] Drogoul, A., Amouroux, Caillou, Gaudou, Grignard, Marilleau, Taillandier, Vavasseur, Vo, Zucker, J.D. – GAMA : a spatially explicit, multi-level, agent-based modeling and simulation platform *Advances on Practical Applications of Agents and Multi-Agent Systems* – Salamanque, Espagne, 2013.
- [44] Grignard, A., Taillandier, P., Gaudou, B., Vo, D. A., Huynh, N. Q., Drogoul, A. – GAMA 1.6 : Advancing the art of complex agent-based modeling and simulation *16th International Conference on Principles and Practice of Multi-agent Systems*, p. 117–131 – Salamanque, Espagne, 2013.
- [45] Le, V.-M., Gaudou, B., Taillandier, P., Vo, D.-A. – A new BDI architecture to formalize cognitive agent behaviors into simulations *7th Conference KES on Agent and Multi-Agent System - Technologies and Applications*, p. 395–403 – Hue City, Vietnam, 2013.
- [46] Nguyen, N. D., Taillandier, P., Drogoul, A., Auger, P. – Inferring equation-based models from agent-based models : a case study in competition dynamics *3th International Conference on Principles and Practice of Multi-Agent Systems*, p. 413–427 – Kolkata, India, 2010.
- [47] Taillandier, P., Gaffuri, J. – Automatic Sampling of Geographic objects. *Sixth international conference on Geographic Information Science* – Zurich, Suisse, 2010.
- [48] Taillandier, P., Gaffuri, J. – Designing generalisation evaluation function through human-machine dialogue. *Sixth international conference on Geographic Information Science* – Zurich, Suisse, 2010.
- [49] Taillandier, P., Therond, O., Gaudou, B. – Une architecture d’agent BDI basée sur la théorie des fonctions de croyance : application à la simulation du comportement des agriculteurs. *Journées Francophones sur les Systèmes Multi-Agents*, p. 107–116 – Honfleur, France, 2012.
- [50] Taillandier, P., Gaffuri, J. – Objective function designing led by user preferences acquisition. *International Conference on Information Technology and Applications* – Hanoi, France, 2009.
- [51] Taillandier, P., Amouroux, Vo, Olteanu-Raimond, A.-M. – Using Belief Theory to formalize the agent behavior : application to the simulation of avian flu propagation. *International Conference on Principles and Practice of Multi-Agent Systems*, p. 575–587 – Kolkata, India, 2010.
- [52] Taillandier, P., Vo, D.A., Amouroux, E., Drogoul. A. – GAMA : a simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. *International Conference on Principles and Practice of Multi-Agent Systems*, p. 242–258 – Kolkata, India, 2010.
- [53] Le, V.M., Adam, C., Canal, R., Gaudou, B., Tuong Vinh, Taillandier, P. – GAMA : a simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. *International Conference on Principles and Practice of Multi-Agent Systems*, p. 604–619 – Kolkata, India, 2010.
- [54] Taillandier, P., Drogoul, A. – From grid environment to geographic vector agents, modeling with the GAMA simulation platform. *25th International Cartographic Conference* – Paris, France, 2011.

- [55] Taillandier, P., Gaffuri, J. – Using human-machine dialogue to refine generalisation evaluation function. *25th International Cartographic Conference* – Paris, France, 2011.
- [56] Taillandier, P., Stinckwich, S. – Using the PROMETHEE multi-criteria decision making method to define new exploration strategies for rescue robots. *International Symposium on Safety, Security, and Rescue Robotics*, p. 321–326 – Kyoto, Japan, 2011.
- [57] Taillandier, P., Amblard, F. – Cartography of multi-agent model parameter space through a reactive dicotomous approach. *25th European Simulation and Modelling Conference* – Guimaraes, Portugal, 2011.
- [58] Taillandier, P., Therond, O. – Use of the Belief Theory to formalize Agent Decision Making Processes : Application to cropping Plan Decision Making. *25th European Simulation and Modelling Conference*, p. 138–142 – Guimaraes, Portugal, 2011.
- [59] Taillandier, P., Drogoul, A. – Supervised feature evaluation by consistency analysis : application to measure sets used to characterise geographic objects. *Second International Conference on Knowledge and Systems Engineering* , p. 63–68 – Hanoi, Vietnam, 2010.
- [60] Taillandier, P., Vo, D.-A., Amouroux, E., Drogoul, A. – GAMA : bringing GIS and multi-level capabilities to multi-agent simulation. *S8th European Workshop on Multi-Agent Systems* – Paris, France, 2010.
- [61] Boucher, A., Canal, Chu, Q.T., Drogoul, A., Gaudou, G., Le, V.M., Moraru, V., Nguyen, V.N., Vu, Q.A.N., Taillandier, P., Sempe, F., Stinckwich, S. – The AROUND project : Adapting robotic disaster response to developing countries. In : *Safety, Security & Rescue Robotics. International Symposium on Safety, Security and Rescue Robotics*, p. 1–6 – Denver, USA, 2009.
- [62] Taillandier, P., Buard, E. – Designing agent behaviour in agent-based simulation through participatory method. *International Conference on Principles and Practice of Multi-Agent Systems* , p. 571–578 – Nagoya, Japan, 2009.
- [63] Taillandier, P. – Heuristic revision by heuristic space exploration. *International Conference on Knowledge and Systems Engineering* , p. 571–578 – Hanoi, Vietnam, 2009.
- [64] Taillandier, P., Domingues, C. – Evaluation automatique de connaissances par la théorie des fonctions de croyance. *9èmes rencontres des jeunes chercheurs en intelligence artificielle* , p. 571–578 – Hammamet, Tunisie, 2009.
- [65] Taillandier, P., Chu, Q.T. – Using participatory paradigm to learn human behaviour. *International Conference on Knowledge and Systems Engineering*, p. 55–60 – Hanoi, Vietnam, 2009.
- [66] Taillandier, P., Duchene, C., Drogoul, A. – Using belief theory to diagnose control knowledge quality : application to cartographic generalisation. *International Conference on Computing and Communication Technologies*, p. 55–60 – Danang, Vietnam, 2009.
- [67] Taillandier, P. – Diagnosis in systems based on an informed tree search strategy. *International Conference on Soft Computing as Transdisciplinary Science and Technology*, p. 55–60 – Cergy-Pontoise, France, 2008.
- [68] Taillandier, P., Duchêne, C., Drogoul, A. – Knowledge revision in systems based on an informed tree search strategy. *International Conference on Soft Computing as Transdisciplinary Science and Technology*, p. 55–60 – Cergy-Pontoise, France, 2008.
- [69] Steiniger, S., Taillandier, P. – Improving Map Generalisation of Buildings by Introduction of Urban Context Rules. *9th International Conference on GeoComputation*, p. 55–60 – Maynooth, Irlande, 2007.

- [70] Taillandier, P. – Acquisition automatique de connaissances de guidage d’un processus de généralisation de données géographiques. *8ième rencontres des jeunes chercheurs en intelligence artificielle*, p. 213–229 – Grenoble, France, 2007.
- [71] Taillandier, P. – Révision à base d’agents des connaissances de guidage d’un processus de généralisation de données géographiques. *Quinzièmes Journées francophones sur les systèmes multi-agents*, p. 243–252 – Carcassonne, France, 2007.
- [72] Taillandier, P. – Automatic Knowledge Revision of a Generalisation System. *11th ICA Workshop on Generalisation and Multiple Representation* – Moscou, Russie, 2007.
- [73] Ruas, A., Dyevre, A., Duchêne, C., Taillandier, P. – Methods for improving and updating the knowledge of a generalization system. *AutoCarto* – Moscou, Russie, 2007.

### A.8.3 Contributions à des ouvrages collectifs

- [74] Banos, A., Corson, N., Lang, C., Marilleau, N., Taillandier, P. – Modélisation multi-échelles : application au trafic routier. *Simulation spatiale à base d’agents avec NetLogo 2 : notions avancées*, ISTE, pp.45–68, – 2017.
- [75] Amblard, F., Daudé, E., Gaudou, B., Grignard, A., Hutzler, G., Lang, C., ..., Taillandier, P. – Introduction à NetLogo. *Simulation spatiale à base d’agents avec NetLogo, partie 1*, ISTE, pp.73–112, – 2015.
- [76] Abrami, G., Amalric, M., Amblard, F., Anselme, B., Banos, A., Beck, E., ..., Daudé, E. – Recueil de fiches pédagogiques du réseau MAPS. *Modélisation multi-Agents appliquée aux Phénomènes Spatialisés*, 258 p. – 2014.
- [77] Drogoul, A., Gaudou, B., Grignard, A., Taillandier, P., Vo, D. A. – Practical Approach To Agent-Based Modelling.. *Water and its Many Issues. Methods and Cross-cutting Analysis*, Nha xuat ban Tri Thucp,k277–300 – 2013.

### A.8.4 Rapports diplômants

- [78] Taillandier, P. – Révision automatique des connaissances guidant l’exploration informée d’arbres d’états : application au contexte de la généralisation de données géographiques. *Doctoral dissertation, Université Paris-Est* – 2008.

### A.8.5 Développements informatiques

- [79] [Logiciel] Plate-forme GAMA- <http://gama-platform.org/> : *plate-forme open-source dédiée à la conception et au développement de modèles à base d’agents* –
- [80] [Logiciel] Toolkit Gen\* - <https://github.com/ANRGenstar> : *toolkit dédié à la génération de populations synthétiques* –